DIGITAL WATERMARKING FOR IMAGE AUTHENTICATION AND RECOVERY

RAFIULLAH

Department of Computer and Information Sciences
Pakistan Institute of Engineering and Applied Sciences,
Islamabad Pakistan
In the name of Allah, the most Beneficent, the most Merciful
Digital Watermarking for Image Authentication and Recovery

By
RAFIULLAH

A dissertation submitted to the
Pakistan Institute of Engineering and Applied Sciences
in the partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer and Information Sciences

Department of Computer and Information Sciences,
Pakistan Institute of Engineering and Applied Sciences,
Islamabad, Pakistan
August 2010
This thesis is prepared under the supervision of

Dr. Asifullah Khan
Associate Professor
of the Department of Computer and Information Sciences,
Pakistan Institute of Engineering and Applied Sciences,
Islamabad Pakistan

Financial support by Higher Education Commission Pakistan through indigenous-5000 PhD program,
Grant No. 17-5-I (Cu-180) HEC/Sch/2004/4343).

Digital Watermarking for Image Authentication and Recovery
Declaration

I declare that all material in this thesis, which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other university.

Signature: __________________
Author’s Name: Rafiullah

It is certified that the work in this thesis is carried out and completed under my supervision.

Supervisor:
**Dr. Asifullah Khan**
PS (DCIS), PIEAS, Islamabad.

Co-Supervisor:
**Dr. Abdul Majid**
PS (DCIS), PIEAS, Islamabad.
To my loving parents
and my wife whose image needs no enhancement
Acknowledgments

First, I would like to thank my Allah Almighty, the most Gracious, and the most Merciful. With His blessing, the completion of my PhD thesis becomes possible. He provided me sufficient opportunities and enabled me to execute my PhD research work.

I would like to thank my parents who care, support, and advice me a lot all the time. The destination was unreachable for me without the prayers, love and moral support of my parent, wife, and siblings. I wish to extend my appreciation to my supervisor Dr. Asifullah Khan for his tireless support, priceless intellectual motivation. He allows me freedom and always support and contribute ideas during my PhD research work.

A lot of credit goes to Higher Education Commission Pakistan for offering me MS leading to PhD scholarship and Pakistan Institute of Engineering and Applied Sciences (PIEAS), where a good environment has been provided for this research work. Many thanks to Dr. Syed Asif Mehmood Gilani, definitely who put me on the path of digital watermarking, and to the Ghulam Ishaq Khan Institute of Engineering Sciences and Technology.

The work may not have been possible without the help and support of many other individuals. Thanks to Dr. Chang Tsun Li (Professor in University of Warwick UK), Dr. Abdul Majid, Dr. Abdul Jalil, Dr. Mutawarra Hussain, Dr. Anila Usman, and Dr. Asmatullah Chaudhry for their kind support and priceless help during my course and research work. Special thanks to Imran Usman and Adnan Idris for their valuable support and encouragement at different stages. Finally yet importantly, thanks to Saeed Ehsan Awan, Asif Mahmood, Manzoor Ahmad, Muhammad Naeem, and all other friends for their encouragement and help in difficult phases of my research work.

Rafiullah, DCIS, PIEAS
List of Publications


<table>
<thead>
<tr>
<th>Acronym</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{W}_1, \hat{W}'_1$</td>
<td>First watermark (authentication watermark), Retrieved first watermark</td>
</tr>
<tr>
<td>$\tilde{W}_2, \tilde{W}'_2$</td>
<td>Second watermark (recovery watermark), Retrieved second watermark</td>
</tr>
<tr>
<td>$g, G$</td>
<td>Group of wavelet Coefficients, Number of groups</td>
</tr>
<tr>
<td>$P_{Rand}$</td>
<td>Pseudo-Random Number Matrix based on a secret key</td>
</tr>
<tr>
<td>$q, q_{Scaled}$</td>
<td>Quantized DCT coefficients and their corresponding scaled coefficients</td>
</tr>
<tr>
<td>$V, V_{permuted}$</td>
<td>Quadruplicating $q_{Scaled}$ vector and scrambled based on secret key</td>
</tr>
<tr>
<td>$LFB(a), LFB(a,b)$</td>
<td>Represents the least five least significant bits of $a$, the substitution of $b$ for five least significant bits of $a$</td>
</tr>
<tr>
<td>$LFB_{th}(a)$</td>
<td>$i^{th}$ least significant bit of $a$</td>
</tr>
<tr>
<td>$D$</td>
<td>Difference in original and extracted authentication watermarks</td>
</tr>
<tr>
<td>$g_s$</td>
<td>Group size (Number of coefficients in a group)</td>
</tr>
<tr>
<td>$\tilde{g}_j$</td>
<td>Weighted Mean of $j$th group</td>
</tr>
<tr>
<td>$\text{Quan}(\tilde{g}_j)$</td>
<td>Quantization of the weighted mean of $j$th group</td>
</tr>
<tr>
<td>$p$</td>
<td>The bipolar random sequence generated by a secret key with uniform distribution $p \in {1, -1}$</td>
</tr>
<tr>
<td>$\text{Quanta (Q)}$</td>
<td>This is the quantization step being used, while embedding the first/authentication watermark ($W_1$)</td>
</tr>
<tr>
<td>$Q_p$</td>
<td>The quantization parameter used in correlating $W_1$ with image coefficients chosen from the LL1 subband. It helps in determining the sensitivity of collage attack</td>
</tr>
<tr>
<td>$\text{Dense}$</td>
<td>Error pixels that determine the malicious attack. If the image is attacked maliciously, then number of dense pixels increases.</td>
</tr>
<tr>
<td>$\text{Sparse}$</td>
<td>Error pixels that determine the incidental attacks like JPEG lossy compression. If the image is attacked incidentally, then the number of sparse pixels increases.</td>
</tr>
<tr>
<td>$\text{Residue}$</td>
<td>The quantity remaining after processing/quantizing the coefficient</td>
</tr>
<tr>
<td>$\text{sign}$</td>
<td>$\text{sign}(x) = 1$ if $x \geq 0$ and $\text{sign}(x) = 0$ if $x &lt; 0$</td>
</tr>
<tr>
<td>$\text{Filtering}$</td>
<td>Removal of noise dots i.e. the noise-like unverified coefficients. Filtering out the isolated unverified coefficients i.e. those coefficients which are not tampered but they belong to the unverified group(s)</td>
</tr>
<tr>
<td>$\Delta_j$</td>
<td>The remainder after dividing the weighted mean of a $j^{th}$ group by $\text{quanta}$</td>
</tr>
<tr>
<td>$\partial$</td>
<td>Difference in expected and original weighted mean of group, while modifying the coefficients of the group</td>
</tr>
<tr>
<td>$\text{idct2}$, $huffmanenco$</td>
<td>Integer DCT applied on approximation of image</td>
</tr>
<tr>
<td>$huffmandeco$</td>
<td>Huffman encoding</td>
</tr>
<tr>
<td>$bchenco$</td>
<td>BCH encoding</td>
</tr>
<tr>
<td>$bchdeco$</td>
<td>BCH decoding</td>
</tr>
</tbody>
</table>
Table of Contents

List of Publications ................................................................................................................. viii
Acronyms ................................................................................................................................. x
Table of Contents ...................................................................................................................... xi
List of Figures ............................................................................................................................. xiv
List of Tables and Algorithms ................................................................................................. xvi

Chapter 1
Semi-Fragile Watermarking for Image Authentication and Recovery....................................... 1
  1.1 Motivation and Objectives .............................................................................................. 2
  1.2 Research Perspective .................................................................................................... 3
  1.3 Contribution ................................................................................................................ 3
  1.4 Thesis Structure .......................................................................................................... 4

Chapter 2
Digital Watermarking Techniques: Literature Review ............................................................. 6
  2.1 Applications of Watermarking ...................................................................................... 7
    2.1.1 Broadcast Monitoring ............................................................................................ 7
    2.1.2 Ownership Assertion ............................................................................................. 7
    2.1.3 Transaction Tracking .............................................................................................. 8
    2.1.4 Content Authentication ........................................................................................ 8
    2.1.5 Copy Control and Fingerprinting ......................................................................... 9
  2.2 Properties of Watermarking ......................................................................................... 10
    2.2.1 Embedding Effectiveness ....................................................................................... 10
    2.2.2 Perceptual Similarity ............................................................................................. 10
    2.2.3 Robustness ........................................................................................................... 11
    2.2.4 Data Payload ........................................................................................................ 11
    2.2.5 Blind and Informed Detection .............................................................................. 11
  2.3 Watermarking Techniques ............................................................................................ 11
    2.3.1 Embedding Locations ........................................................................................... 12
    2.3.2 Robust Watermarking .......................................................................................... 13
    2.3.3 Fragile Watermarking .......................................................................................... 14
    2.3.4 Semi-Fragile Watermarking ................................................................................ 16
    2.3.5 Configuration of Cover Work by Watermark Embedding .................................... 18
    2.3.6 Watermarking with Recovery .............................................................................. 19
    2.3.7 Watermarking Domains ....................................................................................... 20
  2.4 Attacks and their Countermeasures ................................................................................ 21
  2.5 Capacity vs. Robustness vs. Imperceptibility ............................................................... 22
  2.6 JPEG Compression ........................................................................................................ 22
  2.7 Integer Wavelet Transform ............................................................................................ 23
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.8</td>
<td>Huffman Coding</td>
<td>25</td>
</tr>
<tr>
<td>2.9</td>
<td>Integer Discrete Cosine Transform</td>
<td>26</td>
</tr>
<tr>
<td>2.10</td>
<td>BCH Coding</td>
<td>27</td>
</tr>
<tr>
<td>2.11</td>
<td>Chapter Summary</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>Authentication and Recovery of Images using Wavelet Transform</td>
<td>30</td>
</tr>
<tr>
<td>3.1</td>
<td>Proposed Algorithm</td>
<td>32</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Generation of Authentication Watermark</td>
<td>32</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Generation of Recovery Watermark</td>
<td>33</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Embedding of Authentication Watermark</td>
<td>35</td>
</tr>
<tr>
<td>3.1.4</td>
<td>Embedding of Recovery Watermark</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>Integrity Verification</td>
<td>39</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Extraction of Authentication Watermark</td>
<td>39</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Extraction of Recovery Watermark</td>
<td>40</td>
</tr>
<tr>
<td>3.3</td>
<td>Tamper Detection</td>
<td>43</td>
</tr>
<tr>
<td>3.4</td>
<td>Recovery of Approximated Image</td>
<td>44</td>
</tr>
<tr>
<td>3.5</td>
<td>Experimental Results</td>
<td>47</td>
</tr>
<tr>
<td>3.6</td>
<td>Chapter Summary</td>
<td>54</td>
</tr>
<tr>
<td>4</td>
<td>Secure Image Authentication and Recovery using Multiple Watermarks</td>
<td>55</td>
</tr>
<tr>
<td>4.1</td>
<td>The Proposed Algorithm</td>
<td>56</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Generation and Embedding of Authentication Watermark</td>
<td>56</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Generation and Embedding of Recovery Watermark</td>
<td>60</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Watermarks Extraction and Integrity Verification</td>
<td>62</td>
</tr>
<tr>
<td>4.2</td>
<td>Image Recovery</td>
<td>65</td>
</tr>
<tr>
<td>4.3</td>
<td>Experimental Results</td>
<td>65</td>
</tr>
<tr>
<td>4.4</td>
<td>Analysis</td>
<td>70</td>
</tr>
<tr>
<td>4.5</td>
<td>Chapter Summary</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>Image Authenticity with Exact Recovery</td>
<td>71</td>
</tr>
<tr>
<td>5.1</td>
<td>The Proposed Algorithm</td>
<td>72</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Generation of Authentication Watermark</td>
<td>73</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Generation of Recovery Watermark</td>
<td>73</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Embedding of Authentication and Recovery Watermarks</td>
<td>75</td>
</tr>
<tr>
<td>5.1.4</td>
<td>Extraction of Authentication Watermark and Integrity Verification</td>
<td>76</td>
</tr>
<tr>
<td>5.2</td>
<td>Image Recovery</td>
<td>76</td>
</tr>
<tr>
<td>5.3</td>
<td>Experimental Results</td>
<td>82</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Analysis of Authentication Watermark</td>
<td>82</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Effect of JPEG Compression</td>
<td>83</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Malicious Tampering and its Localization</td>
<td>88</td>
</tr>
</tbody>
</table>
List of Figures

Figure 2.1 Three common contradicting parameters of watermarking system: watermark strength, imperceptibility, and robustness .................................................. 22
Figure 3.1 Block diagram for generation and embedding the watermarks ............................. 37
Figure 3.2 Distance diagram ............................................................................................. 37
Figure 3.3 Block diagram of watermarks extraction .......................................................... 42
Figure 3.4 Analysis of the extracted authentication watermark \( W_1 \) .................................. 42
Figure 3.5 Analysis of extracted recovery watermark \( W_2 \) .................................................. 46
Figure 3.6 (a) Original Image (b) Authentication watermark (c) Watermarked image ............ 48
Figure 3.7 (a) Original image (b) Watermarked image (c) Extracted authentication watermark (d) Difference image. Difference is taken between the original and extracted authentication watermarks .......................................................... 49
Figure 3.8 (a) Watermarked image tampered maliciously (b) Recovered recovery watermark (c) Extracted authentication watermark where the tampered regions are detected (d) The difference image locations of tampered regions ................................. 49
Figure 3.9 (a) Original image of cameraman (b) The watermarked image, PSNR 38.2db (c) Recovered recovery watermark (d) Extracted authentication watermark (e) Difference between the original and extracted authentication watermarks .......................... 50
Figure 3.10 (a) Watermarked image which has been manipulated maliciously on feathers (b) recovered recovery watermark which has been degraded a little bit (c) Difference between the original and extracted authentication watermark where the tampered regions are accurately detected and localized .............................................................. 50
Figure 3.11 The first row shows the recovered images. These images degrade according to the JPEG compression quality factor. The second row shows the differences after compressing the watermarked image with quality factors 90%, 80%, 75%, 70% and 60% respectively ................................................................. 51
Figure 3.12 (a) Watermarked image (b) Extracted authentication watermark by using a wrong key ................................................................................................. 52
Figure 3.13 Error pixels versus strength of JPEG compression (a) Image of size 512×512 (b) Image of size 1024×1024 ......................................................................................... 52
Figure 4.1 Embedding process ......................................................................................... 58
Figure 4.2 Selection of coefficient with high magnitude for embedding the watermark bit. The suitable coefficient in the group is indicated. Group size, \( gs = 35 \) .......................... 60
Figure 4.3 Illustration of authentication process ............................................................... 63
Figure 4.4 (a) Original image (b) Watermarked image (c) Recovered image approximation (d) Difference between original and extracted authentication watermarks ......................... 66
Figure 4.5 (a) Original Lichtenstein image (b) Tip of minaret has been removed (c) Detection of the tampered region (d) Recovered image approximation with a tiny degradation because the watermarked image has been manipulated ......................... 67
Figure 4.6 (a) Original London eye image (b) Object moving (airplane) from its original position (c) Detection of location where the object is added as well as from where
the object is removed (d) Recovered image approximation which has been degraded
a little bit because the watermarked image has been altered.................................67
Figure 4.7 PSNR against different Quanta values and group sizes on three different
images, Lena, Cameraman, and Baboon images. The PSNR varies according to
the values of both parameters.........................................................................................69
Figure 5.1 Watermarks generation and embedding diagram ............................................75
Figure 5.2 Watermarks extraction diagram........................................................................79
Figure 5.3 Test images (512 × 512): (a) Lena (b) Baboon (c) Airplane (d) Cameraman (e)
Boat (f) Couple (g) Barbara (h) London eye (i) Lichtenstein (j) Pepper .........................82
Figure 5.4 PSNR with different Quanta and Group Size (From top line: g = 30, 20, 10,
5). Lena image is used as test image .............................................................................83
Figure 5.5 Original image, (b ~ j) Watermarked images with different Quanta
and Group Size, 5 − 5, 5 − 10, 10 − 10, 10 − 30, 20 − 5, 20 − 10, 20 − 30 −
5, 30 − 20 .......................................................................................................................84
Figure 5.6(a) JPEG compressed watermarked images, (b) Recovered approximation
images, (c) Difference images, the difference have been taken between the extracted
and the original authentication watermarks. ................................................................85
Figure 5.7 Number of erroneous pixels (Dense and Sparse pixels) versus JPEG
quantization factors. Lena image is used as test image .............................................86
Figure 5.8 Erroneous pixels,
(a ~ c, Lena Image), QF = 90%, 80%, 70%, (d ~ f, Baboon Image), QF =
90%, 80%, 70%................................................................................................................86
Figure 5.9 PSNR versus Quanta, with different Group Size on three different images
i.e. Cameraman, Lena and Baboon .................................................................................87
Figure 5.10 BER against different JPEG quantization factors for Liu et al [88], Kundur et
al [16] and proposed approaches......................................................................................88
Figure 5.11(a) Tampered watermarked images of Lena and Cameraman, (b) Tampered
regions are determined and localized obviously by Proposed approach, (c) Proposed
by Li et al [30] ..................................................................................................................91
Figure 5.12 (a) London Eye image, (b) Object moving (airplane) i.e. the airplane is
removed from its original position and added nearby, (c) Detection of tampered
regions i.e. both the locations have been detected, (d) Recovered image .......................91
Figure 5.13 (a) Original Lichtenstein Image, (b) Tip of minaret has been removed, (c)
Detection and localization of tampered regions, (d) Recovery of image
approximation ....................................................................................................................92
Figure 5.14 (a) Attacked image (counterfeiting/Collage Attack), (b) The retrieved
watermark which shows that the collage attack in not undetectable..............................92
Figure 5.15 PSNR of 10 test images. Comparison of proposed approach (only W2 and both
W1 and W2 ) with [35]. .................................................................................................93
List of Tables and Algorithms

Table 2.1 The valid wavelet name values [97] .................................................................25
Table 2.2 Generators of primitive BCH codes................................................................28
Table 3.1 PSNR of extracted recovery watermark after JPEG compression of the
watermarked image, with respect to the recovery watermark extracted from a non-
compressed watermarked image. PSNR values are given with the corresponding
JPEG quality factors..................................................................................................47
Table 3.2 Salient features of the proposed algorithm with respect to authentication,
recovery, robustness, and security, against the previous approaches, Piva et al [35]
and Wu et al [36] ......................................................................................................53
Table 4.1 The weighted mean and the expected weighted mean of 10 different groups ....59
Table 4.2 PSNR (dB) against different quanta and group size. ........................................68
Table 4.3 Prominent features and performance comparison of the proposed approach with
previous approaches, Chamlawi et al [33] and Piva et al [35].................................69
Table 5.1 Behavior of the proposed approach by using different BCH codes..............74
Table 5.2 Number of dense and sparse pixels using different frequency content based
images. Corresponding JPEG quality factor (QF) is given.......................................89
Table 5.3 Prominent features and performance comparison with previous approaches....94

Algorithm 3.1 Generation and embedding of $W_1$ and $W_2$.........................................38
Algorithm 3.2 Extraction of authentication and recovery watermarks..........................45
Algorithm 4.1 Generation and embedding procedure of the watermarks $W_1$ and $W_2$ ....61
Algorithm 4.2 Extraction procedure of watermarks $W_1$ and $W_2$..................................64
Algorithm 5.1 Generation and Embedding of $W_1$ and $W_2$........................................78
Algorithm 5.2 Extraction of $W_1$ and $W_2$ ..................................................................80
Algorithm 5.3 Authentication and localization of the proposed watermarking technique .81
Revolution in generation, storage, and communication of digital information has brought about profound changes in our society. The digital information age has evolved with numerous opportunities and new challenges. The goal of this thesis is to provide a framework based on watermarking techniques that can be used for verifying the integrity of a cover work along with the recovery of the intentionally/unintentionally distorted cover work. In this context, multiple semi-fragile watermarking techniques are proposed for not only protecting the digital content from alteration but also to recover it after alteration. In these techniques, watermarks remain intact with the image under minor enhancement and is broken only in case of major manipulations. In this thesis, a hybrid scheme is developed that can both authenticate and recover the altered image. The first phase of this thesis encompasses an improvement of a self-recovery authentication scheme for digital images. The second phase then considers the reduction in computational complexity. Finally, a novel model is proposed in the third phase that not only perform accurate authentication of images but also recovers the altered image.

In this thesis, the concept of multiple watermarking is employed; authentication and recovery watermarks. Both of these independent watermarks strengthen the security aspect of each other and it is user choice to use both of the watermarks or one of them according to the requirement of application. The authentication watermark is correlated to the host image for resisting collage attack and then embedded in the wavelet subbands. Unlike the conventional block-based approaches, it has the ability to determine the regions concisely where the integrity verification fails. The recovery watermark recovers the image with original quality even after manipulation of the watermarked image. Lossless compression (Huffman coding) and BCH (Bose, Ray-Chaudhuri, and Hocquenghem) coding are utilized while generating the recovery watermark. Integer DCT is utilized instead of conventional DCT because the integer DCT contents can be highly compressed by Huffman coding. In addition, integer wavelet transform, which is a fast approach of discrete wavelet transform, also reduces the computational complexity of the proposed algorithm.
In contrast to the earlier authentication algorithms, the proposed techniques exploit flexibility in both of the watermarks, where a trade-off can be made by the user according to the requirement of application. Experimental investigations are performed to evaluate the performance of multiple semi-fragile watermarks. It is demonstrated that the performance of the proposed methods is better compared to the conventional block-based approaches in context of tamper detection.

In summary, efficient techniques have been developed in this thesis which makes trade-off between three contradicting properties of watermarking; imperceptibility, robustness, and capacity. The proposed watermarking techniques are able to answer these questions, i) Has the image been processed? ii) Has the image been processed incidentally or maliciously? iii) Which part of the image has been processed and how much? Additionally, a self-recovery approach makes it possible to recover the exact version of the host image even after the image has been incidentally/maliciously processed.
Chapter 1

Semi-Fragile Watermarking for Image Authentication and Recovery

Information is inevitably physical. Rolf W. Landauer (1927–1999)

**Image/Picture:** One picture is worth more than ten thousand words.

- **Actual or mental picture:** A picture or likeness of somebody or something, produced by a sculptor, painter, or photographer, or conjured in the mind.
- **Somebody closely resembling somebody else:** Somebody who, or something that is very like to somebody or something else in appearance. She is the image of her grandmother.
- **Optics likeness seen or produced:** The likeness of somebody or something that appears in a mirror, through a lens, or on the retina of the eye, or that is produced electronically on a screen.
- **Literature:** Example of figurative language; a figure of speech.
- **Mathematics:** Set of function’s values. The value of a mathematical function corresponding to a specific value of the function’s variable.

However, the image to be discussed in this thesis is:

- Representation of anything else. A stored description of a graphic picture, either a set of brightness and color values of pixels or a set of instructions for reproducing the picture [1].

Now a day, we live in the digital world and the protection of digital content is essential, especially for owners and distributors. Digital watermarking is an information hiding technique, where unlike steganography; the message is related to the cover work. In its initial stage, paper watermark appeared in 1282 in Italy when thin wires were added to the paper mold [2]. However, the commercial use of watermark started in 18th century in Europe and America. The use of watermark then extended beyond the scope of paper and was used in other physical objects (fabrics, garment labels, product packaging etc) and electronic signals (music, photograph, video etc) [3, 4]. Digital watermarking got an extensive popularity in the latter half of 1990s [5].

Initially, informed techniques of watermarking have been used where the original work or any information about the original work is required at the receiving side. In 1997, Piva *et al* [6] have introduced the blind watermarking technique, where the
original or any information about the original work is not required. Now-a-days, most
of the researchers are using blind watermarking approaches [5, 7].

Watermarking can be divided into two broad categories; robust and fragile
watermarking. The robust watermark has the ability to resist the intentional or
unintentional manipulations whereas the fragile watermark is intended to be destroyed
even after unintentional manipulation. The main application of fragile watermarking
is content authentication. A third category is semi-fragile watermarking which is also
employed in this work. Semi-fragile watermarking has the ability to resist the
unintentional manipulations like JPEG compression and is fragile against the
intentional manipulations. Significant research work related to robust [8-15], fragile
[16-23] and semi-fragile watermarking [24-28] techniques has been reported.

This thesis focuses on the authentication and recovery of images by using dual but
independent semi-fragile watermarks. The architectures of the proposed watermarking
systems used in this thesis are such that one can use both or any of the watermarks
based on the application requirement.

The approaches discussed in this thesis avoid the traditional block-based concept
[29-32] for detecting the tampered regions. In block-based approaches, the resolution
of tamper detection is limited to block size. To improve the detection resolution,
block-based approaches require smaller block size and high watermark payload,
which considerably degrades the image quality. Recently, security consideration is
designated as one of the most important aspect of watermarking system [33].
Therefore, the challenge is how to increase the resolution of tamper detection by
embedding less watermark information.

1.1 Motivation and Objectives

We cannot rely on the proverb “Seeing is Believing”, due to the powerful tools
available that can easily manipulate the digital content without leaving any trace. This
issue has number of implications for content security and integrity. For example, if
someone copy the content, then it is impossible to distinguish the copy from the
original content. Such developments in computer resources have also increased the
opportunities for emerging technologies to protect the digital content (image, video,
audio, printed document, piece of art, graphics etc). These emerging techniques are
designed to make the digital content secure. This research addresses the protection of digital image, and is able to detect manipulations and recover the original content if the integrity verification of the image fails. In this regard, multiple semi-fragile watermarks are embedded in proper locations of the image to protect its integrity. While embedding the multiple watermarks, a trade-off has been made between the contradicting properties of a watermarking system such as imperceptibility, robustness, and capacity. In addition, a self-recovering approach is utilized to recover back the original content even if the integrity of the image fails.

1.2 Research Perspective

Advances in computer hardware and software have created threats to the multimedia content (image, audio, and video) integrity. For example with these advancements, images can easily be manipulated and shared. Digital watermarking is a prospective tool for securing the integrity of digital contents. Watermarking is the process of embedding the information into digital content without degrading its visual significance. The watermarked work can be made public or sent through some communication channel. Later on, the detected watermark can be used for authentication, copyright protection etc.

This research endeavors on developing a watermarking system that not only verifies the integrity of the digital images but also recovers the approximated version of the original image on receiving side. This research investigates the limitations of the existing semi-fragile watermarking approaches and strives for designing and developing new schemes to circumvent these limitations. The approaches used in this thesis are designed to protect the digital images and an effort is put to achieve the following properties of watermarking: embedding effectiveness, fidelity, watermark strength, robustness, security, and cost.

1.3 Contribution

The security of digital data is essential in the today’s revolutionary computing world. Many researchers have developed the interesting algorithms for content protection but they either have a security gap or lack some of the watermarking properties. This thesis focuses on the authenticity and recovery of digital images with substantial
consideration of the different watermarking properties. The following contributions have been made towards image authentication and recovery:

Tamper detection localization is improved and unlike conventional block-based approaches, the exact tampered region is detected. The tampering behavior is analyzed and it can be determined that whether image is tampered intentionally or unintentionally by utilizing some parameters.

Exact recovery of the image has been made possible, while keeping low watermark payload. Exploitation of lossless compression technique is performed to reduce payload. An error correction code is utilized for correcting the erroneous bits, if the watermarked image is manipulated either incidentally or maliciously.

The developed techniques have user defined resistivity criteria against JPEG compression. In the embedding stage, some parameters are defined to make the system robust against JPEG compression according to the requirements of the application.

Holliman and Memon [34] have introduced the concept of collage attack. This attack can be mounted by combining blocks where every block is independently authenticated. This attack is thus undetectable by conventional independent block-based approaches. Correlation based watermark-embedding makes the proposed approaches being able to detect the counterfeiting/collage attack. Similarly, security aspects have also been further enhanced by correlating the watermark before embedding. The computational complexity has been reduced by using the integer discrete cosine transform (IntDCT) and integer wavelet transform (IntWT).

1.4 Thesis Structure

In Chapter 2, the literature related to watermarking techniques is provided. The types, applications, and the properties of watermarking are briefly explored. Robust, fragile, and semi-fragile watermarking approaches for different applications are explored. In this chapter, the algorithms that are used to recover the original content are also described. Watermarking approaches are described to demonstrate their effectiveness. Next, detailed discussions about JPEG compression, Huffman coding, BCH coding, Integer Wavelet Transform, and Integer DCT has been provided.
Chapter 1: Semi-Fragile Watermarking for Image Authentication and Recovery

Chapter 3 discusses the first contribution to this thesis. A multiple semi-fragile watermarking approach for secure image authentication and recovery has been presented. In this phase of thesis, two watermarking approaches; Piva et al [35] and Wu et al [36] are combined and thus the advantages of both the approaches with some modifications are exploited which enables the proposed technique to acquire both the attributes of accurate authenticity and recovery. The first watermark is used for authentication and second for recovery. The algorithm lets the users to select one or both of the watermarks based on the requirements of the application. Performance comparison of the algorithm is described by comparing it with previous approaches.

In Chapter 4, we discuss how to overcome the performance limitations of the algorithm presented in Chapter 3. Because the approach presented in Chapter 3 has security holes and the attacker is free to tamper the regions undetectably in transform domain. Now-a-days, security consideration is an important aspect in watermarking systems. The technique described in Chapter 4 has the ability to make the image more secure without any sacrifice on imperceptibility or robustness. Additionally, there is no free zone for an attacker to tamper the image undetectably. In addition, it has the ability to detect the collage attack, which is undetectable by the approach proposed in Chapter 3.

In Chapter 5, the algorithm described in Chapter 4 is improved by utilizing integer DCT for recovery and the ability of Huffman coding for reducing the watermark payload (or imperceptibility). Unlike to the algorithms described in Chapter 3 and Chapter 4, the algorithm in Chapter 5 has the ability to recover the exact version of the original image. In Chapter 3 and Chapter 4, when the tampering strength increases, the recovered image degrades correspondingly. Thus, two major improvements are obtained in this Chapter: one is the exact recovery of original image and the other is to reduce the watermark payload.

Chapters 6 conclude the entire research work and recommend some future directions in the field of digital watermarking.
Chapter 2

Digital Watermarking Techniques: Literature Review

The main reason for development of digital watermarking research is the endeavor for coming up with innovations to protect intellectual properties of the digital world. This is because the recent technological advancement in generation, storage, and communication of digital content has created/generate problems like copying the digital contents without any constraints, forgery, and editing without any prohibitive professional efforts. The absence of protecting techniques makes it doubtful to use the digital communication system in medical, business, and military applications. Watermarking is one of the most common solutions to make the data transferring secure from the illegal interference.

A general watermarking system can be divided into three main components:

1) The watermark signal \( W \), which is generated with the help of generating function \( f_g \). Usually the watermark generation depends on a secret key \( k \) and watermark information \( i \).

\[
W = f_g(i, k)
\]

Sometimes, it also depends on the cover work \( X \), where the watermark is:

\[
W = f_g(i, k, X).
\]

2) The embedding function, \( f_m \), that incorporates the watermark signal \( W \), into the cover work \( X \), and yields the watermarked data \( X_w \). Typically, the watermark incorporation depends on a key \( k \),

\[
X_w = f_m(X, W, k).
\]

3) The extracting function \( f_x \) recovers the embedded watermark \( W' \) on the receiving side. The key used in the embedding process is supposed to be available on the receiving end along with the availability of the original data \( X \) (This is an informed watermarking),

\[
W' = f_x(X, X_w, k)
\]
or without any availability of information about the original data $X$ (This is blind watermarking).

$$W' = f_X(X_w, k).$$

### 2.1 Applications of Watermarking

There are diverse applications of watermarking for which suitable watermarking systems are designed. Digital watermarking is well established research area with plethora of applications [37], but the major applications of watermarking are copyright protection, broadcast monitoring, and authentication of digital content [38, 39]. Significant number of watermarking techniques can be found in the literature used in the variety of applications because of their advantages over the alternative methods. For example, copyright protection needs robust watermarking and authentication needs fragile watermarking, etc. Below we will discuss few of the interesting watermarking applications.

#### 2.1.1 Broadcast Monitoring

Advertisers want to ensure that they receive all of the air time which they purchase from broadcasters (Japan 1997) [5, 40]. A non-technical method in which human observation is used to watch the broadcast and check the originality by seeing or hearing is an error prone and costly. Thus, there should be an auto-identification system, which may store the identification codes to the broadcast. There are several techniques like cryptography that store the identification code in the file header but the data is unlikely to survive any sort of modifications even format change. Watermarking is obviously a suitable technique for information monitoring. The watermark exists within the content and is compatible with the installed base of broadcast equipment. Although, embedding the identification code is very complicated compared to the cryptography where the code is placed in the file header. Moreover, it also, affects the visual quality of the work. Still, many companies protect their broadcasts through watermarking techniques.

#### 2.1.2 Ownership Assertion

A rightful owner can retrieve the watermark from digital content to prove his ownership. There are limitations with textual copyright notices, as they are easily
removable. Copyright notice printed on the physical document cannot be copied along with the digital content. Although, it is possible that text copyright can be placed in an unimportant place of the document to make them unobtrusive [41]. Imperceptible and inseparable watermark is the best solution as compared to the text mark for owner identification. The watermark is not only used for identification of the copyright ownership but for proving the ownership of the document as well. The ownership can be carried out by extracting the embedded information from the watermarked document [42].

### 2.1.3 Transaction Tracking
Transaction tracking is often called fingerprinting, where each copy of the work is uniquely identified, similar to the fingerprint that identifies an individual. The watermark might record the recipient for each legal distribution of the work. The owner embeds different watermarks in each copy. If the work is misused then will the owner be able to find the traitor? Visible watermarking is adopted for transaction tracking but invisible watermarking is much better. For example, in movie making, the daily videos (also called dailies) are distributed to the persons who are concerned with the movie. Sometimes, the videos are disclosed to the press, so the studios use visible text on corner of the screen, which identifies the copy of dailies. Thus, the watermark is preferred as the text can easily be removed.

### 2.1.4 Content Authentication
The procedure to confirm the integrity of watermarked data and to make sure that the data is not being tampered with i.e. act of establishing or confirming whether image is authentic or not. The term authentication has an extensive range of meanings. For instance, an authority that decides whether a portion of art is authentic or not, can a user view or download it? Finally, the decision is to whether the content of an object is staying intact or not after its transmission on the internet.

Many cultural organizations spend time and investing money on new technologies of image documentation and digital libraries construction etc. At the same time, these organizations can guarantee the authenticity of the pieces of art they possess, since they have both the ownership and the experts opinions. When these works of art are digitized and published on the internet, numerous problems take place. Usually
several digital images found on the internet have many differences, but at the same time pretending to represent the same piece of art.

Use of watermarking related to authentication comprises of trusted cameras, video surveillance and remote sensing applications, digital insurance claim evidence, journalistic photography, and digital rights management systems. Commercially, its applications are expected to grow as does the applications of digital content, for example, GeoVision’s GV-Series digital video recorders for digital video surveillance to prevent tampering [37, 43-45].

The digital work can easily be tampered by using computer resources. A solution to the tamper detection is watermarking, where the authentication mark (watermark) cannot stay with the work after slightest modification [46]. Conversely, the system does not matter that the work is compressed or significant changes are made. This leads toward semi-fragile watermarking where the system survive the friendly manipulations and fragile against substantial manipulations [5].

**Authentication Requirements**

The peculiarities of the authentication requirements are set according to the necessity of the particular application. The most common requirements for watermarking based authentication are:

- **Imperceptibility**: How much the original and watermarked images are visually similar?
- **Fragility**: How much the embedded watermark is fragile against different kind of attacks?
- **Security**: How much the watermark and cover work are safe?
- **Efficient Computation**: What is the cost function of algorithm?
- **Capacity**: What is the strength of the watermark?

**2.1.5 Copy Control and Fingerprinting**

Copy control and fingerprinting are used to prevent people from making illegal copies of the content. This issue is very similar to the transaction tracking of the content. An owner can embed a watermark into digital content that identifies the buyer of the copy (i.e. serial number). If unauthorized copies are found later, the owner can trace the origin of the illegal copies.
2.2 Properties of Watermarking

The significance of a watermarking property in any particular application depends upon the requirements of that specific application. Some of the watermarking properties are highlighted in this section. Each of the watermarking property depends on the requirement of the application for which the technique is being designed [5].

2.2.1 Embedding Effectiveness

Effectiveness of watermarking system is the probability of detecting watermark(s), especially at the receiving point. The desired effectiveness is 100% but it is often not possible because of the requirement of perceptual similarity conflicts. Thus, it is application dependent to sacrifice effectiveness for better performance with respect to other characteristics.

2.2.2 Perceptual Similarity

Perceptual similarity is a measure that determines the similarity level between the original and watermarked image, especially at the receiving end. Sometimes the fidelity of the system can be sacrificed for the better performance with respect to other characteristics like higher robustness or low cost. The most commonly used image similarity index measure is PSNR (Peak Signal to Noise Ratio) for two $M \times N$ images, $I$ and $K$ where one is original and the other is watermarked image that can be calculated in Eq. 2.1,

$$PSNR = 20 \log_{10} \left( \frac{\text{max}^2}{\text{MSE}} \right)$$  \hspace{1cm} 2.1

where $MSE$ is defined in the Eq. 2.2,

$$MSE = \frac{1}{MN} \sum_{i=0}^{1-M} \sum_{j=0}^{1-N} [I(i,j) - K(i,j)]^2$$  \hspace{1cm} 2.2

where $\text{max}$ is the possible maximum value of the image i.e. $\text{max} = 255$ for 8-bit gray scale image.
2.2.3 Robustness
Robustness is the ability to detect the embedded watermark after common image processing operations like compression, filtering, geometric distortion etc. Sometimes watermarking systems are developed which have the ability to survive most of the intentional manipulations. Robustness is application dependent and it is not necessary that all the applications require robustness against all the operations. For example, in broadcast monitoring the robustness is required only against the communication related manipulations. In fragile watermarking, robustness is undesirable. However, there is another class of watermarking called semi-fragile watermarking, where robustness is required only against the unintentional manipulations.

2.2.4 Data Payload
The number of bits, a watermarking scheme encodes within a cover work is referred to as data payload and is application dependent. For $N$ bits watermark, the system can encode any of $2^N$ different messages. Increasing the watermark payload will affect the fidelity of the system and vice versa. Thus, it is very important for the researchers to make trade-off between contradicting properties of the watermarking while developing the watermarking systems. The three main contradicting parameters are robustness, imperceptibility, and payload. Increasing the watermark payload will affect the perceptual similarity (fidelity) of the image and robustness is affected by decreasing the watermark payload.

2.2.5 Blind and Informed Detection
In informed watermarking systems (i.e. transaction tracking), the detector requires the original or some information about the original unwatermarked image. However, in blind watermarking systems (i.e. copy control application), there is no need of original or any information about the original image. The terms private and public watermarking systems may be used alternatively for informed and blind watermarking approaches respectively [5].

2.3 Watermarking Techniques
In this section, brief reviews about various contemporary watermarking techniques are given, which deals with the protection of digital content. The algorithms described
in this section are applied to gray scale images but most of the techniques can be extended to color images by embedding watermark in the luminance channel. The watermarking techniques to be presented in this section are categorized based on following criterion:

- Where the watermark needs to be embedded?
- What type of watermark has to be embedded (watermark types are application dependent)?
- What is the robustness of the watermark against different kind of attacks?
- What domain the techniques work in (Spatial or transform domain)?
- Either the watermark is additive or the modified coefficients of the image will represent the presence of the watermark. Further, will the watermark extraction be blind or informed?
- What is strength of watermark in a particular scheme? The watermark strength and the imperceptibility are contradicting in nature.
- What will be the behavior of tamper detection system?
- Does the scheme support data recovery?

2.3.1 Embedding Locations

Human eyes are very sensitive to smooth regions instead of textured regions in the image. The watermarks in the textured regions need significant distortion to be removed [47, 48]. Various watermarking methods are proposed where the watermarks are embedded in perceptually significant parts of image. In [49-51], the image is first transformed into DCT domain and watermark is embedded in mid-frequency coefficients. The middle frequencies are robust against JPEG compression and have less perceptual distortion compared to the regions where the variation of intensities changed gradually. In [52], the wavelet coefficients are utilized to embed the watermark in the proper locations of the image. Nevertheless, the main drawback is that it is an informed technique and need original image at the receiving end. The suitable locations can be pointed out for watermark embedding by using the secret key. In this regard, in [53] the author uses a secret key to select the proper coefficients for embedding the watermark. The selected coefficients are first divided into two halves where first half is incremented by one while the other half is decremented by one. The same key is supposed to be available on the receiving side, which is used to
select the embedding coefficients. This approach is thus based on statistical change in the image, which is very straightforward and it only verifies the existence of the watermark. Perceptual adaptive watermarking techniques are proposed in [54-56], where the watermark is not as straightforward as in [53]. A spread spectrum watermarking approach based on complex wavelet transform is proposed in [57] where the performance is improved by resisting some attacks.

2.3.2 Robust Watermarking

A significant number of robust watermarking approaches are proposed for copyright protection, which aimed to protect the copyrights of the digital images. In [8, 9], the robust watermark is embedded by modifying the mid-high frequency part of the image based on wavelet transform. The visually recognizable pattern like binary or gray scale image is used as watermark and has the ability to protect the image and identify the ownership as well. DCT based robust watermarking approaches where the watermark is perceptually tuned to protect digital images are proposed in [10, 54-56, 58]. The image adaptive watermarks are designed and they are able to protect the image contents. The algorithms find optimal transformed coefficients for the watermark embedding and it simultaneously improves robustness and image quality of the watermarked image. In [59], a robust image watermarking technique is proposed based on an invariant pattern recognition using radon transform. The scheme can have useful applications in the area of medical imaging. The invariant features are used as a watermark and the extracted features are selected for embedding purpose. The root mean square error (RMSE) is used as a similarity measure and this technique has the ability to resist the geometric distortions.

Kundur et al [60] have proposed a robust watermarking approach based on wavelet transform. The original image $I$ and the authentication watermark $W$ where $W(x,y) = \{1, -1\}$, are decomposed up to $Lth$ level by using the wavelet transform. The details of the host image are divided into blocks. Then the watermark is embedded into these blocks in view of significant $S$ which is the numerical measure based on information about human visual system (HVS). The watermark is extracted by the inverse procedure of the embedding process. This is an informed technique where the original unwatermarked image is required.
Similarly, Liu et al [61] have developed a robust watermarking approach based on the wavelet transform using original image $X$, and its reference image $X'$ for watermark embedding. The watermark $W$ is the binary image with $W(x, y) = \{1, -1\}$ and embedded according to the original and its reference image. The reference image is obtained by applying the first level decomposition to the original image and assigns zeros to all its detailed coefficients. It then applies the inverse wavelet decomposition. Proper locations are obtained based on some constraints to make trade-off between robustness and imperceptibility and then the watermark is embedded in the coefficients, which are randomly selected from the proper locations.

### 2.3.3 Fragile Watermarking

The main aim of robust watermarking is to resist attacks and it is difficult for watermarking techniques to detect the friendly manipulations like JPEG compression. Therefore, fragile watermarking approaches have been proposed which are sensitive to all kinds of distortions. In this regard, a wavelet based fragile watermarking algorithm has been proposed in [16] where the authors address the problem of fragile watermarking and embed the watermark by quantizing DWT coefficients. DCT, quantization index modulation, frequency modulation, non deterministic, block wise dependence, and image structure based fragile watermarking approaches are also proposed to protect the digital images and detect tampering [20, 62]. These approaches have capability to detect any sort of manipulation either incidental like JPEG compression or malicious like cut/copy past.

It is worthwhile to reveal the work by Fridrich and Goljan [63], where two schemes are proposed: the first is fragile watermarking technique used to authenticate the digital content and second is used to reconstruct the region where the integrity verification fails. The quality of the reconstructed image degrades after increasing the strength of attack. One of the authentication techniques given by Chen et al [64] in which the encrypted hash values are embedded in the wavelet subbands of the image. This technique applies three-level wavelet transformation to original image and then divides each of the detail subbands into blocks i.e. $8 \times 8$ in level-I subbands, $4 \times 4$ in level-II subbands and $2 \times 2$ in level-III subbands. Then the hash values of each block are calculated by a cryptosystem. The block itself, neighboring blocks, and the corresponding approximations are involved when the hash function is applied to all
these subbands. These values are then embedded in least significant bits (LSBs) of the block. On receiving end, the inverse of the embedding process is applied and the key, used in watermark generation process is supposed to be available at the receiving end. The advantage of this algorithm is that, a tiny change in the watermarked image will destroy the hash value and second advantage is that the detection resolution is not affected by changing the block size.

Yang et al [65] discuss the fragile watermarking for authentication process in a very different manner by introducing the Chinese Remainder Theorem (CRT). CRT provides a unique solution based on prime numbers [66]. The image is divided into \( n \) parts \( m_1, m_2, m_3 ..., m_n \). A set of sequenced distinct prime numbers as public system parameters are chosen i.e. \( p_1, p_2, p_3 ..., p_k \), which are pair wise co-prime. These primes are public and common to all type of images, therefore the number of these primes should be equal to or greater than the maximum number of expected image partitions. Applying hash function such as MD5 (Message Digest) or SHA (Secure Hash Algorithm), denoted as \( h(.) \) to calculate hash value of each part together with its unique index. Message digest of part \( i \) is thus \( h(mi, i) \). Thus \( H \) is computed as given in Eq. 2.3.

\[
H = h(m_1, 1) \mod p_1, \\
H = h(m_2, 2) \mod p_2 \\
\vdots \\
H = h(m_n, n) \mod p_n.
\]

Since \( p_i \) are co-primes, therefore according to the CRT there exists a unique \( H \) which is signed as the digital signature of the image \( M \) denoted here as \( H_s \). \( H \) and \( H_s \) are then archived together with image \( M \) to obtain the watermarked image. Assume that \( m_i \) and \( m_j \) are the requested parts for authentication. With \( H_s \), the verifier can easily determine the authenticity of \( H \). Next, the verifier computes the hash value of the requested part \( m_i \), \( h_i = (m_i, l) \) respectively. To authenticate \( m_i \), the verifier checks to see whether \( H = h_i \mod p_l \), if it holds, \( m_i \) is considered authentic; otherwise \( m_i \) has been tampered with. Authentication of part \( m_j \) follows the similar process.

Another wavelet based fragile watermarking approach is proposed in [16], where the watermark is embedded in DWT domain by quantizing the wavelet coefficients. Different decomposition levels grant the tamper detection within the image in
localized spatial and frequency domain. The aim is to present an authentication technique that hides watermark into some wavelet subbands of the to-be-authenticated image. This scheme is capable to detect the malicious and the incidental manipulations. Furthermore, security is a particular concern that is often overlooked. It is extremely difficult for an attacker to create a faked image that appears to be authentic.

2.3.4 Semi-Fragile Watermarking

Image can be manipulated in two ways: manipulation method i.e. JPEG compression, format transformation, quantization, filtering etc and manipulation purpose i.e. attack. Manipulation method is acceptable but manipulation purpose is not.

Semi-fragile watermarking approaches have been proposed to make the system robust against friendly manipulations (manipulation method), and fragile against malicious manipulations (manipulation purpose) [24, 25]. These approaches have some tolerance against JPEG compression and the system parameters are defined as to how much the approaches are able to survive such manipulations. In communication system, the basic requirement is to transfer the low-size data and for that purpose, the compression technique is used. Usually, in semi-fragile watermarking approaches, the compression parameters are defined in embedding stage to make system tolerable against friendly manipulations [33]. These parameters are flexible and can be set according to the requirement of the application. Because some applications require high compression while in others, we need low or no compression.

The algorithm proposed in [67] hides data in the multimedia document. At least 2 bits of watermark could be hidden into each 8\times8 block of the host image i.e. an estimation of 8192 bits that can be reliably hidden in 512\times512 image. JPEG Compression algorithm is employed for generating the recovery watermark and half-sized copy of the to-be authenticated image is compressed. A quantization factor of around 64 would be required for obtaining 8192 bits payload. The technique is able to survive compression but is fragile against any other information loss. The luminance channel is used to hide information in color images.

Lin et al [32] have introduced another DCT based technique which is robust against JPEG compression. The image is transformed into DCT domain and then divided into blocks i.e. 8\times8. Blocks pairs are made and then the watermark bits are
obtained by taking difference in the corresponding coefficients of the block pairs. The watermark is embedded in each block and the watermarked image is obtained by taking inverse DCT. On verification side, the inverse of the embedding process is used to extract the watermark bits. It survives JPEG compression to a pre-determine quality factor and rejects the intentional manipulations. This technique has also the ability to localize the tampered regions and recover the corrupted blocks approximately. This technique has the limitation that when the malicious tampering occurs then it may not localize it. Because half of the blocks are used for watermark generation and other half of the blocks are used for embedding. If tampered region affects the block, which is used for watermark generation, then it will not be able to localize it.

Hu et al [68] presented a watermarking scheme to authenticate the image by using two semi-fragile watermarks. After wavelet decomposition of the $M \times N$ image up to three levels, wavelet coefficients are denoised with adaptive wavelet threshold method. Adopt sobel edge detection on the $LL3$ component to get $1/8M \times 1/8N$ binary sobel edge map $WL$. It satisfies the need to detect coarse changes in the image and roughly localize the tampered regions. As, $WL$ is used as to-be embedded watermark, compute the hash value of $WL$ and mapped for comparison to a $1/8M \times 1/8N$ binary matrix $WA$ by the session key. Because of the properties of the hashing, the output $WA$ varies with any change to the input, which satisfies the need of tamper detection. Encrypt $WA$ to produce $WE$ of the same size by using an encryption key. Thus, two watermarks, $WL$ and $WA$, are created from the $LL3$ (Approximation at third level wavelet decomposition). Embed the watermarks $WL$ and $WE$ in the middle frequency components of $LH2$, $HL2$ and $HH2$ (Low-High, High-Low, and High-High subbands after two level wavelet decompositions respectively), that gives both the invisibility and robustness. In order to improve the watermark robustness and increase the security, the session key is used to select a stochastic non-overlapping $2 \times 2$ block for embedding a watermark bit by mean of quantization method. This technique is robust against compression and fragile to other content modifications. The scheme is able to classify the incidental and malicious manipulations. A block-based semi-fragile watermarking approach is described in Zhuo et al [69] for protecting the image contents. Input image is divided into $n$ non-overlapping block of size $16 \times 16$ and $k$ bits are extracted from each block. Vector $b_i(k)$ is defined as signature of block $i$. 

17
Thus, the length of signature is $k \times n$ bits and the average gray value of each block is calculated $A_i (i = 1, 2, 3, ..., n)$. The quantization step $\Delta$ is defined in Eq. 2.4.

$$\Delta = \frac{\max(A_i) - \min(A_i)}{2^k}$$

2.4

Gray code is selected for embedding the signature. If $k = 2$, then $b_i$ is calculated in the following figure:

![Gray code figure]

When the length $k$ is longer, the quantization step $\Delta$ is smaller and it tends towards more fragility. In this technique $k = 4$ and the watermark is encrypted and then inserted by modulating the wavelet coefficients. BCH decoding is used to correct the erroneous bits at the receiving end as the watermark bits are encoded by using BCH encoder. At the receiving point, the watermarked image is divided into $n$ number of non-overlapping blocks. Extract the encrypted watermark bits from each block and then decrypt it. If there is an erroneous bit(s) in the extracted watermark then the BCH decoder will correct it and the modified blocks will be localized.

Liu et al [70] proposed the content based watermarking technique using wavelet transform. The watermark is generated from the image features and embedded by quantizing the detailed coefficients. Zernike moments of the low pass wavelet band of the host image are chosen as feature [71]. This approach is able to survive JPEG compression and localize the tampered regions.

### 2.3.5 Configuration of Cover Work by Watermark Embedding

The very common way of watermark embedding is the addition of pseudo random number sequence (based on a secret key) to the amplitude of the digital data either in the spatial or transform domain [72]. Recently, image contents are changed by some other methods i.e. spread spectrum and quantization. In [73], the watermark is embedded by quantizing the image contents and on the receiving end the watermarked components are de-quantized to get the embedded watermark back. The recent techniques circumvent the disadvantages of the additive watermarking methods.
2.3.6 Watermarking with Recovery

Recently besides authentication, some researchers have proposed watermarking techniques that have the ability to protect the image contents as well as recover the image if the watermarked image is being tampered. For example, the altered areas of the image are recovered in [74] using vector quantization technique. The embedded information has the ability to detect the tampered regions and recover it as well. DWT based technique has been utilized in [75] to extract image information from the low-frequency coefficients and then hide in the mid-frequency components for tamper detection and recovery. The eigenvalues are calculated and embedded in some middle frequency band of wavelet transform domain. Based on these eigenvalues, the image can be recovered if it is tampered. In [44], fractal image compression and watermarking schemes are combined to accomplish image authentication as well as recovery. The characteristic values are filtered and then embedded in the least significant bits of the image. The watermark is embedded in a way that the image can be restored. In [76], a block-based hierarchical watermarking approach is used whereby the image can be recovered elegantly but the authenticity of the system is weak as it block wise authenticates/localizes the content. The block-based semi-fragile watermarking techniques are proposed in [32, 77], where the corrupted blocks are recovered. In these techniques, recovery bits of block A are encoded in other block B, which is chosen by using a secret key. The problem with these techniques is that both the blocks (A and its corresponding block B) can be corrupted and in this case, they will be unable to recover the required bits. Furthermore, Radhakrishnan and Memon [78] suggest a possible solution to enhance the security of the authentication systems proposed in [32, 77] as they are not completely secure.

Similarly in [33], a dual watermarking system is designed accurately that authenticates the image and recover it by using self-recovery technique but at the cost of imperceptibility. The embedding of the authentication watermark, which is generated by using a secret key, is weak. The authentication watermark is embedded by modifying the five least significant bits (LSB) of the LL3 subband (Approximation subband after taking third level wavelet decomposition). The attacker can easily modify the significant bits leaving five LSBs intact. The wavelet subbands LL1 (approximation subband at first level wavelet decomposition), HH2 and VV2 (detailed subbands at second level decomposition) are secured by using the second watermark.
The recovery watermark is being secured by using keys $k_1$ and $k_2$. However, subbands other than $LL1$, $HH2$ and $VV2$, are not protected. The attacker is free to modify these areas of the watermarked image without being detected. Therefore, as regards the security aspects of the scheme proposed in [33], it is hard for the attacker to successfully substitute the concerned subbands with another image. However, manipulations to degrade the quality of the watermarked image are possible. Security consideration is designated as one of the most important aspect of watermarking system.

2.3.7 Watermarking Domains

Watermark can be embedded in spatial as well as in transform domains. In spatial domain, the watermark is directly inserted in the intensities of image pixels to modify the image. Nevertheless, in transform domain, first the image is transformed to frequency domain by using some sort of transformation like DCT, Discrete Fourier Transform (DFT), DWT, etc and then embeds the watermark. Watermarked image is obtained by applying the inverse of the corresponding transformation. Spatial domain is sensitive towards modification and that is why now-a-days most of the researches are using different transform domains for their watermarking algorithms. In many approaches [79-81], the DFT domain is used to embed the watermark in magnitude of DFT coefficients and the watermark is shaped according to the human visual system for each frequency band. DCT is very common in today’s research because it is used in JPEG and MPEG compression. The DCT is utilized in the similar manner as the DFT is utilized in watermarking methods. Many DCT based approaches are proposed for watermarking because the extensive use of DCT in compression makes it possible to use for adapting the watermark. Wavelet transform is also an important domain in image processing which split the image into spatial-frequency manner. JPEG2000 compression is based on wavelet transform instead of DCT. Significant watermarking methods are proposed in wavelet domain. Kundur et al [60] embed the watermark in the wavelet coefficients and later in [82], the author describe a technique based on quantization where the wavelet coefficients are quantized by Watson’s [83] standard quantization matrix.
2.4 Attacks and their Countermeasures

There are two ways to interfere in the image: first one is to change the visual meaning of the image and other is to obtain the information about the watermarking algorithm [84]. The former is to manipulate the image by cut/copy past, cloning the image pixels or attempt to mix the pixels with adjacent areas. In the later one, the attacker may know the secrets of the watermarking algorithm and is able to create different strategies to make the owner fool.

Attacks can be divided into two parts based on their strength: incidental and malicious. Incidental manipulations are friendly and sometimes required and are usually cannot be considered as an attack. For example, JPEG compression is very necessary in internet application to save time and to reduce load on the communication channels. Robust watermarking approaches are proposed to allow both the incidental and malicious manipulations. Fragile watermarking does not allow any kind of manipulation and semi-fragile watermarking techniques are designed that are robust against friendly manipulations but fragile against malicious manipulations.

An intentional manipulation is considered as an attack and large numbers of watermarking methods are described to deal with different kind of attacks.

Attacks can be classified in four categories: Geometric, removal, interference, cryptographic, and protocol attacks. It is very difficult to resist the malicious attacks but the watermarking techniques must have the potential to deal with the requirements of robustness [54, 85]. Many techniques are developed to deal with filtering, collage/counterfeiting, removal, copy/past [5]. An attacker can use filtering technique to remove watermark i.e. if the watermark is embedded in the high frequencies of the image and low pass filter is applied then the watermark will be filtered (destroyed), e.g. Weiner filtering is an optimal linear filtering [86]. Holliman and Memon [34] develop an attack called collage/counterfeiting attack which is undetectable by the traditional watermarking algorithms. The techniques proposed in [87, 88], make it possible to detect collage attack by applying the watermark correlation with the original cover work.
2.5 Capacity vs. Robustness vs. Imperceptibility

Conceptually the capacity of the watermark with respect to other two contradicting parameters is described in Figure 2.1, where the curves are theoretically drawn. Increase in any parameter will affect the other. These parameters are inversely proportional and there will be a trade-off between the robustness and the visual quality of the watermarked image while deciding the strength of the watermark. If robustness against attacks is the basic requirement, then we have to embed the larger amount of information, which will degrade the image quality, and less information results the watermarked image visually similar to the original image. The watermarking techniques must have the ability to make a trade-off between embedded information, imperceptibility and robustness. In [89, 90], the algorithms have the ability to generate and embed the watermark in a manner that it become imperceptible and robust against the unintentional manipulations as well. In [33, 87, 91] dual watermarking approaches are developed where multiple semi-fragile watermarks are embedded and a trade-off will be came into existence between watermarks strength and the other two parameters i.e. robustness and imperceptibility.

![Figure 2.1 Three common contradicting parameters of watermarking system: watermark strength, imperceptibility, and robustness](image)

2.6 JPEG Compression

JPEG compression works on color and gray scale images. It does not work on the bi-level (black and white) images. There is a trade-off between compressed image size and reconstructed image quality. The JPEG compression works on $8 \times 8$ blocks of a
matrix. The DCT transform is applied to these blocks where the gradual changes (lower frequencies) are pushed towards the upper left of $8 \times 8$ matrix [92]. These DCT values are then quantized by using the JPEG recommended quantization matrix. The basic process is given in Eq. 2.5.

$$\text{round} \left( \frac{\text{DCT} 8 \times 8 \text{ block}}{\text{Quantization table}} \right) = \text{sparse matrix}$$ \hspace{1cm} 2.5

The sparse matrix contains the rounded DCT values and the rounding errors are obtained which cannot be retrieved. The main pattern (low frequency) is preserved because the DCT places the values in the top left corner. These top left corner coefficients are thus divided by smaller numbers. The quantized DCT matrices are then reordered by using zigzag scanning and give long runs of zeros. The main information is then stored by some arithmetic operation and most of the zeros are removed to decrease the file size. Finally, the original and the compressed images are compared and the size of the compressed version is almost $1/4$th of the original image size. The decompression gives the original image back where some of the image information loses. The decompression is performed by applying the reverse procedure of the compression technique.

### 2.7 Integer Wavelet Transform

Integer wavelet transform (IntWT) is implemented by using an approach called Lifting Scheme (LS) based on a well-established transform domain called discrete wavelet transform (DWT). LS is an efficient implementation of DWT and ensures the perfect reconstruction [93-96]. The perfect reconstruction is possible by exploiting the redundancy between low pass and high pass filters. Three steps are employed in LS: Splitting, Prediction, and Update. The input data is split into even and odd indexed samples by using lazy wavelet. In dual lifting step the even samples are convolved with the lifting filter and the result is subtracted from the odd samples as given in Eq. 2.6,

$$d_{1,l}^i = d_{1,l}^{i-1} - \text{conv}(c_{1,l}^{i-1})$$ \hspace{1cm} 2.6

where $c$ and $d$ are the odd and even samples respectively and $\text{conv}$ is the convolution function. In primary lifting step, unlike to the dual lifting step, the odd samples are
convolved with the lifting filter and the result is subtracted from the even samples as in Eq. 2.7.

\[ c_{1,l}^i = c_{1,l}^{i-1} - \text{conv}(d_{1,l}^{i-1}) \]  \hspace{1cm} 2.7

After \( M \) pairs of dual and primal steps, the even samples become low pass and the odd samples become high pass coefficients up to a scaling factor \( K \) mentioned in Eq. 2.8,

\[ c_{1,l} = \frac{c_{1,l}^{(M)}}{K} \quad \text{and} \quad d_{1,l} = K * d_{1,l}^{(M)} \]  \hspace{1cm} 2.8

where \( c \) and \( d \) are the coefficients obtained after primal and dual lifting steps respectively. By taking reverse operation flipping the signs in the above steps we can get even and odd samples back.

The above values are then round-off by using the dual and primal integer lifting steps, which will be reversible. The dual integer lifting steps round-off the odd samples as given in Eq. 2.9,

\[ d_{1,l}^i = d_{1,l}^{i-1} - \left[ \sum_k p_k \, s_{1,l-k}^{i-1} + 1/2 \right] \]  \hspace{1cm} 2.9

and the primal integer lifting step round-off the even samples as specified in Eq. 2.10,

\[ s_{1,l}^i = s_{1,l}^{i-1} - \left[ \sum_k u_k \, d_{1,l-k}^{i-1} + 1/2 \right] \]  \hspace{1cm} 2.10

where \( p \) and \( u \) are the convolution functions. The process discussed above is invertible and the inverse is the flipping of signs and reverse of the operations.

**Example:**

The original image is decomposed by using the integer wavelet transform in all the algorithms described in this thesis. For example:

```
WaveName = liftwave (wname)
WaveName = liftwave (wname, 'Int2Int')
```

The inverse integer wavelet transform is:

```
Reconstructed Image = iilwt2 (subbands generated after decomposition, WaveName),
```

where \( wname \) can be any wavelet type given in Table 2.1.
2.8 Huffman Coding

In 1952, an efficient lossless data compression technique based on entropy encoding algorithm was developed by David Huffman.

It is good in practice to transfer symbols of data instead of original data. Transmission time is directly proportional to the number of symbols (message code) associated with the original data [98]. Ensemble codes are generated that represent meaning of the message(s) codes for which the transmitter and the receiver are agreed. The restrictions are that, the ensemble codes must be different for different messages and there should be no additional requirements with ensemble codes to specify the message. Kraft [99], derive a technique which gives almost ideal code length. A simple way to derive the Huffman coding is as follows:

The values are first quantized and entropy encoder further compresses the quantized values losslessly. The probabilities of each quantized values are determined and an appropriate code is shaped based on these probabilities. A symbol-by-symbol coding is performed with identified probability distribution of input symbols. In short, the input symbols and their probabilities are given to the system and the output is a
binary code with minimum length. Formally, the Huffman coding can be described as follows:

Let \( I(I_1, I_2, I_3, ..., I_n) \) be the number of input symbols and \( P(P_1, P_2, P_3, ..., P_n) \) are their corresponding probabilities [100]. Then the output is the set of binary codewords, \( C(C_1, C_2, C_3, ..., C_n) \). The length of the output is \( L(C) \leq L(O) \), for any \( O(I, P) \).

### 2.9 Integer Discrete Cosine Transform

The widely used transform domain in signal and image processing called Discrete Cosine Transform (DCT) has the problem of mapping of integer values to floating-point. This property makes it difficult for the researchers to use the floating point DCT (Generally called DCT) in the field of lossless compression and mobile devices [101-103]. Thus, the only solution is to develop a technique that maps the integer-to-integer values. The integer DCT (IntDCT) has become a powerful tool which has been examined for lossless compression and mobile devices [103, 104]. The implementations are simplified by using the Int-to-Int DCT instead of floating point DCT.

The fast algorithm is applied on the DCT where DCT components are supposed to be available. The IntDCT is obtained by factorization of DCT matrices. The DCT matrices are supposed to be able to factorize into a series of elementary integer reversible transformed matrices [105].

For \( 8 \times 8 \) DCT matrix \( A \), the following factorization result is selected from many factorization results through an optimal selection [106] given in Eq. 2.11,

\[
A = P_L S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0 P_R
\]

\[
S_m = I + e_m S_m^T, \quad m = 1, 2, 3, ..., 8, \quad S_0 = I + e_8 S_0^T
\]

where \( e_m \) is the \( m \)th column vector of \( 8 - \text{order} \) identity matrix \( I \), \( S_m \) is the vector where \( m \)th element is 0, \( S_0 \) is the vector where 8th element is 0. The integer transform domain of the given DCT matrices are then given as follow:

Consider Eq. 2.12 present the integer transform,
where \( x \) is the integer matrix. The integer transform can be obtained by using Eq. 2.13,

\[
y_k = \begin{cases} 
  x_k, & k \neq m \\
  x_m + \left[ \sum_{i=1}^{m-1} S_{m,i}x_i + \sum_{i=m+1}^{n} S_{m,i}x_i \right], & k = m 
\end{cases}  
\]

2.13

where \( k = 1, 2, 3, \ldots, N \).

Obviously, the reverse transform is given in Eq. 2.14.

\[
x_k = \begin{cases} 
  y_k, & k \neq m \\
  y_m - \left[ \sum_{i=1}^{m-1} S_{m,i}y_i + \sum_{i=m+1}^{n} S_{m,i}y_i \right], & k = m 
\end{cases}  
\]

2.14

It can be seen that Eq. 2.12 can be implemented reversible for integers, so that \( y = Ax \) can be implemented for integers gradually. The detail discussion about the IntDCT is given in [105, 106].

### 2.10 BCH Coding

BCH is an error detecting and correcting code getting much attention since 1960’s. BCH was developed by Bose and Ray-Chaudhuri in 1960 and BCH comprise the initials of these inventor’s names [107]. It is a random error correcting code which handles the errors located randomly in the data stream [108]. First, the error location is detected and then the correct values are calculated. The major advantage of BCH code is that, it can be decoded easily. Different BCH code pairs are given in Table 2.2.

The number \( n \) represents the number of actual bits used to be decoded, \( k \) represent the number of physical bits and \( t \) corresponds to the erroneous bits, which can be corrected. For example in BCH coding with pair \((31, 16, 3)\), 16 are the actual, and 31 are the physical bits while up to 3 bits can be corrected. If the need is to correct more bits, then the number of physical bits will be high and as for as watermarking is
concerned, the watermark payload will be high. Thus, in the watermarking approaches, a trade-off is made while selecting the BCH code pair. BCH encoding and decoding uses the Galois Array for messages and their corresponding codewords. Please refer [109] for creating and representing elements in Galois Array.

### Table 2.2 Generators of primitive BCH codes

<table>
<thead>
<tr>
<th>$n$</th>
<th>$k$</th>
<th>$t$</th>
<th>$n$</th>
<th>$k$</th>
<th>$t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>4</td>
<td>1</td>
<td>255</td>
<td>247</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>1</td>
<td>239</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>231</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
<td>223</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>26</td>
<td>1</td>
<td>215</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>2</td>
<td>207</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>3</td>
<td>199</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>5</td>
<td>191</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>7</td>
<td>187</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>57</td>
<td>1</td>
<td>179</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>2</td>
<td>2</td>
<td>171</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>3</td>
<td>3</td>
<td>163</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>4</td>
<td>4</td>
<td>155</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>5</td>
<td>5</td>
<td>147</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>6</td>
<td>6</td>
<td>139</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>7</td>
<td>7</td>
<td>131</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td>10</td>
<td>123</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>11</td>
<td>11</td>
<td>115</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>13</td>
<td>13</td>
<td>107</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>15</td>
<td>99</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>120</td>
<td>1</td>
<td>91</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>2</td>
<td>2</td>
<td>87</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>3</td>
<td>3</td>
<td>79</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>4</td>
<td>4</td>
<td>71</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>5</td>
<td>5</td>
<td>63</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>6</td>
<td>6</td>
<td>55</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>7</td>
<td>7</td>
<td>47</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>9</td>
<td>9</td>
<td>45</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>10</td>
<td>10</td>
<td>37</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>11</td>
<td>11</td>
<td>29</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>13</td>
<td>13</td>
<td>21</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>14</td>
<td>14</td>
<td>13</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>15</td>
<td>15</td>
<td>9</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>21</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>27</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>31</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 2.11 Chapter Summary

The detailed discussions about the watermarking approaches are given in this chapter. Different types of watermarking and their applications are discussed in detail. The watermarking techniques are classified according to the following criteria:
Where the watermark will be embedded?

What type of watermark has to be embedded and how much ability the watermark has to resist different kinds of attacks/manipulations?

The watermarking techniques applied in different domains are discussed. The watermark is being embedded either in spatial domain or in transform domain.

Either the watermark is additive or the modified coefficients of the image will represent the presence of the watermark and extraction will be blind or informed.

Watermarking payload and their effects on system performance are also revealed.

Tamper detection resolution based approaches are presented and it is analyzed that if the watermarked image is attacked, then whether it can be recovered or not.

The discussion about IntWT, IntDCT, Huffman Coding, JPEG compression and BCH coding are also presented.

Next chapter will introduce the dual watermarking technique based on wavelet transform that secure the integrity of the image and recover the image approximation.
Chapter 3

Authentication and Recovery of Images using Wavelet Transform

The ease, by which digital data can be manipulated, has always raised many concerns about the reliability of their content [35]. Digital data authentication is thus one of the most important and investigated security applications. The essential role of watermarking is the reliable embedding and detection of information and therefore, it is generally considered as a form of communications. Consequently, the field of watermarking has great potential both in copyright and authentication based applications. However, the security aspect of watermarking is still a difficult issue and many concepts in this perspective have been borrowed from the field of cryptography [43, 110-112].

Authentication is the procedure to validate the integrity of watermarked data, to make sure that the data is not being tampered with. It is the act of establishing or confirming that the image is credible [43]. Imperceptibility, fragility, security and efficient computation are the basic requirements for authentication. The technique described in this chapter strive for imperceptibility, efficient computation, security and also both the fragility and robustness i.e. semi-fragility. The use of two different potential watermarks enhances security of the proposed semi-fragile watermarking based authentication system.

In this chapter, multiple semi-fragile watermarks are exploited to authenticate the image and recover, if authorisation fails. The technique has the ability to accept JPEG lossy compression with the quality factor as low as 70% and reject the malicious attacks. The quality factor is determined at embedding side that how much JPEG compression is acceptable. The brightness adjustment of the image is also acceptable within a reasonable scope. The tampered regions (visible or invisible) are detected and can be recovered. The technique has the ability to detect the exact tampered location instead of traditional block-based approaches where the tampered blocks are
detected. A self-recovering algorithm is employed, that hides the recovery watermark into some wavelet subbands for detecting possible illicit object manipulation undergone in the image and recovers these manipulations.

This chapter focuses on the exploitation of the advantages of two techniques [35, 36] with some modifications enabling the proposed technique of acquiring both authentication and recovery based attributes. Consequently, this approach is based on a comprehensive technique employing two watermarks [32], a recovery watermark and authentication watermark. The recovery watermark is the highly compressed version of the original image and is used for image recovery, and the authentication watermark is used for accurate authenticity of the image. The recovery watermark is computed through a properly modified version of JPEG coding, operating at very high compression ratio on the original image. Thus, recovery watermark helps in obtaining an estimate of the original contents. The modification is introduced in the recovery watermark, to make it insensitive to global innocuous manipulations. The recovery watermark is embedded in the DWT subbands based on the work proposed by Campisi et al [67], where the color information is embedded to improve the compression effectiveness. The authentication watermark is processed with a private key to ensure security. Embedding authentication watermark can help in accurately detecting manipulations made in image, but it cannot ensure recovery of an estimated image. Similarly, embedding the recovery watermark can retrieve the estimated image but leave the users to judge the authenticity by themselves. Thus, embedding recovery watermark as well as authentication watermark can leads to both authentication and recovery. In this chapter, the following contributions are mainly focused towards authentication and recovery of images:

- Embedding multiple watermarks with different intended applications to achieve both authentication and recovery based attributes at the cost of only a small reduction in imperceptibility. Both these watermarks as analyzed strengthen each other in context of security.
- Integer wavelet transform (IntWT) has been utilized instead of DWT, for generating recovery watermark [35] to achieve lower computational complexity.
For the recovery, *recovery watermark* is used as a compressed version of the original image and this technique is referred to as a self-recovery technique. The scheme is flexible enough with the choice of users, to embed either *recovery watermark* or *authentication watermark* or both of them.

### 3.1 Proposed Algorithm

The proposed technique is based on embedding of two watermarks in independent IntWT subbands. The block diagrams for generation and embedding both the watermarks are given in Figure 3.1 and 3.3 respectively. Two watermarks, *authentication watermark* and *recovery watermark* are generated and then embedded into some wavelet subbands. The *authentication watermark* is embedded for authentication purpose and the *recovery watermark* is used for recovering the image. The *authentication watermark* is represented by $\hat{W}_1$ and *recovery watermark* is represented by $\hat{W}_2$. The generation methods for both the watermarks i.e. $\hat{W}_1$ and $\hat{W}_2$ are given in sections 3.1.1 and 3.1.2 respectively.

#### 3.1.1 Generation of Authentication Watermark

Authentication watermark $\hat{W}_1$, which is used for accurate authenticity of the cover work, is preprocessed before being embedded. The meaningful binary image is chosen (The author use his signature as an *authentication watermark* in this chapter). Before embedding, $\hat{W}_1$ is preprocessed for security purpose based on a secret key. Because it is very easy for an attacker to forge the watermark if he has the little bit knowledge about it. The secret key is supposed to be available at the receiving side. Without the availability of the secret key, it is impossible for the receiver to extract the required watermark. Let $W$ be a binary signature of size $X \times Y$,

$$W = w(i, j) \quad (1 \leq i \leq X \text{ and } 1 \leq j \leq Y)$$  \hspace{1cm} 3.1

where $w(i, j) \in \{0, 1\}$ and $P_{Rand}$ be a pseudo-random number matrix of same size based on a secret key $k_1$ is given in Eq. 3.2,

$$P_{Rand} = R_n(i, j)$$  \hspace{1cm} 3.2

where $R_n(i, j) \in \{0, 1\}$ then the formula adopted to get the ultimate watermark $\hat{W}_1$ is given in Eq. 3.3,
\[
\hat{W}_1 = W \oplus P_{Rand}
\]  \[3.3\]

where \(\oplus\) denotes an exclusive OR.

### 3.1.2 Generation of Recovery Watermark

The recovery watermark \(\hat{W}_2\) which is the highly compressed version of the original image, is generated by using the following steps [35].

- One level Integer Wavelet Transform (IntWT) is applied on the original image of size \(N \times N\). The resultant subbands are approximation \(LL1\), Horizontal \(HL1\), Vertical \(LH1\) and Diagonal \(HH1\). The subband \(LL1\) is selected for creating the recovery watermark \(\hat{W}_2\). The \(LL1\) subband is then compressed through a properly modified version of the JPEG compression technique.

- Full-frame DCT on low pass version \((LL1\) of size \(N/2 \times N/2\)) of the original image is then computed. Block distortions take place while using block-based DCT. This problem can be resolved by applying the full-frame DCT, but it is difficult to adapt quantization to local image structure [113].

- DCT coefficients are scaled down by quantization using JPEG quantization matrix, as shown in Eq. 3.4 to decrease the obtrusiveness of DCT coefficients. The DCT matrix is divided into 64 blocks and then divided each block by their corresponding value in quantization matrix. The standard quantization matrix [83] is employed for this purpose. The quantization matrix described in Eq. 3.4 and all other quantization matrices are designed to keep certain frequencies of the image/signal to thwart losing image quality [114]. The human eye cannot differentiate the high frequency variations exactly. Thus, quantization is used to reduce information in high frequency components and this is done by dividing DCT components by a constant for that component.

\[
\begin{bmatrix}
16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\
12 & 12 & 14 & 19 & 26 & 58 & 66 & 72 \\
14 & 13 & 16 & 24 & 40 & 57 & 69 & 76 \\
14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\
18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\
24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\
49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\
72 & 92 & 95 & 98 & 112 & 100 & 103 & 99
\end{bmatrix}
\]  \[3.4\]

- The scaled DCT coefficients are then ordered through zigzag scanning. First \(M\) coefficients are selected and stored in a vector \(q\) as shown in Eq. 3.5,
Chapter 3: Authentication and Recovery of Images using Wavelet Transform

\[ q = (q_1, q_2, q_3, ..., q_M) \]  

where \( M = N^2/32 \). The first \( M \) coefficients are likely to be more significant. For an image of size 512 \( \times \) 512, only 8192 coefficients are selected. The element at first index (Top-Left corner of the DCT transformed image) DC component, which correspond to the average brightness, is not included while selecting the first \( M \) coefficients because of its exceeding high energy. We are not interested in authenticating the gray level of the image. If the DC component is included in recovery watermark generation, then the watermark will be noticeable.

- When the resultant coefficients of the previous step are embedded in the wavelet subbands, then the result in some wavelet values becomes much higher than the other belonging to the same neighbourhood [35]. Because wavelet coefficients contain certain spatial reference, and arise an unpleasant degradation all around the image. To avoid this affect, the quantized values are further scaled down and therefore vector \( q \) is processed based on a secret key \( k_2 \) using Eq. 3.6,

\[ q_{\text{Scaled}}(i) = q(i) \cdot \alpha \cdot \ln (i + 2 + r(i)) \]  

where \( \alpha \) is a strength factor and its value depends upon the watermarked image quality (the value of \( \alpha \) is slightly higher than 1). \( r \) is the shift parameter ranging from \(-0.5 \) to \( 0.5 \) and is randomly generated by using the PRNG (Pseudo-Random Number Generator). \( q(i) \) indicates the DCT coefficient at the \( i \)th position within a zigzag scan and \( q_{\text{Scaled}}(i) \) is its corresponding scaled coefficient.

- Each coefficient obtained from Eq. 3.6, are positioned in the subbands highlighted as dark gray in Figure 3.1. For an \( N \times N \) image, each selected subband has a size of \( N/4 \times N/4 \). Since two subbands are selected for embedding, thus the available positions are \( N^2/8 \) that is four times larger than \( M \) number of coefficients. Due to this fact, the coefficients are then quadruplicated and obtained a new vector described in Eq. 3.7.

\[ V = (q_1, q_2, q_3, ..., q_M, q_1, q_2, q_3, ..., q_M, q_1, q_2, q_3, ..., q_M, q_1, q_2, q_3, ..., q_M) \]  

Before embedding the quadruplicated vector of high-scaled DCT coefficients \( V \), the permutation process is applied based on a secret key \( k_3 \). The four replicas are placed in different positions of the two subbands. This is very important, because whenever the watermarked image is manipulated, the proposed system will be
quite confident that not all the replicas are removed / attacked. The scrambled quadruplicated coefficients are finally obtained and are considered as the recovery watermark $\hat{W}_2$ as shown in Eq. 3.8.

$$\hat{W}_2 = V_{permuted} \tag{3.8}$$

Both the authentication watermarks $\hat{W}_1$ and recovery watermark $\hat{W}_2$ are generated and are ready to be inserted in the independent wavelet subbands of the original image.

### 3.1.3 Embedding of Authentication Watermark

The following steps are used to embed the authentication watermark $\hat{W}_1$ in its respective subband.

- The $LL3$ subband is selected for embedding $\hat{W}_1$ because of the features like: they will be well preserved after common image processing operations like JPEG compression; they ensure the perceptual similarity because of their larger capacity. For the sake of semi-fragility, $LL3$ subband is selected for embedding. The embedding method used in [115] is employed for embedding the authentication watermark. Applying the first level IntWT decomposition on an $N \times N$ image, the approximation and details of the image are obtained. The detailed subbands, horizontal $HL1$ and vertical $LH1$ are further decomposed to obtained the subbands used for embedding the recovery watermark and $LL3$ is obtained by decomposing the approximation ($LL1$) two times further. The embedding subbands are highlighted in Figure 3.1 (Light gray for embedding the authentication watermark $\hat{W}_1$ and Dark gray for embedding recovery watermark $\hat{W}_2$).

- Let $LFB(a)$ denote the five least significant bits of $a$ and $LFB(a, b)$ represent the substitution of $b$ for the five least significant bits of $a$. The two choices ‘11000’ and ‘01000’ representing ‘1’ and ‘0’ respectively, are selected from the distance diagram shown in Figure 3.2. These two choices are selected according to the quality of the watermarked image. If the least five significant bits of wavelet coefficient are simply replaced by the given choices, then the amplitude changes from $-7$ to $+24$ when ‘1’ is embedded and from $-23$ to $+8$ when ‘0’ is embedded. Keeping the performance of invisibility and robustness, the following embedding method has been proposed[115].

When $\hat{W}_1(i,j) = 0$ then Eq. 3.9 is adopted,
\[ f^*(i,j) = \begin{cases} 
\text{LFB}(f(i,j) - 01000, 11000), & \text{if } \text{LFB}(f(i,j)) \leq 01000 \\
\text{LFB}(f(i,j), 11000), & \text{otherwise} 
\end{cases} \]

and when \( \hat{W}_1(i,j) = 1 \) then Eq. 3.10 is adopted,

\[ f^*(i,j) = \begin{cases} 
\text{LFB}(f(i,j) + 10000, 01000), & \text{if } \text{LFB}(f(i,j)) \leq 11000 \\
\text{LFB}(f(i,j), 01000), & \text{otherwise} 
\end{cases} \]

where \( f(i,j) \) is the IntWT coefficient in \( LL_3 \) subband before embedding, \( f^*(i,j) \) is the IntWT coefficient after embedding. With such embedding, the amplitude of the coefficients changes from \(-15\) to \(+16\). The two choices ‘11000’ and ‘01000’ are used to represent the bits ‘1’ and ‘0’ respectively. On the authentication side, only fifth least significant bit is examined of these choices.

### 3.1.4 Embedding of Recovery Watermark

The recovery watermark \( \hat{W}_2 \) is substituted in two details, \( HL_2 \) and \( LH_2 \) subbands highlighted as dark gray in Figure 3.1. Size of \( \hat{W}_2 \) and two subbands, \( HL_2 \) and \( LH_2 \) are equal. The two chosen selected subbands grant a good trade-off between invisibility and robustness of the watermark \( \hat{W}_2 \).

Finally, inverse IntWT is performed on the modified wavelet coefficients and authenticated image is obtained. Both the watermarks show the semi-fragility and have the ability to accept the friendly manipulations and reject the malicious attacks. Although two watermarks are embedded, the to-be checked image and the original image appear very similar in quality point of view. This is because the embedding procedure of authentication watermark \( \hat{W}_1 \) and extreme scaling down of recovery watermark \( \hat{W}_2 \) make it possible to be invisible for the human eye.

Peak Signal to Noise Ratio (PSNR) is used to measure the induced distortion caused by the watermark [116]. PSNR in decibels (dB) for 8-bit gray scale image of size \( M \times N \) is computed using Eq. 2.1 and Eq. 2.2 [117].

The proposed technique uses the parameterize IntWT which is the fast approach of DWT. Meerwald *et al* [118] have proposed the parameterized IntWT for the first time in 2001. However, their scheme is based on conventional DWT. Lifting scheme is an effective method to improve the processing speed of DWT. On the other hand, IntWT allows constructing lossless wavelet transforms and through lifting scheme, such IntWT can be constructed. Consequently, the proposed technique is based on a
novel idea of using IntWT with parameters for the development of secure semi-fragile watermarking for both image authentication as well as recovery. In this chapter, the Daubechies Wavelet Transform has been utilized. The watermarks generation and embedding procedure is given in Algorithm 3.1.

![Figure 3.1 Block diagram for generation and embedding the watermarks](image)

![Figure 3.2 Distance diagram](image)
Algorithm 3.1 Generation and embedding of $\tilde{W}_1$ and $\tilde{W}_2$

**Generation and Embedding of $\tilde{W}_1$**

1. $W = w(i,j)$ \(1 \leq i \leq X\text{ and } 1 \leq j \leq Y\) \(w \leftarrow Random(i)\) // $w$ is the binary matrix.

2. $P_{Rand} = R_n(i,j)$ \(// P_{Rand}\) is the random binary matrix generated through secret key.

3. $\tilde{W}_1 = W \oplus P_{Rand}$ \(// \tilde{W}_1\) is the final watermark ready for embedding in the respective subbands.

4. For embedding $\tilde{W}_1$, Eq. 3.9 and 3.10 are used. // Embedding by using Eq. 3.9 and 3.10, the amplitude of the coefficients changes from $-15$ to $+16$. The two choices ‘11000’ and ‘01000’ are used to represent the bits ‘1’ and ‘0’ respectively.

---

**Generating and Embedding of $\tilde{W}_2$**

1. $d \leftarrow dct2(LL1)$ \(//\) Full-Frame DCT is applied to the approximation, $LL1$ of cover image.

2. $q = quantization(d)$ \(//\) The DCT coefficients are quantized to reduce their obtrusiveness.

3. $z = zigzag(q)$ and $q_M = Select(M\text{ coefficients from } z)$ \(//\) The DCT quantized coefficients are then ordered in as vector through zigzag scanning. After zigzag scanning, the first $M$ significant coefficients are selected and the DC component is discarded.

4. $q_{Scaled}(i) = q_M(i) \cdot a \cdot \ln (i + 2 + r(i))$ \(//\) the quantized values are further scaled down to reduce the affect of DCT quantized values.

5. $V = quadruplication(q_{Scaled})\text{ and } \tilde{W}_2 = V_{Permuted}$ \(//\) The scaled values are then quadruplicated and permuted using the secret key to obtain the recovery watermark $\tilde{W}_2$
# 3.2 Integrity Verification

In the integrity verification phase, the watermarked image undergoes a procedure, where the embedded watermarks ($\hat{W}_1$ and $\hat{W}_2$) are extracted. Both the watermarks are extracted from their respective subbands i.e. the authentication watermark $\hat{W}_1$ is extracted from the $LL3$ subband and the recovery watermark $\hat{W}_2$ is extracted from $HL2$ and $LH2$ subbands. All the private keys and wavelet type(s) i.e. Haar, Daubechies etc are supposed to be available at the receiving end.

## 3.2.1 Extraction of Authentication Watermark

The following procedure is used to extract the watermark $\hat{W}_1'$, which is used for authentication.

Applying a one-level IntWT on $N \times N$ watermarked image, the approximation subband $LL1$ is decomposed two times further and $LL3$ subband, supposed to contain the informative data (authentication watermark $\hat{W}_1$) is selected as shown in Figure 3.3. Let $\hat{W}_1''$ is supposed to the extracted watermark and $LFB_{5th}(a)$ be the fifth least significant bit of $a$ then according to Eq. 3.11,

$$\hat{W}_1''(i, j) = \begin{cases} 1, & LFB_{5th}(f'(i, j)) = 0, \\ 0, & LFB_{5th}(f'(i, j)) = 1, \end{cases} \quad (1 \leq i \leq X, 1 \leq j \leq Y) \quad 3.11$$

where $f'(i, j)$ is the wavelet coefficient of the watermarked image. As the authentication watermark has been processed before embedding, therefore the extracted bits are processed again in the similar way to obtain the ultimate watermark. A Pseudo-Random Number Matrix (PRNM) $P_{Rand}$ is generated using the same key i.e. $k_1$ and taking an exclusive OR with the extracted binary image $\hat{W}_1''(i, j)$ as given in Eq. 3.12,

$$\hat{W}_1'(i, j) = \hat{W}_1''(i, j) \oplus P_{Rand}(i, j) \quad (1 \leq i \leq X, 1 \leq j \leq Y) \quad 3.12$$

where $\hat{W}_1'(i, j)$ is the ultimate extracted authentication watermark. The difference in the extracted authentication watermark $\hat{W}_1'(i, j)$ and original authentication watermark $\hat{W}_1(i, j)$ is expressed in Eq. 3.13. The original authentication watermark is generated again in a similar way as generated before embedding.

$$D(i, j) = |\hat{W}_1(i, j) - \hat{W}_1'(i, j)| \quad (1 \leq i \leq X, 1 \leq j \leq Y) \quad 3.13$$
If $D(i, j) = 1$, then it means that the original and the extracted watermark bits are different (white pixel in difference image) and represents mark extraction error. On the other hand, for $D(i, j) = 0$, it represents the correct mark extraction (black pixel in difference image). Figure 3.11 show detailed experimental results about error pixels and their strength in the difference image in section 3.5.

### 3.2.2 Extraction of Recovery Watermark

To obtain the estimated image $\hat{H}_2$, the reverse procedure of the watermark generation and embedding is performed. The respective subbands are selected i.e. the dark gray colour subbands shown in Figure 3.3 to obtain the estimated image. These subbands are supposed to contain the highly compressed version of the original image. The decompression technique (reverse procedure of compression technique) is applied to the information stored in the selected subbands. The detailed procedure to do so is given as follows:

The watermark stored in the selected subbands $HL2$ and $LH2$ is inversely permuted based on the same secret key $k_3$, which is used in generating recovery watermark. Let $P_{permuted}'$ represents the inverse scrambled coefficients. The estimate of the hidden coefficients is obtained by averaging all the four copies of each extracted coefficient. Because the coefficients were quadruplicated before embedding, through which the attacker cannot remove all the four replicas of the coefficient. A unique set of information (recovery watermark) is obtained having size $M = N^2/32$. By using, the private key $k_2$ used in recovery watermark generation, which allows rightly seeding the PRNG and consequently correctly invert the scaling operation performed during the watermark generation. The inverse scaling operation is given in Eq. 3.14.

$$q_{Reconstructed}(i) = q_{Extracted}(i) \cdot \frac{1}{\alpha} \cdot \frac{1}{\ln(i + 2 + r(i))} \tag{3.14}$$

The anti-zigzag scanning is then performed to put all the coefficients in their correct positions i.e. the vector is converted back to the two dimensional matrix. After applying anti-zigzag scanning, missing elements are replaced by zeros to obtain the DCT version of the reference image. The DC component is replaced by 128. DC component is the average value of the image and changing the DC value will effect the brightness of the image. Thus, it does not matter that what the value of DC
component is, because it is not important to authenticate the gray level of the image. In general, for *Lena* image, replacing the missing DC component by 128 gives best result regarding brightness of the image [35].

It has been analyzed practically, while testing different images including DC component affects the recovered recovery watermark. Also, it has been observed that by including the DC component in recovery watermark, as opposed to [35] where it is approximated by the value of 128, the PSNR value of the recovered image degrades to 38.52db from 38.84db for the test image *Lena*.

The values obtained from Eq. 3.14 are then weighed back to the standard quantization matrix given in Eq. 3.4. The reconstructed matrix is divided into 64 blocks and multiplied by corresponding value in quantization matrix. Inverse DCT is then applied to the resultant matrix to obtain the approximation of the original image of size $N/2 \times N/2$. The quality of the extracted watermark (image approximation) is very satisfactory and permits one to make good comparison with the original image. There is no automatic way of detecting the manipulations if only the recovery watermark is used. This is done by using an authentication watermark, which is used for accurate authenticity of the image. Experimental results (see section 3.5) shows that the tampering regions are accurately detected and localized. The localization of the tampered areas is shown on the difference image $D$ as well as on the extracted authentication watermark $\hat{W}_1$.

Figure 3.4, shows the analysis of the extracted authentication watermark $\hat{W}_1$, which is used for accurate authentication in the proposed technique. If somebody tampers the image, then authentication watermark $\hat{W}_1$ will detect it and in addition, is able to show whether the attack is malicious or incidental. The detailed discussion is given in section 3.3.

Figure 3.5, shows the analysis of the recovery watermark $\hat{W}_2$ (recovered image), which can localize the tampering but not accurately. The image can be recovered after an attack. However, the quality of the recovered image degrades as the strength of the attack increases and vice versa. See section 3.4 for detail discussion. The algorithm for extracting the authentication and recovery watermark is given in Algorithm 3.2.
Chapter 3: Authentication and Recovery of Images using Wavelet Transform

Figure 3.3 Block diagram of watermarks extraction

Figure 3.4 Analysis of the extracted authentication watermark $\hat{W}_1$
Chapter 3: Authentication and Recovery of Images using Wavelet Transform

### 3.3 Tamper Detection

The difference between the original and the extracted watermark is obtained by using Eq. 3.13. If $D(i,j) = 1$ then it means that there exists a difference between the corresponding pixels of original and extracted authentication watermarks. In other words, $D(i,j) = 0$ i.e. black pixel in the difference image corresponds to correctness while $D(i,j) = 1$ i.e. white pixel in the difference image corresponds to erroneous pixel. The proposed approach accurately locates the tampered regions and distinguishes the malicious and incidental manipulations. Two types of parameters are used to evaluate the strength of the manipulations. One is *Dense pixel* and other is *Sparse Pixel*. These parameters can be defined in the following manner:

**Dense Pixel**: For a mark error pixel in the difference image, it is *Dense Pixel* if at least one of its eight neighbor pixels is an error pixel and *Sparse Pixel* otherwise [36]. The parameters defined in Eq. 3.19 and Eq. 3.20, are utilized to calculate the manipulation strength and due to these parameters the incidental and malicious manipulations can be differentiated. In the case of incidental manipulation, most of the watermark error pixels are isolated on the difference image or on the extracted watermark. On the other hand, in case of malicious manipulation, most of the watermark error pixels are gathered together with high probability.

On the basis of above assumption, on the difference image or extracted authentication watermark $\hat{W}_i^{tr}$, the tampered area can be judged. The degree of tampering and the difference between the incidental and malicious manipulations can also be found. This is subjective evaluation. For objective evaluation, the following quantitative method can be employed. Dense and sparse areas are defined in Eq. 3.15 and Eq. 3.16. Eq. 3.17 and Eq. 3.18 define the total number of pixels and the total number of erroneous pixels in the difference image respectively.

$$D_{area} = \text{Total Number of Dense Pixels in Difference image}$$  \hspace{1cm} 3.15

$$S_{area} = \text{Total Number of Sparse Pixels in Difference image}$$  \hspace{1cm} 3.16

$$Area = \text{Total Number of pixels of Difference image}$$  \hspace{1cm} 3.17

$$Total_{area} = Dense\ Area + Sparse\ Area$$  \hspace{1cm} 3.18
Now, the tampering behaviour can be judged that the manipulations are either incidental or malicious. The parameters defined in Eq. 3.19 and Eq. 3.20, are used to differentiate the manipulations.

$$\Delta = \frac{D_{area}}{S_{area}}$$

$$\xi = \frac{Total_{area}}{Area}$$

If $\xi = 0$, it means that there not a single erroneous bit in the difference image and this implies that the image not being tampered, either incidental or malicious. If $\xi > 0$ and $\Delta < \beta$, then the watermarked image is manipulated incidentally i.e. the error pixels are isolated on the difference image. $\beta$ is the threshold selected carefully and generally, the value of $\beta$ is fixed between 0.5 and 1. Finally, if $\Delta \geq \beta$, then the tampering is malicious i.e. the error pixels are gathered together with high probability.

In summary, if the difference image has sparse pixels i.e. $\Delta < \beta$, then the manipulation is considered incidental like JPEG compression, file format change etc. Otherwise, in a case of dense pixels i.e. $\Delta \geq \beta$, the image is maliciously manipulated like cut/copy-past, manipulated geometrically or may be compressed beyond the scope (very high compression etc).

Let suppose the watermarked image if compressed i.e. 80% and further attacked maliciously i.e. a small line is drawn on it. It is undoubtedly a malicious manipulation but by the objective evaluation, the technique will analyze both of the attack. The difference image will show the isolated pixels on the overall difference image and also will show a line which will indicate that the image is tampered maliciously. To improve accuracy of tamper detection, the method that combined subjective evaluation with objective evaluation is recommended. By subjective evaluation, accurate result was given when encountering mild incidental attack and then malicious tampered. With respect to acute incidental attack and then malicious attack, it is the objective evaluation’s work.

### 3.4 Recovery of Approximated Image

Although in [75, 88], the authors use the authentication technique for securing the image but they do not allow one to get the estimated/approximated image back. The
compressed version of host image itself is embedded in this technique and is usually referred to self-recovery technique [35]. The original image is decomposed using integer wavelet transform and then its low level is highly compressed by using standard quantization matrix.

**Algorithm 3.2 Extraction of authentication and recovery watermarks**

**Extraction of $\hat{W}_1$**

1. Select the respective subbands to extract $\hat{W}_1$ // Highlighted as light gray color in Figure 5.2.

2. $\hat{W}_1''(i, j) = \begin{cases} 1, & LFB_{5th}(f''(i, j)) = 0, \\ 0, & LFB_{5th}(f''(i, j)) = 1, \\ (1 \leq i \leq X, 1 \leq j \leq Y) \end{cases}$ // Fifth LSB are extracted to obtain the authentication watermark.

3. $\hat{W}_1'(i, j) = \hat{W}_1''(i, j) \oplus P_{Rand}(i, j)$ // The random binary matrix is generated using the same key as in embedding side and the taking XOR with the extracted watermark. $\hat{W}_1'$ is the ultimate extracted authentication watermark.

4. $D(i, j) = |\hat{W}_1'(i, j) - \hat{W}_1''(i, j)|$ // $D$ is the difference in the original and extracted authentication watermarks. This equation shows that the watermarked image is tampered or not.

**Extraction of $\hat{W}_2$**

1. Select the subbands for extracting recovery watermark $\hat{W}_2$ // Light dark gray color subbands in Figure 5.2

2. $p' = \text{Inverse\_P}(\hat{W}_2')$ // $\hat{W}_2'$ is the extracted information from the respective subbands and $p'$ is the inverse permuted sequence

3. $avg = \text{average}(p')$ // The values are averaged because they are quadruplicated in generation of recovery watermark. The $M$ coefficients are obtained.

4. The $M$ coefficients are then scaled up using eq. 3.14. // The $M$ coefficients are scaled up to obtain the $M$ quantized DCT coefficients. All the missing elements, while selecting the $M$ coefficients, are set to zero and the DC component is set to 128.

5. The scaled values are then inversely quantized and inverse DCT is applied // By inverse quantization and then inverse DCT, the resultant coefficients are similar to the approximation values of the original image. Hence, the recovery watermark is obtained.
On authentication/verification side, the reverse of recovery watermark generation process is applied to obtain the recovered image. The experimental results show that the image can be recovered even after malicious manipulation. The degradation of the recovered image increases when the strength of manipulation(s) increases. This matter is analysed in Table 3.1 where the qualities of the recovered image are shown with different quality factors of JPEG compression.

Table 3.1 shows the PSNR values of the recovery watermark recovered after the watermarked image has been JPEG compressed with respect to the recovery watermark extracted, when the watermarked image has not undergone any compression. For each PSNR value, the corresponding JPEG quality factor is shown. It should also be noted that when quality factor goes down to 70% (the watermarked image is highly compressed), the PSNR is still satisfactory for applications in which data compaction is more important than image detail reconstruction.

The technique proposed in this chapter utilizes the image quality measure PSNR for two purposes. First, the strength of the watermark is analyzed, which shows that how much the watermarked image degrades by embedding the watermark. The PSNR varies according to the watermark strength i.e. PSNR and watermark strength are inversely proportional. Second is to check the degradation of the recovered image when the watermarked image is compressed using different quality factors.
### Table 3.1 PSNR of extracted recovery watermark after JPEG compression of the watermarked image, with respect to the recovery watermark extracted from a non-compressed watermarked image. PSNR values are given with the corresponding JPEG quality factors

<table>
<thead>
<tr>
<th>Quality Factor</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>35.21</td>
</tr>
<tr>
<td>95</td>
<td>33.23</td>
</tr>
<tr>
<td>90</td>
<td>30.53</td>
</tr>
<tr>
<td>85</td>
<td>28.54</td>
</tr>
<tr>
<td>80</td>
<td>27.01</td>
</tr>
<tr>
<td>75</td>
<td>25.50</td>
</tr>
<tr>
<td>70</td>
<td>24.89</td>
</tr>
<tr>
<td>60</td>
<td>22.34</td>
</tr>
</tbody>
</table>

### 3.5 Experimental Results

The algorithms described in this thesis are implemented in MATLAB environment. The technique discussed in this chapter is tested on Lena and Cameraman images of size $512 \times 512$ with different formats like bmp, tiff, etc.

Miss November 1972 in a playboy magazine, Lena is the classical image in the field of image processing [119]. This image contains the smooth region on her shoulder and contrast with the background makes it difficult to embed the watermark without visible distortion. On the other hand, it is easy to modify the textured region of the feathers. The Cameraman image has the large and smooth background and it is difficult to make a trade-off between the capacity and imperceptibility.

The PSNR of the watermarked images is $38 - 40$ db which are quite reasonable. Figure 3.6 shows the original and watermarked images of Lena and a authentication watermark, which is embedded in $LL3$ subband. The signature of the author is used as authentication watermark $\hat{W}_1$. The authenticated image that enclose two watermarks i.e. authentication watermark $\hat{W}_1$ and recovery watermark $\hat{W}_2$ is presented for permitting to carry out the visibility assessment as it can be observed that the quality is well preserved and appear no visual artifacts. The experimental results related to the video surveillance and remote sensing applications will be discussed in Chapter 4 in detail.

The extracted authentication watermark and Recovered image without any attack are shown in Figure 3.7. When the watermarked image has not undergone any attack, is presented. The recovery watermark is recovered through the extraction process.
discussed in section 3.2.3. The recovered image presents the same characteristic of the original image approximation, i.e. \( LL_1 \) of the original image and its quality is sufficient. The difference image, having black colour shows that the image is being tampered neither incidentally nor maliciously. The number of dense pixels and sparse pixel in the difference image is zero.

![Figure 3.6(a) Original Image (b) Authentication watermark (c) Watermarked image](image)

The \textit{Lena} image is tampered maliciously on the eye (cut/past). The image is recovered but the tampered areas are located in the extracted authentication watermark and in the difference image as shown in Figure 3.8. The difference image is the difference in the original authentication watermark and extracted authentication watermark. In proposed scheme, we see that the quality of the recovered image degrades. If the watermarked image is highly tampered or compressed, then it degrades the recovered image accordingly.

The algorithm is applied to the \textit{Cameraman} image with the file format .tiff. Figure 3.9 shows the original, watermarked, recovered and the difference images. There is no white pixel on difference image, which shows that the image is being tampered neither incidentally nor maliciously. The images used for testing the algorithm shown in Figure 3.6 to Figure 3.8 have the .bmp file format.

A result obtained after tampering the image maliciously but not visible to human eye, is shown in Figure 3.10 where the Lena image is tampered maliciously on feathers. The technique proposed in [35] can recover the approximation of the image where the recovered image degrades according to the strength of manipulation. However, their scheme is unable to locate the tampered areas, as there is no automatic system to detect the tampering regions. They just differentiate the watermarked image and the extracted recovery watermark where only the visible tampered regions can be
detected. The proposed scheme recovers the approximated image by using the recovery watermark $\tilde{W}_2$ and locates the tampered area accurately by using $\tilde{W}_1$.

Figure 3.7 (a) Original image (b) Watermarked image (c) Extracted authentication watermark (d) Difference image. Difference is taken between the original and extracted authentication watermarks.

Figure 3.8 (a) Watermarked image tampered maliciously (b) Recovered recovery watermark (c) Extracted authentication watermark where the tampered regions are detected (d) The difference image locations of tampered regions
In [35], it can be observed that, if the image is invisibly tampered or highly compressed then users are left to judge the authenticity of the image by themselves. However, the scheme proposed in this chapter is able to detect and locate the tampered regions accurately as shown in Figure 3.10.

![Figure 3.9](image1.png)

Figure 3.9(a) Original image of cameraman (b) The watermarked image, PSNR 38.2db (c) Recovered recovery watermark (d) Extracted authentication watermark (e) Difference between the original and extracted authentication watermarks

![Figure 3.10](image2.png)

Figure 3.10(a) Watermarked image which has been manipulated maliciously on feathers (b) recovered recovery watermark which has been degraded a little bit (c) Difference between the original and extracted authentication watermark where the tampered regions are accurately detected and localized

Figure 3.11 shows both the recovered image approximation and extracted authentication watermark from the watermarked image, which has been compressed by applying the lossy JPEG compression with different quality factors (QF). The proposed scheme can allow the degree of robustness towards JPEG compression as low as 70%. The quality less than 70% should be considered malicious manipulation and the recovered image degrades seriously. The cameraman.tif image is compressed by using the quality factors 95%, 90%, 80%, 75% and 70%. When the quality factor is 70% or above then the difference image contains the error pixels, which are not dense pixels, but sparse pixels. It means that the proposed scheme has the ability to survive reasonable compression. If quality factor less than 70% is used, then the number of dense pixels increases in the difference image because the error
pixels are gathered together with high probability and it shows that the tampering is malicious. Figure 3.11 present the result in two rows where the first row represent the recovered image approximations after applying JPEG compression based on different quality factors. The second row reveals the difference images between the original and extracted authentication watermarks. In case of compressing the watermarked image, it is very important that the proposed approach is able to offer a degree of robustness against JPEG compression. The dots (erroneous pixels) shown on the difference images are the sparse pixels that the image is incidentally tampered. It can be seen that when the watermarked image is compressed beyond the defined quality factor i.e. below 70% then the number of dense pixels starts appearing in the difference image (last case shown in Figure 3.11). For the quality factor below 70%, the recovered image degrades heavily and the proposed approach considers such type of JPEG compression as a malicious manipulation.

![Figure 3.11](image.png)

**Figure 3.11** The first row shows the recovered images. These images degrade according to the JPEG compression quality factor. The second row shows the differences after compressing the watermarked image with quality factors 90%, 80%, 75%, 70% and 60% respectively.

When the wrong keys are provided to the receiver then the extracted authentication watermark is shown in Figure 3.12. The recovery watermark is corrupted in the similar way i.e. the unauthorized person will try to reveal the reference image (recovery watermark) but he/she will obtain a useless noise like recovery watermark. It means that if an unauthorized person who do not know the secret keys then he will be unable to extract the correct information. In Figure 3.12, the first one is the watermarked image available at the receiving end and the other image is the extracted authentication watermark by using the wrong key.
Figure 3.12 (a) Watermarked image (b) Extracted authentication watermark by using a wrong key

Figure 3.13 illustrates the performance of the proposed technique against JPEG compression. The *Lena* image with different sizes i.e. $512 \times 512$ and $1024 \times 1024$ are watermarked, then JPEG compression is applied to both the watermarked images with different quality factors. In both cases, it is observed that up to quality factor 70%, the number of sparse pixels increases gradually, and the numbers of dense pixels are almost negligible. This shows that the attack is incidental. However, when the quality factor decreases i.e. below 70%, the number of dense pixels increases sharply, which shows that the attack is malicious (last case of Figure 3.11).

![Figure 3.13 Error pixels versus strength of JPEG compression](image)

Performance comparison of the proposed approach with other schemes is given in Table 3.2. Experimental results of the proposed approach are analyzed and compared with the previous techniques Piva *et al* [35] and Wu *et al* [36]. The salient features are given in Table 3.2. It can be observed that the method proposed in [35] can recover the estimated image and localize the tampered regions. However, this approach cannot accurately localize the invisible attacks (intentional or unintentional), i.e. it has
a crude way to tamper localization and does not use any automatic method for localizing the tampered areas. It just computes the difference between the watermarked image and the recovered image, which is based on threshold. Therefore, if the tampering is visible, i.e. cut/copy and paste, then it will give white pixels. Otherwise, black pixels will be shown on the difference image. This issue has been discussed in experimental results i.e. see Figure 3.10. On the other hand, Wu et al [36] cannot recover the approximated image, but there is an automatic system to detect/localize the tampered regions accurately. In this approach, any type of attack can be detected, either it is malicious or incidental, visible or invisible.

<table>
<thead>
<tr>
<th>Features</th>
<th>Piva et al [35]</th>
<th>Wu et al [36]</th>
<th>Proposed Scheme</th>
<th>Supporting Results and Discussions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recovering recovery watermark</td>
<td>Recover the estimated image</td>
<td>Cannot recover the estimated image</td>
<td>Recover the estimated Image</td>
<td>Fig. 3.7</td>
</tr>
<tr>
<td>Accurate authenticity</td>
<td>Cannot accurately authenticate the image in case of invisible attack</td>
<td>Accurately authenticate the image</td>
<td>Accurately authenticates the image</td>
<td>Fig. 3.10</td>
</tr>
<tr>
<td>Compression</td>
<td>Accept JPEG compression</td>
<td>Accept JPEG compression</td>
<td>Accept JPEG compression</td>
<td>Fig. 3.11</td>
</tr>
<tr>
<td>Localization</td>
<td>Cannot localize invisible tampered regions</td>
<td>Localize the tampered regions accurately</td>
<td>Localize the tampered regions accurately</td>
<td>Fig. 3.10</td>
</tr>
<tr>
<td>Security</td>
<td>More Secure: uses two secret keys, for scaling and scrambling</td>
<td>Not substantial security, only one secret key is used during authentication watermark generation</td>
<td>More Secure: uses three secret keys, for scaling, permutation, and authentication watermark generation</td>
<td>Section 3.2.1</td>
</tr>
</tbody>
</table>

Unlike to the previous two approaches shown in Table 3.2, the proposed approach not only recovers the estimated image as in [35], but can also detect any type of tampering as in [36]. If the image is compressed, which is an incidental attack (Figure 3.11), or it is tampered maliciously; either visible (Figure 3.8) or invisible (Figure 3.10), the proposed approach is able to detect and localize all such attacks accurately.
3.6 Chapter Summary

The proposed scheme is able to accurately authenticate the image, distinguish the malicious and incidental attacks, and recovers a good estimate of original contents. In this chapter, the proposed technique sacrifices a little bit on the visual quality of the watermarked image, which is approximately 38db to 40db for different images, but still the quality is satisfactory. The technique is highly secured because of inclusion of three private keys at various stages of watermark generation. The proposed scheme also shows efficient authentication for a smallest scale transformation on an image. Embedding of two watermarks makes the proposed scheme more efficient for accurate detection of tampered region and recovery of estimated image. Invisible tamper detection is another authentication attribute achieved in our proposed semi-fragile secured watermarking scheme. The following contribution has been made towards image authentication and recovery:

- The proposed algorithm is able to make the content secure towards attacks applied in either spatial domain or transform domain.
- Multiple keys are used to make the watermarks safe.
- It is able to recover the exact version of the compressed host image with low overhead of the watermarks.
- Unlike to the traditional block-based approaches, tamper detection and localization is accurate and can differentiate between the malicious and incidental manipulations.
- JPEG lossy compression can be accepted up to a fair extant. Compression survival is flexible and can be set according to the requirement of the application.
- Computational complexity has been reduces by using the lifting scheme (IntWT).
- The architecture of the proposed watermarking system is open. One can use both or any of the watermark according to the application.
Chapter 4

Secure Image Authentication and Recovery using Multiple Watermarks

In this chapter, the dual watermarking technique to secure digital content is presented. Two semi-fragile watermarks are embedded with the choice of user to select both or one of them according to the requirement of application. Similar to the technique discussed in Chapter 3, this technique embed both of the watermarks in the wavelet subbands. Three major limitations of the technique proposed in Chapter 3 have been circumvented in this chapter. The limitations are about the security of the entire wavelet subbands, fragility against counterfeiting/collage attack, and imperceptibility where the visual property has been sacrificed. The counterfeiting attack is also known as Holliman and Memon attack because it is introduced by Holliman and Memon [34]. Security and imperceptibility are considered the most basic requirement in the area of image authentication. Thus, securing the content with reasonable imperceptibility is the challenge of today’s research in the field of multimedia security.

The technique proposed in Chapter 3 has the ability to authenticate the image accurately and can recover the image approximation but at the cost of security and imperceptibility. Furthermore, collage attack cannot be detected by the said technique. Only three wavelet subbands are utilized for embedding the two semi-fragile watermarks. However, subbands other than the selected subbands are not protected. The attacker is free to modify these areas to alter the watermarked image without any trace of detection. Therefore, the security aspect of the scheme has been enhanced and it is hard for the attacker to substitute successfully the concerned subbands with another image. However, manipulations to degrade the quality of the watermarked image are possible.
Very recently, security consideration is designated as one of the most important feature of watermarking system. In fact, watermarking security, for either copyright protection or authentication, is currently one of the unsolved issues in the research community. No one can say that watermarking is more secure than cryptography. Most of the watermarking algorithms have to rely on cryptographic tools for security enhancement, such as hash digest generation, bit pseudo-random permutation \([78, 111, 112, 120]\) and the technique proposed in Chapter 3 is also based on the digest (recovery watermark) generation and use the pseudo-random numbers base on PRNG.

### 4.1 The Proposed Algorithm

The watermarking model discussed in this section make it possible to secure the image against any attack applied in either spatial or transform domain without any sacrifice on the imperceptibility. The image is secured by involving all the wavelet subbands directly or indirectly and the technique has the capability to detect the counterfeiting attack as well. The imperceptibility has been improved by reducing the strength of the authentication watermark without any effect on detection resolution.

Authentication watermark \(\mathcal{W}_1\) and the recovery watermark \(\mathcal{W}_2\) are embedded in the wavelet subbands of the to-be authenticated image. The block diagram for watermarks generation and embedding process is shown in Figure 4.1.

#### 4.1.1 Generation and Embedding of Authentication Watermark

The binary random matrix \(\mathbf{w}\) of size \(LL1/4\) is generated by using a secret key \(k_1\), where \(LL1\) is the approximation subband after taking one level wavelet decomposition of the original image. The binary matrix \(\mathbf{w}\) of other different sizes i.e. \(LL1/2\), \(LL1\) or \(LL1/4\) can also be used. However, with \(\mathbf{w}\) of size \(LL1/4\), the payload of the authentication watermark is reduced. If the size decreases further, then surely the technique will sacrifice on robustness of the watermark. Thus, \(\mathbf{w}\) of size \(LL1/4\) has been utilized in this technique making a trade-off between imperceptibility and robustness. In \([30, 31, 121]\), the authors suggested that an image coefficient used for watermark embedding should be correlated with other positions in the watermarked image to protect against collage attack. An attacker cannot simply use a collage attack due to the correlation because watermark bits are dependent on the image being watermarked. The image features selected for correlation must not belong to the
image features to be watermarked. The \( LL1 \) subband has been selected for correlation with \( w \) because watermarking \( LL1 \) coefficients results in the original image being severely affected and it is thereby free for correlation without introducing any conflict [121]. The wavelet coefficients of wavelet subbands are denoted as \( f_{\text{subband}} \). The four adjacent coefficients \( (f_{LL1}(m,n)), (f_{LL1}(m,n+1)), (f_{LL1}(m+1,n)) \) and \( (f_{LL1}(m+1,n+1)) \) are averaged and quantized using Eq. 4.1. Then the quantized values are XORed with \( w \) to obtain the final watermark \( \tilde{W}_1 \).

\[
\tilde{W}_1 = \frac{\text{avg}}{Q_p} \mod 2 \oplus w \tag{4.1}
\]

where \( \text{avg} \) is the average value of four adjacent pixels in \( LL1 \), \( \oplus \) is the exclusive OR, and \( Q_p \) is the quantization parameter that determines the sensitivity against collage attack. The wavelet coefficients are then modified according to the correlated watermark bits. The image is decomposed further and the light gray color subbands \( (LL2, VV2, DD2, LL3, HH3, DD3, \) and \( DD1) \) are selected for embedding the authentication watermark, as shown in Figure 4.1. The respective subbands are used by taking into consideration the trade-off between robustness and imperceptibility. Selection of subbands for embedding the watermark is application dependent. Because, embedding in the approximation subband enhances the robustness attribute but the capability of localization of the tampered regions is reduced. On the other hand, embedding in the detail subbands reduces the watermark robustness, but increases the accuracy of tamper localization.

The coefficients of the selected subbands (highlighted as light gray color in Figure 4.1) are concatenated into a single sequence \( S \) i.e. coefficients with the same coordinates in the selected subbands are continuously adjacent in the new sequence. The sequence \( S \) is scrambled randomly bases on a secret key \( k_2 \) and then divided into groups \( g \). In every group, one bit of \( \tilde{W}_1 \) is embedded that controls all coefficients of the respective group. The group size \( g \) determines the watermark payload, i.e. larger group increases imperceptibility and vice versa but this will not sacrifice the detection resolution.

The watermark \( \tilde{W}_1 \) is embedded by modifying the weighted means of the groups. Embedding watermark bits in the groups is more robust than the individual coefficients [122]. The weighted mean of every group \( g \) is defined in Eq.4.2,
where $f_j(i)$ is the $ith$ element in the $jth$ group and $p$ is the bipolar random sequence with uniform distribution $p \in \{1, -1\}$. The weighted mean of the groups are further quantized using Eq. 4.3 and Eq. 4.4,

$$
\bar{g}_j = \left\lfloor \frac{\bar{g}_j}{Q} \right\rfloor \cdot Q + \Delta_j
$$

$$
\text{Quan}(\bar{g}_j) = \begin{cases} 
0 & \text{if } \left\lfloor \frac{\bar{g}_j}{Q} \right\rfloor \text{ is even} \\
1 & \text{if } \left\lfloor \frac{\bar{g}_j}{Q} \right\rfloor \text{ is odd}
\end{cases}
$$

where $Q$ is the quantization step and $\Delta_j$ is the quantization residue, while $\text{Quan}$ denotes the binary lattice.

Figure 4.1 Embedding process
The bit pattern $W_1(j)$ is embedded by modifying the weighted mean $\tilde{g}_j$ so that $Quan(\tilde{g}_j)$ becomes equal to $W_1(j)$. The first $G$ number of bits are selected for embedding ($G$ is the number of all groups), because one bit is embedded in each group. The modification of the weighted mean is performed using Eq. 4.5,

$$\tilde{g}_j = \begin{cases} 
\lceil \tilde{g}_j + Q/2 \rceil \cdot Q + \frac{Q}{2}, & \text{if } Quan\left(\tilde{g}_j + \frac{Q}{2}\right) = W_1(j) \\
\lceil \tilde{g}_j + Q/2 \rceil \cdot Q - \frac{Q}{2}, & \text{if } Quan\left(\tilde{g}_j + \frac{Q}{2}\right) \neq W_1(j)
\end{cases}$$

where $\tilde{g}'_j$ is the expected weighted mean of the $j$th group. For example, the weighted means and the expected weighted means of ten groups are given in Table 4.1.

<table>
<thead>
<tr>
<th>$\tilde{g}_j$</th>
<th>-28</th>
<th>-76</th>
<th>-98</th>
<th>10</th>
<th>-91</th>
<th>-37</th>
<th>-19</th>
<th>31</th>
<th>-8</th>
<th>33</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{g}'_j$</td>
<td>-34.5</td>
<td>-82.5</td>
<td>-102.5</td>
<td>12.5</td>
<td>-97.5</td>
<td>-40.5</td>
<td>-22.5</td>
<td>32.5</td>
<td>-10.5</td>
<td>32.5</td>
</tr>
</tbody>
</table>

Those wavelet coefficients, which have the maximum magnitude (suitable wavelet coefficient) in a group, are then modified by embedding the authentication watermark. This causes less noticeable artifacts. In the high frequency subbands, the coefficients with high magnitude will thus represent more textured contents at the corresponding spatial location, and vice versa [88]. Thus, selecting the suitable wavelet coefficients for watermark embedding will produce good results as compare the low textured image. This issue will be discussed in detail in the experimental results. Figure 4.2 shows an example of largest coefficient in a group used for embedding the watermark bit. The larger coefficient offers more robustness at the corresponding spatial location.

The random permutation ensures that there will be at least one large coefficient in each group. Let $\partial$ be the difference in expected and original weighted means calculated using Eq. 4.6.

$$\partial_j = \tilde{g}'_j - \tilde{g}_j$$

The largest coefficient is updated by using Eq. 4.7.

$$f_{j,max}^* = f_{j,max} + p_t \cdot \text{sign}(f_{j,max}) \cdot \partial_j$$

where $f_{j,max}$ is the largest coefficient in $j$th group. If the sign of the coefficient $f_{j,max}$ changes by applying Eq. 4.7 then the coefficient $f_{j,max}$ is left as it is. The value of $\partial$
is then modified according to Eq. 4.8 and the second largest coefficient is updated by applying Eq. 4.7 with \( \partial_{j,\text{residue}} \) instead of using \( \partial_j \). \( \partial_{j,\text{residue}} \) is calculated in Eq. 4.8.

\[
\partial_{j,\text{residue}} = \text{sign}(\partial_j) \cdot |\partial_j| - |f_j,\text{max}|
\]

![Figure 4.2 Selection of coefficient with high magnitude for embedding the watermark bit. The suitable coefficient in the group is indicated. Group size, \( g_s = 35 \).](image)

The process is repeated accordingly until \( \partial_{j,\text{residue}} = 0 \). In [88], the author assigns zero to the sign changing coefficient after applying Eq. 4.7 and select the second largest coefficient to update. Such type of practice made serious modifications and degrades the quality of the watermarked image.

The entire authentication watermark \( \hat{W}_1 \) is embedded in the suitable features (maximum coefficient in each group) of the permuted sequence \( S \). The sequence is then inversely permuted by using the same key \( k_2 \). The coefficients are then placed into the original positions in their respective subbands.

### 4.1.2 Generation and Embedding of Recovery Watermark

The recovery watermark generation and embedding is outlined in Figure 4.1. The recovery watermark \( \hat{W}_2 \) is generated by using the modified version of JPEG compression and then embedded in the dark gray color subbands as shown in Figure 4.1. Generation and embedding of recovery watermark \( \hat{W}_2 \) is similar to the technique proposed in Chapter 3. For detail discussion, see section 3.1.2. The generation and embedding procedure of both the watermarks is given Algorithm 4.1.
## Algorithm 4.1 Generation and embedding procedure of the watermarks $\tilde{W}_1$ and $\tilde{W}_2$

### Generation and Embedding of $\tilde{W}_1$

1. for $i \leftarrow 1$ to $(\text{size } LL1)/4)$
   
   \( w \leftarrow \text{Random}(i) \)
   
   // Other different sizes i.e. $LL1/2$, $LL1$ or $LL1/4$ can also be used. However, with $w$ of size $LL1/4$, the payload of the authentication watermark is reduced and hence making a trade-off between imperceptibility and robustness. $w$ is the binary sequence base on secret key.

2. $\tilde{W}_1 \leftarrow w \oplus LL1$
   
   // XOR each bit with average of four adjacent $LL1$ coefficients.
   
   // $\tilde{W}_1$ is the final watermark ready for embedding in the respective subbands.

3. Select the subbands for $\tilde{W}_1$
   
   // Highlighted as dark gray in Figure 5.1.

4. Concatenate coefficients of the selected subbands, $f_j(i)$ in an array $S$ and generate groups $g$.
   
   // The coefficients of the selected subbands are concatenated into a single sequence i.e. coefficients with the same coordinates are continuously adjacent in the new sequence. One bit will be embedded in each group.

5. $\overline{g}_j \leftarrow \sum_{i=0}^{g-1} p_i |f_j(i)|$
   
   // Computation of weighted mean of each group $g$. $f_j(i)$ is the $ith$ element in the $jth$ group.

6. Quan $\leftarrow \left( \frac{\pi}{2} \right) \cdot Q + \Delta \leftarrow [0,1]$
   
   // $Q$ is quanta and $\Delta$ is quantized residue. The resultants are the binary values, which have to be embedded in their respective subbands.

7. $\overline{g}_j' \leftarrow \text{Modified}(\tilde{W}_1)$
   
   // $\overline{g}_j'$ is expected weighted mean of $jth$ group.

8. $f_{j,\text{max}}^* \leftarrow f_{j,\text{max}}$
   
   // modification of coefficients $f_{j,\text{max}}$ in such a way that $\tilde{W}_1 = \text{Quan}$. (Both are binary sequences).

### Generation and Embedding of $\tilde{W}_2$

1. $d \leftarrow \text{dct2}(LL1)$
   
   // Full-Frame DCT is applied to the approximation, $LL1$ of cover image.

2. $q = \text{quantization}(d)$
   
   // The DCT coefficients are quantized to reduce their obtrusiveness.

3. $z = \text{zigzag}(q)$ and

   \[ q_M = \text{Select}(M \text{ coefficients from } z) \]

   // The DCT quantized coefficients are then ordered in as vector through zigzag scanning.
   
   After zigzag scanning, the first $M$ significant coefficients are selected and the DC component is discarded.

4. $q_{\text{scaled}}(i) = q_M(i) \cdot \alpha \cdot \ln (i + 2 + r(i))$
   
   // the quantized values are further scaled down to reduce the affect of DCT quantized values.

5. $V = \text{quadruplication } (q_{\text{scaled}})$ and

   \[ \tilde{W}_2 = V_{\text{permuted}} \]

   //The scaled values are then quadruplicated and permuted using the secret key to obtain the recovery watermark $\tilde{W}_2$.
4.1.3 Watermarks Extraction and Integrity Verification

The procedure for extracting the authentication and recovery watermarks is given in Algorithm 4.2. The extracted authentication watermark $\hat{W}_1'$ is compared with the original watermark $\hat{W}_1$ for authenticity. The key and the wavelet types are supposed to be available at the receiving side. The original authentication watermark is generated in the similar way as in the embedding side.

After decomposing the watermarked image, the respective subbands are (Light gray color subbands in Figure 4.1) selected, where the required information are embedded. The coefficients are concatenated and permuted by using the same keys, which are used in embedding process. The resultant sequence $S'$ is then divided into groups $g$ of same size. The weighted mean of each group is calculated and watermark bits are extracted by quantizing using Eq. 4.3 and Eq. 4.4. Extraction of watermark appears in Eq. 4.9,

$$W'_1 = \text{Quan}(\hat{g}_j')$$

where $\hat{g}_j'$ is the recalculated weighted mean of $jth$ group and $\hat{W}_1'$ is the extracted watermark and then compared with the original watermark $\hat{W}_1$. If both match then the image is authentic, otherwise the image is tampered. The technique described in this chapter has the ability to exactly detect and localize the tampered areas instead of blocks. The tampered regions can be localized as follows:

If the embedded and the extracted watermarks are not similar then the corresponding group is considered to be unverified (tampered). At this stage, the exact tampered coefficient in the group cannot be determined. The sequence $S'$ is mapped back to their original positions. The entire unverified coefficients will be scattered in all those subbands where the authentication watermark is embedded as shown in Figure 4.3. The locations corresponding to the tampered regions will have high density. The dark areas show the tampered regions in Figure 4.3. This is because, when the sequence $S'$ is mapped back, the unverified coefficients converges together [88]. The other isolated coefficients belonging to the group which are considered unverified, are scattered sparsely in the subbands like random noise.

The matrix $U$ is constructed and $U(m,n)$ is considered as an unverified coefficient according to Eq. 4.10 and Eq. 4.11,
Chapter 4: Secure Image Authentication and Recovery using Multiple Watermarks

\[
U(m, n) = \begin{cases} 
0 & \text{if any of } LL2(m, n), VV2(m, n), DD2(m, n), LL3(m, n), \\
& HH3(m, n), DD3(m, n) \text{ and } d(m, n) \text{ is unverified} \\
1 & \text{otherwise}
\end{cases} \tag{4.10}
\]

\[
d(m, n) = \begin{cases} 
0 & \text{if any of } DD1(m, n), DD1(m, n + 1), DD1(m + 1, n), \\
& DD1(m + 1, n + 1) \text{ is unverified} \\
1 & \text{otherwise}
\end{cases} \tag{4.11}
\]

where \( DD1 \) is the detailed subband as shown in Figure 4.1.

When the matrix \( U \) is generated, black pixel i.e. “0” is considered as an error pixel, while white pixel i.e. “1” corresponds to correct pixels. Dense Pixel and Sparse Pixel are defined as:

An error pixel in \( U \) is Dense Pixel if one of its eight neighboring pixels is also an error pixel otherwise, it is a Sparse Pixel. If the matrix \( U \) has sparse pixels then the watermarked image is manipulated incidentally. On the other hand, in case of high number of dense pixels, the image is attacked maliciously. As shown in Fig. 4.3, the high density of unverified coefficients in \( U \) shows the tampered regions. The other sparse unverified coefficients like random noise are those coefficients, which are not tampered, but belong to the unverified groups.
Algorithm 4.2 Extraction procedure of watermarks $\hat{W}_1$ and $\hat{W}_2$

**Extraction of $\hat{W}_1$**

1. Select the respective subbands to extract $\hat{W}_1$  
   // Highlighted as light gray color in Figure 5.2.

2. Concatenate these extracted coefficients, $f'_j(i)$ in array and generate groups based on secret key.
   // $f'_j(i)$ is the $ith$ wavelet coefficient of $jth$ group of the watermarked image.

3. $\bar{g}'_j \leftarrow \sum_{i=0}^{g-1} p_i |f_j(i)|$
   // Computation of weighted mean of the groups.

4. $Quan \leftarrow (\bar{g}'_j/Q) \cdot Q \cdot \Delta \leftarrow [0, 1]$
   $\leftarrow W'_1$ (Extracted authentication watermark)
   // $Q$ is Quanta and $\Delta$ is quantized residue. After extracting the $\hat{W}_1$, image is checked for authenticity using the extracted watermark.

**Extraction of $\hat{W}_2$**

1. Select the subbands for extracting recovery watermark $\hat{W}_2$
   // Light dark gray color subbands in Figure 5.2

2. $p' = Inverse_P(\hat{W}_2')$
   // $\hat{W}_2'$ is the extracted information from the respective subbands and $p'$ is the inverse permuted sequence

3. $avg = average(p')$
   // The values are averaged because they are quadruplicated in generation of recovery watermark. The $M$ coefficients are obtained.

4. The $M$ coefficients are then scaled up using eq. 3.14.
   // The $M$ coefficients are scaled up to obtain the $M$ quantized DCT coefficients. All the missing elements, while selecting the $M$ coefficients, are set to zero and the DC component is set to 128.

5. The scaled values are then inversely quantized and inverse DCT is applied
   // By inverse quantization and then inverse DCT, the resultant coefficients are similar to the approximation values of the original image. Hence, the recovery watermark is obtained.
As a final step, the noise filter is applied to matrix $U$ so that the tampered regions can be easily picked out. If the attacker attacks on the transform domain of the watermarked image then it is not possible to localize because the coefficients in the transform domain have the one-to-many relationship to the image in spatial domain. However, the technique proposed in this chapter considers the watermarked image as un-authentic because all the wavelet subbands are protected.

### 4.2 Image Recovery

Although, many authentication techniques have been developed [75, 88], where the authors use the authentication watermark for securing the digital images but they do not permit one to recover the estimated/approximated image. The technique proposed in this chapter embeds the highly compressed version of host image itself and such approach usually referred to self-recovering technique [35]. On receiving side, the reverse process of recovery watermark generation is applied to obtain the recovered image even after manipulation either incidental or malicious. The degradation of the recovered image increases when the strength of manipulation(s) increases. The detail discussion is given in Chapter 3, section 3.4.

### 4.3 Experimental Results

All the gray scale images of size $512 \times 512$ with different formats have been chosen for the experiments. As discussed in the embedding process of the watermarks, the suitable locations have been chosen for watermarks embedding. Therefore, the visual quality of the watermarked images is much better compared to the previous technique discussed in Chapter 3. The imperceptibility depends on the two parameters, *Quanta* and *Group Size*. *Quanta* is the quantization factor that is fixed at the watermark embedding side. It determines the resistivity of the system against JPEG compression. By increasing *Quanta*, the survival against JPEG compression will be increased and vice versa. On the other hand, large group size will improve the imperceptibly and vice versa, because each group contain only one watermark bit. The detail discussion about these two parameters is given in next chapter.
Figure 4.4 shows the original, watermarked, recovered, and the difference images. The difference image is the bit-by-bit difference in the original and extracted authentication watermark. The visual quality of the watermarked image is relatively appreciable. The recovery of image approximation is not effected by selecting Quanta and Group Siz. Changing parameters of authentication watermark does not affect the recovering capability of the recovery watermark. Differentiation in different kind of attacks is carried out by using the authentication watermark as discussed in section 4.2. The experimental results in detail are given in Chapter 5.

Figure 4.5 presents the remote sensing application. A particular attention is focused on the tip of the minaret. The tip of minaret has been removed and the proposed system is able to detect the tampering location and recover the original content by using self-recovering approach.

Another case of image manipulation is presented in Figure 4.6 where the effective appearance of the image has been altered. The airplane has been moved and its original position is filled with the neighboring area. In this case, the proposed algorithm is able to extract the image approximation (recovery watermark) where the airplane occupies original position. Both the locations are clearly identified by the automatic manipulation detection. The location has been detected accurately to where the airplane is moved and the location is detected from where the airplane is moved.

![Figure 4.4](image-url)

**Figure 4.4** (a) Original image (b) Watermarked image (c) Recovered image approximation (d) Difference between original and extracted authentication watermarks
Chapter 4: Secure Image Authentication and Recovery using Multiple Watermarks

Figure 4.5 (a) Original Lichtenstein image (b) Tip of minaret has been removed (c) Detection of the tampered region (d) Recovered image approximation with a tiny degradation because the watermarked image has been manipulated

Figure 4.6 (a) Original London eye image (b) Object moving (airplane) from its original position (c) Detection of location where the object is added as well as from where the object is removed (d) Recovered image approximation which has been degraded a little bit because the watermarked image has been altered.
Table 4.2 lists the variations in term of PSNR against different Quanta and Group Sizes. The PSNR is determined empirically and one can see that when the size of the group increases, the PSNR increases. Similarly, by increasing Quanta the PSNR decreases. In addition, since the watermarks are embedded in high magnitude wavelet coefficients, the PSNR for highly textured images (i.e. baboon) is high. Figure 4.7 describe the PSNR values presented in Table 4.2, which presents the performance of the algorithm with various values of both the parameters. The algorithm produces good results for baboon image. Because, the watermarks especially the authentication watermark, are embedded in the suitable wavelet coefficients. This issue will be discussed in detail in the next chapter.

Performance comparison of the proposed technique with previous approaches [33] and [35] is presented in Table 4.3. The comparison has been made with respect to imperceptibility, security, tamper detection etc. The proposed technique modeled a secure watermarking technique producing good results compared to the existing approaches with imperceptibly issue being resolved. The collage attack can be detected by correlating the authentication watermark with the original image contents before embedding. Other salient features of the proposed technique are also given in Table 4.3.

<table>
<thead>
<tr>
<th>Group Size</th>
<th>Quanta</th>
<th>PSNR (Lena)</th>
<th>PSNR(Cameraman) (Cameraman)</th>
<th>PSNR (Baboon)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>30</td>
<td>34.75</td>
<td>34.43</td>
<td>35.17</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>36.23</td>
<td>36.01</td>
<td>37.83</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>37.21</td>
<td>36.97</td>
<td>40.19</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>37.87</td>
<td>37.20</td>
<td>41.75</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>34.51</td>
<td>34.19</td>
<td>34.96</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>35.21</td>
<td>34.65</td>
<td>37.75</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>36.86</td>
<td>36.37</td>
<td>39.28</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>37.01</td>
<td>35.81</td>
<td>40.23</td>
</tr>
<tr>
<td>10</td>
<td>30</td>
<td>33.41</td>
<td>33.01</td>
<td>33.49</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>35.44</td>
<td>34.97</td>
<td>36.12</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>36.58</td>
<td>35.74</td>
<td>37.15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>36.78</td>
<td>36.59</td>
<td>38.28</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td><strong>32.51</strong></td>
<td><strong>32.84</strong></td>
<td><strong>33.23</strong></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>35.23</td>
<td>34.92</td>
<td>33.58</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>36.30</td>
<td>35.72</td>
<td>35.07</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>36.60</td>
<td>36.14</td>
<td>36.54</td>
</tr>
</tbody>
</table>
Figure 4.7 PSNR against different Quanta values and group sizes on three different images, Lena, Cameraman, and Baboon images. The PSNR varies according to the values of both parameters.

Table 4.3 Prominent features and performance comparison of the proposed approach with previous approaches, Chamlawi et al [33] and Piva et al [35].

<table>
<thead>
<tr>
<th>Features</th>
<th>[33]</th>
<th>[35]</th>
<th>Proposed Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Watermark Payload</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>2. Watermark Security</td>
<td>Satisfactory</td>
<td>No</td>
<td>Highly Secure</td>
</tr>
<tr>
<td>3. Tamper Detection</td>
<td>Good</td>
<td>Satisfactory</td>
<td>Good</td>
</tr>
<tr>
<td>4. Localization</td>
<td>Accurate</td>
<td>Not Accurate</td>
<td>Highly Accurate</td>
</tr>
<tr>
<td>5. Imperceptibility (PSNR)</td>
<td>Reasonable</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>6. Survival against JPEG</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (User Control)</td>
</tr>
<tr>
<td>Compression</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Collage Attack Resiliency</td>
<td>No</td>
<td>No</td>
<td>Yes (Correlation of $W_1$)</td>
</tr>
<tr>
<td>8. Attack Classification</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
4.4 Analysis

The issue which effects the performance or relate to the proposed system is Group Size and Quanta. A trade-off has been made while embedding the watermarks in the proper wavelet subbands. Smaller size of the group increases the payload of the watermark and vice versa. On the other hand, the smaller Quanta decreases the robustness of the watermark and the larger Quanta leads to stronger robustness. The proposed technique is able to make the content secure towards attacks applied in both spatial and transform domain. Multiple keys are utilized in watermarks generation and embedding for security purposes. Correlation and permutation of the watermarks enhances the security aspects of the proposed approach. The architecture is open and one can use both or any of the watermark according to the requirement of the application. The correlation of the authentication watermark with the image being watermarked makes it able to handle collage/counterfeiting attack. The tampered regions are concisely determined instead of traditional block-based approaches.

4.5 Chapter Summary

Multiple semi-fragile watermarks are used to focus on both the authenticity and recovery of the image. It classifies the alterations and provides a value-added technique for secure and efficient authentication of digital images.

Each of the embedded watermarks strengthens the security aspect of each other. The processing speed is improved by using parameterized integer wavelet transform. Both of the watermarks are embedded in suitable wavelet coefficients, which increase the imperceptibility and have high potential in video surveillance and remote sensing applications. Similarly, random permutations and scrambling allow one to satisfy the security requirements of the watermarks.
Chapter 5

Image Authenticity with Exact Recovery

Digital watermarking is well-established research area in the field of multimedia security. Many watermarking techniques like fragile watermarking [10, 16, 20, 62, 63, 83, 123, 124], semi-fragile watermarking [24, 25, 32, 33, 67-70, 90], robust watermarking [8-10, 17, 54-56], and other watermarking techniques with recovering techniques [4, 16, 17, 41, 44, 74-76, 95] have been developed in this regard. The techniques described in previous chapters are also developing the semi-fragile watermarking approaches with recovery.

In this chapter, dual semi-fragile watermarking framework using integer transform based information embedding and extraction is presented. This scheme allows accurate authentication with high tamper localization capability and exact recovery of image approximation by using self-recovering method. The proposed technique has the ability to resist the friendly manipulations, better imperceptibility, and the capability of producing better quality recovered image. Security is enhanced by correlating the to-be-embedded watermark with the approximation subband of wavelet transform. Similarly, no free area being left for attacker in either spatial or transform domain. Particularly, the recovering approach is improved by introducing lossless compression and BCH coding of the IntDCT based low-pass version of the cover image itself. Alteration sensitivity has been improved compared to traditional block-based approaches.

Previously described approaches do not offer appreciable imperceptibility due to embedding of heavy digest (recovery watermark). Furthermore, they do not extend high quality recovering of image approximation. In this chapter, the said limitation has been circumvented by utilizing the Huffman and BCH coding in generating and embedding the recovery watermark. Similar to the techniques described in Chapter 3 and Chapter 4, dual watermarking technique is employed to make the image contents secure and recover the image approximation of the original quality.
In this work, two watermarks are embedded in appropriate wavelet coefficients. The first one called the authentication watermark, is generated by quantization technique and then correlated to the original image contents before embedding. The second one called recovery watermark, is computed through the properly modified version of the compressed original image. The authentication watermark helps in accurately detecting the manipulation, but not ensuring the recovery of estimated image. On the other hand, the recovery watermark helps the users to obtain the estimated image but leaves the users to judge the authenticity themselves. Thus, both watermarks in combination can lead to accurate authentication and recovery. As for as the security aspect is concerned, all the subbands are involved in either watermark generation or embedding with no free area being left for an attacker to attack. Strength of the watermarks is controlled in order to obtain the best result. Wavelet coefficients with high amplitude are selected based on their local characteristics and are properly modified while embedding both of the watermarks. In fact, in the high frequency subbands, the coefficients having high magnitude specify that the image have more textures at the corresponding spatial location. Thus, the selective and reduced alteration consequently improves the imperceptibility.

Many block-based watermarking techniques for tamper localization have been reported [29-32], where the image is divided into blocks and the watermark information is embedded into every block. The resolution of tamper detection is limited to the block size. To improve the localization resolution, a smaller block size implies high watermark payload, which degrades the image quality. On the other hand, large block size will affect the tamper detection resolution. Therefore, in addition to improving the security of the watermarking system, the challenge is how to increase the resolution of tamper detection by embedding less watermark information.

5.1 The Proposed Algorithm

In this section, the proposed dual semi-fragile watermarking scheme is discussed in detail; including watermark generation, embedding, and extraction. The block diagrams of watermarks generation, embedding and extraction are shown in Figure
5.1 and Figure 5.3 respectively. Authentication and recovery watermarks denoted as $\tilde{W}_1$ and $\tilde{W}_2$, are embedded in the wavelet subbands of the to-be-authenticated image.

### 5.1.1 Generation of Authentication Watermark

The authentication watermark $\tilde{W}_1$ is correlated to the image coefficients that deal with the counterfeiting/collage attack. Generation and embedding of authentication watermark is similar to the approach discussed in Chapter 4. The process of generating and embedding the authentication watermark in detail is given in section 4.1.1 and section 4.2 respectively. The results for image authentication and high tamper localization are given in detail in sections 5.3.1 and 5.3.3.

### 5.1.2 Generation of Recovery Watermark

Generation of recovery watermark $\tilde{W}_2$ is outlined in Figure 5.1. $\tilde{W}_2$ is the highly compressed version of the original image itself and is referred to self-recovery. After applying the IntWT, the approximation subband $LL1$ is selected for generation of $\tilde{W}_2$. On the receiving side, the extracted $\tilde{W}_2$ will be the image approximation. In this technique, the quality of the recovered image will be same as $LL1$ of the original image. A full-frame IntDCT is applied to the $LL1$ subband using MATLAB function given in Eq. 5.1,

$$d = idct2(LL1)$$  \hspace{1cm} 5.1

where $d$ is the IntDCT coefficient of the image approximation. Huffman coding which is a lossless compression algorithm, is then applied to encode the IntDCT coefficients using Eq. 5.2,

$$h = huffmanenco(d)$$  \hspace{1cm} 5.2

where $h$ is the compressed $LL1$ in binary form. Huffman coding is based on entropy of the input elements and is discussed in detail in section 2.7. The IntDCT is utilized because the Huffman coding works well on integer values produced by IntDCT instead of value produced by floating point DCT. The resultant of Huffman encoding is then XORed with the key-based sequence to ensure the security of the system, using Eq. 5.3,

$$w = h \oplus P_{Rand}$$  \hspace{1cm} 5.3
where $P_{\text{Rand}}$ is the pseudo-random number sequence based on a private key $k_1$ and $\oplus$ is the exclusive OR operator. XORed sequence is then passed through an error correcting code called BCH encoding using Eq. 5.4,

\[
bch = bchenc(o)\]

where $bch$ is the sequence of watermark bits with size greater than $w$, because it include the error correcting bits. The detail of BCH coding is discussed in section 2.10. Different pairs of BCH codes produce different size of the watermark. In the final experiments the BCH code, $(31, 16, 3)$ where 16 are actual bits, 31 are physical bits and 3 bits are to be corrected for each 31 bits, produces better results compared to the other BCH codes that are used throughout in the experiments. The codes $(31, 16, 3)$ and $(127, 64, 10)$ gives almost same result. A trade-off between the imperceptibility and error correction strength has been made while selecting the BCH codes. The behavior of the proposed algorithm varies according to the selection of BCH codes as shown in Table 5.1. *Lena* image is used as test image.

<table>
<thead>
<tr>
<th>BCH codes</th>
<th>PSNR</th>
<th>Survival level against JPEG compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(31, 16, 3)$</td>
<td>40.21</td>
<td>85%</td>
</tr>
<tr>
<td>$(31, 11, 5)$</td>
<td>34.18</td>
<td>80%</td>
</tr>
<tr>
<td>$(127, 64, 10)$</td>
<td>40.20</td>
<td>90%</td>
</tr>
<tr>
<td>$(127, 43, 14)$</td>
<td>33.27</td>
<td>80%</td>
</tr>
<tr>
<td>$(255, 187, 9)$</td>
<td>40.75</td>
<td>95%</td>
</tr>
<tr>
<td>$(255, 179, 10)$</td>
<td>37.43</td>
<td>85%</td>
</tr>
</tbody>
</table>

The final recovery watermark is then obtained by key-based permutation of BCH encoded sequence i.e. $\hat{W}_2 = p(bch)$ where $p$ is used for random permutation based on private key ($k_2$). The permuted BCH encoded values are then embedded in their respective subbands i.e. $HH2$ and $VV3$ as shown in Figure 5.1. Based on the size of the watermark, the embedding positions vary. In case of using $(31, 16, 3)$ BCH codes, every coefficient will be used for embedding two watermark bits. The third and fourth significant positions of the suitable coefficients in $HH2$ and $VV3$ are selected for embedding the watermark bits. The previously discussed algorithms replace the entire
coefficients of the respective subbands i.e. HH2 and VV3 by DCT compressed coefficients and as a result, the PSNR is just satisfactory. On the other hand, the technique proposed in this chapter, use lossless compression (Huffman coding) to generate the recovery watermark and therefore, the imperceptibility has been improved.

5.1.3 Embedding of Authentication and Recovery Watermarks
Both of the watermarks i.e. authentication watermark $\hat{W}_1$ and recovery watermark $\hat{W}_2$ are now ready to be embedded in the suitable features of the to-be-authenticated image. The watermarks are embedded in their respective subbands as shown in Figure 5.1.

![Figure 5.1 Watermarks generation and embedding diagram](image-url)
The embedding procedure of $\tilde{W}_1$ is similar to the process of generation and embedding of authentication watermark described in Chapter 4. The recovery watermark $\tilde{W}_2$ is embedded in third and fourth significant bits of the coefficients in $HH2$ and $VV3$. The inverse IntWT is performed to obtain the watermarked image.

The steps for generating and embedding both the watermarks i.e. $\tilde{W}_1$ and $\tilde{W}_2$ are described in Algorithm 5.1. The left column represents the steps to be taken in the watermarks generation and embedding and the right column shows the short descriptions correspondingly.

5.1.4 Extraction of Authentication Watermark and Integrity Verification

The procedure to extract the authentication watermark to check the integrity of the watermarked image is given in detail in section 4.2. Because the generation, embedding, and extraction of authentication watermark is similar to the technique proposed in Chapter 4. The respective subband, from where the authentication watermark is to be extracted is shown in Figure 5.2. If the attacker attacks even on the transform domain of the watermarked image, then it is not possible to localize because the coefficients in the transform domain have the one-to-many relationship to the image in spatial domain. However, the proposed scheme will consider the watermarked image as un-authentic, because all the subbands are secured. The attack in spatial domain can be detected and localized efficiently. Two parameters $Dense$ and the $Sparse$ pixels are used to differentiate the nature of different kind of attacks. The detail discussion about the extraction of authentication watermarks, authenticity checking, and localization of tampered regions is given in section 4.2. Authentication and localization steps are given in Algorithm 5.3.

5.2 Image Recovery

The image can be recovered in two broad ways: first-one is recovering of the regions/pixels, which have been tampered, and the other is to recover the whole image by using self-embedding algorithms, which is referred to self-recovery based technique. Similar to the previously discussed techniques, the technique proposed in this chapter use the second approach for recovering the image but with the difference that, this technique is able to recover the exact version of the original image approximation. Lossless compression (Huffman coding) method and BCH coding has
been utilized for generating and embedding the recovery watermark. On receiving side, the reverse procedure of generating and embedding the recovery watermark $\hat{W}_2$ is applied to obtain the original image approximation ($LL1$).

Unlike to the previously discussed techniques, where the quality of the recovered image degrades after any manipulation, the exact version of recovery watermark (recovery watermark) can be obtained after JPEG compression and even after tiny malicious manipulations. The experimental results demonstrate the said discussion in section 5.3.2. When the watermarked image is compressed beyond the scope, i.e. 85% $\hat{W}_2$ cannot be recovered. This is because lossless compression (Huffman coding technique) has been used, while generating the $\hat{W}_2$. Therefore, in case of an attack the erroneous bits are corrected by employing BCH coding. In this way, the proposed technique is able to recover the original compressed version of the host image after applying the JPEG compression (85%). The BCH code (31, 16, 3) is used in the experimental results, where 16 are the actual, and 31 are the physical bits while 3 bits can be corrected. The pair (31, 11, 5) is also used, which has the ability to survive up to 80% JPEG compression. However, by increasing the ratio of physical/original bits, the watermark payload becomes high and the PSNR degrades. Thus, the user has the flexibility to make a trade-off between image quality and watermark robustness. Different BCH codes with its performance are given in section 5.1.2.

The simple extraction procedure of recovery watermark $\hat{W}_2$ is given as follows:
The watermarked image is decomposed using IntWT and then the respective subbands containing the recovery information are selected (Light gray color in Figure 5.2). The extracted information is inversely permuted i.e. $p' = Inverse_P(\hat{W}_2')$, where $Inverse_P$ is the inverse permutation function based on based on key $k_2$. The BCH decoding function is then applied on $p'$ to obtain the decoded sequence using Eq. 5.5,

$$ex_{bch} = bchdeco(p') \quad 5.5$$

where $bchdeco$ is the BCH decoding function. Further, the resultant is XORed with the random number sequence based on private key $k_1$ using Eq. 5.6,

$$ex_s = P_{\text{rand}} \oplus ex_{bch} \quad 5.6$$

where $P_{\text{rand}}$ is random number sequence and $\oplus$ is exclusive OR operator.
Chapter 5: Image Authenticity with Exact Recovery

Algorithm 5.1 Generation and Embedding of $\hat{W}_1$ and $\hat{W}_2$

Generation and Embedding of $\hat{W}_1$

1. for $i \leftarrow 1$ to (size (LL1)/4),
   \[ w \leftarrow \text{Random}(i) \] // Other different sizes i.e. LL1/2, LL1 or LL1/4 can also be used. However, with $w$ of size LL1/4, the payload of the authentication watermark is reduced and hence making a trade-off between imperceptibility and robustness. $w$ is the binary sequence base on secret key.

2. $\hat{W}_1 \leftarrow w \oplus LL1$ // XOR each bit with average of four adjacent LL1 coefficients.
   // $\hat{W}_1$ is the final watermark ready for embedding in the respective subbands.

3. Select the subbands for $\hat{W}_1$ // Highlighted as dark gray in Figure 5.1.

4. Concatenate coefficients of the selected subbands, $f_j(i)$ in an array $S$ and generate groups. // The coefficients of the selected subbands are concatenated into a single sequence i.e. coefficients with the same coordinates are continuously adjacent in the new sequence. One bit will be embedded in each group.

5. $\bar{g}_j \leftarrow \sum_{i=0}^{q-1} p_i |f_j(i)|$ // Computation of weighted mean of each group $g.f_j(i)$ is the $ith$ element in the $jth$ group.

6. $\text{Quan} \leftarrow \left( \frac{\bar{g}}{Q} \right) \cdot Q + \Delta \leftarrow [0, 1]$ // $Q$ is quanta and $\Delta$ is quantized residue. The resultants are the binary values, which have to be embedded in their respective subbands.

7. $\bar{g}_j' \leftarrow \text{Modified}(\hat{W}_1)$ // $\bar{g}_j'$ is expected weighted mean of $jth$ group.

8. $f_{j,max}' \leftarrow f_{j,max}$ // modification of coefficients $f_{j,max}$ in such a way that $\hat{W}_1 = \text{Quan}$. (Both are binary sequences).

Generation and Embedding $\hat{W}_2$

1. $d \leftarrow \text{idct2}(LL1)$ // Full-Frame IntDCT is applied to the approximation, $LL1$ of cover image.

2. $h = \text{huffmanenco}(d)$ // Lossless compression (Huffman coding) is used to perform image compression.

3. $w = h \oplus P_{\text{Rand}}$ // Taking and exclusive OR of $h$ with key based random sequence $P_{\text{Rand}}$.

4. $bch = \text{bchenco}(w)$ // BCH encoding of XORed Huffman coded binary sequence with different pairs $(31, 16, 3)$, $(31, 11, 5)$, $(255, 187, 9)$, $(255, 179, 10)$ BCH.

5. $(HH2, VV3) \leftarrow \hat{W}_2$ Key based permutation // $\hat{W}_2$ is embedded in $HH2$ and $VV3$ subbands. // Permutation function is then applied to the BCH encoded values for distributing the malicious attack throughout the recovery watermark on authentication side.
The sequence $ex_s$ is decompressed by using Eq. 5.7.

$$ex_h = huffman(ex_s)$$  \hspace{1cm} 5.7$$

The inverse IntDCT is then applied using Eq. 5.8 to obtain the image approximation of size $M/2 \times N/2$, where $M \times N$ is the size of the original image,

$$f = inverse_idct(ex_h)$$  \hspace{1cm} 5.8$$

where $f$ is the required approximated image. The quality will be the same as the original image approximation even if the image is manipulated by using JPEG.
compression or any other tiny malicious manipulation. The main steps used to extract both the watermarks are described in Algorithm 5.2.

Algorithm 5.2 Extraction of $\hat{W}_1$ and $\hat{W}_2$

**Extraction of $\hat{W}_1$**

1. Select the respective subbands to extract $\hat{W}_1$  
   // Highlighted as dark gray color in Figure 5.2.

2. Concatenate these extracted coefficients, $f_j(i)$ in array and generate groups based on secret key.  
   // $f_j(i)$ is the $ith$ wavelet coefficient of $jth$ group of the watermarked image.

3. $\bar{g}'_j \leftarrow \sum_{i=0}^{g_{x}-1} p_i |f_j(i)|$  
   // Computation of weighted mean of the groups.

4. $Quan \leftarrow (\bar{g}'_j/Q) \cdot Q \cdot \Delta \leftarrow [0, 1]$  
   $\leftarrow W'_1$ (Extracted authentication watermark)  
   // $Q$ is Quanta and $\Delta$ is quantized residue. After extracting the $\hat{W}_1$, image is checked for authenticity using the extracted watermark.

**Extraction of $\hat{W}_2$**

1. Select the subbands for extracting recovery watermark $\hat{W}_2$  
   // Light gray color subbands in Figure 5.2

2. $p' = \text{Inverse}_P(\hat{W}_2')$  
   // $\hat{W}_2'$ is the extracted information from the respective subbands and $p'$ is the inverse permuted sequence

3. $ex\_bch = \text{bcsdeco}(p')$  
   // The BCH decoding of the $p'$, and the size of the resultant bits are reduced to the original size.

4. $ex\_s = P_{\text{Rand}} \oplus ex\_bch$  
   // The exclusive OR is performed based on a secret key.

5. $ex\_h = \text{huffmandeco}(ex\_s)$  
   // The $\text{huffmandeco}$ function provides the original IntDCT coefficients.

6. $f = \text{inverse}\_idct2(ex\_h)$  
   // The Inverse IntDCT is then applied to obtain the image approximation in original form
Algorithm 5.3 Authentication and localization of the proposed watermarking technique

**Authentication**

1. Compare $\hat{W}_1$ and $\hat{W}'_1$ // Original and extracted authentication watermarks $\hat{W}_1$ and $\hat{W}'_1$ are described in Algorithms 5.1 and 5.2 respectively.

2. if $\hat{W}_1 == \hat{W}'_1$ then the watermarked image is authentic otherwise tampered //authentication watermark generation and extraction are given in Algorithms 5.1 and 5.2 respectively.

3. *Dense* and *Sparse* pixels are calculated to check the behavior of the manipulation // *Dense* and *Sparse* are error pixels. These parameters are used to differentiate the intentional and unintentional manipulations.

4. Large number of *Dense* pixels ← Intentional Large number of *Sparse* pixels ← Unintentional // Number of *Dense* pixels increased, if the image is tampered maliciously while increasing in *Sparse* represent the unintentional manipulation. If the image is not tampered then *Dense* = *Sparse* = 0

**Localization**

1. If an element $i$ in $g$ is an error pixel then $g ←$ unverified // $i$ is the wavelet coefficient in the corresponding group $g$.

2. $i$ maps back to original position // The original position in the sequence. The entire unverified coefficients will be scattered in all those subbands where $\hat{W}_1$ is embedded.

3. Tampered($i$) ← Converge // The tampered coefficients converges. See Figure 4.3.

4. Verified($i$) ← Scattered // Verified ($i$) are verified coefficients belong to the unverified groups. See Figure 4.3.

5. $V(i) ←$ Tampered($i$) // All the tampered coefficients are assigned to matrix $V$.

6. $U ←$ filter($V$) // The scattered tampered coefficients are filtered and the location against the tampered regions have high density. See Figure 4.3.
The keys and the wavelet type used in generating and embedding the watermarks are supposed to be available with the watermarked image at the receiving end.

5.3 Experimental Results

A set of images are chosen for experiments, which comprises 10 images of size $512 \times 512$. Figure 5.3 shows the test images being watermarked. The parameters, $Quanta = 20$ and $Group\ Size = 12$ are kept fixed for all the test images. The visual qualities of the watermarked images are quite satisfactory because the proper wavelet coefficients are used for watermarks embedding.

5.3.1 Analysis of Authentication Watermark

The experimental results are analyzed with different scenarios while computing the proposed technique by using two contrast parameters i.e. $Group\ Size$ and $Quanta$. Figure 5.4 shows the quality metric based on PSNR, for different $Group\ Size$ and $Quanta$ while embedding the authentication watermark. By increasing the size of the group, the watermark payload becomes low and vice versa. Consequently, imperceptibility increases and robustness decreases. On the other hand, by increasing $quanta$, imperceptibility decreases and robustness increases. Thus, a trade-off has been made while selecting the $Group\ Size$ and $quanta$.

![Test images](image-url)
Figure 5.4 PSNR with different Quanta and Group Size (From top line: $g = 30, 20, 10, 5$). Lena image is used as test image.

Figure 5.5 illustrates the original image and the watermarked images obtained by applying the proposed watermarking scheme using different Quanta and Group Size. Difference in PSNR is not visually noticeable due to small size of images. However, it is evident from the numerical values in Figure 5.4. In the pair $(x - y)$, $x$ describe the size of the group and $y$ describe the size of the Quanta. These parameters can be selected based on requirement of the application. For example, in case of video surveillance and/or remote sensing applications, small size of the group and big Quanta can be used such as $5 - 20$, because by self-recovering process with such parameters, the requirements of the particular application can be efficiently fulfilled.

### 5.3.2 Effect of JPEG Compression

Figure 5.6 shows the JPEG compressed (85%) watermarked (Lena and Couple images) and the recovered images. It also shows the difference in $\hat{W}_1$ and $\hat{W}'_1$. The usefulness of the approach presented in this chapter is that, both the watermarks are semi-fragile with low payload and have tolerance towards JPEG compression. Figure 5.6(c) shows that the difference between $\hat{W}_1$ and $\hat{W}'_1$. The authentication watermark $\hat{W}_1$ checks the authenticity and proves that the image is not being tampered
maliciously. On the other hand, using the recovery watermark, the approximation of the image can be recovered similar to the embedded recovery watermark in the to-be-authenticated image. It can be observed that proposed approach can obtain the exact version of the compressed host image even when the watermarked image is attacked with JPEG lossy compression as shown in Figure 5.6.

![Figure 5.5 Original image, (b ~ j) Watermarked images with different Quanta and Group Size, 5 – 5, 5 – 10, 10 – 10, 10 – 30, 20 – 5, 20 – 10, 20 – 30, 30 – 5, 30 – 20](image)

When the size of Quanta increases, the survival against JPEG compression increases i.e. the number of affected/error pixels decreases and vice versa. In the Figure 5.7, one can see that up to 75% JPEG lossy compression, the numbers of dense pixels are very low and thus predict that the attack is incidental. However, when watermarked image is compressed more heavily i.e. 70% and onwards, then the number of dense pixels increases which shows that the image is attacked maliciously. When the algorithm is applied to the highly textured image like Baboon image, then the survival against JPEG lossy compression increases. This is probably because the watermarks are embedded in relatively high magnitude IntWT coefficients which are not that much effected in JPEG compression of the fairly texture dominated images. Thus, the algorithm even shows improved attack prediction for images having less smooth regions.

Figure 5.8 shows that the watermarked images, which are compressed by JPEG, are indicated as being acceptably manipulated. The authentication watermark \( \hat{W}_1 \) checks the authenticity and performs the detection of error pixels. The detected error pixels are almost in the smooth regions of the Lena image. The watermarks are
embedded in the suitable wavelet coefficients and the trade-off has been made between the imperceptibility and robustness. Therefore, the JPEG compression effect will be large on the image having major smooth regions. The erroneous pixels against different quality factors (QF) are shown and the empirical determination of the effected pixels after JPEG compression with different quality factors are shown. However, when a same quality factor is applied to the Baboon image then the number of affected pixels decreases. This is because the Baboon image is highly textured image. Thus, one can see that the image with major smooth regions will contain the large number of affected pixels on the extracted authenticated watermark and vice versa. The compression beyond the scope i.e. 70%, are considered the malicious attack as the number of Dense pixels increases.

Figure 5.6(a) JPEG compressed watermarked images, (b) Recovered approximation images, (c) Difference images, the difference have been taken between the extracted and the original authentication watermarks.

Figure 5.9 illustrates the variations in PSNR against different Quanta and Group size. The PSNR are determined empirically. One can see that when the Group Size increases, the PSNR increase, while increasing Quanta, PSNR decreases and vice versa. Moreover, as the watermarks are embedded in high magnitude wavelet coefficients of a group, the PSNR for highly textured images i.e. Baboon image is high. The high magnitude coefficients correspond to the high texture regions in spatial
domain. The embedding strength of the actual watermarks for the proposed scheme is lower than the approaches proposed in previous chapters.

![Graph](image)

**Figure 5.7** Number of erroneous pixels (Dense and Sparse pixels) versus JPEG quantization factors. Lena image is used as test image

(a) (b) (c)

(d) (e) (f)

**Figure 5.8** Erroneous pixels, (a–c, Lena Image), QF = 90%, 80%, 70%, (d–f, Baboon Image), QF = 90%, 80%, 70%
Figure 5.10 demonstrates bit error rate (BER) against different JPEG quantization factors. The comparison is performed with Kundur’s method [16] and Liu’s method [88]. All the approaches are subjected to JPEG compression with different quality factors. We observe that the number of erroneous bits is low for the proposed approach until a high JPEG compression of 30%. The reason for this is that the suitable image features have been selected for embedding the watermark bits.

In Table 5.2, it can be seen that the number of Dense and Sparse pixels after applying JPEG compression with different quality factors for Lena and Cameraman images are almost same. This is because they have roughly same amount of smooth regions. However, the Baboon image gives better results, as this image is more textured compared to the Lena and Cameraman images. Thus, the images with high textured regions in major area can accept the high JPEG compression. Since the parameters; Dense and Sparse pixels have been used for checking the strength and behavior of the attack (incidental or malicious). These parameters are normally used to check the strength of compression that how much one can compress the
watermarked image. When the image is compressed beyond the defined level, then the number of Dense pixels becomes much higher as discussed in Chapter 4, section 4.2.

After applying JPEG compression, the numbers of Dense and Sparse pixels are given in the Figure 5.7. It shows that if the JPEG lossy compression ratio increases more than 75%, the number of Dense pixels increases and the algorithm will consider it as malicious attack. The proposed approach survives reasonable JPEG lossy compression. Note that, in case of recovering the recovery watermark, it cannot be recovered beyond the 85% for BCH layer (31, 16, 3) used in the experiments of the proposed algorithm.

5.3.3 Malicious Tampering and its Localization

Figure 5.11 shows the watermarked image has been tampered maliciously. The Lena image is tampered on the right top and the building is replaced by the background color in the Cameraman image. The difference, Figure 11 (b) shows that the images are tampered maliciously. Malicious modifications in the watermarked image will affect the image recovery i.e. malicious tampering will modify the recovery watermark bits. The algorithm has a strong capability to localize the tampered regions.
instead of traditional approaches [32, 40, 42], where only blocks are located. The approach is able to detect every erroneous pixel instead of erroneous block because image features are highly secured. The comparison analysis of the proposed approach is carried out with Li et al [30] with respect to the resolution of tamper localization as shown in Figure 11 (c). In [30], a trade-off has been made between the tamper localization, security and watermark embedding distortion. The resolution of tamper localization varies according to the watermark payload. The tamper localization capability is improved by reducing the watermark payload and vice versa. As can be observed from Figure 11 (b), the proposed approach has more effective tamper localization resolution (No expansion of altered region as compared to Li et al [30], Figure 11 (c)).

Table 5.2 Number of dense and sparse pixels using different frequency content based images. Corresponding JPEG quality factor (QF) is given.

<table>
<thead>
<tr>
<th>Quanta</th>
<th>QF</th>
<th>Dense</th>
<th>Sparse</th>
<th>Dense</th>
<th>Sparse</th>
<th>Dense</th>
<th>Sparse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q = 30</td>
<td>60</td>
<td>3014</td>
<td>765</td>
<td>517</td>
<td>1646</td>
<td>23</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>2743</td>
<td>987</td>
<td>313</td>
<td>1609</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>1732</td>
<td>1029</td>
<td>170</td>
<td>1524</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>454</td>
<td>1310</td>
<td>24</td>
<td>1340</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>79</td>
<td>829</td>
<td>15</td>
<td>1115</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>23</td>
<td>584</td>
<td>0</td>
<td>818</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>0</td>
<td>212</td>
<td>0</td>
<td>506</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>0</td>
<td>89</td>
<td>0</td>
<td>199</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>26</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Q = 20</td>
<td>60</td>
<td>3312</td>
<td>471</td>
<td>560</td>
<td>1688</td>
<td>34</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>2987</td>
<td>762</td>
<td>431</td>
<td>1635</td>
<td>12</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>1723</td>
<td>1241</td>
<td>297</td>
<td>1583</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>562</td>
<td>1421</td>
<td>89</td>
<td>1399</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>123</td>
<td>1285</td>
<td>36</td>
<td>1120</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>23</td>
<td>756</td>
<td>15</td>
<td>830</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>2</td>
<td>412</td>
<td>0</td>
<td>517</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>0</td>
<td>178</td>
<td>0</td>
<td>265</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0</td>
<td>37</td>
<td>0</td>
<td>147</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In case of malicious attack, the recovery of the image approximation is impossible, because in Huffman coding (Lossless compression) method, if one of the watermark bit is erroneous, then the decompression of the watermark bits to the original image is impossible. Thus, the recovery is not possible after the manipulating of the watermarked image given in Figure 5.11.
5.3.4 Potential Aspects for Remote Sensing and Video Surveillance

The scheme has also been designed for video surveillance applications. Figure 5.12 show that when the effective appearance of the image is altered, then this approach has the ability to detect the alterations. The airplane has been moved from its original position and the proposed algorithm detects both the locations from where the airplane is deleted as well as the location to where the airplane is added. The image used in Figure 5.12 has been taken by the author of this thesis in London during his research visit in 2008 at Warwick University UK. It should be noted that the object is deleted from its own position and copied nearby. After deleting the object, the empty area has been replaced with color taken from a close area. Since the recovery watermark has the property of semi-fragility, therefore the small malicious manipulations can be recovered. This is because the erroneous bits can be corrected by using the error correcting code i.e. BCH coding. The error bits are distributed because of inverse permutation and the BCH decoder has the ability to correct the erroneous bits. By increasing the strength of malicious manipulations, the technique cannot correct the large number of error bits and thus it is not possible to obtain the same embedded recovery watermark back.

Figure 5.13 presents the remote sensing application. The image is manipulated on personal computer using image-editing software. A particular attention is focused on the tip of the minaret. The tip of minaret is removed. When the edited image is presented to the proposed watermark detector, the edited areas are detected and tampered regions are marked in the difference image i.e. Figure 5.12 (c). Effectiveness of the proposed approach has also been tested for counterfeiting attack.

In Figure 5.14, the collage attack has been applied to the watermarked image and the proposed approach is able to detect the attack. Make four copies of visually similar images and then watermark them by using same keys and parameters. On authentication side, the proposed algorithm detects it as a malicious tampering. Here the image can be recovered because it is able to correct the small number of erroneous bit by using BCH coding as discussed in earlier in this section.
Figure 5.11 (a) Tampered watermarked images of Lena and Cameraman, (b) Tampered regions are determined and localized obviously by Proposed approach, (c) Proposed by Li et al [30]

Figure 5.12 (a) London Eye image, (b) Object moving (airplane) i.e. the airplane is removed from its original position and added nearby, (c) Detection of tampered regions i.e. both the locations have been detected, (d) Recovered image
5.3.5 Choice of using two Watermarks: Individually or in Combination

In the earlier discussion, it has been described that both of the watermarks are independent and it is the user choice to use both of the watermarks or one of them according to the requirement of the application. In [35], the author uses only the recovery watermark. In Figure 5.15, the proposed approach is compared with [35],
using only the recovery watermark. Figure 5.15 also describes the PSNR while using both of the watermarks i.e. authentication watermark and recovery watermark.

Performance comparison of the proposed approach with previous approaches [16, 33, 35, 121, 125] is presented in Table 5.3. The comparison is made with respect to imperceptibility, security, tamper detection, localization etc. The approach offers more useful attributes compared to the existing approaches with the security and imperceptibly issues being resolved i.e. in [16], when the resolution of the image is increased, the localization of a tampered region degrades. Since the watermark is not embedded in the proper coefficients, the visual quality is not so good, although PSNR is above 40db.

![PSNR comparison graph](image)

**Figure 5.15** PSNR of 10 test images. Comparison of proposed approach (only $W_2$ and both $W_1$ and $W_2$) with [35].

### 5.4 Analysis

Some of the issues that affect the performance or relate to the proposed system are:

- The size of group and the quantization number $Quanta$ are user dependent. A trade-off has been made when the authentication watermark is embedded into the
original image. Smaller $g$ increases payload of the watermark and vice versa. Similarly, the smaller Quanta decreases the robustness of the watermark and the larger Quanta leads to stronger robustness.

- Multiple keys and the wavelet type i.e. Haar, Daubechies etc are supposed to be available at the receiving side.

The proposed algorithm is able to recover the exact version of the compressed host image with low overhead of the watermarks and can localize the tampered areas with high capability and can differentiate between the malicious and incidental attacks. Survival against JPEG lossy compression is flexible and can be set according to the requirement of the application.

Table 5.3 Prominent features and performance comparison with previous approaches.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Watermark Payload</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Section 5.1.1, Section 5.1.3</td>
</tr>
<tr>
<td>Watermark Security</td>
<td>Satisfactory</td>
<td>No</td>
<td>Satisfactory</td>
<td>Good</td>
<td>Satisfactory</td>
<td>Highly Secure</td>
<td>Section 5.1.1</td>
</tr>
<tr>
<td>Tamper Detection</td>
<td>Good</td>
<td>Satisfactory</td>
<td>Block-Based</td>
<td>Block-Based</td>
<td>Block-Based</td>
<td>Good</td>
<td>Figure 5.11-5.13, Section 4.2</td>
</tr>
<tr>
<td>Localization</td>
<td>Accurate</td>
<td>Not</td>
<td>Accurate</td>
<td>Block-Based</td>
<td>Sacrifice Localization</td>
<td>Block-Based</td>
<td>Highly Accurate</td>
</tr>
<tr>
<td>PSNR</td>
<td>Reasonable</td>
<td>Good</td>
<td>Good</td>
<td>Depends on Block Size</td>
<td>Good</td>
<td>Good</td>
<td>37db–42db</td>
</tr>
<tr>
<td>Compression Acceptance</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No (Fragile)</td>
<td>Yes</td>
<td>Yes (User Control)</td>
<td>Quanta is defined</td>
</tr>
<tr>
<td>Collage Attack Resiliency</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Eq. 1, Fig. 16</td>
</tr>
<tr>
<td>Attack Classification</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Section 2.2.4</td>
</tr>
<tr>
<td>Image Recovery</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Section 2.2.3</td>
</tr>
</tbody>
</table>

The correlation of the authentication watermark enhances the security aspects of the proposed approach, especially in view of collage attack. The technique is able to authenticate the image accurately unlike the traditional block-based authentication.
approaches. The computational complexity has been reduced by using the lifting scheme (IntWT).

The architecture of the proposed watermarking system is open. One can use both or any of the watermark according to the application.

5.5 Chapter Summary

The proposed semi-fragile multiple watermarking approach is able to accurately authenticate the image as well as recover the exact version of the approximated image. The technique has the ability to differentiate malicious and incidental manipulations. Value-added technique has been provided for secure and efficient authentication of digital images with multiple application-specific semi-fragile watermarks. Each of the embedded watermarks has an independent and different purpose; however, they do strengthen the security aspect of each other. Computational complexity has been reduced by using parameterized IntWT. Both of the watermarks are embedded in suitable wavelet coefficients, which increase the imperceptibility without any cost of robustness and security. Similarly, correlation and random permutations allow one to satisfy the security requirements of the watermarks. Huffman coding has been utilized to reduce the size of the recovery watermark, unlike to the approaches discussed in Chapter 3 and Chapter 4, where the recovery watermarks have the coefficients with high obtrusiveness. BCH coding is utilized that is used for correcting the erroneous bits on verification side.
Chapter 6

Conclusion and Future Directions

This chapter summarizes the main contributions of this thesis toward the image authentication and its recovery. Some future directions have been pointed out.

6.1 Summary

Watermarking is an information hiding technique where a secret message is concealed in the digital content. In order to make this hidden information secure, imperceptible and robust, the watermark should be embedded in some proper locations. Improving the security aspect of the watermarking system, without any cost of imperceptibility and robustness, is one of the challenges of today’s research in watermarking. The watermark can be embedded either in spatial or in transform domain. In this thesis, the watermarks are embedded in the suitable transformed coefficients due to which the security, imperceptibility, and robustness of the watermark is quite effective.

The watermarking techniques developed in this research work are able to make the image contents secure and recover the image approximation. The developed semi-fragile approaches can concisely determine the regions of the watermarked image, where the integrity verification may fail.

The work done in Chapter 3 encompasses improvement in a self-recovery authentication scheme for digital images. Dual watermarking approach has been exploited to authenticate and recover the image if authorisation fails. It is able to accept JPEG lossy compression up to 70% and reject the malicious attacks. The brightness adjustment of the image is also acceptable within a reasonable scope. Detection and localization of tampered regions is accurate and the original image can be recovered by using the self-recovering algorithm.

The limitations of the technique proposed in Chapter 3 are circumvented in Chapter 4. The watermark payload capability has been reduced and the security is improved by securing all the subbands of the image without any sacrifice on the
tamper detection resolution and its localization. The correlation based watermark is used to make the content secure from the Holliman and Memon (collage/counterfeiting) attack.

In Chapter 5, a novel model is designed that makes the content secure and recovers the original version of the image. Moreover, the imperceptibility has been improved by generating watermark having low payload. The Huffman and BCH coding are applied on the IntDCT coefficients of the image approximation to generate the recovery watermark. This method makes the strength of the recovery watermark very low as compared to the methods discussed in Chapter 3 and Chapter 4.

Eventually, summary of the main contributions of this research work are presented in summary as follows:

The authentication system proposed in this thesis
- is able to recover the exact version of the image approximation
- can localize the tampered areas with high capability and can differentiate between the malicious and incidental attacks
- imperceptibility has been improved, although two watermarks are embedded in every scheme
- can tolerate JPEG lossy compression up to some extent and compression survival is flexible that can be set according to the requirement of the application
- correlates the authentication watermark which enhances the security aspects, especially in view of collage/counterfeiting attack
- is able to authenticate the image concisely unlike the traditional block-based approaches
- reduces computational complexity by using IntWT, and IntDCT which is described in Chapter 5
- has an open architecture by using two semi-fragile watermarks and one can use both or any of the watermark according to the requirement of the application
- has the capability to answer the following questions:
  i) Has the image been processed?
  ii) Has the image been processed incidentally or maliciously?
  iii) Which part of the image has been processed and how much?
  iv) Can the image be recovered after manipulations?
6.2 Future Directions

In future, use of machine-learning techniques is expected for judicial selection of suitable locations for watermark embedding. In general, the selection of suitable coefficients based on machine learning approaches may result in improvement on the imperceptibility. Thus, the watermarking techniques can be implemented by using any of the intelligent approaches, like genetic programming (GP), genetic algorithms (GA), or artificial neural networks (ANN).

### 6.2.1 Selection of Proper Wavelet Coefficients for Embedding Watermarks

The use of any machine learning technique (most probably GP) for selection of proper location to embed the watermark can be an apparent extension to this work. This will improve the imperceptibility and will make the watermark more secure.

### 6.2.1 Generation of Machine Learning Based Quantization matrices

Although Watson’s perceptual model [83] is fair enough to give us imperceptible alterations, but it is not an optimum one. The focus in future might be on the selection of quantization matrix based on machine learning technique instead of standard quantization matrix. The focus is on quantization matrix based on the local features of the image that varies according to the image features. The elements in the quantization matrices can be set according to the local frequency content in an image.
References


References


41. G.W. Braudaway, K.A. Magerlein, and F. Mintzer. Protecting publicly available images with a visible image watermark. in In SPIE Conf. on Optical Security and Counterfeit Deterrence Techniques. 1996.

42. A. Koz, Digital watermarking based on human visual system, in Department of Electrical and Electronic Engineering. 2002, Middle East Technical University.


101

54. A. Khan, Intelligent perceptual shaping of a digital watermark, in Faculty of Computer Science and Engineering. 2006, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology: Swabi Pakistan.


79. A.D. Roza, M. Barni, F. Bartolini, V. Capalini, and A. Piva. Optimal Decoding of non-additive full frame DFT watermarks. in Proc. 3rd Workshop on Data Hiding. 2006.


84. C.Y. Lin, Watermarking and digital signature techniques for multimedia authentication and copyright protection, in Graduate School of Arts and Science. 2000, Columbia University: Columbia USA.


97. MATLAB Help, Lifting wavelet transform (LWT), liftwave: functions (Wavelet toolbox).


99. L.G. Kraft, A device for quantizing, grouping, and coding amplitude-modulated pulses, in Department of Electrical Engineering. 1949, Massachusetts Institute of Technology.


References
