Digital Image Watermarking Methods for Copyright Protection and Authentication

by

Chaw-Seng WOO

Bachelor of Computer Science
Master of Computer Science

Thesis submitted in accordance with the regulations for Degree of Doctor of Philosophy

Information Security Institute
Faculty of Information Technology
Queensland University of Technology

March 2007
Keywords

Digital watermark, Image watermark, Robust watermark, Semi-fragile watermark, Hybrid watermark, Copyright protection, Content authentication, Tamper detection, Tamper localization, Self-embedding, Self-authentication, Self-recovery.
Abstract

The ease of digital media modification and dissemination necessitates content protection beyond encryption. Information hidden as digital watermarks in multimedia enables protection mechanism in decrypted contents.

The aims of this research are three-fold: (i) to investigate the strength and limitations of current watermarking schemes, (ii) to design and develop new schemes to overcome the limitations, and (iii) to evaluate the new schemes using application scenarios of copyright protection, tamper detection and authentication. We focus on geometrically robust watermarking and semi-fragile watermarking for digital images. Additionally, hybrid schemes that combine the strength of both robust and semi-fragile watermarks are studied.

Robust watermarks are well suited for copyright protection because they stay intact with the image under various manipulations. We investigated two major approaches of robust watermarking. In the synchronization approach, we employed motion estimation for watermark resynchronization. We also developed a novel watermark resynchronization method that has low computational cost using scale normalization and flowline curvature. In another approach, we firstly analyzed and improved a blind watermark detection method. The new method reduces significantly the computational cost of its watermark embedding. Secondly, we created a geometric invariant domain using a combination of transforms, and adapted the blind watermark detection method that we improved. It totally eliminates the need of resynchronization in watermark detection, which is a very desirable achievement that can hardly be found in existing schemes.
On the other hand, semi-fragile watermarks are good at content authentication because they can differentiate minor image enhancements from major manipulations. New capabilities of semi-fragile watermarks are identified. Then, we developed a semi-fragile watermarking method in wavelet domain that offers content authentication and tamper localization. Unlike others, our scheme overcomes a major challenge called cropping attack and provides approximate content recovery without resorting to an original image.

Hybrid schemes combine robust and semi-fragile watermarks to offer deductive information in digital media forensics. We firstly carried out a pilot study by combining robust and fragile watermarks. Then, we performed a comparative analysis on two implementation methods of a hybrid watermarking scheme. The first method has the robust watermark and the fragile watermark overlapped while the second method uses non-overlapping robust and fragile watermarks. Based on the results of the comparative analysis, we merge our geometric invariant domain with our semi-fragile watermark to produce a hybrid scheme. This hybrid scheme fulfilled the copyright protection, tamper detection, and content authentication objectives when evaluated in an investigation scenario.
# Contents

Keywords ................................................................................................................................. i  
Abstract ................................................................................................................................ iii  
List of Figures .................................................................................................................... xi  
List of Tables .................................................................................................................... xv  
List of Abbreviations .......................................................................................................... xvii  
Declaration ......................................................................................................................... xix  
Previously Published Materials ........................................................................................... xxi  
Acknowledgements ........................................................................................................... xxiii

**Chapter 1 Introduction** ......................................................................................................... 1  
1.1 Motivations .................................................................................................................. 1  
1.2 Overview of Digital Watermarking ............................................................................. 3  
1.3 Aims ........................................................................................................................... 6  
1.4 Achievements and Contributions .............................................................................. 8  
1.5 Thesis Outline ............................................................................................................ 9  

**Chapter 2 Digital Image Watermarking** ........................................................................... 11  
2.1 Digital Watermarking .................................................................................................. 11  
2.2 Properties of Digital Watermarks .............................................................................. 13  
2.2.1 Robustness ......................................................................................................... 14  
2.2.2 Capacity and data payload .................................................................................. 15  
2.2.3 Imperceptibility .................................................................................................. 15  
2.2.4 Security .............................................................................................................. 17  
2.2.5 Computational cost ............................................................................................. 19
2.2.6 Watermark detection reliability ................................................................. 19
2.2.7 Blind detection of watermark ................................................................. 21
2.2.8 Trade-off between performance factors ................................................... 21
2.3 Applications of Digital Watermark ................................................................. 22
    2.3.1 Digital Rights Management ................................................................. 22
    2.3.2 Copyright protection ............................................................................. 24
    2.3.3 Authentication ....................................................................................... 24
    2.3.4 Tamper detection and localization ......................................................... 25
    2.3.5 Annotation and privacy control .............................................................. 25
    2.3.6 Media forensics ..................................................................................... 26
    2.3.7 Other applications ............................................................................... 27
2.4 Robust Image Watermarking ........................................................................... 28
    2.4.1 Attacks on watermark robustness ......................................................... 28
    2.4.2 Approaches to robust watermarking ...................................................... 30
    2.4.3 Practical issues ...................................................................................... 33
2.5 Semi-fragile Image Watermarking ................................................................. 34
    2.5.1 Content authentication .......................................................................... 35
    2.5.2 Tamper localization ............................................................................... 36
    2.5.3 Content recovery ................................................................................... 39
    2.5.4 Approaches .......................................................................................... 39
    2.5.5 Remarks ............................................................................................... 41
2.6 Hybrid Methods in Image Watermarking ....................................................... 41
    2.6.1 Existing hybrid methods ....................................................................... 42
    2.6.2 Potential improvements ....................................................................... 45
2.7 Chapter summary ........................................................................................... 46

Chapter 3 Robust Watermarking by Synchronization ........................................ 49
3.1 Introduction .................................................................................................... 49
3.2 Related Work .................................................................................................. 51
3.3 Motion Estimation for Synchronization ........................................................ 52
    3.3.1 Geometrical Transform and Affine Transform ...................................... 53
    3.3.2 Differential Affine Motion Estimation .................................................. 54
    3.3.3 Experimental Evaluations .................................................................... 57
3.4 Flowline Curvature for Synchronization ....................................................... 65
    3.4.1 Scale Normalization .............................................................................. 66
### Chapter 4 Robust Watermarking using Invariant Domain

#### 4.1 Introduction

- Watermark Embedding ................................................................. 87
- Watermark Detection ................................................................. 91
- Analysis of Experiment Results .................................................. 92
- Discussions ................................................................................. 99
- Conclusion .................................................................................. 100

#### 4.3 Geometric Invariant Domain

- The Dual Tree-Complex Wavelet Transform ................................. 102
- Construction of the geometric invariant domain ........................... 104
- The watermarking method ......................................................... 106
- Analysis of Experiment Results .................................................. 110
- Discussions ................................................................................. 115
- Conclusion .................................................................................. 117

#### 4.4 Chapter Summary

- .................................................................................................... 117

### Chapter 5 Semi-fragile Watermarking with Self-Authentication and Self-Recovery

#### 5.1 Introduction

- .................................................................................................... 120

#### 5.2 Challenges in Semi-fragile Watermarking

- .................................................................................................... 121

#### 5.3 Design of Semi-fragile Watermarking

- Watermark Embedding ................................................................. 126
- Watermark Detection ................................................................. 127

#### 5.4 Analysis of Experimental Results

- Experiment settings ..................................................................... 130
- Imperceptibility.......................................................................... 130
- Semi-fragile performance............................................................ 131
- Tamper localization ..................................................................... 134
- Content recovery ........................................................................ 135
5.4.6 False positive condition ................................................................. 136
5.4.7 Watermark security ........................................................................ 137
5.4.8 Comparative analysis ................................................................... 137
5.4.9 Overall performance .................................................................... 137
5.5 Chapter Summary ............................................................................. 138

Chapter 6 Hybrid Watermarking Method ........................................... 139
6.1 Related Work .................................................................................. 140
6.1.1 Content recovery and Watermark Generation ............................. 143
6.2 Pilot Study on Hybrid watermarking method .................................... 144
6.2.1 The Design ................................................................................ 144
6.2.2 Analysis of Experimental Results .............................................. 147
6.2.3 Conclusion ................................................................................. 150
6.3 Comparative Analysis of Hybrid Watermark Implementations ........... 150
6.3.1 The Chosen Method ................................................................. 151
6.3.2 Experimental Results Analysis ................................................ 153
6.3.3 Conclusion ................................................................................. 156
6.4 Hybrid watermarking method Combining Robust and Semi-fragile
Watermarks ......................................................................................... 158
6.4.1 Our Watermarking Methods ....................................................... 158
6.4.2 Analysis of Experimental Results .............................................. 160
6.4.3 Conclusion ................................................................................. 164
6.5 Chapter summary ............................................................................ 164

Chapter 7 Conclusion and Future Work .............................................. 167
7.1 The Need for Watermarking ............................................................ 167
7.2 The Watermarking Problems ........................................................... 168
7.3 Summary of Original Contributions ................................................. 169
7.4 Limitations ..................................................................................... 172
7.5 Future Work .................................................................................. 172

Appendix A Sample Images of Various Attacks ..................................... 175
Appendix B Abstracts of Published Contributions ................................. 181
B.1 Recovery of Watermark Using Differential Affine Motion Estimation 181
B.2 Geometrically Robust Digital Image Watermarking using Scale Normalization and Flowline Curvature ................................................. 182
B.3 Performance Factors Analysis of a Wavelet-based Watermarking Method 182
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A sample image with a visible text watermark</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>A generic watermarking system</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Examples of PSNR inaccuracy for evaluating imperceptibility</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Distributions of watermark detection values</td>
<td>21</td>
</tr>
<tr>
<td>2.4</td>
<td>Contrasting examples of image contents</td>
<td>31</td>
</tr>
<tr>
<td>2.5</td>
<td>Sample of LPM distortion</td>
<td>32</td>
</tr>
<tr>
<td>2.6</td>
<td>Block-based localization</td>
<td>37</td>
</tr>
<tr>
<td>2.7</td>
<td>Sample-based localization</td>
<td>38</td>
</tr>
<tr>
<td>2.8</td>
<td>Wavelet-based localization</td>
<td>38</td>
</tr>
<tr>
<td>3.1</td>
<td>Examples of affine transformation</td>
<td>53</td>
</tr>
<tr>
<td>3.2</td>
<td>Visual appearance and reference grid of RBA</td>
<td>54</td>
</tr>
<tr>
<td>3.3</td>
<td>Images used in the motion estimation experiments</td>
<td>58</td>
</tr>
<tr>
<td>3.4</td>
<td>An image registration example</td>
<td>59</td>
</tr>
<tr>
<td>3.5</td>
<td>Effect of image registration on watermark detection</td>
<td>60,61</td>
</tr>
<tr>
<td>3.6</td>
<td>Effects of RBA bending factor on watermark detection</td>
<td>62,63</td>
</tr>
<tr>
<td>3.7</td>
<td>Effects of image registration on blind watermark detection</td>
<td>64</td>
</tr>
<tr>
<td>3.8</td>
<td>Examples of flowline curvature computation</td>
<td>69</td>
</tr>
<tr>
<td>3.9</td>
<td>Robust watermark embedding using flowline curvature</td>
<td>70</td>
</tr>
<tr>
<td>3.10</td>
<td>Two robust corners selected as reference points</td>
<td>71</td>
</tr>
<tr>
<td>3.11</td>
<td>Robust watermark detection using flowline curvature</td>
<td>72</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.12</td>
<td>Watermark detection of stego image without distortions</td>
<td>74</td>
</tr>
<tr>
<td>3.13</td>
<td>Watermark detection under rotational distortion</td>
<td>75</td>
</tr>
<tr>
<td>3.14</td>
<td>Watermark detection under translational distortions</td>
<td>76</td>
</tr>
<tr>
<td>3.15</td>
<td>Summary of detector responses under scaling distortions</td>
<td>77</td>
</tr>
<tr>
<td>3.16</td>
<td>Image synchronization for combined RST distortions</td>
<td>77</td>
</tr>
<tr>
<td>3.17</td>
<td>Samples of image cropping attacks</td>
<td>78</td>
</tr>
<tr>
<td>3.18</td>
<td>Robust corner detection samples</td>
<td>79</td>
</tr>
<tr>
<td>4.1</td>
<td>Watermark embedding using the three examined methods</td>
<td>87</td>
</tr>
<tr>
<td>4.2</td>
<td>Four-level DWT decomposition</td>
<td>88</td>
</tr>
<tr>
<td>4.3</td>
<td>Watermark and DWT sub-bands</td>
<td>92</td>
</tr>
<tr>
<td>4.4</td>
<td>Set of Baboon embedding results with their respective WPSNR</td>
<td>94</td>
</tr>
<tr>
<td>4.5</td>
<td>Set of Cameraman embedding results with their respective WPSNR</td>
<td>95</td>
</tr>
<tr>
<td>4.6</td>
<td>Set of Lena embedding results with their respective WPSNR</td>
<td>95</td>
</tr>
<tr>
<td>4.7</td>
<td>Set of Pepper embedding results with their respective WPSNR</td>
<td>96</td>
</tr>
<tr>
<td>4.8</td>
<td>Set of Fishing boat embedding results with their respective WPSNR</td>
<td>96</td>
</tr>
<tr>
<td>4.9</td>
<td>PSNR of stego images under different embedding methods</td>
<td>97</td>
</tr>
<tr>
<td>4.10</td>
<td>Samples of attacked images</td>
<td>98</td>
</tr>
<tr>
<td>4.11</td>
<td>The Dual Tree-Complex Wavelet Transform</td>
<td>103</td>
</tr>
<tr>
<td>4.12</td>
<td>The Fourier-Mellin transform</td>
<td>105</td>
</tr>
<tr>
<td>4.13</td>
<td>Consistent response of the DT-CWT</td>
<td>105</td>
</tr>
<tr>
<td>4.14</td>
<td>Consistent relative distance of the DT-CWT input and its wavelets</td>
<td>106</td>
</tr>
<tr>
<td>4.15</td>
<td>Invariant domain watermarking method</td>
<td>107</td>
</tr>
<tr>
<td>4.16</td>
<td>Comparison of rotation with cropping under different implementations</td>
<td>113</td>
</tr>
<tr>
<td>4.17</td>
<td>Lena attacked with a combination of RST and JPEG compression</td>
<td>114</td>
</tr>
<tr>
<td>4.18</td>
<td>Samples from image database</td>
<td>115</td>
</tr>
<tr>
<td>4.19</td>
<td>ROC for mass test</td>
<td>115</td>
</tr>
<tr>
<td>5.1</td>
<td>Histogram of a level-2 DWT horizontal sub-band</td>
<td>124</td>
</tr>
<tr>
<td>5.2</td>
<td>Overview of the semi-fragile watermarking method</td>
<td>125</td>
</tr>
<tr>
<td>5.3</td>
<td>The watermark signal for Lena</td>
<td>130</td>
</tr>
<tr>
<td>5.4</td>
<td>Watermark imperceptibility evaluation</td>
<td>131</td>
</tr>
<tr>
<td>5.5</td>
<td>Watermark detection process</td>
<td>132</td>
</tr>
<tr>
<td>5.6</td>
<td>Tamper localization and self-recovery</td>
<td>135</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Combined Geometrical Distortions</td>
<td>77</td>
</tr>
<tr>
<td>3.2 General Image Processing Tests</td>
<td>78</td>
</tr>
<tr>
<td>4.1 Attacks Used in Robustness Tests</td>
<td>93</td>
</tr>
<tr>
<td>4.2 Embedding Time of the Three Embedding Methods</td>
<td>93</td>
</tr>
<tr>
<td>4.3 Summary of Watermark Detection Result for All Levels of Attacks</td>
<td>98</td>
</tr>
<tr>
<td>4.4 Competency Comparisons of the Three Embedding Methods</td>
<td>99</td>
</tr>
<tr>
<td>4.5 Summary of DT-CWT, DWT, and FFT Properties</td>
<td>104</td>
</tr>
<tr>
<td>4.6 Robustness Attacks using StirMark</td>
<td>110</td>
</tr>
<tr>
<td>4.7 Average Results for Robustness Tests</td>
<td>111</td>
</tr>
<tr>
<td>5.1 Semi-fragile Authentication under Various Attacks</td>
<td>133</td>
</tr>
<tr>
<td>5.2 Alternative Semi-fragile Authentication and Threshold Selection</td>
<td>134</td>
</tr>
<tr>
<td>6.1 General Attacks on Fragile Watermark</td>
<td>148</td>
</tr>
<tr>
<td>6.2 Parameter Values for Non-overlap and overlap Implementations</td>
<td>153</td>
</tr>
<tr>
<td>6.3 Attacks for Robustness Evaluation</td>
<td>154</td>
</tr>
<tr>
<td>6.4 Comparison of Hybrid Watermarking Time</td>
<td>155</td>
</tr>
<tr>
<td>6.5 Effects of Block Size on Fragile Watermarking Time</td>
<td>157</td>
</tr>
<tr>
<td>6.6 Average Score for Robustness Tests</td>
<td>161</td>
</tr>
<tr>
<td>6.7 Semi-fragile Authentication and Threshold Selection</td>
<td>162</td>
</tr>
</tbody>
</table>
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D</td>
<td>One-Dimensional</td>
</tr>
<tr>
<td>2-D</td>
<td>Two-Dimensional</td>
</tr>
<tr>
<td>ACF</td>
<td>Auto-Correlation Function</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BTC</td>
<td>Block Truncation Code</td>
</tr>
<tr>
<td>CLT</td>
<td>Central Limit Theorem</td>
</tr>
<tr>
<td>CMYK</td>
<td>Cyan, Magenta, Yellow, 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>DRM</td>
<td>Digital Rights Management</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete Wavelet Transform</td>
</tr>
<tr>
<td>ECC</td>
<td>Error Correcting Code</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FM</td>
<td>Fourier-Mellin</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System</td>
</tr>
<tr>
<td>IA-W</td>
<td>Image-Adaptive Wavelet</td>
</tr>
<tr>
<td>IDWT</td>
<td>Inverse Discrete Wavelet Transform</td>
</tr>
<tr>
<td>IP</td>
<td>Intellectual Property</td>
</tr>
<tr>
<td>LPM</td>
<td>Log-Polar-Map</td>
</tr>
<tr>
<td>LSB</td>
<td>Least-Significant-Bit</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a Posteriori</td>
</tr>
<tr>
<td>ML</td>
<td>Maximum Likelihood</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Bit</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>NVF</td>
<td>Noise Visibility Function</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</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>RBA</td>
<td>Random Bending Attack</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, Green, Blue</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>ROI</td>
<td>Region Of Interest</td>
</tr>
<tr>
<td>RST</td>
<td>Rotation, Scaling, Translation</td>
</tr>
<tr>
<td>SARI</td>
<td>Self-Authentication-and-Recovery Image</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>SS</td>
<td>Spread Spectrum</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular Value Decomposition</td>
</tr>
<tr>
<td>VOD</td>
<td>Video On Demand</td>
</tr>
<tr>
<td>WPSNR</td>
<td>Weighted-Peak-Signal-to-Noise-Ratio</td>
</tr>
</tbody>
</table>
Declaration

The work contained in this thesis has not been previously submitted for a degree or diploma at any higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signed: . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Date: . . . . . . . . . . . . . . . . . . . . . . . .
Previously Published Materials

The following papers have been presented and published in refereed conferences, and contain material based on the content of this thesis:


Acknowledgements

Many parties have contributed to the completion of this thesis. First and foremost, I would like to thank my supervisors, Dr. Jiang Du and Prof. Binh Pham. They have given me a very rewarding research experience. Dr. Jiang Du has generously shared with me his technical expertise and software tools. He initiated the ideas of geometric invariant domain and hybrid watermarking, which are discussed in Chapter 4 and Chapter 6 respectively. In addition, Prof. Binh Pham has diligently guided me in improving my research and writing skills. I deeply appreciate their supervision, mentoring, and understanding.

Current and former staff at the Information Security Institute (ISI) have been very helpful. Prof. Edward Dawson, Prof. Colin Boyd, and Dr. Andrew Clark have been providing me valuable advice and support. Elizabeth Hansford and Christine Orme have been excellent administrators. Technical support given by Matt Bradford is very efficient. I would like to express my gratitude to Prof. Adrian, Dr. Ernest, Dr. Lauren May, Dr. Juanma, Dr. Kun, Dr. Yvonne, Dr. Jacob, Dr. John, Dr. Matt, Dr. Selwyn, Dr. Greg, and many others for the wonderful learning environment.

I am indebted to the University of Malaya (UM) for the award of this research scholarship. Especially, I would like to thank Prof. Zainab, Prof. Mashkuri, Assoc. Prof. Roziati, and Dr. Rukaini for their encouragement. In addition, Dr. Ling and Omar have offered useful advice and assistance. The Strategic Collaborative Grant on Digital Rights Management (DRM) awarded by QUT, Grant-in-Aid awarded by QUT, ISI funds, and conference grants
awarded by UM have enabled me to present our research outcomes in local and international conferences. The constructive criticisms of anonymous reviewers and conference participants have helped in enhancing our work.

The scrutiny and comments given by the panel members of my PhD defence seminar are invaluable and much appreciated. Therefore, I thank Dr. Jiang Du, Prof. Binh Pham, Dr. Frederic Maire, and Dr. Birgit Planitz.

Medical images were used in one part of this research. Special thanks to Queensland Health for providing the medical images. Dr Anthony Maeder and Dr. Birgit Planitz of e-Health Research Centre in Brisbane have helped to send over the images to me.

The friendship with my fellow students, some of whom have graduated, is unforgettable. An incomplete list with random order would include Dr. Raymond Choo, Dr. Riza Aditya, Dr. Ming Yao, Dr. Jason Reid, Dr. Yousef, Jaimee Brown, Ken Wong, Jason Smith, Mark Branagan, Andrew Marrington, Wenpoh, Alex, Robert, Jared, David, Rong, Steve, Rupinder, Bradley, Sam Zhu, Fabrina, Hamud, Sultan, Quan, Huseyin, and Reza. Additionally, the International Student Services (ISS) and Research Students Centre (RSC) at QUT have made my campus life very enjoyable. Furthermore, many Australian and international friends have helped me to adapt to the life in Brisbane. They include Kelvin, Huey Yee, Alvin, Dr. Linda, Dr. Yong, Dr. Sheng, Ming, Bew, Wai Kuan, and Mun Yee.

My parents, my brothers and their family, and my sister have been my source of strength and faith throughout these years. I must thank all of them for their unconditional love, courage, support, and understanding.

It would be an enormous list to include everyone whom I want to thank. Therefore, please accept my apology if I inadvertently left out your name in this limited space.
Chapter 1

Introduction

1.1 Motivations
Images make up a major component of multimedia content. Examples of images are
digital arts, illustrative diagrams, cultural heritage paintings in digitized form and
digital photographs. Advances in computing hardware, software, and networks have
created threats to copyright protection and content integrity. For instance, images can
be copied, modified, and distributed easily. Digital watermarking is a potentially
good tool in enabling content protection. Encryption can offer confidentiality and
integrity in content protection, and the decrypted content can be further protected
using digital watermarks. The watermarking process embeds a signal into the image
without significantly degrading its visual quality. Then the stego image can be made
public or sent to the end user. Later, the detected watermark can be used for the
purposes of copyright protection and content authentication.

Watermark robustness is one of the major characteristics that influence the
performance and applications of digital image watermarks. Robustness in this
context means the ability of a watermark to resist common image processing.
Watermarks can be categorized into three major groups based on their robustness:
robust, fragile, and semi-fragile watermarks. Robust watermarks should be detected
successfully in images that have been through manipulative distortions. Adversely, fragile watermarks are very sensitive and easily destroyed by image modifications. In the middle of both extreme ends are the semi-fragile watermarks. They can resist legitimate changes while being sensitive to severe tampering.

Copyright protection concerns the positive identification of content ownership in order to protect the rights of the owner. Robust watermarks can be used in copyright protection because they are persistently associated with an image. Attempts to remove the watermark should result in severe degradation of the image’s visual quality. The detection of a watermark in an image can be used to identify the copyright holder. On the other hand, content authentication is the validation of content integrity. This is an emerging field that does not require exact verification of numerical data values. Fragile watermarks are good at the strict level of integrity check. Semi-fragile watermarks are well-suited for content authentication. In this case, a detected watermark is compared with its original content to determine its integrity. Alternatively, semi-fragile watermarks can be applied in content authentication. In contrast to the robust watermark, the successful detection of a semi-fragile watermark indicates that the content has not been tampered with. Therefore, the content can be validated as authentic.

In this thesis, we investigate watermarking methods for copyright protection and content authentication. In addition, we also develop a “self-healing” capability in the semi-fragile watermarking method. This capability enables a modified image to recover its original content. Despite its potential application in media forensics, it is rarely found in existing watermarking methods. To provide an integrated solution in copyright protection and content authentication, we combine two watermarking methods into a hybrid method.

In this chapter, we will give an overview of digital watermarking for images. The classification of robust, fragile, and semi-fragile watermarks will be explained. Then, we will relate various types of watermarks to its applications. We will also describe the aims and objectives of this thesis, and define the scope of the study. Finally, we will describe the organization of this thesis and list our contributions.
1.2 Overview of Digital Watermarking

This section provides an overview of digital watermarking. It covers some basic terms, watermark properties, watermark classifications and its applications. A detailed discussion of these topics is given in Chapter 2.

The list below contains the meaning of standard terms used throughout this thesis.

- **Cover image** is the original image used in watermarking.
- **Stego image** is the cover image following watermark embedding.
- **Test image** is the possibly modified stego image from which the watermark is to be extracted.
- **Reference image** is the image used to assist watermark detection. It could be a cover image, a stego image, or a test image. It is normally used in image registration where watermark information is synchronized to ensure the accuracy in watermark extraction. The image registration process maps each object’s location in a distorted image to its corresponding object’s location the reference image, thus synchronizing the numerical representation of the images.
- **Watermark** can be a simple signal consists of a pseudo-random binary sequence, or a multi-bit message encoded in a transform domain. We will focus on the fundamental type of watermark signal in this thesis.
- **Watermark embedding** is the process of encoding a watermark signal (i.e. the watermark) into an image.
- **Watermark detection** is the process of uncovering a watermark hidden in an image. This process generally consists of a few steps, i.e. extraction of the watermark, decoding of the extracted message, and validation of the decoded information.
- **Blind watermark detection** is a watermark detection which does not require a reference image.
- **Watermark scheme** comprises the embedding and detection methods.
• **Distortions** are changes made to a stego image to evaluate its robustness. These changes could be lossy image compression, geometrical operations, and common image processing. Although the distortions are sometimes named *attacks*, they do not refer to malicious intents with the aim of evaluating the security aspects of a watermark.

To understand watermarking methods and determine their applications, one needs to know the properties of digital watermarks. Listed below are some fundamental watermark properties.

• **Robustness** of a watermark refers to its ability to withstand non-malicious distortions. For example, a robust watermark should be detectable following common image processing operations [1, 2].

• **Data payload** is the encoded message size of a watermark in an image. The simplest form of watermark has no data payload. It only gives a Yes/No answer in watermark detection to indicate the existence of watermark in an image. On the other hand, multi-bit watermarks can carry textual or pictorial information [3].

• **Capacity** is the amount of watermark information in an image. If multiple watermarks are embedded into an image, then the watermarking capacity of the image is the sum of all individual watermark’s data payload [3].

• **Imperceptibility** is the characteristic of hiding a watermark so that it does not degrade the visual quality of an image. A closely related term is fidelity. **Fidelity** is the visual similarity between the stego image and its cover image.

• **Security** of a watermark is the ability of the watermark to resist malicious attacks. These attacks include intentional operations of watermark insertion, modification, removal, and estimation which aim at defeating the purpose of the watermarks [1, 2].

• **Computational cost** is the measure of computing resources required to perform watermark embedding or detection processes. It can be measured using the processing time for a given computer configuration.
1.2 Overview of Digital Watermarking

There are several ways of classifying watermarking methods. One of the most widely adopted classifications is based on watermark robustness. Under this classification, watermark can be grouped into 3 types:

1. **Robust** watermarks are watermarks that can resist non-malicious distortions.
2. **Fragile** watermarks are easily destroyed by all image distortions.
3. **Semi-fragile** watermarks can be destroyed by certain types of distortions while resisting other minor changes.

Besides watermark robustness, watermarks can also be categorized into visible and invisible types. Visible watermarks are perceptible to a viewer. An example of such watermark is depicted in Figure 1.1. On the other hand, invisible watermarks are imperceptible and do not change the visual appearance of images. In this thesis, we are interested in invisible watermarks because they have a wider range of applications compared to visible watermarks. For example, invisible watermarks do not affect the aesthetic value of an image, and privacy infringement is less likely to happen given its obfuscation.

Figure 1.1 A sample image with a visible text watermark. The text “Brian Kell 2006” can be seen on the image center.
(Source: http://en.wikipedia.org/wiki/Digital_watermarking)
Application-wise, robust watermarks are suitable for copyright protection because they can resist common image processing operations. On the other hand, fragile watermarks can be used to detect tampering and authenticate an image because it is sensitive to changes. Semi-fragile watermarks are usually applied in some special cases of authentication and tamper detection. These cases may consider lossy image compression as legitimate changes while highlighting geometrical distortions as intentional attacks.

It should be noted that watermarks can be embedded and detected in different types of domains. The most direct approach is watermarking in the spatial domain where pixel values are modified to encode the watermark signal. Furthermore, frequency domains such as Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT) are widely used in image watermarking. Other domains include Discrete Wavelet Transform (DWT), Radon transform, fractals transform, chirp-Z transform, Hadamard transform, Singular Value Decomposition (SVD), and Fourier-Mellin (FM) transform.

1.3 Aims
The aims of this research are three-fold:

(i) To investigate the strength and limitations of current watermarking schemes,
(ii) To design and develop new schemes to overcome the limitations, and
(iii) To evaluate the new schemes using application scenarios of copyright protection, tamper detection and authentication.

Aimed at the three goals mentioned above, we will find answers to these research questions:

1. What are the major challenges in robust watermarking?
2. How can we reduce the computational cost of a robust watermarking method that is based on resynchronization approach?
3. Is the geometric invariant domain a better option for robust watermarking compared to resynchronization approach?
4. What are the new capabilities of semi-fragile watermarking?
5. Can semi-fragile watermarks offer self-authentication and self-recovery?
6. Is it possible to create a multi-purpose watermarking method for copyright protection, tamper detection, content authentication, and content recovery?

**Research scope**

Research activities in digital image watermarking have become more specialized. Therefore, it is important to identify the focus of study. In this thesis, we investigate robust, semi-fragile watermarking, and hybrid methods. In addition, we also examine hybrid methods that combine the advantages of robust and semi-fragile watermarks.

To preserve the visual appearance of images, we focus on invisible watermarks. The experiments are performed using greyscale images so as to focus on the fundamentals of data embedding. The developed watermarking methods can be easily ported to colour images given the similar pixel representation of both greyscale and colour images. For instance, RGB and YUV models are used in colour displays, and the CMYK model is applied in colour printing. Digital colour display consists of red, green, and blue (RGB) components. Colour prints are blended using cyan, magenta, yellow, and key (CMYK) components. The key component is usually black colour. Some television broadcasts use the YUV colour model. The YUV model has a luminance (Y component) and two chrominance (U and V components) components. Colour images displayed on computer screens consist of red, green, and blue (RGB) components. Therefore, we can apply the watermarking methods on the blue plane of a colour image since human eyes are least sensitive to changes in the blue component [4].

Applications that provide value-added services using image watermarks do not require high level of watermark security. For example, a watermark embedded in an image can be used to provide a link between printed information and web-based information. The printed image can be captured using a camera-phone, and the detected watermark is sent to a web server in order to retrieve extra information associated with the image. This technology could be useful in linking advertisements in printed magazines and time-sensitive materials on web servers. This strategy offers cross-media promotional coverage and dynamic content updates. In addition, given the emerging status of watermark security and the mature status of cryptographic security, we would suggest the adoption of cryptography in securing
watermark applications. Therefore, this thesis will focus on watermark robustness, and place less emphasis on watermark security. Watermark security has become a new branch of watermarking research as can be seen in recent academic conferences, e.g. Information Hiding 2006 (IH2006) and International Workshop on Digital Watermarking 2006 (IWDW2006).

We also need to consider trade-off between watermark properties that have conflicting characteristics, i.e. robustness, capacity, and imperceptibility. We also emphasized the computational efficiency of the algorithms.

1.4 Achievements and Contributions
The major outcomes of this research are: (i) The development of robust watermarking methods, (ii) The development of a semi-fragile watermarking method, and (iii) The development a hybrid watermark method combining robust and semi-fragile watermarks.

The research process started with a thorough literature survey on image watermarking methods for copyright protection and content authentication. The results are reported in Chapter 2 of this thesis. Several robust watermarking methods based on synchronization approach were studied. Firstly, we adopted a motion estimation technique developed by Periaswamy and Farid [5] in watermark synchronization. Secondly, we develop another watermark synchronization method utilising flowline curvature and scale normalization. These studies are discussed in Chapter 3. Then, we investigate robust watermarking using the invariant domain approach. We firstly study a blind watermark detection method reported in [6], and enhance its watermark embedding method to reduce computational cost. Following that, we develop a geometric invariant domain using a combination of transforms, and adapt our enhanced watermark embedding method into that domain. When developing the geometric invariant domain, we refer to the Fourier-Mellin (FM) framework as a guide. The results of invariant domain watermarking are discussed in Chapter 4. In Chapter 5, we created a new semi-fragile watermarking method with self-authentication and self-recovery capabilities. The semi-fragile watermarking method was solely carried out by the author with close supervision from its design to its implementation and evaluation. The final part of this research is described in
1.5 Thesis Outline

Chapter 6. It consists of a hybrid watermarking method that combines our work on robust watermarking and semi-fragile watermarking. We integrate our geometric invariant domain and our semi-fragile watermarking method into a single method.

Original contributions published are listed in Previous Published Materials of the Preliminaries section. There are 6 publications in total. They have been presented in refereed conferences and appeared in the corresponding proceedings. The contents of these publications are discussed in subsequent chapters of this thesis.

1.5 Thesis Outline

The remaining parts of this thesis are organized as follows. Chapter 2 provides literature reviews of the related topics. The chapter begin with discussions on watermark properties and its applications. Then, a review of image distortions that can threaten robust watermarks is presented. This is followed by a comparison of robust watermarking approaches. Following that, we will survey semi-fragile watermark applications and their implementation methods. Finally, an analysis is provided on the advantages and drawbacks of hybrid watermark methods. We also identify new features that can improve their practicality.

In Chapter 3, we will investigate robust watermarking methods that are based on synchronization. Firstly, a robust watermark that resists geometrical distortions at the global and local scale will be studied. It uses differential affine motion estimation to model image distortions as locally affine but globally smooth motions. Secondly, a new method that reduces computational cost significantly while resisting geometrical distortions will be examined. Its synchronization relies on scale normalization and flowline curvature.

To address the limitations experienced in synchronization-based approaches, we will switch to the geometric invariant domain for robust watermarking in Chapter 4. The chapter begins with an evaluation of performance factors for a blind watermark detection method. Blind watermark detection has high practical values because it does not require a reference image. After that, an invariant domain created using a combination of transforms is presented.
Chapter 5 will focus on semi-fragile watermarking. We will describe our novel method that offers content authentication, tamper localization, and approximate content recovery. The concepts of self-embedding, self-authentication, and self-recovery associated with the semi-fragile watermarking method will also be explained.

We will combine our robust and semi-fragile watermarks into a hybrid method and present it in Chapter 6. The complementary features of the robust and semi-fragile watermarks will make the hybrid method very useful in content protection. The advantages of the hybrid method compared to single watermark method will be described. The hybrid watermark method will be evaluated in using a digital media forensics scenario. Its advantages and limitations will be identified, and possible improvements will be suggested.

Conclusion will be drawn in Chapter 7. They will include our achievements in robust watermarking, semi-fragile watermarking, and hybrid watermark method. Future research directions will also be discussed. An appendix at the end of this thesis illustrates various levels of image distortion experimented in content authentication using our semi-fragile watermark.
Chapter 2

Digital Image Watermarking

This chapter gives a detailed explanation of digital watermarking, extending the brief overview of the subject in Section 1.2. Watermark properties and their applications will be covered in Sections 2.2 and 2.3 respectively. We will describe a few types of digital image watermarking: robust watermarking, semi-fragile watermarking, and hybrid watermarking. Robust and semi-fragile watermarks have contradicting properties, thus they are suitable for different applications. Robust watermarks which are typically suitable for copyright protection will be discussed in Section 2.4. On the other hand, semi-fragile watermarks are good for authentication, and will be reviewed in Section 2.5. By combining a robust watermark and a semi-fragile watermark into a single method, an integrated solution for digital content protection can be made. Section 2.6 is devoted to this topic.

2.1 Digital Watermarking

There are a lot of similarities between information hiding, steganography, and watermarking. Information hiding involves the concealment of information so that an observer does not know of its existence. Steganography generally means “covered
writing” where communications are carried out in secret. Watermarking is the embedding of content-dependent information. A hierarchical taxonomy can be made to relate these fields, i.e. information hiding covers both steganography and watermarking. This thesis concerns image watermarking, i.e. embedding of invisible watermarks in images.

An analogy of digital watermark is the paper watermark. Paper watermarks on currency notes and corporate letterheads are used to prove their authenticity. Similarly, digital watermark is embedded into digital media to validate their contents. Although cryptographic methods have long been applied in digital content security, the decrypted content requires further protection. For instance, a piece of artwork may be obtained legitimately but distributed to others unlawfully through peer sharing networks. Digital watermarks can provide extra protection to the decrypted content since it is embedded into the content.

Digital watermarking technologies started to mushroom in the past decades. This is evident through the exponential growth of academic publications in digital watermarking over the years. Some of those articles were published in top rank journals. For example, there are more than 100 watermarking papers in IEEE Transactions on Signal Processing and IEEE Transactions on Image Processing as of October 2006. Research activities in digital watermarking had matured to warrant the establishment of new conferences [7, 8] and new journals [9-11]. The scopes of these publications cover many interests in digital watermarking. They range from theoretical discussions to real-life applications. In addition, research topics are becoming more specialized, e.g. robust watermarking, fingerprinting, benchmarking, steganalysis, and security. Furthermore, watermarking technologies have been commercialized. For example, Digimarc [12] watermark was added into Adobe Photoshop [13] to enable embedding and detection of digital image watermarks. Epson [14] and Kodak [15] produced cameras with image watermarking capabilities.

Research in digital watermarking covers almost all media forms. Examples include audio, video, image, text, 3D model, and software codes. Digital watermarks are signals embedded imperceptibly into the media and can be detected under specific conditions. This thesis focuses on the watermarking of digital greyscale images. Greyscale images are the product of simple sampling where each pixel is assigned a value. Typical greyscale images used for experimentation in the research
community use 8-bits per pixel, thus each pixel has $2^8 = 256$ grey levels. A small number of researchers work on other image formats such as halftone and colour images. Halftone images are suited for printed media due to their binary appearance. Colour images usually consist of three colour channels, e.g. Red, Green, and Blue. Each channel is conceptually similar to greyscale.

Digital image watermarking schemes can be modelled as a communication process involving an embedder and a detector, as depicted in Figure 2.1. Firstly, a watermark signal is imperceptibly embedded into a cover image to produce a stego image. No extra space is required to store the signal. The stego image is then transmitted to the consumer. Distortions due to unintentional modification, malicious attacks, and data compression could occur during this process. Finally, a watermark detector is applied to determine whether the watermark exists in a possibly distorted image.

There have been many watermarking methods proposed by researchers over the years. For instance, watermark embedding can be implemented using additive, multiplicative, quantization or other encoding techniques. In addition, watermarking can be carried out in spatial or frequency domains. Details of these approaches will be discussed later in this thesis.

![Figure 2.1 A generic watermarking system](image)

To understand watermarking systems and determine its applications, one needs to know the properties of digital watermark.

### 2.2 Properties of Digital Watermarks

There are a few important properties associated with watermarking systems concerning digital images, and they are discussed here.
2.2.1 Robustness
Robustness of a watermark is its ability to resist non-malicious distortions. The distortions usually include common image processing, geometrical transforms, and image compression. For example, a watermark is said to be robust against JPEG compression if it can be detected after the image compression. Common image processing operations include noise insertion, contrast adjustment, smoothening, and cropping. Geometrical transforms include rotation, scaling, and translation. Although it is desirable to have watermarks that are robust against all possible distortions, real life applications may only require a subset of the robustness. For instance, images may be archived in databases as compressed format. This application would require those watermarks robust against high quality image compression. However, low quality compression that degrades their visual appearance significantly is not relevant. In other words, robust watermarks normally do not have to cater for extreme conditions. Under these conditions, severe distortions on the image quality will degrade the value of the images.

Among the many types of distortions mentioned, geometrical distortions remain a major challenge in robust watermarking. Geometrical distortions can be carried easily using off-the-shelf image processing software and defeat the purpose of watermarks by making them undetectable. They can cause serious damage to the watermark information through desynchronization effects. Most of the geometrical distortions can be modelled as combinations of three basic transforms: rotation, scaling, and translation (RST). Therefore, much research in robust watermarking have been focusing on geometrical robustness particularly RST robustness.

A watermark can be classified as robust, fragile or semi-fragile depending on its ability to resist distortions. Robust watermarks are normally designed to survive unintentional changes caused by common image processing. For example, unintentional changes may include image smoothening. Early digital watermarking methods embed the watermark into the spatial or transform domain of an image without considering perceptually significant features. New generations of watermarking methods take into consideration the image content and its visual features [16]. This improvement enables higher robustness against geometrical manipulations. More discussions on robust watermarking are presented in Section
2.2 Properties of Digital Watermarks

2.4. On the other hand, fragile watermarks are easily destroyed by slight distortions. The absence of a fragile watermark indicates that changes have been made to the image in which it was originally embedded. Lying in the grey area between the two extremes of robust and fragile watermarks is the semi-fragile watermark. Semi-fragile watermarks have partial characteristics of robust and fragile watermarks. For example, a semi-fragile watermark can be destroyed by image size reduction while being detected after image compression.

2.2.2 Capacity and data payload
The number of watermark bits encoded in a message is the data payload [17], and the maximum repetition of data payload within an image is the watermark capacity. The simplest form of watermarks is the one-bit watermark. (Some researchers prefer to name it zero-bit watermark [17]). In this case, the watermark detector will have 2 possible outputs: “watermark detected” and “watermark not detected”, which are comparable to a simple Yes/No answer. Depending on the application, some watermarking methods require a data payload exceeding 10,000 bits. A watermark may have high capacity but low data payload. For example, we can have a one-bit watermark embedded many times across the image. Determining the upper bound of watermark capacity has attracted the attention of some researchers. This could become a branch of watermarking research given the increase number of publications in this area [18]. However, the interests of this thesis are watermark robustness and integrated content protection.

2.2.3 Imperceptibility
It is normally preferred to have stego images that are perceptually similar to the cover image. Otherwise, the distortions in the stego images caused by watermark embedding would degrade its aesthetic value. Furthermore, they may cause suspicions and jeopardize watermark security. This property is named the imperceptibility of a watermark [2]. It is sometimes called fidelity or perceptual transparency. Human Visual System (HVS) models can be applied during watermark embedding to enhance watermark imperceptibility and robustness. The model specifies that the visual system of human eyes has certain characteristics. The eyes are less sensitive to changes made in highly textured regions compared to flat regions. The textured regions have complex patterns whereas the flat regions are
monotonous. Using the HVS model, a bigger watermarking weight can be used in an additive embedding for image regions that have complex textures compared to those regions with simple textures. The result of increasing the embedding weight would be enhanced watermark robustness.

To evaluate the imperceptibility among watermarking methods, a large number of images should be tested. Due to the huge effort, long time, and high costs of human-based evaluation, an automated measurement of imperceptibility is usually employed. For this, Peak-Signal-to-Noise-Ratio (PSNR) is generally deployed for comparing imperceptibility performance although it is not a perfect metric. The more similar between a stego image and its cover image, the higher is its PSNR. However, a stego image may have high PSNR despite obvious perceptual distortions. The opposite situation is also possible. Figure 2.2 illustrates these cases with the Pepper image obtained from the University of Southern California image database at http://sipi.usc.edu/database/. Comparing the image in Figure 2.2 (b) to the image in Figure 2.2 (a), an obvious black dot near the center of the image does not lower the PSNR value very much because the artifact is small compared to the whole image. Image Figure 2.2 (c) has low PSNR value because the changes are made over all regions. This means PSNR does not model perceptual similarity accurately. Unfortunately, a better perceptual model for imperceptibility measurement is yet to appear in the literature. In the watermarking community, it is generally agreed that minimum PSNR of 38dB is acceptable. Assuming an image with 8-bit greyscale, the PSNR [19] of a stego image compared to its cover image is

\[
\text{PSNR} = 20 \log_{10} \left( \frac{I_{\text{MAX}}}{\text{RMSE}} \right) \tag{2.1}
\]

where \(I_{\text{MAX}}\) is the maximum gray levels of the image. In this case, \(I_{\text{MAX}}\) can have a maximum value of 255. RMSE is the root mean square error given by

\[
\text{RMSE} = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left( f(m,n) - \tilde{f}(m,n) \right)^2} \tag{2.2}
\]

where \(\tilde{f}(m,n)\) is the stego image and \(f(m,n)\) is the cover image. An alternative calculation of PSNR is

\[
\text{PSNR} = 10 \log_{10} \left( \frac{(I_{\text{MAX}})^2}{\text{MSE}} \right) \tag{2.3}
\]
2.2 Properties of Digital Watermarks

where MSE is the mean square error given by

\[
\text{MSE} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[ \tilde{f}(m,n) - f(m,n) \right]^2.
\]  

Weighted-PSNR (WPSNR) was proposed to improve the accuracy of measuring similarity between images [19, 20]. However, it is not widely used in the watermarking community. Using the same notation as above, the computation of WPSNR is given by:

\[
\text{WPSNR} = 10 \log_{10} \left( \frac{(I_{\text{MAX}})^2}{\text{NVF}(\tilde{f}(m,n) - f(m,n))^2} \right)
\]

where NVF is the Noise Visibility Function (NVF). The NVF [21] for each pixel \((n_1, n_2)\) is computed using the DWT sub-bands \(x_{k,l}\):

\[
\text{NVF}_{k,l}(n_1, n_2) = \frac{x_{k,l}(n_1, n_2)}{x_{k,l}(n_1, n_2) + \sigma_{x_{k,l}}^2}
\]

where \(\sigma_{x_{k,l}}^2\) is the global variance of wavelet coefficients in the sub-band \((k,l)\).

Figure 2.2 Examples of PSNR inaccuracy for evaluating watermark imperceptibility. (a) Cover image, (b) Test image with high PSNR (39.59dB) despite an obvious dot near the center of the image, (c) Test image with low PSNR (21.37dB) after a circular row shift.

2.2.4 Security

A secured watermarking method can resist many hostile attacks that try to defeat the watermark’s purpose [17]. These attacks can be unauthorized operations in watermark removal, embedding, modification, and detection (estimation). However, some applications may only need a low level of security. For instance, a watermark that offers tamper localization and content recovery is unlikely to be attacked. In this
case, the enhanced feature of restoring corrupted image regions has added value to the user. Therefore, the user has no incentive to destroy the watermark. The watermark information can be used to correct error bits in images arising from scratches on CDROM surfaces where the images are stored. Tamper localization in an image is the identification of modified regions within the image. Content recovery refers to the restoration of the original image content in a tampered image.

The most direct approach to securing watermark information is to apply Kerckhoffs’ principle as in cryptography. The principle states that the security of a system should rely on a key instead of obscuring the watermarking algorithm. In other words, the algorithm can be known to everyone including an adversary. However, only the authorized use who has the key can reveal the secured watermark. It should be very hard for the adversary to “unlock” the secured watermark without knowing the correct key. In this approach, a watermark is generated using some context information such as the image size and content digest. Then, it is encrypted by a secret key. The secured watermark is then embedded into an image and sent to a receiver. The receiver needs to extract the watermark and decrypt it using the correct key in order to obtain the watermark information. This approach is good at combating unauthorized watermark detection and modification because an adversary cannot read the encrypted watermark information without using the correct decryption key. However, this approach does not prevent unauthorized watermark embedding and removal. For example, the adversary does not need to know the watermark information when performing a collusion attack to remove the watermark. In the collusion attack, several copies of the stego image with different secret keys are used to average out the watermark information.

It should be noted that security is not the same as robustness in watermarking. A robust watermark can survive common image processing, but may not be secure against malicious tampering. This thesis focuses on robustness instead of security. Having mentioned that, it is sometimes confusing to find that “attacks” in the literature could be used in the contexts of watermark robustness and watermark security interchangeably.
2.2 Properties of Digital Watermarks

2.2.5 Computational cost

Watermarking methods with highly complex algorithms will incur more computational costs compared to those with low complexity. Although the processing speed and memory size of consumer equipment have been upgraded throughout the years, algorithm complexity has made applications more resource-hungry.

Computation simplicity is still preferred in resource-limited environments such as mobile devices. Currently, applications in mobile devices have to find a balance between battery power consumption, bandwidth usage, memory allocation and many other factors. Extension of image watermarking into video frame watermarking may also require low complexity algorithms. Watermark detection steps that execute fast enough would ensure smooth transition from one frame to another in real time. If the watermark detection is too complex, then it will affect the practical values of video frame watermarking.

Evaluation of computational cost can be done by measuring the execution time of watermark embedding and detection steps using minimally configured systems. For example, desktop watermarking applications can be executed using a personal computer (PC) with Intel Pentium III processor and 256 megabytes of random access memory (RAM). In addition, the watermarking applications should occupy a small harddisk space if they are implemented in computer software.

2.2.6 Watermark detection reliability

To model robust watermarking in a copyright protection scenario, we can use a watermark that consists of a pseudo-random binary sequence to represent the identity of a copyright holder. The correlation value between the identity and a correctly detected watermark is usually very high compared to the correlation value between the identity and a randomly chosen watermark. In this case, a graph of correlation values plotted against watermarks has a significant peak at the correctly detected watermark which corresponds to the copyright holder’s identity. An example of this graph is shown by Figure 3.5 (b) in Chapter 3.

The situation described above is a simple outcome of watermark detection. A more complete consideration would include false positive, false negative, true
positive, true negative, and Receiver Operating Characteristic (ROC). The following paragraphs explain those terms.

For a given image with a watermark embedded, there are 2 possible results of its watermark detection:

- The successful detection of the watermark is called a **true positive**.
- The unsuccessful detection of the watermark is called a **false negative**.

Likewise, for a given cover image (or un-watermarked test image), there are 2 possible results of its watermark detection:

- The absence of a watermark is called a **true negative**.
- An incorrectly detected watermark causes a **false positive** (a.k.a. false alarm).

The false positive probability, $P_f$ is the chance that a false positive condition will happen, and it is often used to determine the performance of a watermarking method. Depending on the application, robust watermarking usually requires the $P_f$ between $10^{-6}$ and $10^{-12}$ [17].

There is a trade-off between false positive rate and false negative rate because they are interrelated. For example, a reduction in false negative rate would cause an increase in false positive rate. ROC curve can be used to show the relationship between the probability of false positive and the probability of true positive. In order to plot the ROC curve, we need to have the distributions of cover and stego images. By changing the detection threshold, its corresponding false positive rate and false negative rate can be computed. Figure 2.3 illustrates the distributions of watermark detection values for cover and stego images. The lightly shaded area is the probability of false positive. The sum of the lightly shaded and heavily shaded areas is the probability of true positive. Following that, the data points of false positive and false negative rates is used to plot the ROC curve. Figure 4.19 in Chapter 4 shows a ROC.

Ideally, the two distributions should be far apart so that the watermark detection has low false positive. However, real life cases may have the opposite outcomes. In addition, a large number of images may be needed in an experiment to obtain enough data points for the distributions. Therefore, theoretical models may be used to estimate the data points, and certain criteria may be needed to constraint a preferred condition. For instance, the Neyman-Pearson criterion applied in watermarking
detection maximizes the probability of watermark detection while ensuring its false positive probability does not exceed a selected value [22].

![Detection Diagram](image)

Figure 2.3 Distributions of watermark detection values for cover and stego images. The lightly shaded area is the probability of false positive. The sum of the lightly shaded and heavily shaded areas is the probability of true positive.

### 2.2.7 Blind detection of watermark

Blind detection is the detection of a watermark without a reference image. The reference image can be the cover image, a stego image with different watermark, or a non-distorted stego image. To become more practical, a watermarking method should not rely on the availability of a reference image. It should provide blind watermark detection using only the image under test. In other words, we can detect a watermark using only a test image in a blind watermark detection. The watermark detection would take the test image as input, execute an algorithm for the detection, and output the detected watermark. On the other hand, a non-blind watermark detection is similar to a blind watermark detection except that it requires a reference image. The problem is that the reference image may not be readily available.

Informed detection is in opposition to blind detection. In informed watermark detection, the detector must have access to the cover image. This requirement limits the applications of informed watermark detection.

### 2.2.8 Trade-off between performance factors

A basic principle of watermarking is to exploit redundancy in images for embedding the watermark information. Given the fact that many of the existing image compression algorithms are not perfect, watermarking is made possible by embedding extra information in the redundant parts. In addition, enhancing watermark robustness normally requires more image distortions and increased
redundancy. This causes lower imperceptibility and more likely to be removed under malicious attacks.

Many of the watermark properties explained above have conflicting characteristics. For example, increasing the robustness of a watermark would normally lower its imperceptibility due to the higher watermark energy imposed on the cover image. In addition, higher capacity would compromise its imperceptibility because more modifications the cover image are needed to embed the watermark. Therefore, designing a watermarking method usually requires finding a balance among these conflicting factors.

From the viewpoints of watermark robustness and security, there exist many types of attacks. We will categorize these attacks on watermark robustness in Section 2.4.1.

The requirements regarding each of the watermark properties are application-specific. For example, watermarking medical images require a high level of imperceptibility to avoid misjudgement during diagnosis. On the other hand, watermarking artistic pictures for copyright protection might give more attention to its robustness compared to its imperceptibility. These requirements for some application areas are described in the next subsection.

2.3 Applications of Digital Watermark

Digital watermarking technologies have been proposed to be implemented in many applications. Instead of compiling an exhaustive list of digital watermark applications, we describe some major groups of its applications.

2.3.1 Digital Rights Management

Digital Rights Management (DRM) can be defined as “the description, identification, trading, protecting, monitoring, and tracking of all forms of usages over tangible and intangible assets” [23]. It concerns the management of digital rights and the enforcement of rights digitally.

Many factors have contributed to the rise of DRM because they pose a threat to the protection of digital rights. The list below gives some examples.

- The increase amount of digitized content due to technological advancement, e.g. digital photographs, electronic books, video on demand (VOD), downloadable music.
The advances in computer networking technologies created new channels for content distribution in huge quantity and quick manner.

The sophistication of software functionalities enables end users to manipulate digital contents easily.

DRM is required to support these changes and control rights to purchase, consume, edit, store, and distribute digital contents. For example, DRM systems can control access, usage, and distribution of digital contents.

DRM systems have three major components, i.e. the enabling technologies, the business model, and the legislative framework [23]. Concerning the technological implementations, DRM systems are normally used to protect the rights of an intellectual property (IP) holder through copyright protection measures. This protection is necessary to thwart mass reproduction of illegal copies. In this case, watermarking is a tool to secure the digital contents. The embedded watermark remains associated with the content wherever it is distributed and duplicated. Furthermore, watermarks applied in DRM systems also enable copyright protection, copy protection, device control, authentication, and tamper detection. The technological implementation involves watermarking software, hardware and protocols.

Although there are some free media (i.e. MP3 songs, electronic books, and digital videos) on the web, DRM is still an intensely researched area. The free media are usually of lower quality compared to commercial media. For instance, Google Book Search [24] only provides limited number of pages for certain books. Most valuable publications are still protected and not freely available to the public.

There remain a number of open problems in DRM. User acceptance, user privacy, and user friendliness issues of DRM systems are yet to be solved. For instance, users would not like to be tracked during every consumption of digital content. In addition, major content providers and distributors need to adopt a set of standards so that the DRM enabling hardware and software can interoperate. Moreover, watermarking methods need to find a balance between robustness, imperceptibility and computational cost to be practical. More research is required to overcome these obstacles.
2.3.2 Copyright protection
Copyright protection is an important application of digital watermarking. It enables the identification of the copyright holder and thus protects his or her rights in content distribution. Robust watermarks are embedded into an image to protect the rights of the owner. It should be possible to detect the watermark despite common image processing, geometrical distortions, image compression, and many other types of image manipulations. Therefore, deliberate removal of the robust watermark should result in severe degradation of the image’s visual appearance. The successful detection of the watermark can positively identify the owner.

Watermarking is a better option than visible tags of copyright information. For example, a file header with copyright statement can be the target of attack. Robust watermarks, on the other hand, cannot be easily removed from the image without severely degrading its image quality. Therefore, it is suitable for copyright protection and owner identification.

2.3.3 Authentication
Authentication in watermarking should not be confused with authentication in cryptography. While authentication in cryptography means the verification of a message origin or proving the identity of a person [25], authentication in image watermarking refers to the integrity assurance of the image [26]. An image is said to be authentic if it has not been modified. The integrity check using watermark is advantageous compared to cryptographic hash and cryptographic signatures. Firstly, the embedded watermark stays with the image and cannot be removed easily. Secondly, extra space is not required to store the watermark information. Authentication of digital images can be useful in insurance claims by ensuring trustworthy photographs for court evidence. Other reported applications related to image authentication are the validation of cultural heritage paintings, medical records and digital artworks.

To determine whether an image is authentic, either robust watermarks or fragile watermarks can be applied. For instance, information extracted from a robust watermark can be compared with the image features to evaluate its integrity while the absence of a fragile watermark in watermark detection indicates that the image has been changed.
2.3 Applications of Digital Watermark

An emerging field of study in authentication is content authentication (a.k.a. soft authentication, selective authentication). Traditional authentication methods (a.k.a. hard authentication, exact authentication) aim at determining whether an image is 100% authentic or otherwise. However, images that have undergone high quality lossy compression to reduce storage space should be considered authentic in some applications. This requirement has promoted the study of content authentication. Semi-fragile watermarks are suitable for this task because they can tolerate legitimate changes and highlight significant manipulations.

2.3.4 Tamper detection and localization
Tamper detection is used to disclose alterations made onto an image. It is closely related to authentication. If tampering is detected in an image, then the image is considered unauthentic. Tamper localization enables further investigation of an act of tampering by identifying the tampered regions within the image. This information can assist in media forensics. For example, the severity of the tampering and the motives behind it can be established. Similar to authentication, tamper detection and localization can be achieved using robust, fragile, or semi-fragile watermarks according to the applications.

2.3.5 Annotation and privacy control
Multi-bit watermarking can be used to annotate an image. For example, patient records and imaging details related to a medical image can be carefully inserted into the image. This would not only reduce storage space but also provides a tight link between the image and its details. Patient privacy is simply controlled by not keeping the sensitive information as clear text in human readable form, and the watermark can be further secured by encryption. Other usages of annotation watermarking are electronic document indexing and automated information retrieval. In these cases, the watermark information serves as indices and keywords. Imperceptibility is very important in these cases because the images carry vital information for medical diagnosis. Robustness may not be relevant here if the watermarking system resides in a secured and closed environment. Reversible watermarking (a.k.a. erasable watermarking, lossless watermarking) is preferable for such an application since the changes introduced by watermark embedding can be removed. The original image can be accurately restored with the reversal of the embedded watermark.
Selective encryption is a cryptographic method that encrypts selected parts of an image so that they appear as unintelligible noise patterns [27]. It is a rather new method in information security that focus on multimedia data. It reduces the computational cost by eliminating the need to encrypt the whole image. Although selective encryption can be applied to randomise the private information in medical images, the encrypted regions could invite curious users to breach patient privacy by cracking trials. For non-medical images, a simple attack would be cropping off the encrypted regions, provided the rest of the image regions have useful values. On the other hand, the invisible watermarks add a layer of privacy protection by obscurity.

### 2.3.6 Media forensics

Media forensics involves the investigation of digital data in order to unveil scientifically valid information for court evidence. The deleted and hidden data are usually discovered using digital tools [28]. Media forensics has become an important research area due to many factors. For example, the increased number of cyber crimes, the profit loss due to piracy and fraud, and the need for law enforcement.

The application of digital watermarks in media forensics include the trustworthy digital camera [29-32], traitor tracing, transaction tracking, and content recovery. The secure digital camera proposed by Fridrich [29] uses reversible watermarking to embed forensics information into an image in order to qualify as court evidence. The photographer’s iris image, the scene image’s hash value, date, time, and other information are embedded as invisible watermarks into the images taken by the secure camera. Besides that, multimedia fingerprinting methods have been studied extensively at the University of Maryland for traitor tracing [33]. The fingerprint in this context is a specialised watermark with unique keys that enable the identification of the media source. Robustness and security are two essential requirements in fingerprinting in order to overcome malicious attacks. In addition, telltale watermark [34] and self-recovery watermark [35, 36] can specify changes made to an image.

These watermarking methods have some advantages over statistical methods in forensics investigation. For example, the statistical method used to classify images into natural and synthetic classes [37] requires a large training set with a diverse range of contents. Watermarking methods are more straightforward in such authentication of natural images. For example, detecting fragile watermarks can
validate the natural images. Therefore, watermarking eliminates the resource-intensive requirements of training and large data sets. Moreover, tamper detection of re-sampled images based on image statistics only work well when the images has little compression [38]. In contrast, semi-fragile watermarks can offer the same functions in tamper localization for a wider range of compression quality.

2.3.7 Other applications
There are many other applications where digital watermarking methods have been proposed as a technology-enabling tool. Some of them proved to be useful, while others were discarded because they were impractical. Some examples are listed below [26].

- Broadcast monitoring – watermarks embedded into advertisement sections of the broadcasts. It is a cost effective means of monitoring advertisement airtime on television and radio broadcasts. It is format independent and does not consume extra bandwidth. The practicality of this application may be limited by the watermark imperceptibility.

- Device control – watermarks embedded into radio and television signals can be used to control some features of a receiver. This has been proven to be a practical usage of watermarks [26].

- Copy control – watermarks detected in a video content are used to control the recording functionality of a “watermark-compliant recorder”. This was not practical because it involves user acceptance, manufacturer compliance, and enforcement of new laws.

- Communication enhancement – watermarks extracted are used to repair error bits in transmission. Hence, it saves time, costs and bandwidth for re-transmission [39].

Depending on the application, a single watermark or multiple watermarks may be used in a system to serve the system’s purpose. For instance, copyright protection may require a robust watermark, and media forensics may need a combination of robust and fragile watermarks.

To accomplish the objectives of this research, we reviewed both single watermarking methods and hybrid watermarking methods in the literature. The single
watermarking methods cover robust and semi-fragile watermarks. The hybrid watermarking methods include combinations of robust and “non-robust” watermarks. The non-robust watermarks refer to either fragile or semi-fragile watermarks. Such combinations are chosen for the hybrid watermarks because they have conflicting properties, and the combination could provide complimentary functions. The review of robust watermarking is covered in the next section.

2.4 Robust Image Watermarking
Robust watermarks can be used in many applications, e.g. copyright protection. To serve its purpose, a robust watermark should survive common image processing and other non-hostile operations. These modifications on the image are sometimes referred to as “attacks” although they are not the same as hostile attacks targeting watermark security.

2.4.1 Attacks on watermark robustness
Nowadays many attacks on robust watermarks can be performed easily using off-the-shelf image processing software. These attacks can cause removal or desynchronization of watermark information while maintaining the visual quality of the image. Although the appearance of the attacked image does not suffer serious changes, the attack will result in watermark detection failure. The common attacks on watermark robustness can be loosely categorized as follows.

(a) Image degradation
Image degradation operations can damage robust watermarks by removing parts of the image. The image parts that were discarded or replaced may carry watermark information. Examples of these operations are noise insertion, partial cropping, row removal, and column removal. Gaussian noise insertion is a type of signal processing operation. The amount of noise is controlled by its mean and variance. Cropping, row removal, and column removal represent data reduction attack.

(b) Image enhancement
These are generally convolution operations that could desynchronize watermark information in an image. For example, sharpening, histogram equalization, smoothing, median filtering, Gaussian filtering, and contrast enhancement. Median
filtering is a type of non-linear filtering that produces a “smoother” image. Contrast adjustment is part of signal enhancement manipulation. It can be used to change the appearance of an image to be “brighter” or “darker”.

(c) **Image compression**

Image compression is very useful in reducing storage space and thus saving transmission costs. Images processed using high quality compression can retain their aesthetic value, yet reduce their file size. Lossy compression methods are more destructive compared to lossless compression methods. Watermark information can be recovered with an inverse operation if it is losslessly compressed. However, lossy compression such as JPEG and JPEG2000 compressions involve irreversible changes to the image. Therefore, watermark information may be lost and recovery may not be possible. JPEG compression is one of the common compression attacks on digital images. With JPEG compression, one makes a trade-off between image quality and file size by specifying its compression qualities.

(d) **Image transformations**

Image transformations can be useful in restoring distorted images. However, they pose a severe threat to robust watermarks due to their desynchronization effects. Indeed, they are one of the major challenges in robust watermarking. Many types of linear transformations can be modelled as a combination of basic transforms. For example, geometric transforms can be constructed using rotation, scaling, and translation (RST). These transforms are generally known as RST transforms. Some researchers focus on RST robustness because it is the fundamental problem. A broader class of image transformations is affine transformation. It includes other transforms such as aspect ratio change, shearing, reflection, and projection. Global geometrical distortion such as rotation is a big challenge. A small degree of rotation usually retains visual appearance while damaging watermark information. Normally, correlation-based watermark detection is vulnerable to such attack.

One of the vital issues concerning image transformation is global versus local modifications. Early watermarking methods assumed that images are attacked at the global scale. Therefore, new types of attacks targeting local regions do not fit into the transformation model. Random Bending Attack (RBA) in *StirMark* is perhaps the most popular local transformation attack on robust watermarking.
2.4.2 Approaches to robust watermarking
In order to resist the attacks mentioned above, many approaches to robust watermarking have been investigated throughout the years. Some of them are implemented in the spatial domain while others utilize the frequency domains. The frequency domains include DCT, DFT, DWT, and many others. Robust watermarks can also be implemented using a combination of domains. Without considering the domains involved, robust watermarking approaches can be broadly classified as follows.

(a) Redundant information
Block-based watermarking methods divide an image into non-overlapping blocks and process each block separately. These methods are also known as tiling, because the blocks resemble tiles. Their security can be enhanced with local contextual information. This is often carried out using overlapping blocks or linking neighbouring blocks. Obviously, these methods could survive cropping attack.

Another set of methods that exploit the advantages of redundancy is based on Spread Spectrum (SS) coding. They are mostly robust to common image processing because the watermark information is spread across many frequencies. The watermark can be robustly detected using correlation values.

(b) Synchronization
To combat the major threat of geometrical distortions, many robust watermarking methods were developed to synchronize the watermark information, so as to increase the success rate of watermark detection. The concept here is the same as image registration. Some of them use robust features for synchronization in the detection stage. For instance, Bas [40, 41] calculates perceptually significant points (robust corners) in an image and link them using triangular tessellation for resynchronization prior to watermark extraction. In another example [42], robust coefficients (robust moments) are used for resynchronization. Some of these methods are called informed embedding because the watermark embedding decisions are made using the information of image contents. A major threat to these methods is the attack on the robust features. For example, the robust corners could be cropped-off. In addition, images without distinctive contents may not have robust features. Figure 2.4
2.4 Robust Image Watermarking

illustrates examples of two contrasting cases (online at http://sipi.usc.edu/database/).
The *Fishing boat* image has sharp and clear edges, e.g. the poles on the boat can be clearly distinguished from the sky and the clouds. Conversely, the *Straw* image has no distinctive contents.

![Fishing boat and Straw images](image)

Figure 2.4 Contrasting examples of image contents (a) *Fishing boat* image has sharp and clear edges, (b) *Straw* image has no distinctive contents

Besides the robust features, some robust watermarking methods insert a template watermark (a.k.a. reference watermark) for synchronization at the watermark detector. These methods have several drawbacks. Firstly, the template watermark will compromise the available space for embedding the information-carrying watermark. Secondly, the extra distortions caused by embedding the template watermark could degrade the visual quality of the stego image. Thirdly, the template itself could be the target of a hostile attack. Synchronization would fail if the template is removed by the attacker.

An improvement on the template-based method is to combine the information-carrying watermark with the template watermark [21]. This autocorrelation method would overcome the dilemma of balancing the capacity, imperceptibility, and detection success rate of both watermarks. However, the combined watermark could still be targeted and removed, because it has the characteristics of a template [43].

Instead of performing resynchronization at the watermark detection stage, some robust watermarking methods anticipate possible distortions at the watermark embedding stage. The watermark is embedded in the inversion of the distorted domain. Distortions would be cancelled out at the watermark detection stage. These methods may require iterative embedding to ensure that the watermark is effectively embedded. Therefore, they might not be efficient in controlling computational costs.
The robust watermarking methods based on synchronization unavoidably suffer from image imperceptibility loss due to interpolation errors during the image registration step.

(c) Invariant domain
Invariant domain methods were created to eliminate the need for synchronization. They rely on the invariant properties of the transformed domain to resist distortions. They are advantageous compared to the synchronization approach in the sense that they are independent of image features. In addition, blind watermark detection can be implemented easily because there no need for synchronization. One of the most studied methods [44] constructs a RST-invariant domain using a Fast-Fourier Transform (FFT) and a Log-Polar-Map (LPM). It consists of a series of FFT-LPM-FFT transformations. The magnitude component of the first FFT has shift-invariance. Taking this component for a LPM converts rotation and scaling into linear shift. This linear shift is then made invariant with the magnitude component of second FFT. Therefore, the final output of the transforms has RST invariance and can survive most geometrical distortions. However, this framework falls short on visual quality and computational costs. The visual quality of the stego image may degrade significantly due to the interpolation errors in FFT and LPM. Figure 2.5 illustrate a sample of distortion caused by LPM.

![Figure 2.5 Sample of LPM distortion](image)

(d) Combined approaches
Some robust watermarks are implemented using a combination of the approaches discussed. For example, the robust watermark in Deguillaume’s method [21] is a
2.4 Robust Image Watermarking

block-based method with autocorrelation in resynchronization. The watermark message is encoded using Error Correcting Code (ECC) so that transmission error can be detected and corrected. The resynchronization can combat global and local distortions. Then, the robust watermark can be extracted from each block of the resynchronized image, and the watermark message reliably decoded. By this way, the watermark can achieve very high robustness under many attacks.

2.4.3 Practical issues

Given the varying degrees of possible distortions and huge number of possible combined modifications, brute-force search for robust watermarking is not always practical. For instance, a small image of 16×16 pixels with 8-bit greyscale have a maximum of \(256^{16\times16} = 256^{256} = 3.23\times10^{616}\) possible outcomes. Therefore, the computational cost would be prohibitive for an exhaustive search. Even with carefully selected distortion parameters, the reduced search space would not be computationally practical.

Most of the robust watermarking methods embed the watermark into their middle frequencies to trade-off robustness with imperceptibility. If the watermark is embedded into the low frequency coefficients, then visual distortions could be observed in the corresponding flat regions of the image. On the other hand, if the watermark is embedded into the high frequency coefficients, then the corresponding edges in the image is changed, and its robustness can be easily defeated with image smoothening. In addition, increasing the robustness with higher embedding energy will also cause more distortions and hence degrade its imperceptibility.

It is generally agreed that there is no “one size fits all” in robust watermarking. Despite the difficulty in designing robust watermarks that can defeat all possible attacks, it is desirable to tackle as many attacks as possible. Geometrical robustness remains one of the major challenges due to its ease of implementation and destructive effects. A major part of this thesis is devoted to the investigation of geometrical robustness in order to address this challenging problem. However, the application scenario of robust watermarks must be taken into consideration. Therefore, some of the attacks may not be relevant. This is perhaps the major obstacle in establishing a publicly accepted benchmarking system for robust
watermarks. Some of the popular benchmarking tools are CheckMark, CertiMark, OptiMark, and StirMark.

These sections complete the discussion on robust watermarking. The other single watermark category is the semi-fragile watermark. In contrast to robust watermarks, semi-fragile watermarks are sensitive to a majority of image modifications. Therefore, semi-fragile watermarks often serve complementary purposes.

### 2.5 Semi-fragile Image Watermarking

Early watermarking methods focus on exact authentication (a.k.a. hard authentication, complete authentication) using fragile watermarks. In such a case, a single bit change in an image will be classified as unauthentic. An image can be authenticated using cryptographic hash or digital signatures as the fragile watermark. There are several advantages of watermarking over cryptographic methods for this purpose. Firstly, the fragile watermark embedded into an image eliminates the need for extra storage. Secondly, it is also immune to format conversion because it stays intact with the image and undergoes the same transformation as the image. Finally, in addition to the integrity check on the whole image, the watermark can also be designed to determine which part of the image is unauthentic. This is called tamper localization, and will be discussed later. Although robust watermarks can be used in authentication, the simplicity of fragile watermarks makes them a better option [45].

To ensure that the fragile watermark does not interfere with the authentication information of an image, the embedding space can be divided into the watermark-generation subspace and the watermark-embedding subspace. The embedding space here can be a spatial domain or a frequency domain. For example, the watermark can be embedded into the least-significant-bit (LSB) plane after it is generated from the other higher order bit plane. Besides partitioning an image into subspaces for watermark generation and embedding, another approach is to design reversible watermarks that can be “erased” to reveal the original image content. This approach uses the detected watermark information to invert the changes made during watermark embedding.

Both robust and fragile watermarks can be used in authenticating content. However, they are rigid compared to semi-fragile watermarks. Semi-fragile watermarks can be designed to tolerate legitimate changes while highlighting
intentional distortions. This characteristic made the semi-fragile watermarks suitable for a wide range of applications. For example, we might want to allow image compression in order to save storage space. The other applications of semi-fragile watermarks are content authentication, tamper localization, and content recovery. They will be discussed in the following sub-sections.

2.5.1 Content authentication

New applications in digital watermarking require the ability to differentiate legitimate changes and illegitimate modifications. For example, high quality compression that does not affect the visual quality of an image should be tolerated, and region replacement that changes its content meaning should be highlighted. Therefore, semi-fragile watermarks have been created for content authentication. Content authentication in this context is also named soft authentication or selective authentication. To do this, the semi-fragile watermark should be easily destroyed by general image processing, yet robust against legitimate changes. The validation of image integrity is based on its content instead of its numerical representation. The degree of semi-fragility is defined by the application scenario. An example of an application that prefers semi-fragile watermarking to fragile watermarking is reducing storage space using high quality image compression. Another situation that suits semi-fragile watermark is the tolerance of bit errors in media transmission and storage.

There is no publicly agreed standard on the tolerable level of image degradation. Some of the mild processing that may be classified as legitimate changes are listed below.

- JPEG compression with quality factor above 80%. The higher is the quality factor of a JPEG compressed image, then the better is its visual quality. It is commonly mentioned that quality factor below 10% is unacceptable due to visible block artefacts, although the compression reduces storage space significantly [46].
- Salt and pepper noise insertion with maximum intensity of 1%. This can simulate the concept of bit errors due to storage media “scratches”. For example, the surface of a CD-ROM may be scratched due to poor handling, and it causes incorrectly read bits.
• Random bit error in raw data with maximum probability 0.001. This is similar in concept to the *salt and pepper noise* described above.

Depending on the application, the following operations may or may not be considered permitted changes. For instance, we may argue that image region substitution is a forgery and should be detected as tampered region. However, if the substituted region is visually indistinguishable from the original appearance, then this modification may be permitted in some commercial advertisements. Medical images would obviously require higher levels of integrity compared to creative arts.

• Additive White Gaussian Noise (AWGN) that gives minimum signal-to-noise ratio (SNR) of 36 dB
• Image smoothening using a $3 \times 3$ kernel with average weight
• Histogram equalization with uniform distribution
• Image region substitution in which the visual appearance is maintained
• Regional geometrical transform that does not affect the perceptual quality
• Global rotation of less than 1 degree
• Global circular shift of 1 row or 1 column
• Cropping along image borders not exceeding 1% of image area
• File format conversion

There are generally two ways to authenticate the content of an image. One way is to use the significant features of the content as a watermark for self-authentication. This has the advantage of content recovery, and will be described in Section 2.5.3. The other way is to use digital signatures which are content-independent as a watermark for authentication. For image content authentication, the decision is often made by thresholding. For example, a pre-defined threshold value can be used on the count of error bits or the correlation value of the authentication. If the calculated value is above the threshold, then the image is classified as authentic.

### 2.5.2 Tamper localization
In the early stage of the development in fragile and semi-fragile watermarking, the watermark detection result is only a simple answer of authentic or non-authentic. New methods can identify the tampered regions in an unauthentic image. This is the
2.5 Semi-fragile Image Watermarking

tamper localization capability. The results of tamper localization could be useful in media forensics. For example, the motive behind tampering a certain region of an image can be deduced.

A majority of the current semi-fragile watermarking schemes employ block-based processing for tamper localization. This approach divides an image into non-overlapping blocks and processes each of them separately. For example, the mean value of 8×8 pixel blocks can be embedded into a cover image. Later, it can be extracted from the stego image and compared with the computed mean value of a block at the same location to detect tampering. Figure 2.6 illustrates an example of block-based localization. The *Lena* (a.k.a. *Lenna*) image is divided into 16 equal-size blocks. Watermark embedding and detection are performed on each block individually. This approach certainly involves a large amount of computation. Higher accuracy in tamper localization can be achieved using smaller blocks. However, the computational cost will be higher too. The advantage of this approach is that different watermark information can be used for each block, thus giving it more flexibility.

![Figure 2.6 Block-based localization](image)

An alternative to block-based localization is sample-based localization. In this approach, a small size binary logo is embedded in a tiling pattern across an image. Tampered regions are highlighted if the detected watermark pattern is irregular compared to the tiled pattern. An example using the QUT logo is shown in Figure 2.7. The region bounded by the sample at row 2 column 2 indicates tampering has occurred because the detected watermark pattern is damaged. Despite its simple implementation and low complexity, this approach is rigid compared to the previous approach.
Wavelet-based localization is perhaps a more elegant and efficient approach compared to the two approaches described above. A wavelet transform such as the DWT decomposes an image into a low-pass subband and three high-pass subbands. The spatio-temporal information in the wavelet coefficients can easily be exploited in tamper localization. This would eliminate the need to include block location details in a watermark. Advantages of this approach also include the moderate amount of computation and flexible watermarking. For instance, better accuracy in localization can be made by watermarking at the first level of the wavelet decomposition, and higher robustness (less fragility) can be achieved by watermarking at the higher level of the wavelet decomposition. Figure 2.8 depicts a 4-level DWT, and a coefficient at level 2 corresponds to 4 coefficients at level 1 in the same region. In addition, HVS masking can be applied easily using the energy pattern in the low-pass subband of the wavelet decomposition. This is possible because the low-pass subband represents a down-scaled version of the image, and the HVS models image perception of the human eyes.
2.5.3 Content recovery
Content recovery is a rather new challenge in watermarking. After identifying the tampered regions with localization, one would like to reverse the damage and uncover the original content. By comparing the restored content with the tampered image, the job of a forensics investigator could be made easier. Although a full content recovery is theoretically impossible, an approximate content recovery is achievable. Full content recovery cannot be realized practically because watermarking will introduce distortions and occupy a subspace of an image.

To provide content recovery, a condensed set of image information can be embedded as watermark. Then, the detected watermark can be used to reconstruct a tampered region. This is sometimes referred to as self-embedding, self-authentication, and self-recovery. The condensed image information can be a down-scaled version of the image, a compressed set of significant image features, a short description of image regions, or any other abstraction of the image. Image features can be edges, luminance, texture, etc. Essential decisions have to be made in selecting the image information for watermarking, because it directly determines the watermark capacity and the quality of recovered content. It should be noted that robust hashes, visual hashes and digital signatures are not suitable for content recovery because they have too little information to describe the image content. In addition, it is important to avoid embedding the watermark in the same locality in order to overcome cropping attacks. Otherwise, the watermark will be lost with the cropped region, and content recovery will not be possible.

To date, there are very few semi-fragile watermarks which have content recovery capability, e.g. Lin-Chang [36] and Rey-Dugelay [47]. This observation can be explained by the nature of semi-fragile watermarks. Ideally, a watermark should be robust instead of semi-fragile to survive as many distortions as possible so that the original content encoded in the watermark can be used in content recovery. Fragile watermarks can also offer content recovery to a certain extent [48].

2.5.4 Approaches
Having discussed new requirements and directions in semi-fragile watermarking, we review implementation approaches of semi-fragile watermarks in this section. Some of the approaches may not fulfil the latest features, e.g. content recovery. Semi-
fragile watermarking can be studied according to their watermark generation approaches and watermarking domain.

(a) Watermark generation
The way a watermark is generated will affect the information encoded, thus determining its effectiveness in content authentication, tamper localization, and content recovery.

Authentication of images can be simply made using pseudo-random sequences or digital signatures. However, these image-independent watermarks can be easily forged using collage attacks and brute-force search attacks. The attacker can use multiple copies of stego images that have the same watermark key to carry out a collage attack. In this attack, many blocks of authentic image parts can be collected and used to resemble a forged image with authentic response as in the Holliman-Memon attack. In the second type of attack, the attacker can use a publicly available watermark detector to search for a forged image that gives an authentic response. Block-based and sample-based watermarking methods are vulnerable to these attacks. If the block size or sample size is small, then the search space would be small. That means the risk of these attacks would be higher. Therefore, image-dependent watermarks should be employed. The watermark can be generated using such image features as edges and mean intensity of a block. Higher security can be achieved if the watermark of a block is linked with information of its neighbouring blocks. In this way, tamper localization and content recovery is also possible.

To ensure high quality in content recovery, the watermark should carry enough information regarding the image features. Hash-based methods, either robust hashes [49] or visual hashes [50], cannot provide enough information for content recovery because they map image features into a small space. Indeed, they are designed for image indexing and searching purposes. This is also true for some methods based on digital signatures and feature points [51].

(b) Watermarking domain
Various semi-fragile watermarking schemes have been developed in both spatial and transform domains [47]. Spatial domain schemes usually exploit the statistical properties of the image pixels when detecting tampering, and they also provide authentication. As such, their implementations are normally simple and fast. On the
other hand, transform domain schemes offer robustness to compression by using the quantization of frequency information.

JPEG compression has been a major focus of semi-fragile watermarking because many real life applications use it to reduce image storage and transmission cost. The JPEG compression algorithm is a widely accepted compression standard based on quantization of DCT coefficients. Therefore, a huge number of semi-fragile watermarks employ a DCT domain quantization approach. Despite this, quantization of wavelet coefficients has also been providing good performance. Two of the well-discussed works that utilize the wavelet domain are SARI [35, 36] and telltale watermark [34]. SARI is able to perform content authentication, tamper localization, and approximate content recovery.

2.5.5 Remarks
Semi-fragile watermarking is relatively new compared to robust and fragile watermarking. New capabilities offered by semi-fragile watermarking have been the subject of experimentation in recent years. Nevertheless, its performance can be improved, especially in content recovery. We will examine these new capabilities in our work.

This section reviewed single watermarking methods. Hybrid watermarking methods combine the strengths of robust and fragile watermarks. Hence, hybrid watermarking methods can offer more functionalities in a single solution. We will survey hybrid watermarking methods in the next section.

2.6 Hybrid Methods in Image Watermarking
Single watermarking methods, either robust or semi-fragile, can only serve a limited number of purposes. They are bounded by their robustness or semi-fragility. For example, semi-fragile watermarks are not suitable for copyright protection because they can be destroyed by an attacker. To overcome these limitations, a hybrid watermarking method is a good choice, because it combines a robust watermark and a semi-fragile watermark. Complementing the weaknesses of each single watermark, the hybrid watermarking method has a high potential in practical use. Although there are some early hybrid watermarking methods that combine robust and fragile watermarks, we are particularly interested in the combination of robust and semi-fragile watermarks. Semi-fragile watermarks have a distinct advantage compared to
fragile watermarks, because they can differentiate legitimate changes and non-permitted modifications.

**Implementation strategies**

Hybrid watermarking methods can be broadly implemented in two ways. The first way is to overlap the robust and semi-fragile watermarks during watermark embedding, and detect each of them separately. The robust watermark is embedded first, followed the semi-fragile watermark. This is made based on the foundation that the robust watermark should be able to resist distortions introduced by the semi-fragile watermark. This reasoning was discussed by Fridrich [52] and Mintzer-Braudaway [53]. The other way of hybrid watermarking is to ensure that both of the robust and semi-fragile watermarks are not overlapped during watermark embedding. This can be done by dividing an image into blocks for independent processing. The watermark detection is performed separately for the robust and semi-fragile watermarks. This orthogonal arrangement can be made by embedding the watermarks into different sets of coefficients in a transform domain. Such an implementation would reduce the interference between the two watermarks, thus ensuring better watermark detection results compared to the first implementation. However, a practical hybrid watermarking method must find a balance among a few essential factors, e.g. watermark imperceptibility, robustness or semi-fragility of each watermark, accuracy of tamper localization, effectiveness in content recovery, and overall computational costs. For instance, watermark imperceptibility might be worse in the overlapped method given higher level of distortions on each coefficient, but it may have better tamper localization due to higher watermark density.

**2.6.1 Existing hybrid methods**

There are not many hybrid watermarking methods found in the literature compared to single watermark methods. This could be due to the complexity in designing a hybrid method. However, the increased protection functions in a hybrid method may have encouraged some research work to be published. Most of the hybrid watermarking methods use a robust watermark for copyright protection, and a fragile watermark for tamper detection. They are briefly discussed below.
1 Hybrid Method of Fridrich
The earliest hybrid method is probably proposed by Fridrich in 1999 [48]. It consists of a block-based robust watermark and a fragile watermark. An image is divided into non-overlapping blocks of equal size and processed separately. The block size needs to compromise between robustness of the robust watermark and accuracy of tamper localization. For example, large block size has better robustness but is less accurate in tamper localization. The robust watermark is embedded into the middle frequency coefficients of a DCT domain using a spread spectrum method, and secured by a secret key. Detection of the robust watermark is computed using correlation in the DCT domain. The fragile watermark has tamper localization capability but it cannot differentiate significant modifications from innocent image processing. However, using the detection results of both robust and fragile watermarks, significant modifications and innocent image processing can be determined. For instance, a successful detection of the robust watermark and a missing fragile watermark without localized tampering indicate that common image processing may have been performed on the image.

2 Hybrid method of Deguillame-Voloshynovskiy-Pun
In another block-based method, developed by Deguillame-Voloshynovskiy-Pun [21], a robust watermark can be associated with either a fragile or a semi-fragile watermark to form a hybrid method. The robust watermark exploits a self-reference template for resynchronization. It uses additive embedding in the DWT domain. A perceptual mask based on a NVF is also applied during watermark embedding for improved imperceptibility. The robust watermark can be self-synchronized based on equally-spaced peak patterns computed using the Auto-Correlation Function (ACF). On the other hand, the fragile watermark is a hashed signature embedded in LSBs. The hash is computed by linking neighbouring blocks to provide local contextual information for better security. However, the ACF peaks could be removed using an attack proposed by Lu and Hsu [43].

The hybrid method can identify copy attacks and collage attacks based on the detected robust and fragile/semi-fragile watermarks. In these attacks, the robust watermark can be copied from one image to another image in an unauthorized manner. Hence, it creates ambiguity between the watermark signal and the image.
The hybrid method proposed can identify these attacks when the robust watermark is detected but the fragile watermark indicates tampering occurred. In a copy attack, the whole image would be unauthentic. In sophisticated collage attacks, which maintain some context dependent information, some parts of the image would be unauthentic. An example of the context dependent information is the image block location.

3 Hybrid method of Fan-Tsao
A hybrid method reported in Fan-Tsao [54] is designed specifically for the JPEG2000 format. The robust watermark is embedded using scalar quantization at multiple scales in the DWT domain. Its stego images have poor visual quality because the robust watermark was embedded in the low-pass subbands where it causes many distortions. The fragile watermark is similarly embedded into the highpass subbands at multiple scales. Detection of the robust and fragile watermarks is performed at multiple scales in the DWT domain using the same scalar quantization on its respective subbands. The robustness against common image processing and geometrical attacks was not reported.

4 Hybrid method of Habib-Sarhan-Rajab
In Habib-Sarhan-Rajab’s method [55], an image is partitioned into blocks which are watermarked separately in the DCT domain. The robust watermark is embedded into the middle frequency coefficients, and the fragile watermark is embedded into the LSBs. Both of the watermarks are binary pictures. The practicality of using two binary pictures in a hybrid method needs further investigation. Robust watermark detection is based on the correlation value between the extracted binary picture and the validating watermark. The fragile watermark is extracted from the LSBs of the DCT coefficients.

5 Hybrid method of Sharkas-ElShafie-Hamdy
Another hybrid method reported by Sharkas-ElShafie-Hamdy [56] uses two nested watermarks in a non-blind manner. Firstly, the robust watermark, consisting a pseudo-random sequence, is embedded into an image. Then, the stego image is used as a fragile watermark and embedded into another image. The robust watermark is detected by its correlation value with a validating pseudo-random sequence, and the fragile watermark detection is the extraction of the primary stego image. However,
the feasibility of embedding a robust watermark into a fragile watermark is questionable. If the fragile watermark is destroyed by an attacker, then the robust watermark (which is embedded in the fragile watermark) would not be detectable, and would lose its robustness.

6 Hybrid method of Lie-Lin-Cheng
So far all of the hybrid methods consist of a robust watermark and a fragile watermark. Besides the Hybrid Method of Deguillame-Voloshynovskiy-Pun discussed above, there is only one hybrid method that combines a robust watermark with a semi-fragile watermark [57]. It uses a DCT domain for watermarking JPEG images with the informed embedding method. Informed embedding refers to watermarking with image dependent information. The robust watermark is a binary logo modulated with image features, and then embedded using quantization. The semi-fragile watermark consists of channel information that depends on the image’s content. It is also embedded using quantization. Detection of the semi-fragile watermark is carried out first, and its result is applied in a soft-decision decoder to extract the robust watermark. The method is well-balanced in terms of imperceptibility, robustness, and computational costs. However, no tests on its robust watermark under rotation or translation attacks were reported.

2.6.2 Potential improvements
As can be observed in the few methods found, hybrid watermarking is not a well-explored area. Based on the hybrid watermarking methods surveyed in the previous section, they mainly use a robust watermark for copyright protection and ownership claim. Additionally, a fragile/semi-fragile watermark is usually applied in authentication, tamper detection, and tamper localization. A hybrid watermarking method that provides the extra function of approximate content recovery is difficult to find. Such a function is desirable because the recovered content can provide useful forensic information in an investigation.

With the wide adoption of some image compression formats, a few hybrid methods are designed specifically for JPEG [57] and JPEG2000 [54] formats. These methods may not perform well for other file formats. Furthermore, JPEG2000 is not widely supported by web browsers as of 2006. It is also not adopted in many applications. This could be due to the legal issues that may arise. Comparing
JPEG2000 to JPEG, there is not much improvement in compression ratio. Both JPEG and JPEG2000 are designed for images with majority areas consisting smooth features. For example, photographs and medical scans have largely smooth regions. On the other hand, GIF and PNG formats have better compression ratio than JPEG and JPEG2000 for line drawings and diagrams. Therefore, image watermarking methods should be made format independent.

For practicality, hybrid methods should employ blind watermark detection. This means that a reference image is not required during watermark detection. To this end, hybrid methods should have self-embedding and self-authentication capabilities. In addition, self-recovery capability can be realized using self-embedding watermarks. The final part of our work aims to study these potential improvements.

2.7 Chapter summary
Digital image watermarking involves embedding and detection of hidden information in pictorial media. It can complement the functions of cryptography in content protection. The increased consumption of digital content and security circumvention technologies necessitates more research in digital watermarking.

The discussions in this chapter begin with a model of digital watermarking. Then, we described watermark properties and watermark applications. The major properties to be considered when creating a digital watermark are robustness, capacity, imperceptibility, computational cost, detection reliability, and other practical issues. Watermarking technologies can be applied in DRM, copyright protection, authentication, tamper detection, tamper localization, annotation, privacy control, and media forensics. Beside that, other major topics described are robust watermarks, semi-fragile watermarks, and hybrid watermarking methods. Robust watermarks are suitable for copyright protection. Watermark robustness can be achieved using redundancy, synchronization and invariant domains. In contrast, semi-fragile watermarking can be implemented in transform domains and applied in content authentication, tamper localization, and content recovery. Additionally, hybrid watermarking systems that combine robust and semi-fragile watermarks offer integrated solutions for digital content protection.
2.7 Chapter Summary

There are several original work presented in this chapter. For robust watermarking, we categorize attacks on robust watermarking into image degradation, image enhancement, image compression, and image transformations. We also classify robust watermarking methods into 3 approaches: redundant information, synchronization, and invariant domain. In addition, we found that content recovery using semi-fragile watermark is a potentially good research direction. Furthermore, our contributions also include the survey of 6 hybrid watermarking methods and the identification of possible improvements.

These discussions reveal the gap in robust, semi-fragile, and hybrid watermarking methods. Geometric distortion remains a challenging problem in robust watermarking. We will deal with it in Chapters 3 and 4. On the other hand, semi-fragile watermarking that has self-authentication and self-recovery is desirable. Chapter 5 will present our work in solving this problem. Finally, hybrid watermarking is a relatively unexplored field despite the possibility of integrating multiple features into a single method. A new hybrid watermarking method will be described in Chapter 6.
Chapter 3

Robust Watermarking by Synchronization

Having reviewed robust watermarking requirements and some methods in Chapter 2, we will describe our method in this chapter. The advantages of the synchronization approach in robust watermarking will be explained. Then, our watermarking methods will be presented. The methods include watermark embedding and watermark detection processes. We will also evaluate the performance of our methods, identify their limitations, and suggest some improvements.

3.1 Introduction

Robustness of a watermarking method is the ability of a watermark to withstand common image processing. The common image operations may include image smoothening (e.g. low pass filtering), edge enhancement (e.g. image sharpening), cropping, lossy compression, noise insertion, and contrast adjustment. In addition, they may also include some less common operations such as geometrical transforms, histogram equalization, and gamma correction. These distortions are sometimes referred to as “attacks” although it should be differentiated from malicious attacks that target the security aspects of watermarking. Similarly, although some researchers use the term “robust” to include secure watermarks, it should be noted
that watermark robustness is not the same as watermark security. A secure watermark can survive malicious attacks whereas a robust watermark might not. Watermark security is a different research area that is closely related to cryptographic security. Research activities in watermark security cover unauthorized watermark estimation, watermark removal, watermark modification, and watermark insertion. They are often investigated from the information security viewpoint. Hence, a robust watermark should be detectable under unintentional distortions. The main motivation of robust watermarking is to protect the ownership of a digital content. By using redundancy, synchronization, or invariant domains, a robust watermark remains detectable with a high probability when unintentionally distorted.

Because of the ease of carrying out geometrical transforms, and their destructive effects on watermark information, geometrical robustness have been an actively studied area. Watermark synchronization can combat geometrical distortions such as RST operations. These basic operations are generally known as RST distortions, and their combinations make up many types of affine transforms. Synchronization methods realign a distorted image similar to the image registration process so that the embedded watermarks can be detected successfully. Image registration is the process of overlaying two images of the same scene to align the locations of their corresponding points. The images are photographed with different parameters, e.g. photographing time and view angle. The process of image registration resynchronizes watermarks in distorted images so that the watermarks can be detected successfully. Initial work on robust watermarking assumed that image distortions affect the whole image. This is not the case for new attacks that change partial image information. One of the famous attack in this category is the RBA [58, 59]. Watermark synchronization can resist this type of attack by registering local changes instead of inverting distortions globally.

Compared to the invariant domain approach, the design of watermark synchronization is easier. There are many image registration methods that can be applied in watermark synchronization. Furthermore, most images have robust features that can be used in synchronization. For example, feature points of an image form triangular regions in which watermark embedding and detection take place [40, 41]. Other examples of these features are robust moments, and facial feature points. We will investigate two watermark synchronization methods in this chapter. The first
3.2 Related Work

The fundamental method for robust watermark detection is brute force search. Although it can guarantee successful watermark detection in most cases, it requires high level of computational cost when the search space is large. Therefore, efficient robust watermark detection should reduce the search space by guessing the likely distortions and constrain the search within these distortions. However, this is normally not practical due to the huge amount of possible distortions. For example, the combination of rotation, translation, and scaling with varying parameters could produce many different distortions.

A more practical approach to robust watermark detection is to perform image registration prior to watermark detection using synchronization methods. One of the synchronization methods is to use a template as a reference to reverse the distortions before watermark detection. An example of template-based watermarking was proposed by Pereira and Pun [60]. They embedded a template watermark in addition to the actual information-carrying watermark. However, the template can be estimated and removed. Furthermore, the template also reduces the watermark payload. Another example of self-reference method is the use of ACF as a template in Kutter’s work [16]. He embedded repetition of a symmetric pattern into images by flipping and rotating a small watermark in 4 quadrants. In that way, an undistorted peak pattern should have regularly spaced peaks. Therefore, geometric distortions can be reversed by referring to the extracted peak patterns of ACF. However, this method suffers from several weaknesses. The peak patterns can be estimated easily by an adversary without the need of any secret information. Additionally, the computed patterns may have extra peaks, missing peaks, and peaks that are slightly shifted from their correct positions.

Another method of synchronization is based on salient image features. For example, Bas, Chassery, and Macq [40, 41] proposed a watermarking method that considers image contents. They firstly extract salient features from an image, then perform watermark embedding and detection within regions bounded by the salient features. Since the salient features can be robustly detected, the search space for
watermark detection is confined to a set of constant regions. As a result, the
watermark can be detected successfully. Nevertheless, this method would not work
well for images without distinctive features. For example, an image of ocean with
flat regions would not provide robust points as salient features.

To overcome these problems, we will investigate a synchronization approach that
uses motion estimation.

3.3 Motion Estimation for Synchronization
Motion estimation is an image registration technique that compares two images to
estimate the direction and distance of object movements between the images. It is
very useful in video frame coding. By estimating the motion of objects in a sequence
of video frames, the coding process can reduce the storage space required. In
watermark synchronization, the differences between objects in a reference image and
a test image can be computed using motion estimation. These differences are the
objects’ location, orientation, and scale. Objects are robust image features used as
reference points. This will ensure successful detection of the watermark after the
resynchronization.

Early watermark synchronization methods worked on the global scale because
they assumed that geometrical transforms are applied onto an image homogeneously.
For image registration involving RST operations at the global scale, differential
affine motion estimation techniques have been shown to be quite powerful. Later,
local geometrical distortions, such as the RBA, were developed to desynchronize
watermark information without causing noticeable artifacts. As a result, new
watermarking methods are needed to overcome this type of distortion. In this section,
we will investigate a watermarking method that tackles both global and local
geometrical distortions. This method uses differential affine motion estimation for
image registration in its synchronization steps. The geometric distortions between a
reference image and its distorted image are modelled as locally affine but globally
smooth motions. Therefore, both large-scale and small-scale transformations are
captured in its multiscale framework. Then the resynchronized image can provide
correct watermark detection under local distortions. We will experiment with this
approach using two watermarking methods, i.e. correlation-based spread spectrum
3.3 Motion Estimation for Synchronization

watermarking [61] and dynamic thresholding in the wavelet domain [6]. The robustness will be tested using the *StirMark* benchmarking tool [58, 59].

### 3.3.1 Geometrical Transform and Affine Transform

It is useful to model geometrical and affine transformations before describing the motion estimation method. Figure 3.1 depicts some of the basic operations that made up affine transforms. The operations are rotation, uniform scaling, aspect ratio change (a.k.a. non-uniform scaling) translation, flipping (a.k.a. reflection) and shearing.

An affine transform can be modelled by vector addition and matrix multiplication. The affine transformation of $\mathbb{R}^n$ is a mapping $F: \mathbb{R}^n \to \mathbb{R}^n$ of the form:

$$F(p) = Ap + q$$  \hspace{1cm} (3.1)

where $p, q \in \mathbb{R}^n$ and $A$ is a linear transformation of $\mathbb{R}^n$. In a two-dimensional (2-D) space, the affine transformation between two point pairs $(x', y')$ and $(x, y)$ is given by:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = AX + d$$  \hspace{1cm} (3.2)

The affine transformation in equation (3.2) can happen at the global scale and local scale. Global transformation involves the entire image. On the other hand, local transformations operate on small locations of the image, and they may have different parameters. Figure 3.2 illustrates an example of local affine transformation called the RBA in *StirMark* [58, 59]. Although the visual appearance of the distorted image is perceptually similar to the original image, its grid pattern is “bent” and the watermark information is desynchronized.
3.3.2 Differential Affine Motion Estimation

The differential affine motion estimation method estimates image distortions at both global and local scales [5, 62]. The local motion between images can be modelled by an affine transform:

\[ f(x, y, t) = f(m_1 x + m_2 y + m_5, m_3 x + m_4 y + m_6, t - 1) \]  

(3.3)

where \( f(x, y, t) \) and \( f(x, y, t-1) \) are the source and target images respectively. \( m_1, m_2, m_3, m_4 \) are the linear affine parameters, and \( m_5, m_6 \) are the translation parameters. To estimate these parameters, a quadratic error function \( E \) is minimized:

\[ E(\bar{m}) = \sum_{x,y \in \Omega} [f(x, y, t) - f(m_1 x + m_2 y + m_5, m_3 x + m_4 y + m_6, t - 1)]^2 \]  

(3.4)

where \( \bar{m} = (m_1 \cdots m_6)^T \), and \( \Omega \) denotes a small spatial neighbourhood. The differential technique computes the minimization of the error function directly from the image pixel intensities by expanding the right side of the equation in a Taylor series to obtain:
3.3 Motion Estimation for Synchronization

This error function can be expressed more compactly in a vector form:

\[ E(\bar{m}) = \sum_{x,y \in \Omega} \left[ k - \bar{c}^T \bar{m} \right]^2 \]  

(3.6)

where the scalar \( k \) and vector \( \bar{c} \) are given as:

\[ k = f_i + xf_x + yf_y \]

\[ \bar{c} = (xf_x, yf_y, xf_y, yf_y, f_x, f_y)^T \]

This error function can be minimized by differentiating with respect to the \( \bar{m} \):

\[ \frac{dE(\bar{m})}{d\bar{m}} = \sum_{x,y \in \Omega} -2\bar{c}[k - \bar{c}^T \bar{m}] = 0 \]

(3.8)

This solution assumes that the matrix is invertible. It can be guaranteed by integrating over sufficiently large spatial neighborhoods \( \Omega \) with sufficient image content. To account for intensity variations, equation (3.3) takes the form:

\[ m_7f(x, y, t) + m_8 = f(m_1x + m_2y + m_5, m_3x + m_4y + m_6, t - 1) \]

(3.10)

where \( m_7 \) and \( m_8 \) are two new parameters that embody a change in contrast and brightness respectively. The scalar \( k \) and vector \( \bar{c} \) become:

\[ k = f_i - f + xf_x + yf_y \]

\[ \bar{c} = (xf_x, yf_y, xf_y, yf_y, f_x, f_y - f - 1)^T \]

(3.11)

A trade-off exists in choosing the size of the neighbourhood. A large area makes it more likely that the matrix \( \sum_{x,y \in \Omega} \bar{c} \bar{c}^T \) in equation (3.9) will be invertible. On the other hand, a small area makes it more likely that the brightness constancy assumption will hold. To avoid balancing these two issues, we assume that the model parameters \( \bar{m} \) vary smoothly across space. We augment the error function in equation (3.6) as follows:
Chapter 3 Robust Watermarking by Synchronization

\[ E(\tilde{m}) = E_b(\tilde{m}) + E_s(\tilde{m}) \quad (3.12) \]

where \( E_s(\tilde{m}) \) is defined as in equation (3.6) without the summation:

\[ E_b(\tilde{m}) = \left[ k - \tilde{c}^T \tilde{m} \right]^2 \quad (3.13) \]

with \( k \) and \( \tilde{c} \) as in equation (3.11). The new quadratic error function \( E_s(\tilde{m}) \) embodies the smoothness constraint:

\[ E_s(\tilde{m}) = \sum_{i=1}^{8} \lambda_i \left[ \left( \frac{\partial m_i}{\partial x} \right)^2 + \left( \frac{\partial m_i}{\partial y} \right)^2 \right] \quad (3.14) \]

where \( \lambda_i \) is a positive constant that controls the relative weight given to the smoothness constraint on parameter \( m_i \). To minimize this error function, we have:

\[ \frac{dE(\tilde{m})}{d\tilde{m}} = \frac{dE_b(\tilde{m})}{d\tilde{m}} + \frac{dE_s(\tilde{m})}{d\tilde{m}} = 0 \quad (3.15) \]

where

\[ \frac{dE_b(\tilde{m})}{d\tilde{m}} = -2\tilde{c}\left[ k - \tilde{c}^T \tilde{m} \right] \]
\[ \frac{dE_s(\tilde{m})}{d\tilde{m}} = 2L(\overline{\tilde{m}} - \tilde{m}) \quad (3.16) \]

with \( \overline{\tilde{m}} \) is the component-wise average of \( \tilde{m} \) over a small spatial neighbourhood, and \( L \) is a \( 8 \times 8 \) diagonal matrix with diagonal elements \( \lambda_i \), and zero-off the diagonal. In equation (3.15), solving for \( \tilde{m} \) at each pixel location yields an enormous linear system which is intractable. Instead of solving that, \( \tilde{m} \) is expressed as:

\[ \tilde{m}^{(j)} = (\tilde{c}^T \tilde{c} + L)^{-1}(\tilde{c}k + L\overline{\tilde{m}}^{(j)}) \quad (3.17) \]

An iterative scheme to solve for \( \tilde{m} \) is employed. On each iteration \( j \), \( \tilde{m}^{(j)} \) is estimated from the current \( \tilde{m}^{(j)} \). The initial estimate \( \tilde{m}^{(0)} \) is estimated from the closed-form solution of equation (3.10).

To implement the formulation above, a more accurate estimate of the actual error function can be determined using a Newton-Raphson style iterative scheme. In each iteration, the estimated transformation is applied to the distorted image, and a new transformation is estimated between the newly warped distorted and reference image. A coarse-to-fine scheme is adopted to cope with large motion. A Gaussian pyramid is built for both distorted and reference images, and the local affine and contrast parameters estimated at the coarsest level. These parameters are used to warp the distorted image in the next level of the pyramid. A new estimate is computed at this level, and the process is repeated through each level of the pyramid. The
3.3 Motion Estimation for Synchronization

Transformations at each level of the pyramid are accumulated to form a single final transformation.

3.3.3 Experimental Evaluations

The effectiveness of the image registration method was tested on two watermarking methods. The first method is a correlation-based spread spectrum method that operates in the DCT domain [61]. It embeds watermarks into the DCT coefficients across a broad range of frequencies, and correlation values are computed for watermark detection in a non-blind manner. The second method uses dynamic thresholding in the DWT domain [6]. It embeds watermarks into DWT coefficients, and computes threshold values dynamically for blind watermark detection.

Several 8-bit grayscale images of 256×256 pixels were used in the experiments. They are illustrated in Figure 3.3 and identified by name: Lena, Fishing boat, F16 airplane, Baboon (a.k.a. Mandrill), Pepper, and Cameraman. Baboon represents images with large areas of complex texture (i.e. the fur) and homogeneous areas (i.e. the face); Cameraman is chosen for its flat regions (i.e. the sky) and high contrast regions (i.e. the shirt and its background); Lena has a mixture of characteristics (e.g. smooth background, while the hat has complex textures and big curves); Pepper provides luminosity changes (i.e. light reflection surfaces); Fishing boat contains smooth parts (i.e. the clouds) as well as other feature combinations. The image registration was performed using four-level pyramids constructed with a five-tap lowpass filter. At each level, a single global affine transform is first estimated by setting $\Omega$ to be the entire image. Then, the local affine parameters are estimated with $\Omega = 5\times5$ pixels. In each iteration, $\lambda_i = 1\times10^{11}$, $i = 1, \cdots, 8$ and $\bar{m}_i$ is computed by convolving with a 3×3 kernel. After forty iterations, the distorted image is warped according to the final estimate, and this process is repeated five times. This entire process is repeated at each level of the pyramid.

Figure 3.4 shows an example of registering Lena, which has been distorted with the RBA with a relatively high bending factor. Comparing the images in (a) and (b), the distortions are perceptually distinctive as depicted in (d). Using the image registration method described, the registered image in (c) has little differences compared to the original image, and this is illustrated in (e).
(i) Non-blind Watermark Detection
The first watermarking method developed by Cox et al. requires cover images for watermark detection [61]. It transforms both test image and the cover image into DCT domain, and subtracts the chosen set of coefficients to extract the watermark. Therefore, it is a non-blind watermark detection. The method detects watermarks by calculating correlation values and comparing it to predefined threshold values. The watermarks are considered detected if the correlation values exceed the threshold.
values. This can be visually verified by the presence of significant peaks in the graph plots of correlation values. The correlation calculation involves the watermark extracted from a test image and the watermark under examination. The peak appears when the extracted watermark matches that of the embedded watermark, thus giving a high correlation value. Under other circumstances, correlation values are low for randomly chosen watermarks.
To demonstrate that image registration can resynchronize watermark information and assist in successful detection of watermarks in distorted images, we performed watermark detection in three stages, i.e. before distortion, after distortion, and after synchronization. Figure 3.5 shows the detection results for a stego image, a distorted stego image and a registered stego image. As shown in (b), watermark detection was successful for the stego image in (a). The stego image was distorted by the RBA. The distortion was minimized with a very small bending factor of 0.3 to preserve the image’s perceptual quality. This distortion desynchronized watermark information and defeated watermark detection despite the perceptual similarity between the stego image in (a) and its distorted version in (c). This means RBA is an important threat to robust watermarking. In the absence of image registration, the watermark cannot be detected even at very low levels of distortion. This is shown by the absence of a significant peak in the graph (d). However, the image registration method tested can resynchronize the watermark information and improve the detector performance significantly. This is evidenced by the significant peak in (f) for the detection result of the registered image in (e).

Figure 3.5 Effect of image registration on watermark detection

(a) Stego image

(b) Detector response of the image in (a)

(c) Distorted stego image

(d) Detector response of the image in (c)
To evaluate the effects of RBA bending factor on the image registration performance and the resultant watermark detection, we used bending factors ranging from 0.1 to 1.5 with 0.2 intervals. The test results for Pepper and Cameraman are shown in Figure 3.6. The stego images depicted in (a) and (c) has good visual appearance compared to their cover images in Figure 3.3. The correlation values for the stego images, distorted stego images, and their registered stego images are plotted in the graphs (b) and (d) together with their legends. Although the watermark detector performance drops rapidly with an increase in bending factor, in all cases, the watermark still can be detected after registration. The experiment was repeated using Fishing boat, F16 airplane, and Baboon. In all tests, the watermark can be detected after image registration because they have significant peaks in their correlation graphs.

These results show that having the cover images for non-blind watermark detection cannot guarantee successful detection in distorted stego images. The desynchronized watermark information in the distorted stego images need to be resynchronized through image registration. They also show that image registration can improve the detector performance significantly.

(ii) Blind Watermark Detection
The second watermarking method developed by Barni et al. does not need reference images for watermark detection [6]. Hence, it is a blind watermark detection. However, we performed image registration using their cover images prior to watermark detection in order to demonstrate the effects of watermark synchronization. This method transforms test images into DWT domain, and
computes correlation and threshold values for watermark detection. If the correlation values exceed the threshold values, watermarks are considered detected. Otherwise, watermarks are not detected. Details of this watermark detection method are described in Section 4.2.2.

Figure 3.6 Effects of RBA bending factor on watermark detection
3.3 Motion Estimation for Synchronization

(c) Stego *Cameraman*

Figure 3.6 Effects of RBA bending factor on watermark detection (cont.)

Figure 3.7 shows the effects of image registration on blind watermark detection using *Pepper* and *Cameraman*. The graphs are plotted for RBA with bending factors ranging from 0.1 to 1.5 with 0.2 intervals. They show that the correlation values in the distorted stego images are improved (i.e. higher values) after image registration. These values are higher than their dynamically calculated threshold values, thus watermarks were detected with the aid of image registration. Test results of other images are also improved significantly after the distorted stego images are registered.
Figure 3.7 Effects of image registration on blind watermark detection

(a) Pepper watermark detection

(b) Cameraman watermark detection
Summary

The differential affine motion estimation method worked well in image registration for both blind and non-blind watermark detections. Modelling the random geometric distortions between images as locally affine but globally smooth, the method correctly resynchronized the watermarks. As a result, the watermarks were detected in the registered images. Nevertheless, this method involves high computational cost given the iterative multi-level processing. In the next section, we will examine another image registration method with lower computational cost.

3.4 Flowline Curvature for Synchronization

Despite the good results offered by the motion estimation method in watermark synchronization, it has high computational cost. One way to reduce the algorithm complexity dramatically is to use a small number of robust features for watermark synchronization. For instance, a sharp edge between two homogeneous regions with different textures may have a robust corner, and this robust corner can be used to realign its surrounding regions so that the robust corner is positioned at its original location prior to a geometrical distortion. The correctly registered image will enable successful detection of its resynchronized watermark. Focusing on the fundamental geometrical operations, we will investigate a new image registration method to combat RST distortions. It uses flowline curvature as robust features to invert the distortions. It also utilizes scale normalization to attain robustness against the distortions.

Recalling the equation (3.2) for affine transformation, \((x',y')\) is the transformed coordinate of \((x,y)\) with parameters \((A,d)\) \(\in\mathbb{R}\). By setting the moment centroids of images as the origin \((0,0)\) of its coordinate system, we can eliminate the translation parameter \(d\). Details of this operation will be explained in the “Scale Normalization” section below. With this, we only need 4 sets of parallel equations to solve for the rotation and scaling parameter \(A\) to perform a RST resynchronization. Although we model the RST distortions at the global scale here, it can be easily adapted to local distortions by dividing the image into non-overlapping blocks for individual processing.
3.4.1 Scale Normalization

The scale normalization process transforms an image into a format with standard geometrical properties [42]. Scale normalization provides scale invariance and translation invariance. Given the geometric moments, $m_{p,q}$, of a gray scale image $I(x,y)$ as follows:

$$m_{p,q} = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} x^p y^q I(x,y).$$  \hfill (3.18)

The central moments are

$$\mu_{p,q} = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q I(x,y),$$  \hfill (3.19)

where

$$\bar{x} = \frac{m_{1,0}}{m_{0,0}}, \quad \bar{y} = \frac{m_{0,1}}{m_{0,0}},$$  \hfill (3.20)

and $(\bar{x}, \bar{y})$ is the centroid of the image. During scale normalization, an image is transformed into unit aspect ratio with a predefined area. To obtain unit aspect ratio, we use $g l_x = h l_y$, where $l_x$ and $l_y$ are the pixel count of height and width of the image $I(x,y)$, $g$ and $h$ are scaling factors to produce the rescaled image $I\left(\frac{x}{g}, \frac{y}{h}\right)$ with area $\alpha = (gl_x)(hl_y)$. From equation (3.18), the zero order moment $\beta$ of the scaled image becomes

$$\beta = ab m_{0,0}$$  \hfill (3.21)

where $m_{0,0}$ is the zero order moment of the cover image. We can solve for $a$ and $b$:

$$a = \sqrt{\frac{\beta y}{m_{0,0}}}, \quad b = \sqrt{\frac{\beta}{m_{0,0}}}$$  \hfill (3.22)

where

$$\gamma = \frac{l_y}{l_x}.$$  \hfill (3.23)

Besides rescaling the image into aspect ratio 1 with area $\alpha$, we need to translate its origin to the centroids $(\bar{x}, \bar{y})$ and change the coordinates into $(x', y')$:

$$x' = \frac{(x - \bar{x})}{a}, \quad y' = \frac{(y - \bar{y})}{b}.$$  \hfill (3.24)
3.4 Flowline Curvature for Synchronization

3.4.2 Flowline Curvature

In addition to scale normalization, our watermarking method also exploits an important property known as flowline curvature. The concept of flowline curvature can be explained as follows.

The derivatives of an image can provide RST invariant features in designing a robust watermarking method. For example, an approximation of an image’s intensity surface can be done using a first derivative such as its gradient. Normally, higher order derivatives give a more precise approximation of a local surface. For digital images, derivatives are computed using weighted kernel convolution. The convolution process involves filtering an image by moving a filter across the image in the horizontal and vertical directions. The weighted kernel is the filter, which is a small matrix with various values. We apply Gaussian derivatives, \(G\), at multiple scales (i.e. many kernel sizes) to obtain more precise approximations:

\[
G = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.25)
\]

where \(\sigma\) is the scale of the filter. Multi-scale derivatives at a point \(p\) of an image are obtained by filtering the image with Gaussian derivatives. The multi-scale two-jet is

\[
J^2_{\{\sigma_1,\ldots,\sigma_k\}}[I] = \{J^2[I](p,\sigma_1), J^2[I](p,\sigma_2), \ldots, J^2[I](p,\sigma_k)\} \quad (3.26)
\]

for set of scales \(\sigma_1,\ldots,\sigma_k\) and it is translation invariant. Given a coordinate frame \((u,v)\) centered at point \(p(x,y)\) in an image \(I\), the local gradients \(I_x\) and \(I_y\) are obtained using

\[
I_x = I \ast G_x, \quad I_y = I \ast G_y. \quad (3.27)
\]

where \(G_x\) and \(G_y\) are the Gaussian derivatives in the \(x\) and \(y\) directions. The first order and second order derivatives using a rotation matrix \(A\) are:

\[
\begin{bmatrix}
I_u \\
I_v
\end{bmatrix} = A^T \begin{bmatrix}
I_x \\
I_y
\end{bmatrix}
\]

\[
= \frac{1}{\sqrt{I_x^2 + I_y^2}} \begin{bmatrix}
I_y & -I_x
\end{bmatrix} \begin{bmatrix}
x \\
y
\end{bmatrix}
\]
We define the image flowline curvature as

\[ F_i(p) = \frac{I_x I_y (I_{xx} - I_{yy}) + I_{xy} (I_{y}^2 - I_{x}^2)}{(I_x^2 + I_y^2)}. \]  

(3.30)

The flowline curvature corresponds to curvature of the gradient integral curve at a point \( p \). Figure 3.8 (a) illustrates the computation results of flowline curvature for \( Lena \) at scale 1. The intermediate results \( I_x, I_y, I_{xx}, I_{xy}, \) and \( I_{xy} \) are shown in Figure 3.8 (b) – (f) respectively.

With the basic concepts explained, we move on to describe the watermarking method. The embedding process and detection process are described separately.

### 3.4.3 Watermark Embedding

Figure 3.9 depicts an overview of the watermark embedding method. We find the most robust corners of the cover image and normalize its scale before embedding the watermark into the image. Beside that, flowline curvatures of the robust corners are also computed and recorded for use in watermark detection. The marked image is then restored to its original size and sent to its destination.
Figure 3.8 Examples of flowline curvature computation (a) Flowline curvature $F_1(p)$, (b) $I_x$, (c) $I_y$, (d) $I_{xx}$, (e) $I_{xy}$, (f) $I_{yy}$
The first step in the watermark embedding process is Robust Corner Detection. We use the modified Harris Corners Detector [63] proposed by Simitopoulos [64] to detect the robust corners. There are a few changes made to the method of extracting robust corners. Instead of using the radius-shape enhancements proposed by Simitopoulos [64], a square area is applied to eliminate weak corners within a fixed neighborhood. In this way, we trade-off a little bit of robustness with simplicity in computation. In addition, the corners near the edges of the image are discarded. For example, a corner that falls within 16% of the image width or height from the edges is not considered as robust corner. Such rule effectively eliminates image rotation, edge cropping, and minor translation distortions. Furthermore, we enlarge the corners detected in each scaling and rotation distortions before it is sent to the union operation. This operation retains the corners after the interpolation operation of image scaling and rotation. For instance, if this step is not performed, then a corner in a scaled-up image may be lost when it is rescaled to its original size. In addition, we control the number of robust corners by adjusting the parameters in Harris Corner Detector, i.e. sigma and threshold values. These corners serve well as reference points in RST resynchronization of the watermark detection because they are based on salient features of an image. We apply five rotation distortions and four scaling distortions on the images to find the robust corners. The rotation angles are $3^\circ$, $5^\circ$, $10^\circ$, $15^\circ$, and $30^\circ$. The scaling factors are 0.5, 0.75, 1.25, and 1.5. These values represent small changes normally applied by an attacker to remove watermarks while preserving the image quality. From the set of robust corners computed, we select two robust corners with highest output values as reference points because important features of an image are usually placed here. An example of robust corners selected in Lena is showed in Figure 3.10. The first robust corner is located at the left eye.
3.4 Flowline Curvature for Synchronization

and the second robust corner is located at her chin. Both of them are marked with plus signs (+) in light colour.

Scale normalization is performed on the cover image to produce an image with unit aspect ratio and predefined area. Then, the corresponding coordinates of the two robust corners selected in the normalized image are calculated using the equation (3.24).

The next step in the watermark embedding process is flowline curvature computation. Flowline curvature of the normalized image is computed as described in Section 3.3.3. We compute the local two-jet, $(I_x, I_y, I_{xx}, I_{xy}, I_{yy})_\sigma$ at fixed scales $\sigma_1 = 1.0$ and $\sigma_2 = 1.5$, where $I_x$ and $I_y$ are the filter response of $I$ to the first derivative of Gaussian in directions $x$ and $y$ respectively; $I_{xx}, I_{xy},$ and $I_{yy}$ are the second derivative responses [65]. After that, we compute flowline curvatures for each robust corner selected with the two fixed scales using equation (3.30). The reference point set, $REF_p$, in spatial domain is stored for use in watermark detection.

![Figure 3.10 Two robust corners selected as reference points. They are marked by the white cross signs.](image)

$$REF_p = \{REF_{p_1}(x,y,F_1(p_1)_{\sigma_1,\sigma_2}), REF_{p_2}(x,y,F_2(p_2)_{\sigma_1,\sigma_2})\} \quad (3.31)$$

where $p_1$ and $p_2$ are the robust corners at coordinate $(x,y)$ selected, and $F_i(p_i)$ is the flowline curvature for robust corner $i$.

Following the three previous steps, a watermark signal $m$ is encrypted with a key $K$ and embedded into the DFT domain of the scaled image. It is assumed that cryptographic protocols and their related infrastructures for key sharing between the
watermark embedder and the watermark detector can be used. The watermark is embedded in the 1000 DFT coefficients with highest magnitude using the classical spread spectrum watermarking method [61]. Instead of employing a complex embedding method, this is done to simplify the embedding process and focus on image geometry recovery. The additive embedding formula is

\[ x'_i = x_i + \alpha m_j \]  

(3.32)

where \( \alpha \) is the strength factor, \( x' \) is the marked coefficient of \( x \), \( m \) is the watermark signal bit. The value of \( \alpha \) must be carefully chosen to maximize imperceptibility and detection of the watermark. The embedding of watermarks must avoid the two robust corners selected because we will take it as reference points in RST resynchronization of the watermark detection. After that, the marked image is restored to its original size using simple interpolation operation before it is sent out to its destination.

### 3.4.4 Watermark Detection

The watermark detection part is illustrated in Figure 3.11. The beginning steps, Robust Corner Detection, Scale Normalization, and Flowline Curvature Computation, are similar to those of the embedding process. The parameters extracted from the received image are then compared to those of the cover image in order to achieve automated RST resynchronization. Then, the watermark can be detected in the synchronized image with ease.

![Figure 3.11 Robust watermark detection using flowline curvature](image)

The watermark detection process takes the marked image as input. After robust corner detection, scale normalization, and flowline curvature computation, all of the robust corners detected in the received image are compared to the two robust corners of watermark embedding process using brute force search. This is made possible with the small number of robust corners given by the modified Harris Corner
3.4 Flowline Curvature for Synchronization

Detector. The search is done using the reference point set information for the corners at the same scales. The most similar pair of robust corners are then selected and used in RST resynchronization. This is carried out by finding the minimum error function $E$:

$$
E = |I_{REF} - I_{wREF}|^2
$$

(3.33)

where

$$
I_{REF} = [F_i(p_1)_{\sigma_1}, F_i(p_1)_{\sigma_2}]^T, \quad I_{wREF} = [F_{i_w}(p_1)_{\sigma_1}, F_{i_w}(p_1)_{\sigma_2}]^T.
$$

(3.34)

RST resynchronization is obtained with parameter $A$ in equation (3.2). We can solve four parallel equations for the parameters using two robust corners in the marked image and its two corresponding robust corners in the cover image.

Once the distorted image is synchronized with the cover image, we can detect the watermark in its DFT domain using identical computations. The 1000 highest magnitude DFT coefficients are selected for watermark extraction using a reversed operation:

$$
m_j = \frac{x_j - x_i}{\alpha}.
$$

(3.35)

Finally, the secret key $K$ is used to decrypt the signal extracted. Successful watermark detection and its signal extraction are measured using a correlation technique. For example, the correlation value between the extracted signal and its embedded signal should be very high compared to the correlation value between randomly generated signals with the embedded signal. This can be visualized by plotting a graph of the correlation values.

### 3.4.5 Analysis of Experiment Results

We tested the watermarking method under various common distortions such as rotation, scaling, translation, JPEG compression, Gaussian noise addition, smoothing by low pass filtering, region cropping, and contrast adjustment. Imperceptibility of the stego image and distorted images are measured using PSNR and WPSNR values. WPSNR have been shown to be more accurate in measuring the perceptual similarity between images [19]. However, it should be noted that WPSNR is not widely accepted in the watermarking community. An initial test on Lena shows a stego image in Figure 3.12 with PSNR measuring 54.38 dB and WPSNR measuring
60.86dB. These values indicate that the stego Lena has good watermark imperceptibility. The watermark was detected successfully with a significant peak in its response graph.

![Stego Lena](image)

Figure 3.12 Watermark detection of stego image without distortions

(i) RST Distortions

Single distortions of rotation, scaling, and translation were carried out. In all cases, the method described was able to recover image geometry for watermark detection.

Rotational distortions at 5, 10, 15, 20, and 25 degrees around image center were done. These distortions produced rotated images with WPSNR ranging from 21.10 dB down to 17.12 dB. The method was able to recover the original orientation with the two robust corners correctly selected. The robust corners are marked with “⊕”. The results are shown in Figure 3.13. However, no peaks exist in the watermark detector response graph because the simplified watermarking method employed requires exact synchronization. The reverse rotation caused interpolation errors
although the image orientation was correctly restored. Therefore, the watermark cannot be extracted precisely.

Translational distortions ranging from “2 rows and 4 columns shift” to “6 rows and 12 columns shift” can be recovered with response peaks valued at about 30. In these cases, the distorted images have WPSNR value ranging from 22.29 dB to 18.39 dB. Distortions greater than “6 rows and 12 columns shift” cannot be recovered although the robust corners can be correctly located. The watermark information lost due to the translation is too large, and it caused the correlation value to fall below a significant peak. Figure 3.14 shows a “6 rows and 12 columns” translational distortion. Summary of the tests are graphed in the figure.

(a) Matched robust corners denoted by two dots. Zoom-in versions of the images are shown in the upper panels

(b) Watermark detector response.
Figure 3.13 Watermark detection under rotational distortion
Scaling distortions were done using scales ranging from 0.85 to 1.85. Peaks were clearly distinguishable at values ranging from 18 to 20. The graph in Figure 3.15 shows that peaks were clearly detected in all tests. Randomly chosen watermarks have very low correlation values compared to these peaks.

(ii) Combined RST Distortions
Following the tests using single distortions, a set of combined RST distortions were performed. Table 3.1 summarized the combination of RST distortions performed.

This method was able to register the distorted images correctly and provide a peak in its response graphs. Figure 3.16 shows the Combined Distortion Set 4 and its resynchronized image.

Figure 3.14 Watermark detection under translational distortions
Figure 3.15 Summary of detector responses under scaling distortions

Table 3.1 Combined geometrical distortions

<table>
<thead>
<tr>
<th>Set</th>
<th>Combined RST distortions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Translate (4,2) Rotate 10 degree -</td>
</tr>
<tr>
<td>2</td>
<td>Scale 0.85 Rotate 5 degree -</td>
</tr>
<tr>
<td>3</td>
<td>Translate (6,2) Scale 1.15 -</td>
</tr>
<tr>
<td>4</td>
<td>Translate (6,2) Rotate 10 degree Scale 1.05</td>
</tr>
</tbody>
</table>

(a) Stego image distorted with manipulation in set 4 of Table 3.1

(b) Registered stego image

Figure 3.16 Image synchronization for combined RST distortions
(iii) General Image Processing

In addition to the RST distortions, experiments were also conducted using a set of general image processing operations listed in Table 3.2.

Table 3.2 General image processing tests

<table>
<thead>
<tr>
<th>Set</th>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>JPEG compression</td>
<td>Quality factor ranging from 90% to 50%</td>
</tr>
<tr>
<td>2</td>
<td>Noise insertion</td>
<td>Gaussian noise with mean 0 and variance ranging from 0.0002 to 0.0010</td>
</tr>
<tr>
<td>3</td>
<td>Cropping</td>
<td>Clip out $11 \times 11$ pixels region, and crop off 25 pixels along each border of an image</td>
</tr>
<tr>
<td>4</td>
<td>Contrast adjustment</td>
<td>Gamma ranging from 0.98 to 0.90</td>
</tr>
</tbody>
</table>

For JPEG compression with quality factor declining from 90% to 50%, WPSNR between the stego image and the distorted image degraded from 64.38 dB to 51.71 dB, and the peak of detector response decreased from 28.48 to 14.62. Despite the deteriorated image quality, the watermark was detected in all cases with obvious peaks in detector responses.

Under Gaussian noise insertion, WPSNR reached a minimum of 45.02 dB when variance is set to 0.0010. However, the watermark was detected with peak response at 8.31.

Figure 3.17 depicts two cropping samples. Pictured on the left is a square region of $11 \times 11$ pixels cropped out. Depicted on the right is a 25-pixel stripe along each border cropped out and replaced with those of the unmarked image. This region is indicated by the dark colour lines. In both tests, the watermark was detected.

The contrast adjustment test provided good result with peak response at 7.79 when WPSNR degraded to 35.62 dB.

(a) Square region cropped off  
(b) Edges cropped and replaced with cover image sections

Figure 3.17 Samples of image cropping attacks
(iv) Tests on other images
Besides Lena, the experiments on Cameraman and Pepper also gave good results in robust corner detection. The robust corners are marked with cross sign (“+”) in white colour. Figure 3.18 (a) shows two robust corners detected in Cameraman, one on the left stand and another on the right stand of the tripod. In Figure 3.18 (b) for Pepper, the robust corners are at the tip of the pepper stem, and at the border between two peppers.

![Figure 3.18 Robust corner detection samples](image)

(a)              (b)

Figure 3.18 Robust corner detection samples (a) Robust corners detected in Cameraman, (b) Robust corners detected in Pepper.

Summary
The image registration method using flowline curvature and scale normalization can assist in watermark detection by synchronizing the watermarks. Robust corners are computed using flowline curvature of the image. Two of the most robust corners are selected to resynchronize an image that underwent RST transformations. The invariant properties of feature points are exploited in image resynchronization to minimize computational cost. Instead of searching through hundreds of corners [41], this method only needs to work on four robust corners. They are two robust corners recorded during watermark embedding, and another two robust corners computed in watermark detection. Compared to the motion estimation method in Section 3.3, this method reduces the computational cost dramatically. For example, a cycle of watermark embedding and detection in Matlab scripts using flowline curvature and scale normalization takes less than a minute on a Pentium III 800MHz machine with 256 MB of memory.
The developed watermarking method is able to resist single geometrical operations as well as combined geometrical manipulations. In addition, the method is able to resist general image manipulation operations such as JPEG compression, noise insertion, regional cropping, and contrast adjustment.

Interpolation in scaling and rotation caused imprecision in geometric resynchronization through registration. Therefore, perfect resynchronization is impossible. However, the objective of geometric resynchronization is to aid the detection of watermark, not the perfect restoration of image quality.

The dependency on the two robust corners selected makes this watermarking method robust against global transformations. However, local transformations need further corrections. To overcome this limitation, we can divide an image into non-overlapping blocks and process each block separately [106]. A major drawback of this method is the requirement of a reference image for synchronization.

### 3.5 Chapter Summary

Geometrical robustness remains as one of the major challenges. This is because geometrical distortions can be implemented easily, and the distortions can cause destructive effects on the robust watermark. Therefore, many robust watermarks can be defeated by geometrical distortions. There are two major approaches to robust watermarking, i.e. synchronization and invariant domain. We investigated the first approach in this chapter.

We developed two synchronization methods to warrant successful watermark detection. The first method uses differential affine motion estimation for image registration. It can capture both global and local distortions concurrently using a multi-level framework. However, it requires high computational cost. Therefore, we developed the second method for robust watermarking through synchronization. It uses flowline curvature and scale normalization to combat RST distortions at the global scale. It can be adapted to tackle local distortions by processing small blocks of an image. Experimental results showed that the synchronization methods performed well for robust watermarking.

The methods based on synchronization unavoidably suffer from image fidelity loss due to interpolation errors during the image registration step. Moreover, the robust features could be cropped-off inadvertently. Therefore, we will discuss
3.5 Chapter Summary

another approach of robust watermarking that utilizes invariant domains in the next chapter. This approach does not rely on robust features and eliminates synchronization. Furthermore, we will also investigate blind watermark detection to overcome the requirement of reference images during watermark detection.
Chapter 4

Robust Watermarking using Invariant Domain

The robust watermarking methods discussed in the previous chapter used non-blind watermark detection. They require reference images for watermark synchronization. Although the methods are robust against many image distortions, reference images may not be readily available during watermark detection in real life scenarios. Therefore, they have limited practical values. In this chapter, we will discuss a robust watermarking method that offers blind watermark detection. The method also provides good imperceptibility and low computational cost. Section 4.2 describes this blind watermark detection method. We will also adapt the blind watermark detection method in a geometric invariant domain we created. Our geometric invariant domain is very robust against RST distortions and common image processing operations. In addition, we will also evaluate the false positive probability of the invariant domain using an image database.

4.1 Introduction

Robust watermarking methods can be broadly categorized into synchronization-based and invariant domain approaches. The former approach is easy to design and implement. There exist many image registration methods that can be adopted for
Chapter 4 Robust Watermarking using Invariant Domain

Synchronization. Their algorithms could be simple but the amount of computation can be high. One such example is the differential affine motion estimation method studied in Chapter 3. The algorithm simply repeats similar steps in an iterative loop. As a result, the computational cost of this iteration is high. On the other hand, synchronization using robust features has lower computational cost but the robust feature itself could be the target of attack. For example, the flowline curvature method discussed in the previous chapter fit into this type of watermarking method.

The other approach of robust watermarking utilizes transform invariant domain to achieve robustness. This approach eliminates the need of synchronization steps and does not depend on image features. We investigate a geometric invariant domain in Section 4.3. Before developing the geometric invariant domain, we will firstly inspect a blind watermark detection method. Then we will adapt the blind watermark detection method for use in the invariant domain. The geometric invariant domain watermarking with blind watermark detection that we developed is highly practical because there is no synchronization required and reference images are not necessary during watermark detection. The watermarking method also performs well in terms of robustness and imperceptibility.

4.2 Blind Watermark Detection in the Wavelet Domain

The blind watermark detection method we inspected was developed by Barni, Bartolini, and Piva [6]. The method embeds and detects watermarks in the wavelet domain. It also utilizes perceptual masking during watermark embedding to enhance the imperceptibility of the watermark. In order to have good performance in terms of robustness, imperceptibility, and computational simplicity, we created an improved embedding method based on Barni’s algorithm with minimal effects on the blind watermark detection. Our method maintained the watermark robustness and imperceptibility while significantly reduces the computational costs.

Wavelet-based image watermarking methods exploit the frequency information and spatial information of the transformed data in multiple resolutions to gain robustness. Digital watermarks that use wavelet transforms have been experimented by some researchers recently [66-70]. DWT is widely explored in watermarking due to its simplicity. For instance, every highpass sub-bands and lowpass sub-bands in a wavelet packet transform is further decomposed at each resolution, whereas only the
4.2 Blind Watermark Detection in the Wavelet Domain

The lowpass sub-band in DWT is further decomposed at each resolution. The advantages of wavelet transform compared to DCT and DFT were mentioned in [68]. For example, DWT has multi-resolution information and spatial information. These information are not available in DCT and DFT. However, there is still room for improvement. For instance, HVS properties can be exploited to enhance watermark embedding strength. Although the HVS model offers imperceptibility in wavelet-based watermarking, it suffers from high computational cost. We examine embedding strength determined by a HVS model, a constant, and our simplified method. The novel simplified embedding method significantly reduces embedding time while preserving the performance of imperceptibility and robustness. The fast embedding method exploits implicit features of DWT sub-bands, i.e. the luminosity information in the low pass band, and the edge information in the high pass bands. It achieves embedding speed comparable to a constant energy embedding process. Robustness is demonstrated with a few conventional attacks, e.g. JPEG compression, Gaussian noise insertion, image cropping, contrast adjustment, median filtering, and global geometrical distortion. Experimental visual quality is measured in WPSNR for high accuracy. Robustness and imperceptibility of HVS-based embedding could be trade-off with computational simplicity of a fast embedding method.

The performance of DWT-based watermarking methods depends on the overall watermarking method as well as embedding and detection methods. Recalling the discussions in Chapter 2, we focus on three essential factors of a watermarking scheme, i.e. robustness, imperceptibility, and computational cost. Enhancing watermark robustness often requires a trade-off in watermark imperceptibility and computational cost. A well-tuned watermarking method that offers robustness, imperceptibility, and computational simplicity remain a big challenge [71]. One of the factors that steers the balance point is the embedding strength in an additive embedding method. Hence, an analysis of DWT-based watermarking method focusing on its embedding strength would provide useful insight in how to improve its performance. The performances measured include robustness, imperceptibility, and computational cost. Embedding strength refers to the magnitude of watermark signal inserted into the wavelet domain of a cover image. Particularly, we are interested in analysing the performance of the watermark under HVS-based embedding method and our simplified embedding method. HVS models enable
adaptive strength embedding of a watermark to gain robustness while maintaining its imperceptibility. It considers sensitivity of the human eye to noise, luminosity and textures. For instance, higher embedding strength can be applied to complex regions of an image. The novel method aims at achieving similar results with less computation. The method mimics a HVS model in adjusting the watermark embedding strength using the implicit features of DWT sub-bands.

We evaluate three of the essential elements of a robust watermarking method, i.e. robustness, imperceptibility, and computational cost under different embedding strengths. High robustness often offsets the imperceptibility of a watermark. Furthermore, a computationally simple watermarking method usually cannot attain desirable robustness and imperceptibility. Ideally, a watermarking method should achieve a balance among these mutually conflicting requirements. While the variance-based mask [68, 72] uses local sub-band variance to increase watermarking energy, its non-overlapping blocks with fixed block size made it “rigid”. Furthermore, the non-blind watermark detection of the method made it less practical than blind watermark detection. Another wavelet-based watermarking method that exploits HVS is mentioned as image-adaptive wavelet (IA-W) method in [73]. The major drawback of the method is the requirement of an original test image in watermark detection, thus reduces its practicality in real life scenario.

In this study, a recent DWT-based image watermarking method [6] is chosen based on its overall performance. The method embeds a watermark in all the high pass bands of the DWT domain. This is due to the good imperceptibility provided by high pass bands. The human eyes are less sensitive to changes made to the wavelet coefficients in these bands. On the other hand, embedding watermark in the low pass band could easily cause visual artefacts because the low pass bands correspond to flat regions in an image. Changes made on these regions can be easily spotted by the human eyes. The embedding method exploits an adaptive weighting of the HVS model. Although the HVS model offers imperceptibility in wavelet-based watermarking, it suffers from high computational cost. We investigate the performance factors of the watermarking method and analyse the results. The performance of three cases is compared: embedding with a HVS model, a constant strength [73], and our adjustable-strength based on a simplified model. The constant
4.2 Blind Watermark Detection in the Wavelet Domain

strength embedding method is used as a baseline comparison for the performance factors.

4.2.1 Watermark Embedding

Common watermark embedding strategies are additive, multiplicative, and quantization. Additive embedding is simple and fast. It usually takes the form \( I' = I + \alpha m \) where \( I' \) is the stego image, \( I \) is the cover image, \( \alpha \) is the embedding strength, and \( m \) is the watermark signal. Using the same set of notations above, multiplicative embedding is implemented with \( I' = I \times \alpha m \). Quantization embedding can be implemented using a quantization method. For example, the coefficients of a DWT can be divided into 2 sets similar to the odd/even number concept. One set of the coefficients represent bit ‘0’ while the other set of the coefficients represent bit ‘1’. This embedding method is more complex compared to the additive and multiplicative embedding. For instance, the step size which determines the gap between the 2 sets of coefficients must be carefully chosen by analysing the range of coefficient values. A large step size would degrade the watermark imperceptibility while a small step size would affect the watermark robustness. In this study, we will use additive embedding for low computational cost.

The watermark embedding steps for the three methods studied are similar. It begins with an image decomposition using DWT, followed by embedding strength computation using the respective methods, and finishes with an inverse DWT (IDWT) that reconstructs the stego image. Figure 4.1 depicts the processes involved.

![Figure 4.1 Watermark embedding using the three examined methods](image-url)
The robust DWT-based watermarking method [6] embeds watermark information in the DWT domain. It also incorporates an embedding weight factor that exploits the HVS characteristics. This adapts embedding strength according to the changes of image texture, edge distance, noise sensitivity and local luminosity. Therefore, the method gained robustness and imperceptibility simultaneously. Firstly, an image is decomposed into its highpass and lowpass sub-bands using DWT with a Daubechies-6 filter. Other types of wavelet filters can also be used. A 4-level decomposition with its sub-bands is sketched in Figure 4.2. Each of the \( l \)-th level of the decomposition consists of three directional high pass bands \( I^0_l, I^1_l, I^2_l \) and a low pass band \( I^3_l \). To avoid analytical attacks, a watermark is usually embedded in all the high pass bands instead of some of the sub-bands.

![Figure 4.2 Four-level DWT decomposition](image)

(i) HVS Embedding Method
Details of the HVS-based watermarking method are presented in [6], and summarized here. The watermark is embedded in the three high pass bands at level 0 using the following equation.

\[
I^\theta_0 (i, j) = I^\theta_0 (i, j) + \alpha w^\theta (i, j) x^\theta (i, j)
\]  

(4.1)

where

- \( \theta \in \{0,1,2\} \) is the high pass sub-band selection,
- \( I^\theta_0 (i, j) \) is the original sub-band coefficients,
- \( I^\theta_0 (i, j) \) is the stego sub-band of \( I^\theta_0 (i, j) \),
- \( \alpha \) is a global energy parameter that determines watermark embedding strength,
- \( w^\theta (i, j) \) is a weight function derived from local noise sensitivity which provides masking characteristics of the HVS,
4.2 Blind Watermark Detection in the Wavelet Domain

\( x^0(i, j) \) is a pseudorandom binary sequence, \( m_h \in \{+1, -1\} \) coded in two-dimensional array using equation (4.2).

\[ x^0(i, j) = m_{0(iN+1+j)} \]  

(4.2)

where \( \theta \in \{0, 1, 2\} \) is the high pass sub-band selection, and \( 2M \times 2N \) is the size of the cover image. The weight function \( w \) is an adaptation of DWT coefficient quantization used in image compression [74]. Considering noise sensitivity of the human eye, [6] proposed the weight calculation below:

\[ q^0_l(i, j) = \Theta(l, \theta) \Lambda(l, i, j) \Xi(l, i, j)^{0.2} \]  

(4.3)

where

\( \Theta(l, \theta) \) denotes noise sensitivity as shown in equation (4.4),

\[ \Lambda(l, i, j) = 1 + L(l, i, j) \]  

(4.5)

\[ K(l, i, j) = \frac{1}{256} I_3 \left( 1 + \left( \frac{i}{2^{3-l}} \right), 1 + \left( \frac{j}{2^{3-l}} \right) \right) \]  

(4.6)

\[ L(l, i, j) = \begin{cases} 1 - K(l, i, j) & \text{if } K(l, i, j) < 0.5 \\ K(l, i, j) & \text{otherwise} \end{cases} \]  

(4.7)

\[ \Xi(l, i, j) = \sum_{k=0}^{3-l} \sum_{\theta=0}^{1} \sum_{x=0}^{1} \sum_{y=0}^{1} I^0_{i+j} \left[ y + \frac{i}{2^k}, x + \frac{j}{2^k} \right]^2 \]  

\[ \cdot \text{Var} \left[ I_3 \left( 1 + y + \frac{i}{2^{3-l}}, 1 + x + \frac{j}{2^{3-l}} \right) \right]_{y=0, l, y=0, l} \]  

(4.8)

Assuming changes smaller than half of the calculated weights are imperceptible, the weight function \( w \) gives maximum embedding energy in the quantization of DWT coefficients using

\[ w^0(i, j) = q^0_l(i, j) \]  

(4.9)

From equation (4.1), it is apparent that the computed weight function \( w \) at each pixel enables HVS-based watermarking to obtain high level of imperceptibility and
robustness. However, the computations in equations (4.3) to (4.9) consume a large amount of resources, e.g. computation cycles and computer memory. Finally, IDWT is performed after the watermark embedding to produce the stego image.

(ii) Constant Energy Embedding Method
A constant energy embedding method is realized using a similar method, with the weight function \( w \) omitted.

\[
I_0^\alpha (i, j) = I_0^\theta (i, j) + \alpha x_0^\theta (i, j)
\]  

(4.10)

The value of \( \alpha \) in equation (4.10) has to be bigger than that of equation (4.1) to guarantee high embedding energy and warrant successful watermark detection. Obviously, this embedding method requires the least amount of computation compared to the HVS and the simplified methods. The constant energy embedding method is chosen as a baseline in this comparative study.

(iii) Simplified Embedding Method
To achieve a balance between the two extremes of the HVS and constant energy methods, we created a simplified embedding method. The Simplified embedding method significantly reduces embedding time while preserving the performance of imperceptibility and robustness. The fast embedding method exploits implicit features of DWT sub-bands. The DWT coefficients in the low pass band provide a good approximation of an image’s luminosity information. Also, the DWT coefficients in the high pass bands give an estimation of edges information of an image. Referring to equation (4.1), the Simplified embedding method employs a different weight function \( s \).

\[
I_0^\alpha (i, j) = I_0^\theta (i, j) + \alpha x_0^\theta (i, j)
\]  

(4.11)

where \( x_0^\theta (i,j) \) denotes the luminosity and edge information in an image; and other terms are the same as mentioned in equation (4.1). The weight function \( s \) is calculated using equations (4.12) and (4.13) below.

\[
s_0^\theta (i, j) = \frac{q_n^\theta (i, j)}{2}
\]  

(4.12)

\[
q_n^\theta (i, j) = \Theta (l, \theta) \Lambda (i, j) \Xi (i, j)
\]  

(4.13)

where \( \Theta (l, \theta) \) considers noise sensitivity as shown in equation (4.4),
4.2 Blind Watermark Detection in the Wavelet Domain

$\Lambda''(i, j)$ considers luminosity for gray levels in the lowpass subband with reference to equation (4.14). It makes sense to take a fraction of approximation values from the low pass band because the values indicate luminosity information. Our experimental outcome shows $\beta = 0.01$ gives good results,

$\Xi'^{\theta}_{0} (i, j)$ considers edges information using equation (4.15). As opposed to (4.3), this value is squared in (4.13) to provide a fast reduction of a value while considering edges information in each of the sub-bands. Our experiments show that $\delta = 0.005$ provides good results.

We traded-off texture information in (4.15) for computation speed.

An enhanced version of the Simplified embedding method omits the edge information totally by taking out $\Xi'^{\theta}_{0} (i, j)$ for faster computation. Our experiments show that the performance is similar to the original version because the small values of edges information in equation (4.13) have little effect on the weight function $s$.

The watermark capacity in all of the embedding methods mentioned above is the same. This is due to the same size of watermark pattern $x$ applied.

4.2.2 Watermark Detection

Regardless of the embedding method used, a cross-correlation method is adopted in blind watermark detection. This provides a fair comparison among the three embedding methods for robustness under various attacks. To detect the presence of a watermark signal $x$, we begin with a DWT operation on the stego image similar to the embedding process. Then, a cross-correlation value between the stego sub-band coefficients $I'$ and the watermark pattern $x$ is calculated using equation (4.16).

$$\rho = \frac{1}{3MN} \sum_{\theta=0}^{2} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I'^{\theta}_{0} (i, j) x^{\theta}(i, j)$$

(4.16)

It is worth mentioning that an adaptive threshold value $T_{p}$ is computed dynamically, avoiding the requirement of embedding strength factor $\alpha$. If $\rho > T_{p}$, then the watermark $x$ is present; otherwise it is absent. To ensure that the false positive probability does not exceed $10^{-8}$, the threshold $T_{p}$ is chosen as follows:
Chapter 4 Robust Watermarking using Invariant Domain

\[ T_p = 3.97 \sqrt{2\sigma^2_{\rho \sigma}} \]  

(4.17)

\[ \sigma^2_{\rho \sigma} \approx \frac{1}{(3MN)^2} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I_0^0(i, j))^2 \]  

(4.18)

The calculation of this threshold value is adapted from Neyman-Pearson criterion. The value 3.97 in equation (4.17) is based on mass tests result of [6].

4.2.3 Analysis of Experiment Results

(i) Experiment Settings

Using the watermark embedding and detection procedures explained in Section 4.2.2, a set of five common images were tested. The images are illustrated in Figure 3.3 of Chapter 3. They are all gray scale images with standard dimension 256×256 pixels. The images are identified by name: Baboon, Cameraman, Lena, Pepper, and Fishing boat.

Computational costs are compared by measuring the embedding time taken by each of the embedding methods. Intuitively, the HVS model has the highest amount of computation because the weight function calculation involves many summation/convolution operations. On the contrary, the constant energy embedding method should be the fastest because there is no computation of weight function.

To evaluate the imperceptibility quality of stego images, WPSNR of each stego image is measured. The collected data are interpreted in graphical form in the next section. WPSNR is chosen due to its higher accuracy over PSNR metric [19, 20].

Robustness tests were carried out with six types of conventional attack listed in Table 4.1.

By applying the embedding steps illustrated in Figure 4.1, a binary watermark pattern shown in Figure 4.3 below is embedded into the three high pass bands (i.e. the dark quadrants) of Lena.

Figure 4.3 Watermark and DWT sub-bands (a) Watermark Pattern \( x \), (b) Level 0 sub-bands of DWT for Lena
4.2 Blind Watermark Detection in the Wavelet Domain

Table 4.1 Attacks Used in Robustness Tests

<table>
<thead>
<tr>
<th>No</th>
<th>Attack type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>JPEG compression</td>
<td>Quality factor 85%, 70%, 55%, 40%, and 25%</td>
</tr>
<tr>
<td>2</td>
<td>Gaussian noise insertion</td>
<td>Zero mean, variance are 0.0002, 0.0004, 0.0006, 0.0008, and 0.0010</td>
</tr>
<tr>
<td>3</td>
<td>Cropping</td>
<td>8×8, 16×16, 32×32, 64×64, and 128×128 squares at image centre cropped out</td>
</tr>
<tr>
<td>4</td>
<td>Contrast adjustment</td>
<td>Gamma 0.8</td>
</tr>
<tr>
<td>5</td>
<td>Median filtering</td>
<td>2D median filtering using 3×3 neighbourhood</td>
</tr>
<tr>
<td>6</td>
<td>Global geometrical</td>
<td>3 degree rotation at image centre, random bending.</td>
</tr>
</tbody>
</table>

(ii) Embedding Time Evaluations

Experiments showed that the Simplified embedding method takes as little time as the constant energy embedding. On the other hand, the HVS embedding method requires more than 55 times longer processing duration. Table 4.2 shows the embedding time for the images processed. The detection of watermark prior to attacks was done on each of the embedded images. In all cases, the watermarks were detected successfully.

Table 4.2 Embedding Time of the Three Embedding Methods

<table>
<thead>
<tr>
<th>Image</th>
<th>HVS</th>
<th>Simplified</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baboon</td>
<td>62.370</td>
<td>1.222</td>
<td>1.111</td>
</tr>
<tr>
<td>Cameraman</td>
<td>61.398</td>
<td>1.172</td>
<td>1.071</td>
</tr>
<tr>
<td>Lena</td>
<td>61.209</td>
<td>1.182</td>
<td>1.072</td>
</tr>
<tr>
<td>Pepper</td>
<td>61.089</td>
<td>1.182</td>
<td>1.072</td>
</tr>
<tr>
<td>Fishing boat</td>
<td>61.209</td>
<td>1.182</td>
<td>1.072</td>
</tr>
</tbody>
</table>

(iii) Imperceptibility Evaluations

The stego images produced by each of the embedding methods are measured their WPSNR value. Figures 4.4 to 4.8 depict the stego images and its WPSNR values. For these tests, we used the enhanced version of the Simplified embedding method because the influence of edge information is very small on the total embedding strength \( \alpha^*(i,j) \). The \( \alpha \) values selected for the HVS, constant energy, and Simplified
embedding methods are 4.5, 1.5 and 2.2 respectively. In addition, the Simplified embedding method uses $\beta = 0.01$. Such arrangements are necessary since the major interest is in performance factors comparison. Although the $\alpha$ values of the embedding methods are different, the effective embedding strengths after multiplication with its respective weight functions do not differ much.

A visual comparison of imperceptibility in graphical form is presented in Figure 4.9. Constant energy embedding method has the lowest visual quality overall, and the HVS embedding method achieves the highest visual quality in general. It is also noticed that the Simplified embedding method obtained visual qualities slightly lower than the HVS embedding method.

![Figure 4.4 Set of Baboon embedding results with their respective WPSNR](image)

Figure 4.4 Set of *Baboon* embedding results with their respective WPSNR
4.2 Blind Watermark Detection in the Wavelet Domain

Figure 4.5 Set of *Cameraman* embedding results with their respective WPSNR

Figure 4.6 Set of *Lena* embedding results with their respective WPSNR
Figure 4.7 Set of Pepper embedding results with their respective WPSNR

Figure 4.8 Set of Fishing boat embedding results with their respective WPSNR
(iv) Robustness Evaluations
All of the embedded images were attacked with the six operations listed in Table 4.1. For the first 3 types of attacks, five levels of attack described in the table were performed. Samples of various attacked images are presented in Figure 4.10. The original Lena image is shown in 4.10 (a). A JPEG compression with quality factor 25% on HVS embedded image is depicted in 4.10 (b). Gaussian noise with variance 0.001 is inserted into the Simplified embedded image and illustrated in 4.10 (c), and 4.10 (d) represents a 32×32 pixels region cropping on constant energy embedded image. Contrast adjustment with Gamma set to 0.8 on the HVS embedded image is printed in 4.10 (e). Figure 4.10 (f) is a two-dimensional median filtered image on the constant energy embedded Lena, and it uses a 3×3 neighbourhood kernel. Lastly, a global rotation at 3 degree around the image centre in the anticlockwise direction on the Simplified embedded image is given in 4.10 (g).

Experimental results for all of the five images under all attack conditions listed in Table 4.1 are compiled. The robustness tests results are summarized in Table 4.3. For each of the JPEG compression, Gaussian noise insertion, and cropping attacks, every detection method is tested with 25 stego images. There are 5 levels of attack performed on every image. Therefore, there are 25 (5×5=25) test images for each embedding method. For each of the three remaining attacks listed in the table, every detection method is tested with 5 stego images. There is 1 level of attack performed on every image. Therefore, there are 5 (1×5=5) test images for each embedding
method. For JPEG compression attacks, constant energy embedding performed excellently since the energy chosen is strong enough in a trade-off for visual quality. However, HVS and Simplified embedding methods cannot resist high level of lossy compression. 

![Attacked Images](image)

*Figure 4.10 Samples of attacked images (a) Original Lena, (b) JPEG compression, (c) Gaussian noise, (d) Cropping, (e) Contrast adjustment, (f) Median filtering, (g) Rotation.*

<table>
<thead>
<tr>
<th>No</th>
<th>Attack type</th>
<th>Number of watermarks detected</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HVS</td>
</tr>
<tr>
<td>1</td>
<td>JPEG compression</td>
<td>21</td>
</tr>
<tr>
<td>2</td>
<td>Gaussian noise insertion</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>Cropping</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>Contrast adjustment</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>Median filtering</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Global geometrical distortion</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.3 Summary of Watermark Detection Result for All Levels of Attacks

We use the value $C = (\rho - T_p)$ to measure the “Competency” of watermark detection. A positive $C$ value indicates a watermark is detected and a negative $C$ value otherwise. A higher $C$ value means a higher “strength” of successful watermark detection. From Table 4.4, it is noted that HVS embedding method does not produce detectable watermark in *Cameraman* under JPEG compression attack.
with quality factor 55%. The watermark is also not detected in the HVS embedded Cameraman for lower quality factors 40% and 25%. In the experiments, watermarks were not detected for Cameraman images produced by Simplified embedding method when JPEG compression attack quality factor is set to 40% or 25%. These observations can be explained by the large areas of flat regions in Cameraman. Under HVS and the Simplified embedding methods, the watermark embedding strengths for flat regions are lower compared to highly textured regions. Therefore, the watermark embedding strength is too weak to resist severe JPEG compression.

All of the embedding methods were able to give positive results under three subsequent attack types: Gaussian noise insertion, regional cropping, and contrast adjustment. Some positive Competency values, $C$ were obtained under median filtering for all the embedding methods. In fact, all detections in Lena were successful under the attacks for HVS, constant energy, and Simplified embedding methods. However, none of the embedding methods supplied detectable stego images under median filtering for Baboon, Cameraman, Pepper, and Fishing boat. It is evident that global geometrical distortion remains a big challenge because none of the embedding methods is able to warrant a successful detection for all the test images.

Table 4.4 Competency Comparisons of the Three Embedding Methods under JPEG Compression Attack with Quality Factor 55%

<table>
<thead>
<tr>
<th>Image</th>
<th>Competency value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HVS</td>
<td>Simplified</td>
</tr>
<tr>
<td>Baboon</td>
<td>1.1961</td>
<td>1.3458</td>
</tr>
<tr>
<td>Cameraman</td>
<td>– 0.0016</td>
<td>0.1142</td>
</tr>
<tr>
<td>Lena</td>
<td>0.1740</td>
<td>0.1830</td>
</tr>
<tr>
<td>Pepper</td>
<td>0.6264</td>
<td>0.6225</td>
</tr>
<tr>
<td>Fishing boat</td>
<td>0.4942</td>
<td>0.5867</td>
</tr>
</tbody>
</table>

4.2.4 Discussions

The Simplified embedding method offers efficient computation with similar performance as the HVS embedding method. It offers a moderate option between two extremes of HVS and constant energy embedding methods.

Obviously, the HVS embedding method is very slow compared to constant energy embedding and the Simplified embedding method. This can be referred to the fact that weight function computation in the HVS embedding method involves many
complex convolution operations. Such calculation definitely increases with an increase in image size. The constant energy embedding method requires the least amount of computation compared to the HVS and Simplified methods. The Simplified embedding method significantly reduces embedding time while preserving the performance of imperceptibility and robustness. Its embedding speed is comparable to those of the constant energy embedding method.

Comparing the WPSNR values in the stego images of each embedding methods, it is clear that constant energy embedding gives lowest visual quality. This is due to the rigid energy level used. HVS embedding has highest visual quality since it has adaptive advantage in visual masking with its weight function calculations. The Simplified embedding method achieves HVS-comparable levels of imperceptibility, especially for Pepper and Fishing boat images.

Table 4.3 shows the constant energy embedding method is the most robust method. Due to the high level of embedding energy, it survives all levels of JPEG compression attacks. However, it traded-off imperceptibility for robustness.

Gaussian noise insertion, cropping, and contrast adjustment do not pose a treat to all embedding methods. Therefore, partial information retained in the attacked images helped with successful watermark detection. Severe level of median filtering caused the embedding methods to fail in watermark detection for most images. The major changes in filtered images caused its threshold values $T_p$ goes lower than its correlation value $\rho$. Although the HVS embedding method can resist partial geometrical manipulations such as implode and pinch operations [6], it cannot survive global geometrical distortions. The reason behind this is that the watermark detection step only requires a small piece of unchanged image area in order to succeed. However, such a requirement is not met in a global geometrical distortion.

Despite the simplicity in correlation-based watermark detection, its major drawback is its weakness under global geometrical transformations. Beside this, RBA and JPEG compression remain a big challenge for robust watermarking.

### 4.2.5 Conclusion

An efficient embedding method for DWT-based watermarking is demonstrated. The Simplified embedding method promises fast embedding speed with its computational simplicity. It attained competitive performance in terms of imperceptibility and
robustness in par with the HVS-based model. In addition, the practical advantages also lie in its fast embedding speed and blind watermark detection.

The DWT-based watermarking scheme is vulnerable to severe levels of JPEG compression and median filtering attacks. Furthermore, it is particularly weak under global geometrical attacks. This is due to the correlation nature of the detection method. Nevertheless, these weaknesses will be overcome using the geometric invariant domain described in the next section. The Simplified watermark embedding method developed will be adapted in the geometric invariant domain.

4.3 Geometric Invariant Domain

Basic geometrical attacks such as RST operations pose significant threats to image watermarking due to its ease of implementation and desynchronization effects. Compared to image resynchronization methods [41, 75], transform invariant approaches [76] has a major advantage. The latter is independent of image contents and its features. This property is desirable especially for images without distinctive features.

Invariant domain methods rely on the invariant properties of a transformed domain to resist distortions. Ruanaidh [76] developed a framework for RST invariant domain using the FM transform. However, the watermark detection described was a non-blind method. Later, another invariant domain method was developed in [77] that works on one-dimensional (1-D) signals with a small search space. However, it was not designed to resist cropping attack. Based on the FM framework, phase information was used to construct an invariant domain [78, 79]. Later, it was improved by using LPM and phase-only filtering [80]. However, the method still requires a resynchronization step. Following that, Radon and Fourier transforms were experimented in [81] but they require exhaustive search to resist scaling attack. In another attempt [82-84], the first FFT step in the FM framework was replaced with a robust centroid but the centroid itself could be the target of attack.

In summary, many attempts to create a RST invariant domain had been reported throughout the years. However, many methods are not truly invariant because they still need resynchronization in a small search space.

We developed a geometric invariant domain to resist RST distortions with three transforms; FFT, LPM, and Dual Tree - Complex Wavelet Transform (DT-CWT).
Shift invariance is obtained using FFT. Rotation and scaling invariance are achieved by taking the DT-CWT of a LPM output. Unlike most invariant schemes, our method eliminates explicit resynchronization. The method resists geometric distortions at both global and local scales. It is also robust against JPEG compression and common image processing. In addition, we adapted the blind watermark detection method investigated in the previous section in this geometric invariant domain. Therefore, the high computational cost of DT-CWT is compensated with our fast watermark embedding method. The geometric invariant domain also exploits perceptual masking property of the DT-CWT sub-bands to improve watermark imperceptibility. In addition, our work is also motivated by the lack of literature in robust watermarking that incorporates shift invariant wavelets. There are not many wavelets that have shift invariant property. Experiment on a large set of natural images is performed to demonstrate the robustness of the new scheme.

The DT-CWT is an important component of the Geometric invariant domain. Therefore, we will describe the DT-CWT and its advantages before discussing the watermark embedding and detection steps.

### 4.3.1 The Dual Tree-Complex Wavelet Transform

The conventional DWT decompose a signal into low and high frequencies using a binary tree structure. DT-CWT consists of two trees, each of it has linear phase filters that give the real and imaginary coefficients in its forward transform. Odd-length filters in one tree are paired with even-length filters in another tree. The final outputs are averaged to give approximate shift invariance. In its inverse transform, biorthogonal filters are applied in each tree separately. The filters used in the forward transform and inverse transform are almost orthogonal. Figure 4.11 depicts the structure of the DT-CWT decomposition and reconstruction processes. We can only achieve approximate shift invariance with DT-CWT because filters with compact support will not have zero gain in its stop bands in real life. This is also due to the little differences between the frequency responses of odd-length and even-length filters [85, 86].

Compared to the other FM-based watermarking methods, our method does not require any resynchronization. Our method also enables blind watermark detection...
4.3 Geometric Invariant Domain

through dynamic thresholding. In addition, perceptual masking can be implemented easily by using the DT-CWT subbands during watermark embedding.

DWT-based watermarking methods enjoy multi-scale analysis and spatial information which are not available in Fourier transform-based methods. Nevertheless, DWT-based methods lacked shift invariance. Kingsbury and Selesnick designed wavelet transforms with such property [86]. One of the best transforms is the DT-CWT. The application of DT-CWT in robust watermarking was reported by several researchers recently [87, 88]. However, they all require resynchronization to combat geometric distortions.

Figure 4.11 The Dual Tree-Complex Wavelet Transform

We use shift invariance of DT-CWT to achieve geometric invariance without resynchronization. This aims at exploiting the advantages of DT-CWT compared to FFT and DWT. Two major shortcomings of FFT are the lack of multi-resolution sampling and perceptual masking property. Therefore, multi-resolution analysis and HVS modeling are not implicitly present in FFT methods [89]. Adversely, wavelet-based methods can be implemented with HVS masking easily. This is possible because wavelet-based methods encode spatial and frequency information in its transform domain, and they are superior compared to DCT and DFT approaches which only store frequency information. The DT-CWT provides two properties that
are absent in DWT, i.e. approximate shift invariance and directional selectivity [90]. Besides that, the perfect reconstruction property eliminates block artifacts in the reconstructed stego image. Although undecimated wavelet transform can offer shift invariance, it requires a huge amount of computation and high redundancy. Compared with the steerable pyramids method, DT-CWT offers well-balanced properties of shift invariance, directional selectivity, and redundancy [85]. Moreover, DT-CWT had been reported to offer better fidelity and higher robustness compared to DWT [91]. Table 4.5 summarizes the comparisons.

<table>
<thead>
<tr>
<th>Property</th>
<th>DT-CWT</th>
<th>DWT</th>
<th>FFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift invariance</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Perfect reconstruction</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Perceptual masking</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Multi-resolution sampling</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

### 4.3.2 Construction of the geometric invariant domain

Given an image \( I(x,y) \), and its FFT as \( I(u,v) \), we can write the LPM of FFT as \( I(\rho,\theta) \) where \( u = e^\rho \cos \theta \) and \( v = e^\rho \sin \theta \). Let \( \alpha, \beta, \) and \( (x_0,y_0) \) be the parameters of rotation, scaling, and translation respectively. Then, image rotation in \( I(x,y) \) corresponds to cyclical shift of \( \alpha \) along the angle \( \theta \) axis in LPM of the magnitude component of FFT. Also, image scaling in \( I(x,y) \) corresponds to translational shift of \( \ln \beta \) along the log-radius \( \rho \) axis in LPM of the magnitude component of FFT. Finally, translation in \( I(x,y) \) does not change the coefficients in LPM of the magnitude component of FFT because the FFT’s magnitude component is shift invariant. Therefore, RST operations in \( I(x,y) \) are transformed into linear shift in the FFT-LPM output. This transformation of FFT-LPM is called the FM transform. Figure 4.12 illustrates the mapping of rotation and scaling operations from spatial domain to the FM domain. By sending the FM output to a DT-CWT, we obtain RST invariance with the magnitude component of the final output due to its shift invariant property. The consistent response of DT-CWT to linear shift allows us to use a correlator detector with dynamic thresholding in blind watermark detection. Figure 4.13 and Figure 4.14 show the consistent response of the DT-CWT. A parallel shift in the input resulted in an almost parallel shift in its wavelets. A relative distance of \( \delta_1 \) in the input will result in a relative distance of \( \delta_2 \) in the wavelets, and they are not affected by the
parallel shift. Therefore, the correlation value between a correctly extracted watermark and a copyright holder’s watermark is consistent.

\[ I(x, y) \]
\[ \text{FFT} \]
\[ I(u, v) \]
\[ \text{LPM} \]
\[ I(\rho, \theta) \]

\[ u = e^\rho \cos \theta \]
\[ v = e^\rho \sin \theta \]

\[ \theta \]
\[ \rho \]

Figure 4.12 The Fourier-Mellin transform

Figure 4.13 Consistent response of the DT-CWT

4.3.3 The watermarking method

We investigated a geometric invariant domain watermarking scheme as shown in Figure 4.15 to exploit the advantages of DT-CWT. It uses the properties of FFT, LPM, and DT-CWT to achieve RST invariance.

(i) Watermark Embedding

To embed a watermark pattern \( W \) into an image \( X \), we perform a series of forward transformation as shown in Figure 4.15(a). The cover image \( X \) is sent through FFT, LPM, and finally DT-CWT to produce invariant domain coefficients. Then, the watermark pattern \( W \) is embedded using an additive embedding method with a global weight factor \( f \) and a simple HVS masking as follows.

\[
I_0^\theta(i, j) = I_0^\theta(i, j) \times [1 + f \times w^\theta(i, j)]
\]

where \( I_0^\theta(i, j) \) is the stego subband coefficients with the subbands \( \theta \in \{0,1,2,3,4,5\} \), \( I_0^\theta(i, j) \) is the DT-CWT subband coefficients transformed from the cover image \( X \), the embedding weight factor \( f \in \{\mathbb{R}^+\} \), and \( w^\theta(i,j) \) is the watermark pattern \( W \) arranged in the subbands \( \theta \) dimension.

The watermark pattern \( W \in \{-n, +n\}, n \in \{\mathbb{R}^+\} \) is a pseudo random pattern to mimic random noise. It has zero mean in order to minimize the changes made to the cover image. The magnitude components of FFT and DT-CWT are sent to its subsequent steps because they have the invariant properties. Its corresponding phase information is used in the inverse transformation steps to construct the stego image \( Y \). In addition, the backward transformation steps must take the reverse order because they are not commutative with the forward transformation steps, i.e. inverse DT-CWT is carried out first, followed by inverse LPM, and finally inverse FFT. The following pseudocode summarize the watermark embedding steps:
4.3 Geometric Invariant Domain

Figure 4.15 Invariant domain watermarking method

1. Perform FFT on the cover image.
2. Perform LPM on the magnitude component of the FFT output.
3. Perform DT-CWT on the LPM output.
4. Embed watermark into the magnitude component of the DT-CWT output using equation (4.20).
5. Perform IDT-CWT using the embedded magnitude component of DT-CWT and the phase component of DT-CWT. The phase component of DT-CWT is obtained from step 3.
6. Perform ILPM on the IDT-CWT output.
7. Perform IFFT using the ILPM output and the phase component of FFT to produce the stego image. The phase component of FFT is obtained from step 1.

We expect the transformed domain to be RST invariant. However, interpolation error introduced by the transform may affect its performance. This is particularly true for LPM where complete and unique mapping is difficult. As a result, robustness to rotation and scaling attacks may be degraded.

(ii) Watermark Detection
To determine whether the watermark \( W \) exist in a given test image \( Y' \) which could possibly be attacked, a series of forward transformation depicted in Figure 4.15(b) is
performed. It consists of a FFT, followed by a LPM, and finally a DT-CWT. The magnitude components of FFT and DT-CWT steps are sent to its subsequent steps because it has shift invariant property. Then, all of the 6 subband coefficients in the invariant domain are used in a dynamic thresholding computation. These steps are similar to the embedding process because we need to transform the test image into the same domain for RST invariance.

Blind watermark detection is enabled through a cross correlation computation based on the Neyman-Pearson criterion \([6, 92]\). We adapted the computation of the correlation value \(\rho\) and its threshold value \(T_\rho\) to cater for 6 subbands in the invariant domain with false positive probability \(P_f \leq 10^{-8}\). If the calculated value \(\rho\) is greater than its corresponding threshold \(T_\rho\), then the watermark \(W\) is detected, otherwise \(W\) is absent. The commonly chosen false positive probability \(P_f\) range from \(10^{-6}\) to \(10^{-12}\) \([17]\), and we take an intermediate value of \(10^{-8}\). Besides the RST invariance provided by the combination of three transforms mentioned above, the correlation detector with thresholding can discard changes resulting from amplitude scaling \([77]\).

Instead of working on 3 subbands of DWT as reported in \([6]\), we adapted the dynamic thresholding method to cater for 6 subbands of DT-CWT. As a result, we change the computation of \(\rho\) and \(T_\rho\) as follows. The correlation between the invariant domain coefficients and the watermark pattern \(W\) is given by Eq.(4.21).

\[
\rho = \frac{1}{6MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I_0^\theta (i,j) w^\theta (i,j)
\]

where \(I_0^\theta (i,j)\) is the DT-CWT subband coefficients for the test image \(Y'\), \(w^\theta (i,j)\) is the watermark pattern \(W\) arranged in the subbands \(\theta\) with \(\theta \in \{0,1,2,3,4,5\}\), and \(2M\times2N\) is the size of the test image \(Y'\).

The computation of the threshold value \(T_\rho\) is also adapted to DT-CWT subbands \([6]\). The probability of missing the watermark at a false positive probability is minimized using the following cases:

- Case A: the image has no watermark.
- Case B: the image is watermarked with \(W'\), \(W' \neq W\).
- Case C: the image is watermarked with \(W\).
The watermark embedded $w(i,j)$ is binary valued independent random variables with zero mean. Using the Central Limit Theorem (CLT), we assume $I_0^\theta(i,j)$ to be independent variable with Gaussian distribution, and the random variable $\rho$ also has Gaussian distribution. Then, the false positive probability $P_f = \text{Prob}(\rho > T_\rho | \text{Case A OR Case B})$ is estimated using the variance of $\rho$ for Case A

$$\sigma_{\rho A}^2 = \frac{\sigma_w^2}{(6MN)^2} \sum_{\theta=0}^5 \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} E\left[ (I_0^\theta (i,j))^2 \right]$$ (4.22)

and Case B

$$\sigma_{\rho B}^2 = \frac{\sigma_w^2}{(6MN)^2} \sum_{\theta=0}^5 \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} E\left[ (I_0^\theta (i,j))^2 \right] + 2 \sigma_w^2 E\left[ (I_0^\theta (i,j))^2 \right]$$ (4.23)

Given a higher variance, Case B has higher probability. Therefore,

$$P_f \leq \frac{1}{2} \text{erfc} \left( \frac{T_\rho}{\sqrt{2\sigma_{\rho B}^2}} \right)$$ (4.24)

From mass experimental results for $P_f \leq 10^{-8}$[6], the threshold value is given by

$$T_\rho = 3.97 \sqrt{2\sigma_{\rho B}^2} \cdot$$ (4.25)

To estimate the variance for Case B, the mean square value of watermarked coefficients is

$$E[I_0^\theta (i,j)^2] = E[I_0^\theta (i,j)^2] + f^2 E[w_0^\theta (i,j)^2] + 2f E[I_0^\theta (i,j) w_0^\theta (i,j)]$$ (4.26)

With $\sigma_w^2 = \sigma_w^4 = 1$, $E[I_0^\theta (i,j)] = E[w_0^\theta (i,j)] = 0$, and that $w_0^\theta (i,i)$ is independent of $I_0^\theta (i,j)$, we obtain

$$\sigma_{\rho B}^2 = \frac{1}{(6MN)^2} \sum_{\theta=0}^5 \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} E\left[ (I_0^\theta (i,j))^2 \right]$$ (4.27)

An unbiased estimate is

$$\sigma_{\rho B}^2 \approx \frac{1}{(6MN)^2} \sum_{\theta=0}^5 \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I_0^\theta (i,j))^2$$ (4.28)

where $I_0^\theta (i,j)$ is the DT-CWT subband coefficients for test image $Y'$ with the subbands $\theta \in \{0,1,2,3,4,5\}$, and $2M \times 2N$ is the size of the test image $Y'$. The following pseudocode summarize the watermark detection steps:
1. Perform FFT on the test image.
2. Perform LPM on the magnitude component of the FFT output.
3. Perform DT-CWT on the LPM output.
4. Take the magnitude component of the DT-CWT output and the watermark pattern to compute $\rho$ and $T_{\rho}$ using equations (4.21) and (4.25).
5. If $\rho > T_{\rho}$
   - The watermark is detected
   Else
   - The watermark is not detected

### 4.3.4 Analysis of Experiment Results
To evaluate the robustness of the implemented watermarking scheme, we performed a set of attacks on the stego images using *StirMark 3.1* [58, 59] and carried out the watermark detection steps. Most of the complex image manipulations such as projection can be modeled as local RST operations. Some of the basic attacks are listed in Table 4.6. All attacks were performed using *StirMark* except the RST-JPEG combined attack marked with asterisk (*) which was implemented in *Matlab*. Details of each attack are provided for completeness.

**Table 4.6 Robustness attacks using *StirMark***

<table>
<thead>
<tr>
<th>Attack</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation with cropping</td>
<td>Rotation angle from $-2^\circ$ to $90^\circ$ with cropping</td>
</tr>
<tr>
<td>Scaling</td>
<td>Scaling factor from 0.5 to 2.0</td>
</tr>
<tr>
<td>Translation</td>
<td>Circular shift 50% of image size</td>
</tr>
<tr>
<td>JPEG compression</td>
<td>Quality factor of 10% to 90%</td>
</tr>
<tr>
<td>Random bending</td>
<td>Random bending attack</td>
</tr>
<tr>
<td>Row and column removal</td>
<td>Remove 1 to 17 rows and columns.</td>
</tr>
<tr>
<td>Median filtering</td>
<td>Kernel size from 2×2 to 4×4</td>
</tr>
<tr>
<td>Cropping</td>
<td>Crop off 1% to 75% of image size</td>
</tr>
<tr>
<td>Gaussian filtering</td>
<td>Kernel size 3×3</td>
</tr>
<tr>
<td>Linear transform</td>
<td>General linear transformation</td>
</tr>
<tr>
<td>Aspect ratio change</td>
<td>Change aspect ratio in x and y directions</td>
</tr>
<tr>
<td>Rotation with cropping and scaling</td>
<td>Rotation angle from $-2^\circ$ to $90^\circ$ with scaling</td>
</tr>
<tr>
<td>Sharpening</td>
<td>Kernel size 3×3</td>
</tr>
<tr>
<td>Shearing</td>
<td>Shear in x and y directions</td>
</tr>
<tr>
<td>Combination of RST and JPEG compression*</td>
<td>Circular shift 10 columns to the right, scale down to 220×280 pixels, rotate at 15° anticlockwise, and JPEG compress with quality factor 50%</td>
</tr>
</tbody>
</table>
4.3 Geometric Invariant Domain

The images are selected to represent various characteristics, and are illustrated in Figure 3.3 of Chapter 3. They are all gray scale images with standard dimension 256×256 pixels. The images are identified by name: Lena, Baboon, Cameraman, Pepper, and Fishing boat.

The experiments begun with watermark embedding, followed by attacks mentioned in Table 4.6, and finally watermark detection. A watermark of 128×128 pseudo-random binary values was generated and embedded into all of the DT-CWT subbands in the invariant domain. The embedding weight factor $f$ was computed using the average values of all subband coefficients.

Table 4.7 lists the average results of robustness tests. The column of “5 images” contains average score of all the test images mentioned above whereas the column of “3 images” contains average score of Baboon, Lena, and Fishing boat images. The scores are normalized to the range from 0 to 1. A score of 1.000 means the watermark was detected in all images for all levels of attacks in that category. Adversely, a score of 0.000 indicates no watermark was detected in all cases. The results in “3 images” column is used to compare the performance of our method with Kim’s method [84], which was reported to outperform several state of the art watermarking methods. Under rotation with cropping attack, our method scored

<table>
<thead>
<tr>
<th>Attack</th>
<th>Average results</th>
<th>Kim</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5 images</td>
<td>3 images</td>
</tr>
<tr>
<td>Rotation with cropping</td>
<td>0.875</td>
<td>0.875</td>
</tr>
<tr>
<td>Scaling</td>
<td>0.700</td>
<td>0.722</td>
</tr>
<tr>
<td>Translation</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>JPEG compression</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Random bending</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Row and column removal</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Median filtering</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Cropping</td>
<td>0.667</td>
<td>0.667</td>
</tr>
<tr>
<td>Gaussian filtering</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Linear transform</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Aspect ratio change</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Rotation with cropping and scaling</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Sharpening</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Shearing</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Combination of RST with JPEG compression*</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>
0.875 whereas Kim’s method has 0.95. Our method suffers from much information lost under cropping. This can be improved by watermarking small blocks of the image instead of the whole image. For scaling attack, our method achieved 0.722 compared to Kim’s 0.87. Although our scores were slightly lower than Kim’s scores in these categories, the situation was reversed in the other two attacks, i.e. our scores for RBA and JPEG compression are 1.000. Kim’s scores for RBA and JPEG attacks were 0.95 and 0.90 respectively. These could be due to the threshold value chosen in the experiments. We fixed the false positive probability to be less than $10^{-8}$ for all types of attacks whereas Kim’s mass tests yielded best result of $7.8 \times 10^{-2}$. Therefore, we could safely conclude that our results are better.

(i) **Rotation**

Rotation operation is transformed into linear shift operation in LPM, and later transformed into the invariant domain with DT-CWT operation. Therefore, our method can resist most of the rotation levels. The lowest score appeared at 45° rotation because it has the least correlation value and biggest cropped area compared to other levels of rotation. However, in another set of experiments implemented using Matlab, watermark was detected in all degrees of rotation. This can be explained by the difference in the cropping methods between StirMark and Matlab. The former causes too much information lost compared to the latter. Figure 4.16 illustrates a comparison between them for a 45° rotation with cropping.

(ii) **Scaling**

Scaling in the spatial domain is transformed into linear shift in the LPM output. Then, it is further transformed into invariant coefficients in the DT-CWT output. As a result, the implemented method can resist scaling attack. It is generally agreed that scaling smaller than half of the original image size would ruin the commercial value of its output. This is the same for scaling larger than twice the original size. The high level of information lost at scaling factor 0.5 caused the method to fail for all test images. This is further deteriorated by the LPM. However, the method performed well for other scaling factors tested.
(iii) Translation
The magnitude component of FFT is invariant to translation attacks. Consequently, the watermark was detected in all images under circular shift operation. The dynamically computed $\rho$ value stays above the threshold $T_\rho$ for all the tests.

![Figure 4.16 Comparison of rotation with cropping under different implementations.](a) Lena rotated using StirMark; (b) Lena rotated using Matlab.

(iv) JPEG Compression
The watermark was detected in all levels of compression quality with JPEG algorithm. Due to the chosen false positive probability of $10^{-8}$, the dynamically computed $\rho$ value always stays above the threshold $T_\rho$ for all compression quality factors.

(v) Random Bending Attack
Local geometrical distortions were carried out on the stego image using the RBA. Although the stego image and its corresponding distorted images appear similar to human eyes, the distorted regions in the attacked images desynchronized pixel locations. This type of attack presents a tough problem to many robust watermarking schemes because the assumption that distortion is uniform globally does not hold. However, the watermark was detected in all cases using our watermarking scheme. The local shifting in spatial domain caused by RBA has little effect on the magnitude component of FFT and DT-CWT coefficients due to its shift invariance.

(vi) Common Image Processing Operations
Many common image processing operations can be done easily with off-the-shelf software packages, yet these operations can cause desynchronization in the watermark information and affect watermark detection. Such operations include cropping, median filtering, Gaussian filtering, linear transform, aspect ratio change, image sharpening, and shearing. The test results show that cropping has the most
severe effect on watermark detection, especially when 75% of the stego image is cropped off. The remaining stego image area can only provide limited information for watermark detection. On the other hand, the implemented scheme attained 100% successful detection under many other common attacks. The robustness of wavelet-based methods against these attacks is provided by the multiscale and spatial information encoded. Using DT-CWT with a correlation-based watermark detection, the scheme can resist modifications caused by these attacks.

(vii) Combined RST and JPEG Compression
The combination of RST operations followed by JPEG compression mentioned in Table 4.6 cannot defeat the watermarking scheme. The attacked stego image of Lena is illustrated in Figure 4.17. In addition, the watermark was also detected in many other combinations with varying levels of distortions. This shows that the scheme is extremely robust to geometrical attacks.

![Figure 4.17 Lena attacked with a combination of RST operation and JPEG compression](image)

(viii) Mass test
Using 1000 test images obtained from an image database at the Pennsylvania State University (source: http://wang.ist.psu.edu/docs/related/), we embedded watermarks into it and perform watermark detection. The images consist of human body, natural scenes, buildings, transports, animals, and plants. Figure 4.18 shows some samples of the images. The results provided very high reliability. Figure 4.19 illustrates the ROC of the method. It shows a high level of true detection and low level of false alarm.
4.3 Geometric Invariant Domain

We investigated a geometric invariant domain watermarking scheme that explored the many advantages offered by DT-CWT. However, some limitations arise from the adoption of the RST invariant framework.

(i) **Advantages**
The major contribution of this work lies in the application of DT-CWT properties in developing a geometric invariant domain. Our method does not require any
resynchronization. It also enables perceptual masking and blind watermark detection. Although the magnitude component of DT-CWT is only approximately shift invariant, the experimental results proved the high level of robustness. The attacks tested include the basic RST operations, JPEG compression, some common image processing operations, and local geometrical distortion. Images with various characteristics were experimented in the mass test.

The multi-resolution samples of DT-CWT provided fine-tuning capabilities. For instance, watermarking in the high level subbands of the transform increases the robustness at the cost of stego image fidelity. Since the LPM and ILPM operations introduce much interpolation errors, the scheme requires a trade-off between robustness and fidelity. Therefore, we chose to embed the watermark in the lowest level subbands.

To improve stego image fidelity, perceptual masking was applied during watermarking embedding by adjusting the embedding weight according to local coefficient values. A simple approach of such masking was carried out using the DT-CWT subbands.

The perfect reconstruction feature of DT-CWT can compensate the visual quality degradation caused by LPM. Despite the interpolation errors introduced in LPM and ILPM, the Lena stego image achieved good visual quality of 38dB in terms of PSNR.

(ii) Limitations

There are a few limitations attached to the scheme. One important issue to be resolved is to overcome interpolation errors caused by LPM and improve stego image fidelity. LPM and its inverse operation cause image quality degradation. This is due to interpolations involved in the transform. Ruanaidh [76] minimize such impact with a one-way transform for embedding and another way for detection. Lin [77] avoided this by eliminating strong invariant requirement and simplified the data complexity into 1-D stream. They mentioned that invertibility offered by strong invariant is not essential because they can substitute it with a watermark extraction function which gives approximately similar results. However, we prefer strong invariant for its invertibility and to achieve robustness. Therefore, we suggest the use of a large mapping space with redundancies to overcome under-sampling in LPM. This will improve the visual quality of the stego image. We found that over-sampling
an image with LPM into a space 5 times the original size could give nearly perfect
inversion when ILPM is applied. Nevertheless, this will increase the computation
cycles.

The adoption of RST invariant framework in the watermarking scheme
inherently required large amount of computation. This is caused by 2-D FFT and
LPM operations. In addition, DT-CWT also involves a certain amount of
computation. Despite all of these, the embedding and detection process performed
within acceptable time frame on an average desktop computer. Using a Pentium III
800MHz machine with 256 MB of memory, one loop of embedding and detection
process implemented in Matlab scripts does not exceed 2 minutes.

4.3.6 Conclusion
Invariant domain that resists RST attacks is a promising approach to robust
watermarking. We developed a geometric invariant domain taking advantages of DT-
CWT properties and HVS masking. In addition, DT-CWT offers high robustness
with multi-resolution sampling, perceptual masking, and perfect reconstruction. We
also adapted our fast watermark embedding method and blind watermark detection
method in the invariant domain watermarking. The invariant domain not only
resisted basic RST attacks, it also survives JPEG compression, common image
processing operations, and local geometrical distortion. The watermark scheme is
also robust under extreme distortions created with combination of these attacks. In
short, our robust watermarking method does not require any resynchronization,
enables blind watermark detection and implicit perceptual masking.

4.4 Chapter Summary
Blind watermark detection has higher practical values compared to non-blind
watermark detection. Therefore, we investigated a blind watermark detection method
in this chapter. A simplified watermark embedding method for the blind
watermarking method was created to reduce computational cost. The method
provides watermark imperceptibility and robustness as good as the HVS model. It
also has processing time as fast as a watermark embedding method that uses constant
embedding energy.

Geometrical distortions are the fundamental operations of many image
manipulations. They can be implemented easily using off-the-shelf software, and
have destructive impact on robust watermarks. Therefore, we focus on RST robustness. We investigated the invariant domain approach to robust watermarking in this chapter. The geometric invariant domain was created using a combination of FFT, LPM, and DT-CWT. The method exploits approximate shift invariant and perceptual masking properties of the DT-CWT. We also adapted our simplified watermark embedding method in the invariant domain for fast processing. The method also offers good watermark imperceptibility using perceptual masking, and enables blind watermarking detection.

This chapter completes the discussions on robust watermarking. We will combine our geometric invariant domain with a semi-fragile watermarking method to form a hybrid watermarking method. Therefore, we will describe our semi-fragile watermarking method in the next Chapter. After that, the hybrid watermarking method will be explained in Chapter 6.
Chapter 5

Semi-fragile Watermarking with Self-Authentication and Self-Recovery

Semi-fragile watermarks offer flexibility in content authentication where malicious attacks can be separated from legitimate changes according to application scenario. Section 5.1 discusses developments in semi-fragile watermarking. Major challenges exist in creating semi-fragile watermarking methods that enable blind watermark detection, self authentication, and self recovery. To address these challenges, the requirements for semi-fragile watermarking are listed in Section 5.2. We created a new method using a dynamic quantization method after analyzing the distribution of wavelet coefficients. The watermark embedding and detection details are discussed in Section 5.3. Self embedding with random mapping is suggested for secured self recovery. Unlike others, our method overcomes a major challenge called cropping attack. Detection decision is adjustable based on the thresholding of correlation result. The method finds a balance among conflicting factors of imperceptibility, semi-fragility, computational costs, and security. Experiments in Section 5.4 validated its capabilities in content authentication, tamper localization, and approximate content recovery.
5.1 Introduction
This section revisits the developments in semi-fragile watermarking covered in Chapter 2. Robust watermarks are suitable for copyright protection because they persistently stay intact with the image. On the other hand, fragile watermarks are good for tamper detection applications due to their ability to highlight changes in images. Recent developments in the watermarking world have witnessed the rise of semi-fragile watermarks. As the name suggests, semi-fragile watermarks reside in the gray area between the two extremes of robust and fragile watermarks, i.e. they possess some properties of both robust and fragile watermarks. The need for semi-fragile watermarks arises from the requirements of content authentication. In content authentication, the watermark must highlight malicious attacks while tolerating legitimate changes that do not severely alter the content of the stego image. This is important because most content is stored, transmitted, and consumed in digital form. For example, a semi-fragile watermark should provide evidence of forgery under a cropping attack, and resist high quality image compression where the meaning of the content is preserved. Content authentication, in this context, is also named soft authentication or selective authentication. Fragile watermarking is usually implemented in the spatial domain using a statistical approach in a similar way to digital signatures or fingerprints. Semi-fragile watermarking is normally implemented in transform domain using a quantization approach.

As described in Chapter 2, there are a number of approaches to semi-fragile watermarking. The block-based approach divides an image into blocks and watermarks each block separately. This approach obviously suffers from high computational cost associated with processing a large number of blocks. The feature-based approach extracts robust features from an image and embeds that information as authentication watermarks. This approach must make a compromise between robustness and detection accuracy. Although an improvement was suggested by Rey and Dugelay [47], it requires high computational costs. In addition, some researchers have suggested a hashing approach for semi-fragile watermarking. Hash-based methods, either robust hash [49] or visual hash [50], cannot provide enough information for content recovery because they map image features into a small space.
5.2 Challenges in Semi-fragile Watermarking

Indeed, they are designed for image indexing and searching purposes. This is also true for some methods based on digital signatures and feature points [51].

Although some recent watermarking methods can localize tampering by highlighting the attacked regions, very few of them can recover the original content. The recovered contents could provide useful investigative information in media forensics. Such application scenarios include courtroom evidence and journalistic photography. Among the many semi-fragile watermarking methods, only a few have both tamper localization and content recovery capabilities [36, 47, 48, 93]. For example, Fridrich [48] has self-recovery capability only; On the other hand, Lin-Chang [36] and Rey-Dugelay [93] have both tamper localization and content recovery capabilities. However, a major shortcoming of these methods is the high amount of computation due to their iterative embedding process.

The shortcomings of current semi-fragile watermarking methods motivated us to address the challenges described in the next section. Following that, we discuss the design issues in creating a balanced watermarking method to fulfill the requirements identified. Watermarking embedding and detection methods are then presented in the subsequent sections. Finally, we provide experimental results in analyzing the effectiveness of the watermarking method in meeting the requirements identified.

5.2 Challenges in Semi-fragile Watermarking

As discussed earlier, semi-fragile watermarks have been studied in recent years and improvements have been made. However, there remain challenges that need to be addressed. For instance, blind watermark detection and cropping resistance are hard to achieve using most watermarking methods. In addition, the ability to reconstruct a cropped region using semi-fragile watermarks is rarely found. Furthermore, it is hard to construct a semi-fragile watermark with self-recovery capability [48]. To overcome these limitations, we defined the following requirements for our design of a semi-fragile watermarking method:

1. Mild image modifications and compression that do not change the perceptual quality of the image result in the test image being classified as authentic.
2. Malicious changes that affect the visual quality of the image result in the test image being classified as unauthentic.
3. The watermark detection process must operate in a blind manner, i.e. without resort to a reference image. The watermark detection process includes the detection, extraction, and decoding of watermark signal. The reference image could be the cover image or the unattacked stego image.

4. When tampering is detected, the tampered regions must be located and highlighted correctly using tamper localization ability of the method.

5. An approximate content recovery must be made without the need of a reference image. This recovery could be necessary under cropping attack or region modifications that change the image’s original content. The recovered content could provide useful investigative information for media forensics.

6. The watermark information must be sufficiently secured so that adversaries cannot easily modify it without being noticed. The objective here is not to create a highly secure watermarking method. Secure watermarking would require a separate study.

7. The watermarking method must strike a balance between semi-fragility, watermark imperceptibility, computational cost, and security.

With regard to the first 2 requirements above, the authentication decision often depends on application scenario. For instance, watermarking distortions in medical images must be strictly controlled to avoid misjudgment in diagnosis whereas smoothening of artistic pictures may be allowed in commercial environment. In our case, we classify innocent changes as those operations that produce minimal differences compared to the cover image while other operations are considered malicious.

5.3 Design of Semi-fragile Watermarking

Various semi-fragile watermarking methods have been developed in both the spatial and transform domains. Spatial domain methods usually exploit the statistical properties of the image pixels in detecting tampering and providing authentication. As such, their implementations are normally simple and fast. On the other hand, transform domain methods offer robustness against compression by using frequency information.

We chose to utilize spatio-temporal information in the wavelet domain in our semi-fragile watermarking method. Compared to DWT, DCT and DFT lack spatial
5.3 Design of Semi-fragile Watermarking

information. Although DWT will certainly increase the computational cost, we compensate it with simple processing steps in the embedding and detection processes. For example, we use the low pass band and a down-scaled image in authentication and tamper localization instead of the commonly used block-based approach.

To fulfil the requirements listed in Section 5.2, we embed a down-scaled version of the image in the highpass bands of the wavelet coefficients. Embedding the watermark in the low level subbands provides mild robustness to image compression. A higher level of robustness can be achieved by embedding the watermark in the higher level subbands. However, this will degrade the visual quality of the stego image. Therefore, we embed the watermark at the second lowest level subbands to obtain semi-fragility.

A majority of the current semi-fragile watermarking methods employ block-based processing for authentication and tamper localization. For example, the mean values of 8×8 pixel blocks can be embedded into a cover image. Later, the values can be extracted from the stego image and compared with the computed mean values of the blocks at the same location to detect tampering. This approach certainly involves a large amount of computation. We reduce the computational cost of the authentication and tamper localization by processing a down-scale version of the image and the wavelet transform subbands. The effect of such an approach is the same as block-based approaches because each of the elements in the down-scaled image or the subband corresponds to a certain block of pixels in the stego/test image. For the same reason, minor changes in the stego image would have minimal effect on the element values. Therefore, we can apply a simple correlator to detect tampering and localize it in the spatial domain.

Quantization is chosen as the embedding method in the wavelet domain because it provides high level of robustness [94]. The watermark is usually embedded in the highpass subbands at low level wavelet decompositions for better imperceptibility. We use the histogram of wavelet subbands to perform quantization for reduced computation. Quantization also allows fine tuning of watermark detection accuracy and imperceptibility by varying the number of quantization bins. A larger number of bins offer better imperceptibility at the cost of watermark extraction accuracy. This is because the bin size becomes smaller with a larger number of bins for a fixed range
of coefficient values, and this means the changes made during watermark embedding
will be smaller. At the same time, the watermark extraction accuracy would be
degraded because the distinction between the bins becomes smaller. To find a
balance point between these contradicting requirements, we created a method
utilising varying bin size. Figure 5.1 illustrates the histogram of a level-2 DWT
horizontal subband. Analyzing the histogram of highpass subbands of level-2 DWT,
it is noticed that most of the wavelet coefficients have near-zero values because they
correspond to flat regions in the image. Also, these coefficients occupy only a small
range of the values in the histogram. We can use a small number of bins for these
middle range coefficients because the changes made would be small. That means
good imperceptibility and high watermark extraction accuracy. As for both ends of
the histogram with large-value coefficients, we use a large number of bins to
minimize changes in coefficient values for good imperceptibility. Although this
would degrade the watermark extraction accuracy, the total effect is minimal because
these coefficients only occupy a small fraction of the total count of coefficients.
Referring to Figure 5.1, we use a small number of bins (NB1) for the center part of
the middle range coefficients (1-2B), and a large number of bins (NB2) for the both
ending fractions (B).

![Figure 5.1 Histogram of a level-2 DWT horizontal subband](image)

The watermark bits are embedded in locations far away from its original position
in order to combat cropping attack and enable content recovery. For example, the
watermark information of the lower right corner of the cover image can be embedded
into the upper left corner of a wavelet subband. This way, a cropped area in the lower
right corner of a stego image can be recovered by extracting the watermark information from the un-affected upper left corner of the same image. In addition, the watermark embedding positions can be made random using a secret key to offer security.

An overview of the watermarking method is depicted in Figure 5.2. A watermark is generated by taking the down-scaled version of the cover image. The cover image is transformed into the wavelet domain by DWT. Then, the 4 most significant bits (MSBs) of the watermark are embedded into one wavelet subband, and the 4 least significant bits (LSBs) are embedded into another wavelet subband. Following that, the stego image is obtained by an inverse transform from the wavelet domain into the spatial domain. To authenticate a test image which could have undergone changes, a DWT is performed and the watermark is extracted from the wavelet subbands. The watermark is then compared with a down-scaled version of the test image. If the similarity between them exceeds a threshold value, then the test image is classified as authentic. Otherwise, tampered regions will be highlighted and content recovery is carried out using the watermark information extracted.

![Figure 5.2 Overview of the semi-fragile watermarking method](image)
5.3.1 Watermark Embedding
The watermark signal is generated by taking a down-scaled version of the cover image in order to enable content authentication and self-recovery. Watermark bits should be embedded far away from their original positions to combat cropping attacks. For enhanced security, a secret key can be used to randomize pixel positions of the watermark to form a secure watermark. Existing cryptographic protocols and their related infrastructure can be applied in sharing of the secret key between the watermark embedder and the watermark detector. Using a 8-bit grayscale watermark, we embed the 4 MSBs into the horizontal subband, and the 4 LSBs into the vertical subband. Then, an IDWT is performed using the embedded subbands to obtain a stego image. There are 7 user-defined parameters:

1. Let $f(m,n)$ be the cover image.
2. Let $w(p,q)$ be the watermark signal for authentication.
3. Let $L \in \{1,2\}$ be the wavelet decomposition level for watermark embedding.
4. Let $N1 \in \mathbb{Z}^+$ be the count of quantization bins in the middle range of the wavelet subband histogram.
5. Let $N2 \in \mathbb{Z}^+$ be the count of quantization bins in both ends of the wavelet subband histogram.
6. Let $B=[0, 0.3]$ be the boundary fraction for both ends of the wavelet subband histogram. This value is determined by inspecting the histogram of wavelet subbands.
7. Let $skey(p,q)$ be the secret key used to randomize pixel.

The watermark embedding begins with a DWT of $f(m,n)$. Let $g_{k,l}(m,n)$ be the subbands $k \in \{a,h,v,d\}$ at level $l \in L$ of the wavelet coefficients where $a$, $h$, $v$, and $d$ are the approximate, horizontal, vertical, and diagonal subbands respectively. An initial watermark is generated by taking a down-scale version of $f(m,n)$ having the same size as $g_{k,l+1}(m,n)$. Then its pixel positions are mapped using $skey(p,q)$ to produce the secure watermark $w(p,q)$ for authentication and content recovery. A $N1$-bin quantization table for each $h$ and $v$ subbands is constructed by taking the $(1-2B)$ middle range of the wavelet coefficients. Then $N2$-bin is appended within $B$ range to the quantization tables at both ends. A quantization function $Q$ is used to map each wavelet coefficient to a binary value,
5.3 Design of Semi-fragile Watermarking

\[
Q(f) = \begin{cases} 
0 & \text{if } r\Delta < f < (r+1)\Delta \text{ for } r = 0, \pm 2, \pm 4, \ldots \\text{ or } \pm 1, \pm 3, \pm 5, \ldots \n\end{cases} \quad (5.1)
\]

where \( \Delta \) is the quantization parameter: \( \Delta = [(f_{\text{max}} - f_{\text{min}}) \times (1 - 2B)] / N1 \) for the middle range of wavelet coefficients; and \( \Delta = [(f_{\text{max}} - f_{\text{min}}) \times B] / N2 \) for both the ending range of wavelet coefficients. Following that, the 4 MSBs of \( w(p,q) \) are embedded into the \( h \) subband, and the 4 LSBs of \( w(p,q) \) are embedded into the \( v \) subband. For example, we can choose to embed the 4 MSBs of \( w(m,n) \) into \( g_{h,L}(m,n) \), \( g_{h,L}(m,n+1) \), \( g_{h,L}(m+1,n) \), and \( g_{h,L}(m+1,n+1) \). The embedding process ensures that each wavelet coefficient maps to the correct bit value by assigning a value in the middle part of the quantization bin. It also ensures minimal changes when flipping a bit value by moving the current value to its next bin or previous bin by examining its current value. Lastly, IDWT is performed using the embedded \( h \) and \( v \) subbands, and the original \( a \) and \( d \) subbands to obtain the stego image. The pseudo-code below outlines the steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Initialize user-defined parameters: ( f(m,n) ), ( w(p,q) ), ( L ), ( N1 ), ( N2 ), ( B ), ( skey(p,q) ).</td>
</tr>
<tr>
<td>2.</td>
<td>Decompose ( f(m,n) ) using Haar filter for ( L ) levels.</td>
</tr>
<tr>
<td>3.</td>
<td>Generate ( w(p,q) ) using ( f(m,n) ) and ( skey(p,q) ).</td>
</tr>
<tr>
<td>4.</td>
<td>Construct quantization tables for each ( h ) and ( v ) subbands using ( Q(f) ).</td>
</tr>
<tr>
<td>5.</td>
<td>Embed ( w(p,q) ) into ( h ) and ( v ) subbands using the quantization table with these rules:</td>
</tr>
<tr>
<td></td>
<td>if ( Q(g_{k,L}(m,n)) = w(m,n) )</td>
</tr>
<tr>
<td></td>
<td>set ( g_{k,L}(m,n) = (r+0.5)\Delta )</td>
</tr>
<tr>
<td></td>
<td>else</td>
</tr>
<tr>
<td></td>
<td>if ( Q(g_{k,L}(m,n)) &gt; (r+0.5)\Delta )</td>
</tr>
<tr>
<td></td>
<td>set ( g_{k,L}(m,n) = (r+1.5)\Delta )</td>
</tr>
<tr>
<td></td>
<td>else</td>
</tr>
<tr>
<td></td>
<td>set ( g_{k,L}(m,n) = (r-1.5)\Delta )</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
<tr>
<td>6.</td>
<td>Perform IDWT on ( g_{k,L}(m,n) ) to get the stego image.</td>
</tr>
</tbody>
</table>

### 5.3.2 Watermark Detection

The detection begins with a DWT on the test image. Then 4 bits of gray level information are extracted from the horizontal and vertical subbands respectively to form an 8-bit grayscale watermark. The extracted watermark is compared with the down-scaled version of the test image to determine its authenticity. Alternatively, the
lowpass subband can be used to replace the down-scaled test image for authentication. If the test image is not authentic, then tamper localization and approximate content recovery are carried out. Authentication is carried out in a blind detection manner because it does not require a reference image. However, due to its fragile nature, the watermark would be destroyed if the test image is severely distorted. Therefore, tamper localization and content recovery would require a reliable reference in addition to the extracted watermark. To enable blind detection in this case, we use a down-scaled version of the cover image as the secret key for this reference.

To detect the watermark correctly, the correct set of private information \{L, NB1, NB2, B, skey\} must be used. Let \(f(m,n)\) be the test image. This is the stego image that could have undergone attacks. Let \(w(p,q)\) be the secret key for tamper localization and content recovery. The \(w(p,q)\) is only necessary if test images are severely distorted. Let \(t_1 \in \mathbb{R}^+\) be the threshold value for authentication, and \(t_2 \in \mathbb{R}^+\) be the threshold value for tamper localization and content recovery. The process starts with a DWT on \(f(m,n)\) to obtain \(g_{k,l}(m,n)\) with the subbands \(k \in \{a,h,v,d\}\) at level \(l \in L\). Then, quantization tables for each \(h\) and \(v\) subbands are constructed in a similar manner to those in the watermark embedding steps. The watermark \(w'(p,q)\) is extracted from the subbands at level \(L\), taking 4 MSB from the \(h\) subband, and 4 LSB from the \(v\) subband. The quantization function \(Q\) as in the watermark embedding steps is used in the watermark extraction. After that, the pixel positions in \(w'(p,q)\) are re-mapped using \(skey(p,q)\) to produce the watermark \(w''(p,q)\) for authentication. The watermark \(w''(p,q)\) should appear as a down-scaled version of the cover image, and it may has some error bits due to attacks. To reduce the error effects, we can perform a smoothening operation on \(w''(p,q)\). This will also enhance the semi-fragile characteristic of the watermark for authentication by introducing some “fuzziness”. Thresholding with a two-dimensional correlation coefficient \((corr2)\) is used to determine the authenticity of the test image.

\[
corr2 = \frac{\sum_p \sum_q (w'' - \bar{w})(u - \bar{u})}{\sqrt{\left[\sum_p \sum_q (w'' - \bar{w})^2\right]\left[\sum_p \sum_q (u - \bar{u})^2\right]}}
\]  

\hspace{1cm} (5.2)
where \( u(p,q) \) is the down-scaled version of \( f'(m,n) \), \( \bar{w} \) is the mean value of \( w'' \), and \( \bar{u} \) is the mean value of \( u \). To locate tampered regions in an unauthentic test image, a tampering matrix \( y(p,q) \) is computed,

\[
y(p,q) = u(p,q) - \bar{w}(p,q) \tag{5.3}
\]

where \( \bar{w}(p,q) \) is the down-scaled version of the cover image. Then the threshold value \( t_2 \) is used in a thresholding for tamper localization. To approximately recover the contents of the tampered regions, \( y(p,q) \) is up-scaled to the size of the test image and the tampered regions are replaced by an up-scaled version of \( \bar{w}(p,q) \). If the attack is not severe, then \( w''(p,q) \) can be used instead of \( \bar{w}(p,q) \). The steps are outlined in the pseudo-code below.

1. Initialize user-defined parameters: \( L, NB1, NB2, B, skey, f'(m,n), \bar{w}(p,q), t1, t2 \)
2. Decompose \( f'(m,n) \) using Haar filter for \( L \) levels.
3. Construct quantization tables for each \( h \) and \( v \) subbands.
4. Extract the watermark \( w'(p,q) \) from the \( h \) and \( v \) subbands using \( Q(f) \).
5. Reverse the mapping of pixel positions in \( w'(p,q) \) using \( skey(p,q) \) to obtain \( w''(p,q) \).
6. Compute the 2D correlation coefficient \( corr2 \) and authenticate the test image:
   
   \[
   \text{if } corr2 > t1 \\
   \quad \text{The image is authentic} \\
   \text{else} \\
   \quad \text{The image is not authentic} \\
   \text{end}
   \]
7. If image is not authentic, locate tampered regions using a tampering matrix \( y(p,q) \)
   
   \[
   \text{if } |y(p,q)| > t2 \\
   \quad y(p,q) \text{ is tampered} \\
   \text{else} \\
   \quad y(p,q) \text{ is not tampered} \\
   \text{end}
   \]
8. Recover tampered regions using \( y(p,q) \) and \( \bar{w}(p,q) \).

The extracted watermark \( w''(p,q) \) can be used to replace \( \bar{w}(p,q) \) in steps 8 and 9 for tamper localization and content recovery under minor attacks. This is possible because minor attacks do not cause too much information loss in the extracted watermark. Therefore, the watermark detection can be carried out in a strictly blind manner.
5.4 Analysis of Experimental Results

This section describes the experiment settings and analyzes the experimental results with regard to several performance metrics. The metrics include imperceptibility, semi-fragile performance, tamper localization, and content recovery.

5.4.1 Experiment settings

Four images with different characteristics are used in the experiment. *Baboon* has complex textures, *Lena* has clear boundaries between regions, *Pepper* has smooth surfaces, and *Fishing boat* has high contrast areas and tiny objects. These images are illustrated in Figure 3.3 of Chapter 3. The 512×512 pixel cover image $f(m,n)$ is down-scaled to 64×64 pixel to form the watermark $w(p,q)$. The other settings are $L = 2$, $N1 = 22$, $N2 = 400$, and $B = 0.25$. The watermark signal is embedded into level 2 of the DWT subbands ($L = 2$) for moderate robustness. The value of the parameters $N1$, $N2$, and $B$ are determined through experiment. For simplicity and ease of manual verification, $skey(p,q)$ is chosen as a circularly shifted matrix in both horizontal and vertical directions. This shift at half of its size will produce a watermark with 4 quadrants having maximum distance from its original position, and can be powerful in fighting cropping attacks. An example of the watermark $w(p,q)$ for *Lena* produced by $skey(p,q)$ is shown in Figure 5.3. A randomly permuted $skey(p,q)$ is preferred for high security system.

![Figure 5.3 The watermark signal for Lena](image)

5.4.2 Imperceptibility

The difference between a cover image and its stego image is minimal and does not reveal any information about the watermark because it appeared as random noise. Figure 5.4 illustrates an example. The PSNR of the stego images are 41.26 dB for *Lena*, 41.14 dB for *Baboon*, 40.15 dB for *Pepper*, and 40.11 dB for *Fishing boat*. Better imperceptibility can be obtained by increasing the number of quantization bins so that smaller bin size results in less modification on the image. This is done at the
cost of watermark extraction accuracy because the small bin size produces more quantization error during watermark detection.

![Watermark imperceptibility evaluation](image)

**Figure 5.4 Watermark imperceptibility evaluation** (a) The cover image; (b) The stego image; (c) The magnified difference between the cover image and its stego image

### 5.4.3 Semi-fragile performance

The parameters applied in watermark detection must be the same as its embedding procedures because this is a symmetric key watermark system. In addition, the threshold values $t_1$ and $t_2$ are determined through experiments. Higher threshold values increase the watermark’s fragile nature and make it more sensitive to changes. For example, $t_1 = 0.86$ and $t_2 = 30.0$ for Lena shows optimal performance. Figure 5.5 illustrates interim watermark detection results. Some error bits in watermark extraction can be seen when comparing the original watermark in Figure 5.3 with the extracted one in Figure 5.5 (a). Smoothening operation was applied to reduce the error effects. The smoothening operation also provides “fuzziness” for its semi-fragility because exact comparison of content is not required. This is opposed to hard authentication of a fragile watermarking method. The “fuzziness” serves the purpose of reducing the probability of false positive in watermark detection. Furthermore, the computational cost of our method is relatively low. In contrast, the methods proposed by Lin-Chang [36] and Rey-Dugelay [93] suffered from high computational costs due to their iterative embedding process.
Figure 5.5 Watermark detection process (a) The 8-bit grayscale watermark \( w'(p,q) \) extracted with some error pixels; (b) The error-reduced \( w''(p,q) \) produced from remapping \( w'(p,q) \) and smoothening; (c) The down-scaled version of the test image for authentication.

Table 5.1 lists suitable authentication threshold values \( t_1 \) for each test image after examining its corresponding correlation \( corr_2 \) values. All of the test images were watermarked using the parameter values mentioned in Section 5.4.1. Local shift attack was performed by shifting the region \((130:220,115:125)\) five columns to its right, and shifting the region \((382:392,260:340)\) two rows upwards. Noise attacks involved adding “salt and pepper” noise with varying density. JPEG compression attacks used quality factors of 90, 80, and 70. Shift attacks involved circular shift with varying row and column. Rotation attacks are rotation at 1.2, and 4 degrees with auto-cropping. Cropping attacks cropped off a rectangular region of the stego images by setting its pixels to zero value. Mean filtering attacks have kernel sizes ranging from \(2 \times 2\) to \(5 \times 5\). Sample images of these attacks are included in the Appendix A. To allow high quality modifications that do not affect the visual quality of the images, threshold values for each image were selected so that the test images underwent local shift, low level of noise insertion, and high quality JPEG compression, are classified as authentic. It is observed that Baboon has the lowest threshold at 0.70 whereas Pepper has the highest threshold at 0.88. This can be explained by the complexity of image texture. Overall, Baboon has the most complex texture and it caused the lowest correlation value \( corr_2 \) in authentication because the extracted watermark has more distortion compared to those of other images. Adversely, Pepper has smooth textures, thus its watermark has the highest correlation value. The use of correlation-based thresholding is inherently weak to shifting attacks. For example, circular shift of one row does not affect the visual quality of the image but the result will be classified as non-authentic.
### 5.4 Analysis of Experimental Results

Table 5.1 Semi-fragile authentication under various attacks and threshold selection

<table>
<thead>
<tr>
<th>Attack</th>
<th>Attack level</th>
<th>corr2 value</th>
<th>Lena</th>
<th>Baboon</th>
<th>Pepper</th>
<th>Fishing boat</th>
</tr>
</thead>
<tbody>
<tr>
<td>No attack</td>
<td></td>
<td></td>
<td>0.88</td>
<td>0.72</td>
<td>0.88</td>
<td>0.83</td>
</tr>
<tr>
<td>Local shift</td>
<td></td>
<td></td>
<td>0.87</td>
<td>0.72</td>
<td>0.88</td>
<td>0.83</td>
</tr>
<tr>
<td>Histogram equalization</td>
<td></td>
<td></td>
<td>0.29</td>
<td>0.22</td>
<td>-0.21</td>
<td>-0.14</td>
</tr>
<tr>
<td>Noise</td>
<td>0.0005</td>
<td></td>
<td>0.87</td>
<td>0.72</td>
<td>0.88</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td></td>
<td>0.87</td>
<td>0.72</td>
<td>0.88</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>0.005</td>
<td></td>
<td>0.84</td>
<td>0.67</td>
<td>0.85</td>
<td>0.78</td>
</tr>
<tr>
<td>JPEG compression</td>
<td>90</td>
<td></td>
<td>0.87</td>
<td>0.71</td>
<td>0.88</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td></td>
<td>0.68</td>
<td>0.59</td>
<td>0.87</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td></td>
<td>0.68</td>
<td>0.49</td>
<td>0.82</td>
<td>0.15</td>
</tr>
<tr>
<td>Shifting</td>
<td>[1 0]</td>
<td>-0.10</td>
<td>-0.01</td>
<td>0.02</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[0 2]</td>
<td>0.20</td>
<td>-0.01</td>
<td>-0.18</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[3 0]</td>
<td>0.06</td>
<td>-0.02</td>
<td>0.09</td>
<td>-0.08</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[2 2]</td>
<td>-0.05</td>
<td>0.02</td>
<td>-0.01</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>Rotation</td>
<td>1° and crop</td>
<td>0.02</td>
<td>0.02</td>
<td>-0.05</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2° and crop</td>
<td>0.06</td>
<td>0.08</td>
<td>-0.13</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4° and crop</td>
<td>0.01</td>
<td>-0.03</td>
<td>0.08</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>Cropping</td>
<td>(1:50,460:512)</td>
<td>0.84</td>
<td>0.70</td>
<td>0.85</td>
<td>0.78</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1:90,420:512)</td>
<td>0.79</td>
<td>0.65</td>
<td>0.79</td>
<td>0.71</td>
<td></td>
</tr>
<tr>
<td>Mean filtering</td>
<td>2x2</td>
<td>-0.46</td>
<td>-0.02</td>
<td>0.55</td>
<td>0.37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3x3</td>
<td>-0.34</td>
<td>-0.13</td>
<td>0.54</td>
<td>-0.35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4x4</td>
<td>0.12</td>
<td>-0.03</td>
<td>-0.06</td>
<td>-0.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5x5</td>
<td>0.02</td>
<td>0.05</td>
<td>0.07</td>
<td>-0.07</td>
<td></td>
</tr>
</tbody>
</table>

**Threshold t1**

<table>
<thead>
<tr>
<th>Lena</th>
<th>Baboon</th>
<th>Pepper</th>
<th>Fishing boat</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.86</td>
<td>0.70</td>
<td>0.88</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Besides correlation, PSNR value can also be used in authentication because it is based on the same principle of measuring the content likelihood between two images. The PSNR of the stego image compared to its cover image is given by the equations (2.1). Therefore, the PSNR value calculated using the extracted watermark and the down-scaled version of the test image can replace the correlation value in image authentication.

Table 5.2 lists the PSNR value for each image under various attacks. Based on those results, suitable threshold values \( t1 \) for each image are also suggested.
### Table 5.2 Alternative semi-fragile authentication and threshold selection

<table>
<thead>
<tr>
<th>Attack level</th>
<th>PSNR value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lena</td>
</tr>
<tr>
<td>No attack</td>
<td>20.98</td>
</tr>
<tr>
<td>Local shift</td>
<td>20.69</td>
</tr>
<tr>
<td>Histogram</td>
<td>10.08</td>
</tr>
<tr>
<td>equalization</td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>20.85</td>
</tr>
<tr>
<td>0.0005</td>
<td>20.70</td>
</tr>
<tr>
<td>0.001</td>
<td>19.88</td>
</tr>
<tr>
<td>0.005</td>
<td>20.84</td>
</tr>
<tr>
<td>JPEG compression</td>
<td>17.07</td>
</tr>
<tr>
<td>80</td>
<td>17.25</td>
</tr>
<tr>
<td>70</td>
<td>11.96</td>
</tr>
<tr>
<td>Shifting</td>
<td>12.52</td>
</tr>
<tr>
<td>[1 0]</td>
<td>11.39</td>
</tr>
<tr>
<td>[0 2]</td>
<td>12.59</td>
</tr>
<tr>
<td>[3 0]</td>
<td>13.09</td>
</tr>
<tr>
<td>[2 2]</td>
<td>13.54</td>
</tr>
<tr>
<td>Rotation</td>
<td>13.22</td>
</tr>
<tr>
<td>1° and crop</td>
<td>19.81</td>
</tr>
<tr>
<td>2° and crop</td>
<td>18.63</td>
</tr>
<tr>
<td>4° and crop</td>
<td>11.35</td>
</tr>
<tr>
<td>Cropping</td>
<td>11.53</td>
</tr>
<tr>
<td>(1:50,460:512)</td>
<td>11.71</td>
</tr>
<tr>
<td>(1:90,420:512)</td>
<td>10.51</td>
</tr>
<tr>
<td>Mean filtering</td>
<td></td>
</tr>
<tr>
<td>2x2</td>
<td></td>
</tr>
<tr>
<td>3x3</td>
<td></td>
</tr>
<tr>
<td>4x4</td>
<td></td>
</tr>
<tr>
<td>5x5</td>
<td></td>
</tr>
<tr>
<td>Threshold t1</td>
<td>20.00</td>
</tr>
</tbody>
</table>

### 5.4.4 Tamper Localization

Tamper localization is performed if a test image is not authentic. Tampered regions are detected by comparing the extracted watermark with the down-scaled version of the test image. Instead of up-scaling the watermark to the size of the test image for authentication, we down-scaled the test image to the size of the watermark to reduce computation. Figure 5.6 illustrates an example of tamper localization. The unaltered stego image is in Figure 5.6 (a). Tampering was done by copying the flower knot near the edge of the hat and pasting its magnified version onto the centre of the hat. The result is showed in Figure 5.6 (b). Tamper localization correctly highlighted the tampered region as depicted in Figure 5.6 (c). However, a small area of the tampered region was not classified as a tampered region due to the selected threshold value $t_2$. This demonstrates the semi-fragile nature of the watermarking method. In order to achieve a high level of fragility, a high value of threshold can be chosen.
5.4 Analysis of Experimental Results

5.4.5 Content recovery

Although correlation-based authentication is not new, semi-fragile watermarking methods that offer content recovery under cropping attack are very rare. This watermarking method provides tamper localization and approximate content recovery. The tampered regions can be identified correctly, and the approximately recovered contents give the user an idea of the image regions altered. Such information can be useful for human judgment in determining the severity of tampering. Figure 5.6 (d) depicts the approximately recovered content of the tampered region. The recovery is done using the extracted watermark information after localizing the tampered region. Comparing the recovered image in Figure 5.6 (d) with the original stego image in Figure 5.6 (a), the recovered content was nearly identical. However, due to the limited amount of watermark information embedded, the recovery cannot provide detail information such as complex textures and crisp
edges. For example, Figure 5.7 illustrates a stego image with its top right corner cropped off, and the approximately recovered content. The self authentication and self recovery capabilities of this watermarking method made it practical in a real life scenario where a reference image may not be available. By comparing the attacked image with its recovered image, an investigator would be able to deduce forensic information.

![Figure 5.7 Approximate content recovery](image)

(a) Top-left corner cropping on the stego image (b) Approximately recovered content without edge details

One major weakness of semi-fragile watermarks for content recovery is their fragile nature. If a large region of the test image was cropped off, or underwent severe distortion, then the watermark information is lost and content recovery is impossible. To have good performance in content recovery, a robust watermark is needed.

It should be noted that direct comparison with other work is not always possible because some researchers focused on specific subsets of the image restoration problem. For instance, Kundur [34] used a deconvolution method to undo localized linear blurring while many others proposed robust watermarks for image recovery. We aim at general image restoration using the full grayscale information in a semi-fragile watermark without any reference images. The full greyscale information can give high quality content recovery, and the independence of reference images make our method practical.

### 5.4.6 False positive condition

To evaluate the watermarking method under false positive condition, the 4 test images were not embedded with any watermark and sent to watermark detection
5.4 Analysis of Experimental Results

step. Each of the images was tested 10 times in watermark detection. The results appeared as random noise instead of a down-scaled version of the cover image. These indicated that the watermarking method works correctly.

5.4.7 Watermark security

Security of the watermark is achieved by randomizing watermark pixel positions using the secret key $skey(p,q)$. This is necessary to deter malicious attacks when the watermarking algorithm is made public. For a watermark of $64 \times 64$ pixels and 256 grayscales, there are $64 \times 64 \times 256 = 2^{20}$ possible combinations. If we simplify the problem with a binary watermark, then there will be $64 \times 64 \times 2 = 2^{13}$ possible combinations. In addition, the quantization parameters $B$, $N1$, and $N2$ make it secure against quantization estimation attack. Together with other watermarking parameters such as the threshold value, an adversary would have to try a huge number of combinations in order to break the symmetric key system.

5.4.8 Comparative analysis

Focusing on general image restoration using the full grayscale information without any reference image, a close comparison is made with a few closely similar methods. We obtained stego image PSNR of 41.26 dB for the 512×512-pixel Lena using a single watermark whereas Tang-Hung [95] can only achieve 30.2dB with two watermarks, i.e. one for authentication and another for image recovery. Our result is comparable to a similar level of robustness (QRmode 0) and visual quality (42.60 dB) in SARI 1.0 of Lin-Chang [35, 36] where authentication and weak recovery watermarks are embedded in the Lena image.

In Rey-Dugelay [93] and their previous work in recoverable watermarks[96], their watermarks consist of ASCII text, binary logo, and random binary sequences whereas our watermark is the full grayscale image.

The content recovery capability in Fridrich [48] was achieved using a very fragile watermark, and authentication is done using a semi-fragile watermark. We offer an all-in-one solution with a single well-balanced watermark.

5.4.9 Overall performance

Our watermarking method was designed to balance a few conflicting requirements such as semi-fragility, imperceptibility, tamper localization, content recovery, and
computational costs. As a result, there are a few advantages and limitations associated with the watermarking method.

Our method provides content authentication by allowing semantic-preserving changes such as high quality JPEG compression, minor local distortion, and minimal noise insertion. Other changes such as histogram equalization, cropping, rotation, and mean filtering are classified as malicious attacks because they affect the visual quality of the image. The method is practical because it does not require a reference image during content authentication. Tampered regions can be located correctly, and an image’s original content can be recovered. The approximately recovered content could give the user some indication of the cover image appearance. The watermark information is secured by a secret key that randomizes the watermark pixel positions. The single transform, correlator detector, and down-scaled processing spaces of the method offer low computational cost.

The watermarking method is inherently unable to recover image content if exposed to severe attacks such as a major cropping. This vulnerability must be overcome by a robust watermark. In addition, due to the adoption of correlator detector, the watermarking method cannot classify minor shift as legitimate modification. To balance imperceptibility and semi-fragility, the watermark is embedded in the second level of wavelet subbands. This will result in a downgrade of accuracy in tamper localization.

5.5 Chapter Summary

Semi-fragile watermarks are suitable for content authentication where legitimate modifications are allowed and malicious attacks are alerted. Based on the limitations of current watermarking methods, a list of design objectives was created to address the challenges. Then, we developed and evaluated a semi-fragile watermarking method that offers self authentication and self recovery. The method is designed to achieve a balance among several conflicting requirements. Overall, the watermarking method achieved its objectives in providing self authentication and self recovery in a flexible manner.

In the next chapter, we will combine the robust watermarking method described in Chapter 4 with the semi-fragile watermarking method in this chapter to form a hybrid watermarking method.
Chapter 6

Hybrid Watermarking Method

Robust watermarks are normally used in copyright protection because they are hard to be removed from the associated image. On the other hand, fragile watermarks are suitable for authentication because they are sensitive to changes made on the image. Between these two types of watermarks is the semi-fragile watermark. Semi-fragile watermarks are usually applied in content authentication because they allow legitimate changes and can detect malicious tampering.

Hybrid watermarking methods combine robust and fragile watermarks to enable copyright protection and authentication in an integrated solution. In addition, the hybrid watermarking methods can also combine robust and semi-fragile watermarks to offer deductive information in digital media forensics. Section 6.1 analyzes related work in hybrid watermarking method.

We investigated hybrid watermarking methods in 3 phases. In the first phase, a pilot study was performed in combining a robust watermark and a fragile watermark into a hybrid watermarking method. The main objective of this study is to evaluate the watermark imperceptibility in the stego images. In addition, we also study the feasibility of annotation using the robust watermark, and practicality of tamper detection using the fragile watermark. This pilot study is described in Section 6.2.
In the second phase, a hybrid watermarking method with good performance [21]
was chosen for comparative analysis. Section 6.3 discusses this comparative analysis.
It has a robust watermark and a fragile watermark that can be combined into a hybrid
watermarking method using two implementation methods. The first method has the
robust watermark and the fragile watermark overlapped while the second method
uses non-overlapping robust and fragile watermarks. We compared these two
methods in terms of watermark imperceptibility, robustness/fragility, and
computational cost. The result of this analysis is used as a guide for the third phase of
developing a hybrid watermarking method.

In the final phase, we merge our geometric invariant domain with our semi-
fragile watermark to produce a hybrid method. In doing so, the implementation
method of overlapping the two watermarks is adopted. The robust watermark utilizes
a geometric invariant domain to eliminate synchronization problem. The semi-fragile
watermark uses dynamic quantization of wavelet coefficients in self-embedding to
enable self-authentication and self-recovery. The hybrid method offers an integrated
protection for digital images, where detection of the watermarks does not require any
reference image. This hybrid method fulfilled the copyright protection, tamper
detection, and content authentication objectives when evaluated in an investigation
scenario. The final phase of the hybrid watermarking study is explained in Section
6.4.

6.1 Related Work
This section discusses existing hybrid watermarking methods and their drawbacks.
Firstly, we describe some hybrid watermarking methods that are designed for
medical images. This is to prepare for our pilot study on hybrid watermarking.
Secondly, we explain hybrid watermarking methods that combine robust and fragile
watermarks. The design and development of our hybrid watermarking method will
address the shortcomings found.

There are not many hybrid watermarking methods found in the literature
compared to single watermark methods. This could be due to the complexity in
designing a hybrid method. However, the increased protection functions in a hybrid
method may have encouraged some research work published. Robust digital image
watermarks are suitable for copyright protection because they remain intact with the
protected content under various manipulative attacks. The annotation watermark can take the robust form in order to preserve data integrity. Annotation information can be patient name, hospital name, date and time of imaging process, and image dimension. On the other hand, the fragile watermarks are good for tamper detection. Hybrid watermarking methods that are designed for medical images are very few compared to those methods that cater for general types images. This is due to the stringent requirement of high visual quality in watermarking medical images. Three methods are found, and analyzed in the following paragraphs.

- Wakatani [97] proposed a watermarking method that avoids embedding watermark in the region of interest (ROI). Although it preserves the image quality in that region, the major drawback is the ease of introducing copy attack on the non-watermarked regions. In contrast to that method, we embed a fragile watermark that covers the entire central region of an image. In this way, tampering in small regions can be located easily.

- Giakuomaki et al. [98] proposed a wavelet-based watermarking scheme to embed multiple watermarks in medical images. Although the scheme offers medical confidentiality and record integrity, the visual quality of stego images can be improved to achieve higher PSNR values.

- Another approach is to create a virtual border by inserting extra line of pixels around image borders in order to embed watermarks within it [99]. This approach increases file size and storage space. Such approach is in contrast to space saving objective of watermarking. In addition, the absent of a fragile watermark makes it vulnerable to tampering.

To improve these weaknesses, we will create a simple hybrid watermarking method in a pilot study.

We analyze hybrid watermarking methods that cater for general types of images in order to prepare for the third phase of this study. Most of the hybrid watermarking methods use a robust watermark for copyright protection, and a fragile watermark for tamper detection. The following paragraphs discuss some methods found.

- One early hybrid method was proposed by Fridrich [52]. It consists of a block-based robust watermark and a fragile watermark. In another block-based method developed by Deguillaume et al. [21], the robust watermark
can be self-synchronized using an ACF, and its fragile watermark is a hashed signature embedded in LSBs. However, the ACF peaks could be removed using an attack proposed by Lu and Hsu [43].

- A hybrid method reported in Fan-Tsao [54] was designed specifically for JPEG2000 format. Its stego images have poor visual quality because the robust watermark was embedded in the low-pass subbands where it caused much distortions. Furthermore, the robustness against common image processing and geometrical attacks is not tested.

- In Habib-Sarhan-Rajab [55], an image is partitioned into blocks and watermarked separately in the DCT domain. The robust watermark is embedded into the middle frequency coefficients, and the fragile watermark is embedded into the LSBs. Both of the watermarks are binary pictures. The robust watermark detection is based on the correlation value between the extracted binary picture and the validating watermark. The fragile watermark is extracted from the LSBs of the DCT coefficients.

- Another hybrid method reported by Sharkas-ElShafie-Hamdy [56] uses two nested watermarks in a nonblind manner. Firstly, the robust watermark, consists of a pseudorandom sequence, is embedded into an image. Then, the stego image is used as a fragile watermark and embedded into another image. The robust watermark is detected by its correlation value with a validating pseudorandom sequence, and the fragile watermark detection is the extraction of the primary stego image. However, the feasibility of embedding a robust watermark into a fragile watermark is questionable. If the fragile watermark is destroyed by an attacker, then the robust watermark (which is embedded in the fragile watermark) would not be detectable, and lose its robustness.

So far most of the hybrid methods consist of a robust watermark and a fragile watermark. There is only one hybrid method that combines a robust watermark with a semi-fragile watermark [57]. They use a DCT domain for watermarking JPEG images. However, no rotation or translation attacks were tested on its robust watermark. Additionally, hybrid methods that are designed specifically for JPEG [57] and JPEG2000 [54] formats may not perform well under other file formats.
None of the hybrid methods surveyed was designed to perform content recovery. We focus on hybrid methods that combine a robust watermark with a semi-fragile watermark because they have conflicting characteristics that made them suitable for different purposes. To the best of our knowledge, there is no hybrid method that combines the features of a robust watermark and a semi-fragile watermark to offer all these functionalities with blind watermark detection: copyright protection, content authentication, tamper localization, and approximate content recovery. The recovered content can provide useful forensic information in an investigation.

6.1.1 Content recovery and Watermark Generation

Content recovery is the restoration of cover image features. For example, a cropped region can be recovered to its original pattern and intensity. To date, content recovery is not investigated in hybrid methods although it appeared in single watermark methods.

They are generally self-embedding and self-recovery, without the need of a reference image. Most of them applied block-based processing to attain good localization in tamper detection.

There are three major approaches to content recovery, i.e. fragile watermark, robust watermark, and semi-fragile watermark. Robust watermark in DWT domain was proposed by Li, Chen and Wu [100] and Wang et al. [101]. Fragile watermarking approaches for content recovery are mainly based on LSB embedding and detection (e.g. [48], [102]). The third approach is a semi-fragile watermark that offer approximate content recovery (e.g. [48], Self-Authentication-and-Recovery Image Watermarking System (SARI) [36],[35]). While fragile watermarks offer good quality content recovery, it can be easily destroyed. Therefore, we choose semi-fragile watermark in our hybrid method.

Among the many semi-fragile watermarking methods, only a few have both tamper localization and content recovery capabilities [47] (e.g. [36], [93]). However, a major shortcoming of these methods is the high amount of computation due to their iterative embedding process.

An item worth noting is the method of generating a watermark. Compression is usually used to reduce the watermark payload in the generated watermark (e.g. fractal compression [102], block truncation code (BTC) [101]). On the other hand,
the redundant information in an uncompressed watermark could be useful in combating severe attacks and cropping.

6.2 Pilot Study on Hybrid watermarking method

In this section, we create a hybrid watermarking method that combines a robust watermark with a fragile watermark. The robust watermark is used for annotation instead of copyright protection. The emphasis here is to demonstrate a workable robust watermark and not the high level of watermark robustness. The fragile watermark is used for tamper detection and tamper localization. The main objective of this study is to evaluate the watermark imperceptibility in the stego images and the overall feasibility of hybrid watermarking. Although medical images were used in this pilot study, other images can be used as replacement.

6.2.1 The Design

We created a hybrid watermark method as shown in Figure 6.1 below. The annotation watermark and the fragile watermark are embedded separately into different regions of the image.

(i) Annotation watermark for privacy control

To provide data security and patient privacy, patient information can be encrypted and carried by an annotation watermark. In addition, the identity of the medical practitioner involved in the imaging process can be digitally signed using a digital signature which is also carried by the annotation watermark for authentication.

The annotation watermark is embedded into the border pixels of the image using a robust embedding method described in Section 4.2 in Chapter 4. A watermark signal is arranged in a frame pattern as illustrated in Figure 6.2. Then, it is embedded
6.2 Pilot Study on Hybrid watermarking method

using a linear additive method into the three high pass bands of DWT of the cover image borders. This is carried out at the first level of the DWT sub-bands. An inverse DWT is performed on the marked coefficients to obtain the marked image border. This is depicted in Figure 6.3. Although the illustrations use fixed size borders for a square image, this method can be easily adapted to rectangular images of any sizes.

![Figure 6.2 An annotation watermark arranged in frame pattern](image)

Figure 6.2 An annotation watermark arranged in frame pattern

![Figure 6.3 Image borders for annotation watermarking](image)

(a)          (b)

Figure 6.3 Image borders for annotation watermarking (a) Cover image borders used in annotation watermark embedding. (b) Stego image borders.

(ii) Fragile watermark for tamper detection

The integrity of the medical image can be authenticated using a fragile watermark. Tampering on the image can be detected by examining the tiled fragile watermark patterns.

The fragile watermark is embedded into the central region of the cover image using the LSB method. In this way, we can ensure that distortion is not too severe for majority parts of the ROI in the image. As a result, the stego image has good level of watermark imperceptibility. The image border is reserved for annotation watermark.
embedding. A binary watermark pattern is tiled to cover the whole image, and its binary pixel values are used to overwrite the corresponding LSBs of the cover image pixels. Figure 6.4 gives an example of the process using X-ray image of the chest.

![Image](image1.png)

Figure 6.4 Fragile watermark embedding (a) Cover image, (b) Fragile watermark embedded into central region of an X-ray chest image.

After the annotation watermark and fragile watermark are embedded, the two parts are combined to form a complete hybrid stego image. Figure 6.5 shows a hybrid watermarked image.

![Image](image2.png)

Figure 6.5 Hybrid stego image

(iii) Watermark detection

For watermark detection, the annotation watermark and the fragile watermark are detected separately, similar to their embedding steps. The detection of annotation watermark takes a few steps similar to its embedding process. Firstly, the border of the stego image is decomposed into its DWT sub-bands. Then, the correlation value is calculated using the three high pass band coefficients. Finally, the calculated value is compared with a dynamically computed threshold value to determine successful watermark detection as explained in Section 4.2 in Chapter 4. The fragile watermark
is detected using a simple LSB detection method. The LSBs of each pixel in the stego image is read to form the tiled binary watermark pattern. Figure 6.6 shows the correctly tiled fragile watermark detected in the central region of the image, and the annotation watermark patterns around the image borders.

### 6.2.2 Analysis of Experimental Results

Three types of medical images that represent soft tissues and hard tissues characteristics were used in the experiment, i.e. X ray image of the chest, MR image of the skull, and CT image of the brain. These images are depicted in Figure 6.7.

![Detected Fragile Pattern](image)

**Figure 6.6** Fragile and annotation watermark patterns detected without attacks

**(i) Visual quality of stego images**

The visual quality of a stego image is measured in WPSNR because it is generally more accurate than PSNR [19, 21]. A test on X ray chest image provided very good imperceptibility of 60.78dB, well above the 50dB benchmark. The annotation part and fragile part were detected correctly.

MR of the skull gives WPSNR of 60.80dB, and the CT brain image gives WPSNR 60.70dB. Figure 6.7 provides a visual quality comparison between the cover image and the stego image.

**(ii) Tamper detection using the fragile watermark**

Some of the general image manipulations were performed as attacks to evaluate the effectiveness of the fragile watermark. These attacks are easy to perform using off-the-shelf image processing software, and they pose a significant threat to the integrity of medical images. The effects of these attacks are hard to be identified by human eyes. Fortunately, it can be detected using the fragile watermark. The attacks are tabulated in Table 6.1.
Figure 6.7 Test images and their hybrid stego image with its respective WPSNR: from top to bottom are X ray image of the chest, MR image of the skull, and CT image of the brain.

Table 6.1 General attacks on fragile watermark

<table>
<thead>
<tr>
<th>No.</th>
<th>Attack</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Noise insertion</td>
<td>Gaussian noise with zero mean and variance 0.0002.</td>
</tr>
<tr>
<td>2</td>
<td>JPEG compression</td>
<td>Quality factor 90%.</td>
</tr>
<tr>
<td>3</td>
<td>Copy attack</td>
<td>Copy a region and paste it on another region with similar texture.</td>
</tr>
</tbody>
</table>

Gaussian noise with zero mean and variance 0.0002 was inserted into the stego image to evaluate the effectiveness of the fragile watermark in tamper detection. Figure 6.8 illustrates the test results.
6.2 Pilot Study on Hybrid watermarking method

Figure 6.8 Gaussian noise attack on the fragile watermark. (a): Original CT brain image; (b): Gaussian noise added to the stego image; (c): Fragile watermark pattern destroyed by the Gaussian noise.

A test on JPEG compression with quality factor 90% on the CT brain image is shown in Figure 6.9. The JPEG compressed stego image looks very similar to the cover image. However, the fragile watermark tile pattern is destroyed by the JPEG compression. This alerts us that the image is not authentic.

Figure 6.9 JPEG compression attack on the fragile watermark. (a): Original CT brain image; (b): JPEG compressed stego image; (c): Fragile watermark tile pattern destroyed by the JPEG compression attack.

Figure 6.10 shows an example of copy attack detected by the fragile watermark. Although it is hard for human eyes to identify the tampered regions, this method makes it possible to do so by highlighting the distorted tiled patterns.
Figure 6.10 Detection of a copy attack. (a) Copy attack on a stego image. (b) Two tampered regions are detected by the fragile watermark (the circled regions).

6.2.3 Conclusion
The hybrid-watermarking method presented has shown to be suitable for use in medical images. The annotation watermark can be used to embed patient information in a private and secure manner, while the fragile watermark offers tamper detection. The visual quality of stego image is very good. In addition, the effectiveness of the fragile part in tamper detection has been proven under some general image manipulation attacks. On the other hand, the annotation watermark is meant to store context information in a private manner without increasing storage space requirement. Although a single bit of robust watermark is experimented, multibit watermark can be embedded by dividing the image into multiple regions and processed separately.

The performance factors of a watermarking method are mutually conflicting. For instance, increasing watermark robustness normally degrades its imperceptibility and limits its embedding capacity. To achieve a desirable balance among the performance factors, a designer must understand the influence of one factor on another. Therefore, it is important to evaluate the effects of system architecture on a hybrid method.

6.3 Comparative Analysis of Hybrid Watermark Implementations
This section analyses an overlap and a non-overlap implementation of the robust and fragile parts in a hybrid watermarking method. The hybrid watermarking method
6.3 Comparative Analysis of Hybrid Watermark Implementations

proposed by Deguillaume [21] was chosen in this study due to its good performance. Its robust watermark can resist many types of image distortions, and its fragile watermark can locate tampered regions correctly. The difference between the two implementation methods lies in the robust and fragile watermarks embedding positions. The first method ensures that robust and fragile watermarks are embedded in non-overlapping positions, and will be called “non-overlap” implementation. The second method overlaps both watermarks, and will be called “overlap” implementation. The overlap implementation has the advantage of full embedding capacity and higher localization in tamper detection. However, the compromise in its robustness and computational cost need to be investigated. The comparison include watermark embedding capacity, computational cost, robustness of the robust part, and tamper detection effectiveness of the fragile part. In addition, the effects of block size on the hybrid method’s performance are also studied.

6.3.1 The Chosen Method

The hybrid method chosen in this analysis [21] embeds a periodic robust watermark pattern in the DWT domain. It uses robust estimation technique with superior performance to enable watermark detection. Thus, it is suitable for real life application where the cover image may not be available during watermark detection. It also has a state-of-the-art fragile part that embeds watermark blocks in the LSB of pixels. The following two paragraphs describe the robust and fragile parts. Detail steps can be found in [21].

The robust part uses a self-reference method to recover from geometrical distortions. Firstly, the watermark signal is encoded using an ECC for reliable decoding, and encrypted for confidentiality. Secondly, the signal bits are spread in a symmetric pattern to cover the whole image size. This provides regularly-spaced peaks in geometrical resynchronization for watermark detection. Finally, the watermark is embedded in DWT domain for robustness. Instead of employing HVS masking for imperceptibility, we simplified it with constant energy embedding. To detect a watermark in an attacked image, it exploits the periodic peaks of magnitude spectrum for image resynchronization. These steps are detailed in Deguillaume’s work [21]. We applied thresholding on the magnitude spectrum to extract the peaks for simplicity. Figure 6.11 (a) depicts peaks obtained from the magnitude spectrum
of the embedded watermark. Figure 6.11 (b) shows peaks extracted from a rotated stego image. Assuming local distortions are restricted by the acceptable image quality change, local and non-linear transformations can be recovered using the same approach at the local level. Details of such approaches are described in Voloshynovskiy’s article. After that, a watermark estimation based on Maximum a Posteriori (MAP) probability is applied on the resynchronized image. Then, a correlator detector is used in watermark decoding with a threshold value.

The fragile part uses a block-wise scheme to locate tampered regions. It computes a key-dependant hash value for overlapping blocks of an image and embeds the value into the LSB of pixels inside that block. By comparing the estimated signatures of the fragile blocks, tampered regions can be highlighted.

(i) Non-overlap Implementation
In the non-overlap implementation, the embedding of its robust and fragile parts is performed simultaneously as described above. The robust positions do not overlap with the fragile positions within each block. Hence it was named “orthogonal” in Deguillaume’s work [21]. The detection of watermarks in the non-overlap implementation is the same as its embedding part where the robust and fragile parts are processed independently. In our implementation, the robust blocks and fragile blocks are chosen to have the same size.

The block-wise hashing of fragile part takes the current block with its eight neighboring blocks as input. The computed hash code is then embedded into the current block. This provides local contextual dependency. However, this approach not only detects modifications within the block but also modifications in its neighboring blocks. Compensation steps mentioned in [21] are not implemented in this study. Computational simplicity is preferred in this study.

(ii) Overlap Implementation
In this implementation, the robust part is embedded prior to the fragile part. By definition, the robust part must survive distortions caused by the fragile part. Therefore, we embed the fragile part in all positions, overwriting the LSBs of the robust stego image. As a result, both the robust and fragile parts can be embedded into all positions, achieving maximum watermarking coverage. Therefore, it gives the highest possible localization for tamper detection. In addition, it also reduces computation by eliminating position tracking of the robust and fragile parts.
6.3 Comparative Analysis of Hybrid Watermark Implementations

Figure 6.11 Peak pattern samples (a) Peaks obtained from the magnitude spectrum of the embedded watermark. (b) Peaks extracted from a stego image with 30 degree rotation and auto-crop

The watermark detection in the overlap implementation is similar to those in the non-overlap implementation. However, all pixel locations in overlap implementation are processed because both the robust and fragile parts are embedded in every position. This requires an examination of the compromise in computational cost.

6.3.2 Experimental Results Analysis

To compare the overlap and non-overlap implementations in a hybrid method, the parameters listed in Table 6.2 were applied. Two test images with 256 gray levels were used. They are Lena and Cameraman of 256×256 pixels. A set of general image manipulation operations listed in Table 6.3 was used to evaluate the performance of the robust watermark. Three types of attacks were used in fragile watermark evaluation: local tampering modify the pixel values of a small area, copy attack copies a small region from an image and pastes it onto the same image whereas collage attack pastes it onto another image.

Table 6.2 Parameter values for non-overlap and overlap implementations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Non-overlap implementation</th>
<th>Overlap implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block size $t_1 \times t_2$</td>
<td>$16\times16 = 256$ pixels</td>
<td>$16\times16 = 256$ pixels</td>
</tr>
<tr>
<td>Robust positions</td>
<td>178</td>
<td>256</td>
</tr>
<tr>
<td>Fragile positions</td>
<td>42</td>
<td>256</td>
</tr>
<tr>
<td>Empty positions</td>
<td>36</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 6.3 Attacks for robustness evaluation

<table>
<thead>
<tr>
<th>Attack</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation followed by cropping</td>
<td>Rotate 30 degree with auto-crop</td>
</tr>
<tr>
<td>Scaling followed by shearing</td>
<td>Uniform scaling factor 0.98; Shear 2%</td>
</tr>
<tr>
<td>JPEG compression</td>
<td>Quality factors at 50%</td>
</tr>
<tr>
<td>Gaussian noise insertion</td>
<td>Zero mean; variance 0.02</td>
</tr>
<tr>
<td>Contrast adjustment</td>
<td>Gamma value 0.6</td>
</tr>
<tr>
<td>Median filtering</td>
<td>2×2 smoothening kernel</td>
</tr>
</tbody>
</table>

(i) Analysis of the Robust Part Results

Using block size of 32×32, the stego images of non-overlap and overlap implementations give PSNR of 37.02dB and 37.29dB respectively. This indicates image qualities of both implementations are very close. Such observation can be explained by the small difference of un-marked positions between the two implementations, i.e. 256 – 178 = 78 bits ≈ 30.47% in each block.

Regularly-spaced peaks can be observed after thresholding the magnitude spectrum of the non-attacked stego images. These patterns are very similar to those of the embedded watermark. Therefore, the peak patterns can be used in geometrical resynchronization, and the robust watermark signal can be extracted successfully in both non-overlap and overlap implementations. The non-attacked stego image of overlap implementation gives better peak patterns compared to those of the non-overlap implementation because it has full embedding capacity. To improve the robustness of the implementation modes, the watermark embedding energy can be increased to warrant better peak patterns, but it will degrade the visual qualities. In the non-overlap implementation, compromise must be made between the densities of robust part and fragile part. Increasing fragile watermark positions to enhance its localization in tamper detection will reduce those of the robust watermark, thus degrade its robustness.

To evaluate the robustness of the watermark, the attacks listed in Table 6.3 were carried out. With the obvious axes in the peak patterns, distortions can be compensated with a resynchronization step to enable successful watermark detection. This is done using Hough transform to estimate the rotation angle, and Maximum Likelihood (ML) to estimate peak periods. Details of the recovery steps can be found in Deguillaume’s work [21]. There are two items worth description here. Firstly, the estimation outcome of Hough transform may deviate one degree. Therefore, brute
force search need to be applied in finding the correct parallel lines for period estimation. Secondly, a predefined period range must be specified in the estimation of period between peaks as mentioned in Deguillaume’s work [21]. Overall, both of the implementation modes are equally robust to the attacks. The robust watermark was detected in both non-overlap and overlap implementations after resynchronization.

Computational costs for the implementation methods are listed in Table 6.4 for block size 16×16 pixels. The overlap implementation requires more processing time because it embeds robust watermark into every pixel in each block whereas the non-overlap implementation only need to process about 70% of the pixels in each block. The savings of not tracking robust and fragile watermark positions in an overlap implementation does not offset the overall computational costs.

Table 6.4 Comparison of hybrid watermarking time (seconds)

<table>
<thead>
<tr>
<th></th>
<th>Non-overlap implementation</th>
<th>Overlap implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robust embedding</td>
<td>3.10</td>
<td>3.12</td>
</tr>
<tr>
<td>Robust detection</td>
<td>0.49</td>
<td>0.50</td>
</tr>
<tr>
<td>Fragile embedding</td>
<td>5.70</td>
<td>6.35</td>
</tr>
<tr>
<td>Fragile detection</td>
<td>5.60</td>
<td>6.34</td>
</tr>
</tbody>
</table>

(ii) Analysis of the Fragile Part Results
The fragile watermark evaluation for both implementation modes were done using local tampering, copy attack, and collage attack. Local tampering was easily detected and highlighted as shown in Figure 6.12. The cover image of Lena is shown in Figure 3.3 in Chapter 3.

A copy attack on Cameraman stego image and its fragile watermark detection results are given in Figure 6.13. Its cover image is shown in Figure 3.3 in Chapter 3. In the test, a dark color region was copied and pasted onto another region on the cloth of the same image. A similar operation was performed on a textured region, i.e. the lawn. The results of a collage attack involving Lena and Cameraman stego images gave similar results. The fragile watermark in both implementations highlighted tampered regions correctly.

Since the overlap implementation employs full capacity embedding, it was able to highlight modifications at each pixel. Conversely, the non-overlap implementation...
embedded its fragile watermark in about 30% pixels of each block. As a result, it was not as accurate as the overlap implementation.

Besides the three types of attacks above, the effects of block size on the fragile watermark are also examined on the non-overlap implementation. As tabulated in Table 6.5, larger block size requires less processing time. This is due to the convolution operation in hashing neighbouring blocks. Also, large block size allows high security with long signatures. On the other hand, the smaller the block size, the more blocks are involved. Thus, the more computing cycles are needed. Nevertheless, smaller block size offers better localization in tamper detection.

### 6.3.3 Conclusion

The overlap and non-overlap implementations of a hybrid method were analyzed and compared. We have found that both implementations generally produce similar results. This is due to the fact that the robust part in the overlap implementation resisted distortions introduced by the fragile part. Although the overlap implementation reduces computational by not tracking robust and fragile watermark positions, its embedding time and detection time is slightly longer compared to those of the non-overlap implementation. This is caused by the extra processing load of embedding fragile and robust watermarks in all pixel positions.

![Figure 6.12 Example of a tamper detection](image)
(a) Tampered Lena stego image. (b) Tampered regions highlighted by fragile watermark detection
6.3 Comparative Analysis of Hybrid Watermark Implementations

![Image of Tamper Detection](image)

Figure 6.13 Tamper detection in a copy attack scenario (a) Copy-attacked Cameraman stego image. (b) Tampered regions highlighted by fragile watermark detection

<table>
<thead>
<tr>
<th>Block size</th>
<th>Embed time</th>
<th>Detect time</th>
</tr>
</thead>
<tbody>
<tr>
<td>4×4</td>
<td>15.60</td>
<td>15.30</td>
</tr>
<tr>
<td>8×8</td>
<td>7.53</td>
<td>7.51</td>
</tr>
<tr>
<td>16×16</td>
<td>5.70</td>
<td>5.60</td>
</tr>
<tr>
<td>32×32</td>
<td>5.00</td>
<td>4.90</td>
</tr>
</tbody>
</table>

Table 6.5 Effects of block size on fragile watermarking time (seconds)

The overlap implementation offers higher watermark capacity for both the robust and fragile watermarks compared to the non-overlap implementation. Hence, the overlap implementation gives better peak patterns than the non-overlap implementation in robust watermark extraction. Due to the same reason, the overlap implementation has better localization in tamper detection compared to the non-overlap implementation.

Finally, a balance between tamper detection localization and computational cost must be determined when selecting an optimum block size for both implementation modes.

In summary, the overlap implementation can meet high integrity requirements in digital contents while the non-overlap implementation is suitable for commercial applications where processing speed is a preference. Therefore, we will apply the
overlap implementation in our hybrid watermarking method to obtain good authentication results.

6.4 Hybrid watermarking method Combining Robust and Semi-fragile Watermarks

In this section, we merge our geometric invariant domain with our semi-fragile watermark to form a hybrid method. The geometric invariant domain was discussed in Chapter 4, and the semi-fragile watermark was explained in Chapter 5. The implementation method of overlapping the two watermarks is adopted. The hybrid method offers an integrated protection for digital images with blind watermark detection. This hybrid method fulfilled the copyright protection, tamper detection, content authentication, and approximate content recovery objectives when evaluated in an investigation scenario.

6.4.1 Our Watermarking Methods
(i) Robust Watermarking
The robust watermark in our method is based on the geometric invariant domain developed in Chapter 4, and used for copyright protection. The invariant domain is constructed using a combination of FFT, LPM, and DT-CWT. Since geometrical distortions can be modelled as combinations of basic operations, we focus on robustness against RST. The RST invariance is further explained in the next paragraph. Unlike most invariant methods, our method eliminates explicit resynchronization. The robust watermark can survive geometric distortions, common image processing, and JPEG compression. In addition, it exploits perceptual masking property of the DT-CWT subbands, and its watermark detection step does not require the cover image.

(ii) Semi-fragile Watermarking
The semi-fragile watermark in our hybrid method enables blind detection, self-authentication, and self-recovery. We use a dynamic quantization method in DWT domain developed in Chapter 5. This approach has lower computation complexity compared to block-based approaches. Self-embedding with random mapping is suitable for secured self-recovery. The redundancy here can combat cropping attack. This attack may be targeted at robust features in feature-based approaches. Detection decision is adjustable based on the thresholding of correlation result. The authentication decision often depends on application scenario. For instance,
watermarking distortions in medical images must be strictly controlled to avoid misjudgment in diagnosis whereas smoothening of artistic pictures may be allowed in a commercial environment. We classify innocent changes as those operations that produce minimal differences compared to the cover image while other operations are considered malicious. The differences are measured by comparing the extracted watermark to the stego image. The tampered regions are localized and approximate content recovery is carried out using the watermark information extracted.

(iii) **Hybrid Watermarking**

The hybrid watermarking method consists of the robust watermark and the semi-fragile watermark described in the preceding sections. Figure 6.14 depicts the embedding and detection process of the two watermarks.

![Figure 6.14 Hybrid watermark embedding and detection](image)

The robust watermark is embedded first, followed by the semi-fragile watermark. This is logical because the robust watermark should be able to resist distortions caused by the semi-fragile watermark. The same reasoning was discussed by Fridrich [52] and Mintzer-Braudaway [53]. Compared to orthogonal arrangement of the
robust and semi-fragile watermarks, this overlapping design provides better tamper localization and content recovery results as analyzed in Section 6.3 [103].

The stego image may subject to unintentional changes or malicious attacks. To determine whether watermarks exist in a test image, separate detections steps are carried out for each of the watermarks. The detected robust watermark would be used in copyright protection. The extracted semi-fragile watermark can be use in global authentication, tamper localization, and content recovery. This hybrid method provides an integrated protection solution for digital content.

6.4.2 Analysis of Experimental Results
To evaluate the performance of the implemented method, five images representing various characteristics were used. They are shown in Figure 3.3 of Chapter 3, and are identified as Lena, Baboon, Cameraman, Pepper, and Fishing boat. They are all 256×256 pixel with 8-bit greyscale.

The difference between a cover image and its stego image was minimal and did not reveal any information about the watermark because it appeared as random noise.

(i) Evaluation of the Robust Watermark
To evaluate the robustness of the implemented watermarking method, we performed a set of attacks on the stego images using StirMark 3.1 [58],[59] and carried out the watermark detection steps. All attacks listed in Table 4.6 were performed using StirMark except the RST-JPEG combined attack marked with an asterisk (*) which was implemented in Matlab. A watermark of 128×128 pseudo-random binary values was generated and embedded into all of the DT-CWT subbands in the invariant domain. The embedding weight factor f was computed using the average values of all subband coefficients. The false positive probability was set to be less than 10\(^{-8}\) for all types of attacks. The RBA was carried out with bending factor 0.50. The scores in Table 6.6 were normalized to the range from 0 to 1. A score of 1.000 means the watermark was detected in all images for all levels of attacks in that category. Adversely, a score of 0.000 indicates no watermark was detected in all cases. Under rotation with cropping attack, our method scored 0.975. This was due to the lost of watermark information under cropping. For scaling attack and cropping attack, our method achieved 0.933. Many of the robust watermarks were lost at scaling factor
0.5, and rotation angles 45° and 90°. It can be concluded that the robust watermark was reasonably robust because it survived most of the attacks.

Table 6.6 Average score for robustness tests

<table>
<thead>
<tr>
<th>Attack</th>
<th>Average score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation with cropping</td>
<td>0.975</td>
</tr>
<tr>
<td>Scaling</td>
<td>0.933</td>
</tr>
<tr>
<td>Translation</td>
<td>1.000</td>
</tr>
<tr>
<td>JPEG compression</td>
<td>1.000</td>
</tr>
<tr>
<td>Random bending</td>
<td>1.000</td>
</tr>
<tr>
<td>Row and column removal</td>
<td>1.000</td>
</tr>
<tr>
<td>Median filtering</td>
<td>1.000</td>
</tr>
<tr>
<td>Cropping</td>
<td>0.933</td>
</tr>
<tr>
<td>Gaussian filtering</td>
<td>1.000</td>
</tr>
<tr>
<td>Linear transform</td>
<td>1.000</td>
</tr>
<tr>
<td>Aspect ratio change</td>
<td>1.000</td>
</tr>
<tr>
<td>Rotation with cropping and scaling</td>
<td>1.000</td>
</tr>
<tr>
<td>Sharpening</td>
<td>1.000</td>
</tr>
<tr>
<td>Shearing</td>
<td>1.000</td>
</tr>
<tr>
<td>Combination of RST with JPEG compression*</td>
<td>1.000</td>
</tr>
</tbody>
</table>

(ii) Evaluation of the Semi-fragile Watermark
The 256×256 pixel cover image \( f(m,n) \) was down-scaled to 32×32 pixel to form the watermark \( w(p,q) \). The other settings were \( L = 2, N1 = 22, N2 = 400, \) and \( B = 0.25 \). For simplicity and ease of manual verification, \( skey(p,q) \) was chosen as a circularly shifted matrix in both horizontal and vertical directions. This shift at half of its size produced a watermark with 4 quadrants having maximum distance from its original position, and can be powerful in fighting cropping attacks. A randomly permuted \( skey(p,q) \) is preferred for high security system.

Table 6.7 lists suitable authentication threshold values \( t/l \) for two test images after examining its corresponding correlation values. Local shift attack was performed by shifting the region \((65:110, 57:62)\) two columns to its right, and shifting the region \((191:196,130:170)\) one row upwards. Noise attacks involved adding “salt and pepper” noise with varying density. JPEG compression attacks used quality factors of 90, 80, and 70. Shift attacks involved circular shift with varying row and column. Rotation attacks were rotation at 1, 2, and 4 degrees with auto-cropping. Cropping attacks cropped off a rectangular region of the stego images by setting its pixels to zero value. Mean filtering attacks used kernel sizes ranging from \( 2\times2 \) to \( 5\times5 \). To
allow high quality modifications that do not affect the visual quality of the images, threshold values for each image was selected so that the test images underwent local shift, low level of noise insertion, and high quality JPEG compression were classified as authentic. Baboon has the lowest threshold at 0.61 whereas Cameraman has the highest threshold at 0.82. The other images have threshold values ranging from 0.68 to 0.73. Overall, Baboon had the most complex texture and it caused the lowest correlation value \( corr^2 \) in authentication because the extracted watermark suffered more distortions compared to those of other images. Adversely, Cameraman had many flat regions, thus its watermark achieved the highest correlation value. The use of correlation-based thresholding was inherently weak to shifting attacks. Minor shifts caused low correlation values despite the well-preserved visual appearance.

Tamper localization was performed if a test image was not authentic. Tampered regions were detected by comparing the extracted watermark with the down-scaled

<table>
<thead>
<tr>
<th>Attack</th>
<th>Attack level</th>
<th>( corrl )</th>
<th>( corr^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>No attack</td>
<td></td>
<td>0.74</td>
<td>0.63</td>
</tr>
<tr>
<td>Local shift</td>
<td></td>
<td>0.74</td>
<td>0.63</td>
</tr>
<tr>
<td>Histogram equalization</td>
<td></td>
<td>– 0.03</td>
<td>– 0.03</td>
</tr>
<tr>
<td>Noise</td>
<td>0.0005</td>
<td>0.74</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td>0.73</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>0.005</td>
<td>0.70</td>
<td>– 0.41</td>
</tr>
<tr>
<td>JPEG compression</td>
<td>90</td>
<td>0.74</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>0.60</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>0.00</td>
<td>0.15</td>
</tr>
<tr>
<td>Shifting</td>
<td>[1 0]</td>
<td>– 0.05</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>[0 2]</td>
<td>0.10</td>
<td>– 0.01</td>
</tr>
<tr>
<td></td>
<td>[3 0]</td>
<td>0.07</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>[2 2]</td>
<td>– 0.10</td>
<td>– 0.07</td>
</tr>
<tr>
<td>Rotation</td>
<td>1° and crop</td>
<td>0.09</td>
<td>– 0.10</td>
</tr>
<tr>
<td></td>
<td>2° and crop</td>
<td>– 0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>4° and crop</td>
<td>– 0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>Cropping</td>
<td>(1:25,230:256)</td>
<td>0.71</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>(1:45,210:256)</td>
<td>0.61</td>
<td>0.55</td>
</tr>
<tr>
<td>Mean filtering</td>
<td>2×2</td>
<td>– 0.18</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>3×3</td>
<td>– 0.36</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>4×4</td>
<td>– 0.06</td>
<td>– 0.05</td>
</tr>
<tr>
<td></td>
<td>5×5</td>
<td>0.03</td>
<td>0.02</td>
</tr>
<tr>
<td>( Threshold \ t1 )</td>
<td></td>
<td>0.73</td>
<td>0.61</td>
</tr>
</tbody>
</table>
version of the test image. Figure 6.15 illustrates an example of tamper localization. The unaltered stego image is in Figure 6.15 (a). Tampering was done by cropping a region containing a stem near the top left corner of the stego image. The result is showed in Figure 6.15 (b). Tamper localization correctly highlighted the tampered region as depicted in Figure 6.15 (c).

![Figure 6.15 Approximate content recovery](image)

Figure 6.15 Approximate content recovery (a) The unaltered stego image; (b) The test image with tampered hat; (c) Detected tampered region in black color; (d) Recovered image

Figure 6.15 (d) depicts the approximately recovered content of the tampered region. The recovery was done using the extracted watermark information after it had localized the tampered region. Comparing the recovered image in Figure 6.15 (d) with the original stego image in Figure 6.15 (a), the recovered content was nearly identical. The self-authentication and self-recovery capabilities of this watermarking method made it practical in a real life scenario where a reference image may not be available.
6.4.3 Conclusion
A hybrid watermarking method combining a robust watermark and a semi-fragile watermark have been developed and evaluated. The robust watermark was based on a geometric invariant domain constructed using FFT, LPM, and DT-CWT. The semi-fragile watermark based on DWT embeds a down-scaled version of the cover image using dynamic quantization. The extracted semi-fragile watermark can assist in global authentication, tamper localization, and approximate content recovery. The hybrid method offers multiple functionalities: copyright protection, authentication, tamper localization, and content recovery. To the best of our knowledge, this is the first of its kind in hybrid watermarking that provides integrated content protection. The blind watermark detection, non-synchronized, self-authentication, and self-recovery features contributed to the merits of this method.

Two major aspects that need to be improved are the visual quality of the stego image and the effectiveness of content recovery. The many transforms involved in robust watermarking have caused interpolation errors. Thus, better sampling and compensations are required to reduce the errors. The effectiveness of content recovery depends on the quality of extracted semi-fragile watermark. Embedding the semi-fragile watermark in higher level subbands can improve its robustness and ensure better extraction results. This must be carefully tuned by adjusting the quantization step size.

6.5 Chapter summary
Robust and fragile watermarks are suited for different purposes given their contrasting characteristics. Robust watermarks are good for copyright protection due to its robustness against image distortions. On the other hand, fragile watermarks are sensitive to changes in images. Therefore, they are suitable for authentication. Semi-fragile watermarks provide an intermediate alternative to the two extremes of robust and fragile watermarks. They can differentiate legitimate changes and major modifications. As a result, semi-fragile watermarks are usually applied in content authentication.

Hybrid watermarking methods can combine a robust watermark with either a fragile or a semi-fragile watermark to offer an integrated solution for copyright protection and authentication. This chapter discussed our pilot study on a hybrid
watermarking method that combine a robust watermark with a fragile watermark. A comparative analysis of two implementation methods in a hybrid watermarking method is also presented. Based on the analysis result, the final part of this chapter described a novel hybrid method that combines our work in Chapter 4 and Chapter 5. It has geometric invariant domain for robust watermarking, and a semi-fragile watermark for content authentication and content recovery.

This chapter ends the discussion of our work in robust, semi-fragile, and hybrid watermarking. Conclusion and future research will be presented in the next chapter.
Chapter 7

Conclusion and Future Work

7.1 The Need for Watermarking

The economical mass storage and the increasing speed of transferring files over the internet have been promoting the amount of digital content consumption. Digital media can be copied, modified, and distributed easily thanks to the advancement of computer software, hardware, and communication technologies. With the introduction of HP Memory Spot Chip in 2006 [104], content integrity become more important because digital content can be obtained easily through the conventional channel. A recent scenario of image tampering is the doctored photograph in which smoke condition of a war zone was deliberately thickened [105] to increase the seriousness of a conflict. All these factors have made digital media security a prominent research area.

Cryptographic methods can offer confidentiality and integrity of digital media, and the decrypted content can be further protected using steganographic methods. One such method is digital watermarking. Digital watermarks enable protection mechanism on the decrypted contents so that illegal use, modifications, and distribution of the contents can be prevented or detected. Digital watermarking is an emerging field that attracted the attention of many researchers over the past decades. Digital images make up a major component of digital contents. Therefore, we focus our research on digital image watermarking.
7.2 The Watermarking Problems

The aims of this research are three-fold:

1. To investigate the strength and limitations of current watermarking schemes.
2. To design and develop new schemes to overcome the limitations.
3. To evaluate the new schemes using application scenarios of copyright protection, tamper detection and authentication.

Robust watermarks have been proposed to protect the copyright of the media owner because it can resist media manipulations. However, there exist geometrical distortions that can easily desynchronize watermark information and defeat the purpose of robust watermarking. Focusing on the fundamental geometrical operations of RST, we developed several robust watermarking methods within 2 approaches. The first approach resynchronizes the watermark information to enable successful watermark detection. The second approach utilizes a geometric invariant domain for robust watermarking.

On the other hand, semi-fragile watermarks have been developed to detect malicious tampering while tolerating legitimate modifications. However, image forensics require more than content authentication. Content recovery can provide useful investigative information in modified images. We created a semi-fragile watermark method that offers self-authentication and self-recovery.

Hybrid watermarking methods combine robust watermark and semi-fragile watermarks to offer complementing functionalities. Despite the advantages of hybrid watermarks over single watermarks, very little work on hybrid watermarking can be found. We combined our geometric invariant domain with our semi-fragile watermark into a hybrid watermark method. It can provide an integrated solution in copyright protection, content authentication, tamper localization, and approximate content recovery. The watermark extracted in the detection steps could provide useful investigative information in media forensics. The method is practical given its blind watermark detection capability.
7.3 Summary of Original Contributions

This research resulted in 4 major contributions:

1. The development of robust watermark methods by synchronization
   a. Robust watermarking using differential affine motion estimation
   b. Robust watermarking using flowline curvature and scale normalization

2. The development of a robust watermark method using an invariant domain
   a. Simplified watermark embedding and blind watermark detection
   b. Geometric invariant domain for robust watermarking

3. The creation of a semi-fragile watermark
   a. Semi-fragile watermarking with self-authentication and self-recovery

4. The development of a hybrid watermark method
   a. Multiple watermarking for annotation and tamper detection in medical images
   b. Comparative analysis of overlapped and non-overlapped implementations in a hybrid watermark method
   c. Hybrid watermarking combining robust and semi-fragile watermarks

We grouped robust watermarking methods into 3 main approaches in Chapter 2. They are redundancy-based, synchronization-based and invariant domain watermarking. Redundancy-based watermarks such as the SS method improve the probability of watermark detection by providing excess amount of watermark signal. Synchronization-based watermarks rely on robust features in an image for image registration. They can also use self-embedded templates for image registration. Invariant domains offer robustness against image distortions without the need to synchronize watermark signals. In addition, we categorized image distortions that robust watermarks need to resist into image degradation, image enhancement, image compression, and image transformations. On the other hand, semi-fragile watermarking is identified as a new trend evolving from fragile watermarking. Furthermore, we discussed a rather unexplored usage of semi-fragile watermarking in content recovery. Lastly, we described potential improvements for hybrid watermarking at the end of Chapter 2.
The first original contribution of this research is the development of robust watermark methods that rely on synchronization. In Chapter 3, we explored robust watermarking using differential affine motion estimation for image registration. We adopted the motion estimation method in watermarking, and showed that the multi-resolution processing is able to tackle both global and local distortions. However, we discovered that multi-resolution and iterative processing of this method involved high computational cost. Therefore, in the second contribution, we created a new method using flowline curvature and scale normalization which has low computational cost while offering robustness and imperceptibility. The method reduces computational cost significantly because it works on two pairs of robust corners instead of numerous feature points.

In another part of robust watermarking, we created a new watermark embedding method that reduces computational cost while maintaining watermark robustness and imperceptibility. This is the third contribution, and we named it the Simplified Embedding method. This work is based on a blind watermark detection method that uses the HVS model. The Simplified Embedding method exploits implicit visual masking information in wavelet subbands to reduce computational cost. We also evaluated the performances of our Simplified method with the HVS model and a constant energy embedding method. In the fourth contribution, we constructed a geometric invariant domain using FFT, LPM, and DT-CWT. It is described in Chapter 4. The advantages of DT-CWT exploited are approximate shift invariance, perceptual masking, and multi-resolution sampling. The new invariant domain eliminates synchronization totally, and was showed to be very robust against many types of image distortions. This is in contrast to many invariant domain watermarks which are not truly invariant. They require a small amount of search for synchronization so that the watermarks can be detected. In addition, we also adapted our Simplified Embedding method to reduce computational cost. Using the implicit visual mask of the DT-CWT subbands, we eliminated huge amount of computation in HVS masking. As a result, the novel method executes significantly faster without compromising watermark imperceptibility and robustness. Moreover, the method is highly practical because it enables blind watermark detection.

Another contribution was made in semi-fragile watermarking. In Chapter 5, we created a new method that performs self-embedding, self-authentication, and self-
recovery. The self-authentication and self-recovery features are rather new in semi-fragile watermarking. Our watermarking method embeds a down-scaled version of an image in the wavelet domain using a dynamic quantization approach. Therefore, content authentication and tamper localization can be made without the need of a reference image. Moreover, approximate content recovery can be carried out using the detected watermark on tampered regions. This is advantageous in combating minor cropping.

A pilot study on hybrid watermarking was made. We developed the hybrid method by combining a robust watermark with a fragile watermark. The novel idea in this contribution is the segregation of watermarking regions to achieve high level of watermark imperceptibility without incurring huge computational cost. We did it by embedding the fragile watermark into the central region of an image, and embedding the robust watermark into the border region of an image. In this way, our hybrid watermark does not scarify the robustness of the robust part, and the tampering sensitivity of the fragile part. Additionally, embedding the fragile watermark in the central region ensures good performance in watermark imperceptibility. Evaluation of the method was performed using image annotation and tamper detection scenarios. In another contribution, comparative analysis of two implementation methods for a chosen hybrid watermark was carried out before we developed our own hybrid method. The implementations involve embedding a robust watermark and a fragile watermark in an overlapped and a non-overlapped manner. In this contribution, comparisons were made on watermark robustness, tamper detection effectiveness, and computational cost. Following that, a new hybrid watermark was developed by combining our work on invariant domain and semi-fragile watermarking. The overlapped implementation was used in the hybrid method because it offers better performance in robust watermarking and tamper localization. We also adapted our simplified watermark embedding method in the hybrid watermark to improve watermark imperceptibility and lower computational cost. The hybrid watermark offers integrated content protection by combining robust and semi-fragile watermarks. The hybrid method was evaluated for copyright protection, tamper detection, and content authentication. Our hybrid watermark provides content recovery in addition to copyright protection and content authentication normally
offered by hybrid watermarks. These works on hybrid watermarking were explained in Chapter 6.

Evaluations were made on our watermark methods in their respective Chapters and sections. The evaluations employ scenarios of copyright protection, tamper detection and content authentication.

The original contributions published are presented in Appendix B together with their abstracts. They have been presented in refereed conferences and appeared in the corresponding proceedings.

7.4 Limitations
Despite the good performances of our watermarking methods as explained in the previous section, some limitations exist. Firstly, there is a high computational cost in differential affine motion estimation due to the multi-resolution and iterative processing involved. Secondly, our work on synchronization using flowline curvature and scale normalization was designed to combat image distortions at the global scale. Local image distortions would require different treatment. Our invariant domain also has its shortcomings. For example, DT-CWT has large amount of computation, and LPM produces many interpolation errors. Limitations also exist in the semi-fragile watermarking.

The parameter values of the semi-fragile watermark are determined manually through experimentations. Different images have different features in their contents, and they would produce different parameter values.

Furthermore, overlapping the robust watermark and semi-fragile watermark in our hybrid watermarking method unavoidably affects the robustness of the robust watermark. This is due to the distortions introduced by the semi-fragile watermark.

7.5 Future Work
It is generally agreed that there is no “one size fits all” watermark methods. The performances of watermark methods usually depend on their applications. For example, a robust watermark may has high level of robustness but poor imperceptibility. Similarly, the watermark methods we developed can be improved further.

In Chapter 3, the robust watermarking that rely on two robust corners for image registration was designed for global geometrical distortions. To combat local
geometrical distortions, a cover image can be divided into blocks and be processed separately. The smaller the block size, then the more accurate the image registration result. However, smaller block size will also increases the computational cost because there will be more blocks to work on.

The geometric invariant domain requires huge amount of computation, especially in the LPM and DT-CWT steps. Future work can look into simplifying their computations. In addition, the interpolation errors of LPM and DT-CWT degrade visual quality of the stego images. Compensations of these transformations can be made to improve watermark imperceptibility. For instance, a large mapping space of LPM can improve the stego images’ visual quality.

For the semi-fragile watermarking discussed in Chapter 5, the parameter values are currently determined manually through experiments. They should be determined automatically using a suitable algorithm. For example, the number of quantization bins for the middle range of wavelet coefficients should be computed using the shape of a histogram. More analysis work is needed to automate the computations because every image has its own features.

The combination of the invariant domain and the semi-fragile watermark in our hybrid watermark method also requires more investigation. For example, study on how the semi-fragile watermark affects the robustness of the invariant domain can be carried out, and an efficient non-overlapping implementation can be developed to improve the watermark robustness, tamper localization accuracy, and content recovery effectiveness.

Although we focus on watermarking greyscale images, extensions into color images and video frames are possible because they all have similar data representation. However, the effects on the visual quality and watermark robustness need further investigation.

Overall, watermarking studies require a breakthrough in theoretical research. For instance, the upper limits of watermark capacity and robustness are mostly determined through experiments currently. In addition, the adoption of watermarking technology in commercial applications can be expanded with good supporting infrastructures. For instance, practical business models and legislative controls are necessary in the implementation of DRM frameworks.
### Appendix A

#### Sample Images of Various Attacks

<table>
<thead>
<tr>
<th>Attack</th>
<th>Attack level</th>
<th>Authentic</th>
<th>Attacked image</th>
</tr>
</thead>
<tbody>
<tr>
<td>No attack</td>
<td>Yes</td>
<td></td>
<td><img src="image1" alt="Attacked image" /></td>
</tr>
<tr>
<td>Local shift</td>
<td>Yes</td>
<td></td>
<td><img src="image2" alt="Attacked image" /></td>
</tr>
<tr>
<td>Attack</td>
<td>Attack level</td>
<td>Authentic</td>
<td>Attacked image</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------</td>
<td>-----------</td>
<td>----------------</td>
</tr>
<tr>
<td>Histogram equalisation</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>0.0005</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>0.005</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>JPEG compression</td>
<td>90</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>JPEG compression</td>
<td>80</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Attack</td>
<td>Attack level</td>
<td>Authentic</td>
<td>Attacked image</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------</td>
<td>-----------</td>
<td>----------------</td>
</tr>
<tr>
<td>JPEG compression</td>
<td>70</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Shifting</td>
<td>[1 0]</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Shifting</td>
<td>[3 0]</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Shifting</td>
<td>[2 2]</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Rotation</td>
<td>1° and crop</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Attack</td>
<td>Attack level</td>
<td>Authentic</td>
<td>Attacked image</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>-----------</td>
<td>----------------</td>
</tr>
<tr>
<td>Rotation</td>
<td>2° and crop</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Rotation</td>
<td>4° and crop</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Cropping</td>
<td>(1:90, 420:512)</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Mean filtering</td>
<td>2x2</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Mean filtering</td>
<td>3x3</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Attack</td>
<td>Attack level</td>
<td>Authentic</td>
<td>Attacked image</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------</td>
<td>-----------</td>
<td>----------------</td>
</tr>
<tr>
<td>Mean filtering</td>
<td>4x4</td>
<td>No</td>
<td><img src="image" alt="Attacked image" /></td>
</tr>
<tr>
<td>Mean filtering</td>
<td>5x5</td>
<td>No</td>
<td><img src="image" alt="Attacked image" /></td>
</tr>
</tbody>
</table>
B.1 Recovery of Watermark Using Differential Affine Motion Estimation

Abstract
Digital watermarking techniques have been proposed to protect the copyright of multimedia data. Robustness against geometric distortion is one of the most important issues to be solved to increase the robustness of digital image watermarking systems. Such attacks are very simple to implement, so they can defeat most existing watermarking algorithms without causing serious perceptual distortion. In this paper, a method for the recovery of watermarks based on differential affine motion estimation is presented. This method models the geometric distortion between images as locally affine but globally smooth. This approach is built upon a differential multi-scale framework, allowing us to capture both large-scale and small-scale transformations. Experimental results show that the described method can estimate the distortions quite accurately and allow correct watermark detection.
B.2 Geometrically Robust Digital Image Watermarking using Scale Normalization and Flowline Curvature

Abstract
The growth of internet communications, multimedia storage capacity, and software sophistication triggered the need to protect intellectual property in digital media. Digital watermark can be inserted into images for copyright protection, copy protection, tamper detection and authentication. Unfortunately, geometrical robustness in digital image watermarking remains a challenging issue because consumer software enables rotational, scaling and translational attacks on the watermark with little image quality degradation. To balance robustness requirements and computation simplicity, we propose a method to re-synchronize watermark information for its effective detection. The method uses scale normalization and flowline curvature in embedding and detection processes. Scale normalization with unit aspect ratio and predefined area offers scale invariance and translation invariance. Rotational robustness is achieved using the flowline curvature properties of extracted robust corners. The watermark is embedded in Discrete Fourier Transform (DFT) domain of the normalized image using fixed strength additive embedding. Geometric properties recovery is simplified using flowline curvature properties and robust corners as reference points prior to watermark detection. Despite the non-blind nature and vulnerability to local transformations of this approach, experimental results indicate its potential application in robust image watermarking.

B.3 Performance Factors Analysis of a Wavelet-based Watermarking Method

Abstract
The essential performance metrics of a robust watermark include robustness, imperceptibility, watermark capacity and security. In addition, computational cost is important for practicality. Wavelet-based image watermarking methods exploit the frequency information and spatial information of the transformed data in multiple resolutions to gain robustness. Although the Human Visual System (HVS) model offers imperceptibility in wavelet-based watermarking, it suffers high computational
cost. In this paper, we examine embedding strength determined by a HVS model, a constant, and a simplified technique. The proposed simplified embedding technique significantly reduces embedding time while preserving the performance of imperceptibility and robustness. The fast embedding technique exploits implicit features of discrete wavelet transform (DWT) sub-bands, i.e. the luminosity information in the low pass band, and the edge information in the high pass bands. It achieves embedding speed comparable to a constant energy embedding process. Robustness is demonstrated with a few conventional attacks, e.g. JPEG compression, Gaussian noise insertion, image cropping, contrast adjustment, median filtering, and global geometrical distortion. Experimental visual quality is measured in Weighted-Peak Signal to Noise Ratio (W-PSNR) for high accuracy. Robustness and imperceptibility of HVS-based embedding could be trade-off with computational simplicity of a fast embedding technique.

B.4 Geometric Invariant Domain for Image Watermarking

Abstract
To enable copyright protection and authentication, robust digital watermark can be embedded into multimedia contents imperceptibly. However, geometric distortions pose a significant threat to robust image watermarking because it can desynchronize the watermark information while preserving the visual quality. To overcome this, we developed an invariant domain with three transforms; Fast Fourier Transform (FFT), Log-Polar Mapping (LPM), and Dual Tree-Complex Wavelet Transform (DT-CWT). Shift invariance is obtained using FFT. Rotation and scaling invariance are achieved by taking the DT-CWT of a LPM output. Unlike most invariant schemes, our method eliminates explicit re-synchronization. The method resists geometric distortions at both global and local scales. It is also robust against JPEG compression and common image processing. In addition, it exploits perceptual masking property of the DT-CWT sub-bands, and its watermark detection step does not require the cover image. Experiment on a large set of natural images shows the robustness of the new scheme.
B.5 Multiple Watermark Method for Privacy Control and Tamper Detection in Medical Images

Abstract
Medical images in digital form must be stored in a secure way to preserve stringent image quality standards and prevent unauthorised disclosure of patient data. This paper proposes a multiple watermarking method to serve these purposes. A multiple watermark consists of an annotation part and a fragile part. Encrypted patient data can be embedded in an annotation watermark, and tampering can be detected using a fragile watermark. The embedded patient data not only save storage space, it also offers privacy and security. We also evaluate the images’ visual quality after watermark embedding and the effectiveness of locating tampered regions.

B.6 System Architecture Analysis of a Hybrid Watermarking Method

Abstract
A hybrid watermark that consists of a robust part and a fragile part can be used to serve multiple purposes. The robust part can protect copyright information, the fragile part can detect tampering, and their combination enables identification of attacks encountered. This paper analyses an overlap and a non-overlap implementation of the robust and fragile parts in a hybrid scheme. The difference between the two implementation methods lies in the robust and fragile watermarks embedding positions. Embedding capacity, computational costs, watermark robustness, and tamper detection localization of the two implementations are analyzed. In addition, optimization issues of block size in the hybrid scheme are discussed.
Glossary

- **Attacks** on watermarks are manipulations that aimed at destroying the watermark and therefore defeat their purposes. Attacks could refer to the robustness or security aspects of a watermarking method.

- **Authentication** in image watermarking is the integrity assurance of an image.

- **Blind watermark detection** is a watermark detection which does not require a reference image.

- **Capacity** is the amount of watermark information in an image. If multiple watermarks are embedded into an image, then the watermarking capacity of the image is the sum of all individual watermark’s data payload.

- **Computational cost** is the measure of computing resources required to perform watermark embedding or detection processes.

- **Constant energy embedding** is the application of equal embedding strength in every coefficient during watermarking.

- **Content authentication (soft authentication)** is the authentication of image contents in which legitimate changes are differentiated from illegitimate modifications.

- **Content recovery** is the restoration of the original image features onto a corrupted image.
• **Copyright protection** is the mechanism to protect the rights of a copyright holder by preventing illegal duplication of the protected work.

• **Cover image** is the original image used in watermarking.

• **Data payload** is the encoded message size of a watermark in an image.

• **Digital Rights Management (DRM)** is the “description, identification, trading, protecting, monitoring, and tracking of all forms of usages over tangible and intangible assets [23]”.

• **Distortions** are changes made to a stego image to evaluate its robustness.

• **Fragile watermarks** are easily destroyed by image distortions.

• **Human Visual System (HVS)** model is a description that mimics the sensitivity of the human eyes to image characteristics.

• **Hybrid watermarks** consist of robust and fragile/semi-fragile watermarks.

• **Image registration** is used to synchronize watermark so that it can be detected. The registration process maps each object’s location in a distorted image to its corresponding object’s location the reference image during synchronization.

• **Imperceptibility** is the characteristic of hiding a watermark so that it does not degrade the visual quality of an image.

• **Invariant domain** is a set of coefficients which are not affected by distortions.

• **Media forensics** involves the investigation of digital data in order to unveil scientifically valid information for court evidence.

• **Motion estimation** is the estimation of the direction and distance of object movements between two images.

• **Privacy control** is the regulation of concealing secret from intrusion.

• **Reference image** is the image used to assist watermark detection. It could be a cover image, a stego image, or a test image.

• **Robust watermarks** are watermarks that can resist non-malicious distortions.

• **Robustness** of a watermark refers to its ability to withstand non-malicious distortions.
• **Security** of a watermark is the ability of the watermark to resist malicious attacks.

• **Self-authentication** is the authentication of an image’s content without resorting to any reference images.

• **Self-recovery** is the restoration of image contents without resorting to any reference images.

• **Semi-fragile** watermarks can be destroyed by certain types of distortions. Thus, they can differentiate between legitimate changes and illegitimate modifications.

• **Stego image** is the cover image following watermark embedding.

• **Synchronization** is the process of registering a distorted image using a reference image in order to align the watermark information for successful watermark detection.

• **Tamper detection** is the disclosure of alterations made onto an image.

• **Tamper localization** is the identification of tampered regions within the altered image.

• **Test image** is the possibly modified stego image from which the watermark is to be extracted.

• **Watermark** can be a simple signal consists of a pseudo-random binary sequence, or a multi-bit message encoded in a transform domain.

• **Watermark detection** is the process of uncovering a watermark hidden in an image.

• **Watermark embedding** is the process of encoding a watermark signal into an image.

• **Watermark scheme** comprises the embedding and detection methods.
Bibliography


191


46. JPEG Image Compression, 2006,online at http://www.vectorsite.net/ttdcmp2.html, last accessed 27 November 2006


192


193


