Reversible Watermarking and Its Applications

Sana Ambreen Malik

Department of Computer and Information Sciences,
Pakistan Institute of Engineering & Applied Sciences,
Islamabad, Pakistan.

2013
In the Name of Allah, the Most Beneficent, the Most Merciful

The Rasool has believed in the guidance which has been revealed to him from his Rabb and do so the believers. They all believe in Allah, His angels, His books, and His rasools. They say: We do not discriminate against anyone of his Rasools. And they say: “We hear and we obey. Grant us Your forgiveness, O Rabb; to You we shall all return.”

Allah doest not burden any human being with more than he can bear. Everyone will enjoy the credit of his deeds and suffer the debits of his evil-doings. The believers say: “Our Rabb! Do not punish us if we forget or make a mistake. Our Rabb! Do not place on us a burden as You placed on those before us. Our Rabb! Lay not on us the kind of burden that we have no strength to bear. Pardon us, Forgive us, Have Mercy upon us. You are our Protector, Help us against the unbelievers.”

(Al-Baqarah 285-286)
Reversible Watermarking and Its Applications

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

By
Sana Ambreen Malik

Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Islamabad, Pakistan
2013
This thesis is prepared under the supervision of

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

Financial support by Higher Education Commission Pakistan through indeginous-5000 PhD fellowship program Batch-VI,
Grant No. 106-1555-Eg6-012.

Reversible Watermarking and Its Applications
DECLARATION OF ORIGINALITY

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: Sana Ambreen Malik

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

Supervisor:

Dr. Asifullah Khan
Associate Professor
DCIS, PIEAS, Islamabad.
Dedicated to my loving Mother & my Husband
ACKNOWLEDGEMENT

All praises to Almighty ALLAH, Creator of Universe, Who made us the super creature, blessed us with knowledge. I am grateful to Almighty ALLAH, the most Benevolent and Merciful, Who blessed me throughout my life and gave me the ability to undertake such a challenging task and proceeding towards completion.

I extend my sincerest thanks to my supervisor, Dr. Asifullah Khan for his generous guidance and moral support during my PhD. I appreciate his endless patience, positive attitude, ability to provide assistance and especially his willingness to put his students before his work. I thank him greatly for his meticulous proof reading of all of my published work. His valuable suggestions and persuasive criticism has led me to complete my goal successfully.

A very special note of thanks goes to my mother, my husband Mr. Mumtaz Ali, my brothers and sisters, my father-in-law and mother-in-law, whose heart felt prayers, appreciation, and support have always been a valuable asset and a great source of inspiration for me. They always encouraged me, whenever I was demoralized during my academic career. They really deserve special thanks for enduring all my problems with great patience and love.

I am also indebted to Dr. Abdul Jalil, Dr. Muttawarra Hussain, Dr. Abdul Majid, and all other teachers of the department for their corporation and encouragement to attain my goal. Thanks are due to Dr. Javeed Bashir, Ms. Rubina Shaheen, Dr. Nasir Khan, Dr. Ghulam Shabbir, Mr. Ghulam Sarwar, Mr. Riaz Malik, and Mr. Najm ul Haq, as this work would not have been possible without their encouragement and moral support.

I gratefully acknowledge Higher Education Commission of Pakistan for the financial support provided through Indigenous PhD scholarship program.

Last, but certainly not the least, I would like to thank my dearest friends (Ms. Raabya Rauf, Ms. Fatima Tahir, Ms. Munirah Atiq, Ms. Saima Rathore, Ms. Nabeela Kausar, Ms. Aysha Siddiqa, and Ms. Summuyya Munib). They helped me in times of troubles, praised me on my achievements, and cheered me when I was depressed.

Sana Ambreen Malik
LIST OF PUBLICATIONS


# TABLE OF CONTENTS

Abstract.............................................................................................................................................xvii
Abbreviations ........................................................................................................................................xix

Chapter 1  Introduction ............................................................................................................................1
  1.1  Motivation and Objectives .............................................................................................................3
  1.2  Research Perspective .....................................................................................................................3
  1.3  Contributions ..................................................................................................................................3
  1.4  Thesis Structure ...............................................................................................................................4

Chapter 2  Literature Review ...................................................................................................................6
  2.1  Histogram Modification based Reversible Watermarking ............................................................6
  2.2  Compression based Reversible Watermarking ..............................................................................7
  2.3  Quantization based Reversible Watermarking ............................................................................8
  2.4  Expansion based Reversible Watermarking ..................................................................................8
  2.5  Chapter Summary ..........................................................................................................................10

Chapter 3  Compression based Reversible Watermarking ..................................................................11
  3.1  Proposed GA-revWM Approach ......................................................................................................11
  3.2  Training Phase (GA based Optimization) .......................................................................................11
  3.3  Testing Phase ..................................................................................................................................13
    3.3.1  Information Embedding Procedure .........................................................................................13
    3.3.2  Information Extraction Procedure .............................................................................................18
    3.3.3  Restoration of the Original Image .............................................................................................19
  3.4  Experimental Results and Discussions .........................................................................................20
    3.4.1  GA Related Implementation Details .........................................................................................21
    3.4.2  GA based Tradeoff between Imperceptibility and Capacity .....................................................21
    3.4.3  Performance Comparison with Existing Approaches .............................................................25
    3.4.4  Performance on Medical Images ...............................................................................................27
  3.5  Chapter Summary ...........................................................................................................................29

Chapter 4  Reversible Watermarking for Images with 3D Information ...............................................30
4.1 Depth Map Computation .................................................................................................................. 31
4.2 Tmap Generation ................................................................................................................................. 35
  4.2.1 Adaptive Threshold based Reversible Watermarking (ATRW) .................................................. 35
  4.2.2 Genetic Algorithm based Reversible Watermarking (GARW) .................................................. 38
4.3 Information Embedding ....................................................................................................................... 40
4.4 Information Extraction ......................................................................................................................... 42
4.5 Authentication of the Host Image through Tmap.............................................................................. 45
4.6 Results and Discussion ......................................................................................................................... 46
  4.6.1 Preparation of Data for Experimental Analysis ........................................................................ 46
  4.6.2 Reversible Hiding of Depth Map using ATRW ........................................................................ 48
  4.6.3 Lossless Compression of Depth Maps ....................................................................................... 51
  4.6.4 GARW Results ............................................................................................................................. 52
  4.6.5 Authentication Analysis of a Host Image .................................................................................... 56
4.7 Chapter Summary ................................................................................................................................. 57

Chapter 5  Reversible Watermarking for 3D Cameras ................................................................................ 58
5.1 Time of Flight (TOF) Principle ............................................................................................................. 58
5.2 Watermark Embedding Procedure ..................................................................................................... 60
  5.2.1 Threshold Matrix (Tmap) Generation using DERW Approach ................................................ 60
  5.2.2 Threshold Matrix (Tmap) Generation using HPDRW Approach ............................................ 62
  5.2.3 Information Embedding .............................................................................................................. 66
5.3 Watermark Extraction Procedure ...................................................................................................... 67
5.4 Image Authentication through Auxiliary Information .......................................................................... 68
5.5 Results and Discussion ......................................................................................................................... 69
  5.5.1 Performance Analysis in terms of Watermark Imperceptibility .............................................. 69
  5.5.2 Performance Analysis of the proposed Techniques on Image Database ................................ 71
  5.5.3 Authentication Analysis of a Host Image .................................................................................... 72
  5.5.4 Temporal Analysis of DERW and HPDRW .............................................................................. 73
  5.5.5 System Related Discussion .......................................................................................................... 74
  5.5.6 Security Analysis ......................................................................................................................... 74
5.6 Chapter Summary ................................................................................................................................. 74
Chapter 6  Histogram based Reversible Watermarking ........................................................... 76
  6.1  Reversible Data Hiding using Block Skipping Scheme (RDHBS) .................................. 77
  6.2  Proposed Improved RDHBS Approach ................................................................. 80
  6.3  Proposed RW-HPBS Approach .............................................................................. 82
       6.3.1  Embedding using RW-HPBS ................................................................. 83
       6.3.2  Extraction using RW-HPBS ................................................................ 85
       6.3.3  Embedding of Location Map as an Auxiliary Information ...................... 86
       6.3.4  Additional Use of Location Map for Authentication ............................... 88
  6.4  Results and Discussion ......................................................................................... 88
       6.4.1  Effect of Reducing the Block Size .......................................................... 91
       6.4.2  Effect of Modified Down-Sampling ...................................................... 93
       6.4.3  Effect of Changing the Embedding Level ............................................. 98
       6.4.4  Performance Analysis when the Location Map is embedded as an Auxiliary Information 98
       6.4.5  Image Authentication related Analysis ............................................... 101
  6.5  Chapter Summary ............................................................................................... 104
Chapter 7  Conclusions and Future Directions ............................................................ 105
  7.1  Research Summary ............................................................................................ 105
  7.2  Future Directions ............................................................................................... 106
       7.2.1  Improving computational Time ............................................................. 106
       7.2.2  Detection of Tampered Region ............................................................. 107
       7.2.3  Developing Tracing Capabilities ......................................................... 107
       7.2.4  Multiple Watermarking ...................................................................... 107
References ................................................................................................................. 108
LIST OF FIGURES

Figure 1.1: Reversible Watermarking Framework .......................................................... 2
Figure 3.1: Block diagram of GA-revWM Approach .................................................. 12
Figure 3.2: Block diagram of watermark embedding procedure .................................. 13
Figure 3.3: Histogram modification. (a) Original Image Data, (b) Modified Image Data ....... 14
Figure 3.4: Sub-bands split of an X×Y image .................................................................. 15
Figure 3.5: Threshold matrix Tmap and sub-bands ....................................................... 17
Figure 3.6: Block diagram of information extraction procedure .................................... 19
Figure 3.7: Images considered for accessing the performance of the GA-revWM .......... 20
Figure 3.8: GA simulation for Lena image ........................................................................ 22
Figure 3.9: Threshold Matrix for a 16×16 block of Lena image (payload=0.1) .............. 22
Figure 3.10: Column (a) original input images, (b) watermarked images, (c) difference between original and watermarked images (difference of (a) and (b)), (d) restored images and (e) difference between original and restored images (difference of (a) and (d)) .................. 23
Figure 3.11: PSNR vs. Payload for block size 8×8 to 32×32 using Lena image ............... 24
Figure 3.12: Performance comparison with existing approaches for Lena image .......... 26
Figure 3.13: PSNR versus payload for 16-bit grayscale images ...................................... 26
Figure 3.14: Performance comparison of GA-revWM approach with Xuan et al. approach [46] for payload 0-1 ......................................................... 27
Figure 4.1: Optical microscopic system for SFF approach ........................................... 33
Figure 4.2: IDWT decomposition of an N×N block showing different sub-bands .......... 35
Figure 4.3: Block diagram of Tmap generation using ATRW approach ....................... 37
Figure 4.4: Block diagram of Tmap generation using GARW approach ...................... 39
Figure 4.5: The basic block diagram of the embedding process .................................. 41
Figure 4.6: The basic block diagram of the extraction process ..................................... 43
Figure 4.7: Column (a) original input images, (b) watermarked images, (c) difference between original and watermarked images (difference of (a) and (b)), (d) restored images and (e) difference between original and restored images (difference of (a) and (d)) .......... 44
Figure 4.8: Processing Tmap to detect collage attack ............................................... 46
Figure 4.9: (a-e) Sample images of TFT-LCD color filter, human face, human body, and simulated cone respectively. (f-j) show their corresponding depth maps. ............................. 47
Figure 4.10: Effect of varying the block size from 8×8 to 64×64 on face image shown in Figure 4.9 (b), using ATRW approach .............................................................. 48
Figure 4.11: Performance comparison of proposed GARW and ATRW techniques with other existing approaches, for the image shown in Figure 4.9 (b) ..................... 49
Figure 4.12: Performance comparison of GARW and ATRW techniques (after employing lossless compression) with other existing approaches, for the image shown in Figure 4.9 (b) .................. 50
Figure 4.13: Average PSNR plot for 255 images using ATRW approach ...................... 51
Figure 4.14: Performance comparison of GARW and ATRW approaches using database of 75 images .... 52
Figure 4.15: Column (a) watermarked images, (b) tampered watermarked images, (c) difference of Tmaps for non-tampered form of watermarked image (column (a)), (d) difference of Tmaps for tampered form of watermarked image (column (b)). ................................................. 56

Figure 5.1: Basic principle of TOF. .................................................................................................................. 59

Figure 5.2: Column (a) 2D Images, Column (b) Depth map images of Column (a). Row (1) man Image, Row (2) Glass Image, Row (3) Toy Image, Row (4) Face Image. ........................................... 59

Figure 5.3: Block diagram of computing Tmap using DE. .............................................................................. 61

Figure 5.4: Block diagram of computing Tmap using hybrid PSO-DE. .......................................................... 65

Figure 5.5: Block diagram of watermark embedding procedure. .................................................................... 67

Figure 5.6: Block diagram of watermark extraction procedure. ..................................................................... 68

Figure 5.7: Performance comparison of proposed HPDRW and DERW with adaptive threshold [50], fixed threshold [48] and variable threshold [49] approaches, against different values of payload (for Lena image). ................................................................. 70

Figure 5.8: Performance comparison of proposed HPDRW and DERW techniques with that of Adaptive Threshold [50], Fixed Threshold [48] and Variable Threshold [49] approaches using a database of 100 images. ...................................................................................... 71

Figure 5.9: Column (a) watermarked images, (b) tampered form of watermarked images, (c) difference of Tmaps for non-tampered watermarked images (column (a)), (d) difference of Tmaps for tampered watermarked images (column (b)). ......................................................... 73

Figure 6.1: Pixel gray scale values distribution and conditions for block classification in different categories [42]. .......................................................................................................................... 77

Figure 6.2: Distribution of pixels in sets S1 and S2. ......................................................................................... 78

Figure 6.3: Frame work of the bit embedding scheme [42]. ............................................................................. 79

Figure 6.4: Category transferring diagram when bit "1" is embedded [42]. ..................................................... 79

Figure 6.5: Images used in simulation. ........................................................................................................... 81

Figure 6.6: (a) Original image (b) Reference sub-sampled image (Lref) (c) Data hiding sub-sampled version (Lhd) ........................................................................................................................................ 83

Figure 6.7: Selection of pixels from reference sub-sampled verion and data hiding version (a) First block pixels(b) Second block pixels ...................................................................................................... 83

Figure 6.8: Flow chart of embedding process using RW-HPBS technique. ....................................................... 84

Figure 6.9: Flow chart for message extraction in the RW-HPBS method. .......................................................... 86

Figure 6.10: Watermarking using proposed RW-HPBS approach (a) Original images (b) Watermarked images (c) Restored images (d) Difference between (a) & (c). ......................................................... 91

Figure 6.11: Performance comparison of RW-HPBS technique with other existing approaches, using Lena image........................................................................................................................................ 99

Figure 6.12: Performance comparison of proposed RW-HPBS technique with Thodi et al. [58], Xuan et al. [48], and Gao et al. [42] approaches using database of 300 images (Capacity is taken as 1000 bits). .......................................................... 99

Figure 6.13: Performance comparison of proposed RW-HPBS technique with Thodi et al. [58] and Xuan et al. [48] approaches using database of 300 images (Capacity is taken as 0.4 bpp). 100

Figure 6.14: Performance comparison of proposed RW-HPBS technique with Thodi et al. [58] and Xuan et al. [48] approaches using database of 300 images (Capacity is taken as 0.5 bpp). 100
Figure 6.15: Performance comparison of proposed \textit{RW-HPBS} technique with Thodi et al. [58] and Xuan et al. [48] approaches using database of 300 images (Capacity is taken as 0.7 \textit{bpp}).

Figure 6.16: Column (a) watermarked images, (b) tampered form of watermarked images, (c) difference of location maps for non-tampered watermarked image (column (a)), (d) difference of location maps for tampered watermarked image (column (b)).
LIST OF TABLES

Table 3.1: PSNR comparison of Xuan et al. [46] with proposed technique, against different values of payload.................................................................28
Table 3.2: SSIM comparison of Xuan et al. [46] with proposed technique, against different values of payload.................................................................28
Table 4.1: Compression results for different depth maps of various images .................................................................51
Table 4.2: Performance comparison of proposed GARW technique with that of ATRW approach at fixed effective payload.................................................................53
Table 4.3: PSNR results for proposed GARW and ATRW approaches using the dataset of 75 images ..........54
Table 5.1: Image quality (PSNR and SSIM) based comparison with existing approaches .................................................................69
Table 6.1: Block category changing conditions [42] .................................................................................80
Table 6.2: Comparison of block classification by RDHBS [42] and proposed improved RDHBS method .................................................................82
Table 6.3: Image and block size for each test image .................................................................................88
Table 6.4: PSNR (dB) and payload comparison of RDHBS [42] and proposed improved RDHBS method .................................................................89
Table 6.5: Imperceptibility and payload comparison of proposed improved RDHBS method for different block sizes .................................................................................92
Table 6.6: Imperceptibility and payload comparison of the proposed improved RDHBS method (block size 8×8) and the proposed RW-HPBS method .................................................................................93
Table 6.7: Imperceptibility and payload comparison of the proposed improved RDHBS method (block size 2×2) and the proposed RW-HPBS method .................................................................................95
Table 6.8: PSNR comparison of Luo’s approach [102] and the proposed RW-HPBS method .........................96
Table 6.9: PSNR comparison of Luo’s approach [102] and proposed RW-HPBS method. .........................97
LIST OF ALGORITHMS

Algorithm 4.1: RSA Algorithm ................................................................. 34
Algorithm 4.2: Tmap Generation using ATRW .................................................. 37
Algorithm 4.3: Reversible Hiding of Depth Information (Embedding Procedure) .............. 41
Algorithm 4.4: Reversible Hiding of Depth Information (Extraction Procedure) .................. 43
Algorithm 6.1: Generation of sub-sampled versions ($I_{ref}$ and $I_{hd}$) of image ....................... 83
ABSTRACT

In the last decade, watermarking applications have increased considerably. The main reason is that watermarking has emerged as a prospective technique, which can provide copyright protection and authentication of digital content. However, the disadvantage of watermarking is that it introduces small modifications in the original work and thus causes slight degradation. These modifications may be undesirable in some sensitive applications, like medical imagery, 3D reconstruction, and military applications. As a remedy to this problem, researchers have introduced the concept of reversible watermarking.

The main objective of reversible watermarking scheme is to restore the watermarked image to its original state after watermark extraction. In this thesis, new reversible watermarking techniques as well as their novel applications are presented. In some of these techniques, computational intelligence (CI) approaches have been employed to improve the watermark capacity versus imperceptibility tradeoff. The research work is carried out in four phases.

In the first phase, reversible watermarking is employed on medical imagery, which comprises regions of sensitive information. Slight modification in these regions affects the diagnostic analysis and thus can lead to wrong decisions. For this purpose, a novel reversible watermarking technique has been developed that utilizes genetic algorithm (GA) to improve capacity versus imperceptibility tradeoff. The algorithm makes use of block based companding technique, which helps in increasing the watermark capacity. Experimental analysis depicts that the developed watermarking technique provides good performance compared to the existing approaches. As the technique is reversible, therefore, it is even capable of embedding in the sensitive regions of the image.

Reversible watermarking of images with depth information is discussed in the second phase of this work. 3D imaging is widely used in 3D gaming, robotics, controlling and routing devices etc. Different techniques and algorithms are reported to compute the depth information of an object. In this phase, depth information is computed through shape from focus algorithm. The depth information is reversibly embedded in its corresponding 2D image. This technique also utilizes GA to compute near-optimal threshold matrix for performance improvement in terms of capacity versus imperceptibility tradeoff. An additional attribute is achieved by using the threshold matrix for authentication purpose.

The third phase focuses on reversible watermarking of 3D camera images. 3D cameras work on different principles for depth map computation. Cameras working on time of flight principle for depth
map calculation are used in the experimental analysis of the proposed technique. The developed technique utilizes 3D information to embed as a watermark. In this way, protection and secure transmission of an image along with its corresponding depth map is provided. Two CI approaches, namely, differential evolution and a hybrid approach (comprising particle swarm optimization and differential evolution) are utilized to optimize the capacity and imperceptibility tradeoff. This technique is also able to provide authentication capability against manipulation and collage attacks.

In the first three phases, CI is exploited to improve the performance of the proposed reversible watermarking techniques. However, CI approaches are more time and resource consuming. Therefore, in fourth phase, a novel and fast reversible watermarking technique is proposed based on histogram processing and down sampling. Histogram based reversible watermarking techniques are easy to implement and are computationally less expensive. A concept of down sampling is employed to generate a reference image and thus create more space for hiding bits. Block selection is used to generate a location map. However, the location map is required at the receiving side to perform extraction and recovery processes. An additional use of location map is devised, which makes the technique capable of authenticating digital images.
## ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCH</td>
<td><em>Bose-Chaudhuri-Hoequenghem</em> algorithm</td>
</tr>
<tr>
<td>bpp</td>
<td>Bit per pixel</td>
</tr>
<tr>
<td>BDS</td>
<td>Bookkeeping Data Store</td>
</tr>
<tr>
<td>CDF</td>
<td><em>Cohen-Daubechies-Feauveau</em> wavelet family.</td>
</tr>
<tr>
<td>CI</td>
<td>Computational Intelligence</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DE</td>
<td>Differential Evolution</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processing</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field-Programmable Gate Array</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GP</td>
<td>Genetic Programming</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System</td>
</tr>
<tr>
<td>HH</td>
<td>Diagonal Detail sub-band</td>
</tr>
<tr>
<td>HL</td>
<td>Vertical Detail sub-band</td>
</tr>
<tr>
<td>IDWT</td>
<td>Integer Discrete Wavelet Transform</td>
</tr>
<tr>
<td>IWT</td>
<td>Integer Wavelet Transform</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group, use for image compression.</td>
</tr>
<tr>
<td>LH</td>
<td>Horizontal Detail sub-band</td>
</tr>
<tr>
<td>LL</td>
<td>Approximation sub-band</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bit.</td>
</tr>
<tr>
<td>MED</td>
<td>Median Edge Detector</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>MFLOPS</td>
<td>Mega Floating Point Operations per Seconds</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal to Noise Ratio</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest</td>
</tr>
<tr>
<td>RSA</td>
<td><em>Rivest-Shamir-Adlema</em> algorithm</td>
</tr>
<tr>
<td>SFF</td>
<td>Shape from Focus</td>
</tr>
<tr>
<td>SSIM</td>
<td>Structured Similarity Index Measure</td>
</tr>
<tr>
<td>TFT-LCD</td>
<td>Thin-film-transistor liquid-crystal display</td>
</tr>
<tr>
<td>Tmap</td>
<td>Threshold Matrix</td>
</tr>
<tr>
<td>TOF</td>
<td>Time of Flight</td>
</tr>
</tbody>
</table>
Chapter 1  INTRODUCTION

During the past few decades, advancement in digital communication and internet facilities gave rise to certain issues related to copyright violation, malicious tampering, and illegal reproduction of multimedia information. Consequently, security of digital content (image, audio, video, graphics, printed document, databases, piece of art etc.) becomes important not only for the owners but for the distributors as well. Digital watermarking is a promising solution to the abovementioned problems. The term digital watermarking can be defined as: “imperceptibly altering a digital content to embed a message about that work” [1].

History of watermarking is quite old. Approximately, one thousand years ago, China invented paper marking. In 1282, first paper watermark appeared in Italy. In paper watermarks, marks were created by embedding thin wire patterns in paper molds. However, the actual use or purpose of paper watermark was uncertain at that time. In 18th century, America and Europe used paper watermark as trademarks and as anti-counterfeiting measures for money and other documents. In 1988, Komatsu and Tomingo [2] for the first time introduced the term “digital watermarking”. Generally, there are two types of watermarking techniques; namely, informed and blind watermarking. In informed watermarking technique, some information about the original work at the receiving side is required. However, blind watermarking technique, some information about the original work at the receiving side is required. However, blind watermarking techniques do not require any information for watermark extraction. In the beginning, almost all of the watermarking techniques were informed watermarking techniques, as they required some information about the original work at the receiving side.

Digital watermarking is generally classified into three categories; robust [3–11], fragile[12–19], and semi-fragile [20–27]. A watermarking system is robust, if it can withstand illicit and normal signal processing based operations. On the other hand, in fragile watermarking, a watermark is destroyed if it undergoes any type operations. Semi-fragile watermarking lies in between fragile and robust watermarking systems. It can resist some common signal processing operations but the watermark is destroyed if it undergoes any intentional attack.

Due to easy distribution and frequent manipulation of the content over the transmission channel, authentication capability of a watermarking system has become an important property [27]. In various applications related to the medical imagery, multimedia archives, and military imagery, it is important to identify whether the received signal has undergone any type of attack or not. In this regard, watermarking research community reported different techniques to identify any malicious activity performed on the
received image [17], [26–30]. Performance of any watermarking application can be analyzed through the parameters such as robustness, imperceptibility, and payload. Researchers are working to improve the tradeoff between any two parameters, while keeping the third one constant.

Digital watermarking introduces slight quality degradation in the original work, which is undesirable in many applications like military and medical imagery. To overcome this problem, the concept of reversible watermarking is introduced in literature. Reversible watermarking is a technique in which the watermarked image is reverted to its original state after the watermark extraction. A general framework of reversible watermarking technique is shown in Figure 1.1.

![Figure 1.1: Reversible Watermarking Framework.](image)

In 2001, Honsinger et al. [31] reported their work on reversible watermarking, which was one of the initial works presented in this field. After that, Macq [32] published his work in the field of reversible watermarking by modifying a patchwork algorithm. Performance of the above-mentioned approaches was poor due to the use of modulo-256 operation, which introduces salt and pepper noise. Fridrich et al. [25] alleviated the noise problem but their approach offers low embedding capacity. In 2002, Fridrich et al. [33] presented another approach that provides relatively good imperceptibility versus capacity tradeoff. After this, the field of reversible watermarking evolved rapidly. Various researchers proposed different techniques in this field either by introducing novel ideas or by extending and hybridizing the existing
techniques. However, the major concern of each research is to create more and more space for data hiding in the cover work without decreasing its imperceptibility.

1.1 Motivation and Objectives

Due to advancements in multimedia tools, it is easy to manipulate the digital content without leaving a trace. Consequently, there is a considerable need of authentication and copyright protection of digital content. Watermarking is widely used to secure a digital work, but this phenomenon introduces quality degradation to the digital content. In some applications, like military and medical imagery, even a small change in the original work is not acceptable. Therefore, it is required to revert this degradation to get the original state of an image at the receiving side. In this regard, a notion of reversible watermarking is presented. It is a newly evolving field and many researchers are striving for introducing innovative ideas in this field. Aim of this research is to develop advance reversible watermarking techniques for the protection of digital content.

1.2 Research Perspective

Proliferation of powerful multimedia tools has created the issues of illegal reproduction, manipulation, malicious tampering, and copying of the digital content. These issues thus activated the research in the area of content security and integrity. In this regard, various techniques have been developed to provide security to the digital content.

This research focuses on developing new reversible watermarking techniques as well as new applications of watermarking. The research strives for designing and developing the techniques that are not only reversible but also outclass various existing schemes by providing better performance in terms of capacity versus imperceptibility tradeoff.

1.3 Contributions

This thesis concentrates on authentication and recovery of digital images, while keeping a good imperceptibility versus capacity tradeoff. The research presented in this thesis contributes in the following domains:

- The proposed techniques are not only designed to provide protection to digital contents but are also reversible. The reversibility property of the proposed techniques is ascertained through experimental analysis.
• To improve the performance of the developed techniques, computational intelligence (CI) is integrated with reversible watermarking. CI algorithms run in the training phase and optimize the performance of the watermarking system, while avoiding overflow/underflow problem. The intelligence and learning capability of these algorithms provides improved performance compared to the existing approaches.

• The proposed reversible watermarking techniques are applied on medical images as well as on images used for 3D reconstruction. 3D images are very important in robotics, controlling and routing devices, etc.

• Authentication capability introduced in the reversible watermarking approach, which can detect manipulation as well as collage attacks. Detection of collage attack is a challenging task, but experimental analysis depicts that the developed techniques can successfully detect the collage attack in the received watermarked image.

1.4 Thesis Structure

Chapter 2 presents a recent survey of reversible watermarking approaches. In this chapter, the ideas and techniques reported by different contemporary researchers have been presented. To make it more understandable, the literature study is explained by dividing it into different groups.

In Chapter 3, a technique is presented for the protection of medical images. Medical image watermarking is a challenging issue because these images contain sensitive information that needs to be preserved. For this purpose, compression based reversible watermarking in conjunction with genetic algorithm is presented. The proposed technique is not only reversible but also performs well compared to the existing approaches.

A reversible watermarking approach for the protection of both 3D information and its corresponding 2D image is presented in Chapter 4. Different techniques and algorithms are reported in literature to compute the depth information of an object. The technique developed in this chapter is for images whose 3D information is computed by employing shape from focus (SFF) algorithm. The work presented in this chapter utilizes the learning capabilities of genetic algorithm (GA) for making a suitable tradeoff between watermark imperceptibility and capacity. It also provides authentication capability against manipulation and collage attacks.

Chapter 5 presents a watermarking approach for 3D cameras. 3D cameras utilize different concept for depth computation. In this work, we considered 3D camera that uses time of flight (TOF) principle for depth map calculation of an object. The proposed technique utilizes the capability of differential evolution
(DE) algorithm and a hybrid algorithm (comprising particle swarm optimization (PSO) with DE) for performance improvement. This technique also provides authentication capability against manipulation and collage attacks.

In Chapter 6, a novel histogram based reversible watermarking technique is presented. The techniques developed in previous chapters utilize CI based approaches for performance optimization. The main drawback of using CI based approaches is that they consume high computation time in training phase. Therefore, for real time applications, a high performance histogram based technique is reported in this chapter. The main advantages of histogram based reversible watermarking are the simplicity of implementation and less computational time. This technique is also able to authenticate the received image against manipulation and collage attacks.

Chapter 7 concludes the work presented in this dissertation. Additionally, it discusses some future directions of the work presented in this thesis.
Chapter 2 \textbf{LITERATURE REVIEW}

Various reversible watermarking approaches have been reported in the literature. Their performance is compared in terms of capacity versus imperceptibility tradeoff. Feng et al. [34] have published a survey on reversible watermarking approaches. They have divided the field of reversible watermarking in three categories; namely, histogram shifting, data compression, and difference expansion. In 2009, Pan et al. [35] reported a comparative study of some reversible watermarking applications on medical imagery. Caldelli et al. [36], in 2010, have published a survey on reversible watermarking approaches. They have categorized the field of reversible watermarking according to fragile, semi-fragile, and robust.

Various reversible watermarking approaches have been proposed in literature, some are based on new techniques, and some are the modification or hybridization of already existing ideas. According to the literature study, we divide the field of reversible watermarking in four groups based on the characteristics of algorithms. These are:

1. Histogram Modification based Reversible Watermarking
2. Compression based Reversible Watermarking
3. Quantization based Reversible Watermarking
4. Expansion based Reversible Watermarking

Some research work related to each group is discussed in the upcoming sections.

2.1 \textbf{Histogram Modification based Reversible Watermarking}

In this group, initial work is done by Vleeschouwer et al. [37]. In their work, image is divided in blocks and each block is split into two zones. Histogram of each zone is computed. To embed watermark bit, histogram is shifted. The histogram shift introduces salt and pepper noise in the image, which results in low imperceptibility. Vleeschouwer et al. dealt with this problem in their work reported in 2003 by introducing bijective transformation [38].

Ni et al. [39] reported another novel reversible watermarking approach, which is based upon histogram modification. In their work, initially histogram of the whole image is generated and then pixels that fall at histogram peak are selected for embedding. However, their technique provides limited capacity, i.e. embedding can be done only in the pixels that fall at the peak. In 2008, Lin et al. [40] published their work about multilevel reversible watermarking. Their work increases the embedding
capacity of histogram modification based reversible watermarking, but it requires some extra information at the extraction side to perform recovery process. Ni et al. [41], in 2008, presented another reversible watermarking approach which is based upon histogram modification. Gao et al. [42] have investigated Ni et al.’s [41] technique and highlight some shortcomings of their approach. In their work, they proposed an approach by suggesting the solution to the problems present in Ni et al.’s [41] technique.

In 2009, Kim et al. [43] proposed a reversible watermarking approach, in which image is first divided into two sub-sampled images. Watermark is embedded in the difference histogram of the sub-sampled images.

### 2.2 Compression based Reversible Watermarking

In this type of algorithms, part of the image is compressed to create more space for embedding. Yang et al. [44] reported companding based reversible watermarking approach. In their approach, discrete cosine transform (DCT) of image block is computed first and then companding is performed over the DCT coefficients. To prevent overflow and underflow problem, a block discrimination structure is used to select suitable blocks for embedding. In 2005, Celik et al. [45] reported a technique, which involves compression of image features. In their approach, compression is applied on a part of the signal that is more susceptible for distortion. Description of this compression is also transmitted with the payload, which is used in recovery at the receiving side. This technique provides high capacity and low distortion compared to the previous approaches.

In 2005, Xuan et al. [46] published a reversible watermarking approach, which is based on companding process. In their technique, image is first transformed to integer wavelet transform (IWT) domain, which divides the image into four bands; approximation sub-band LL, horizontal sub-band LH, vertical sub-band HL, and diagonal sub-band HH. Only HH, HL, and LH sub-bands are selected for embedding, because, any change introduced in LL sub-band cause huge distortion in the image. Before embedding, coefficients of these sub-bands are compared with an empirically selected threshold. If, coefficient value is greater than the threshold then companding process is applied to compress the value. This process improves the embedding capacity; however, it also increases the amount of auxiliary data. In 2011, Memon [47] enhanced Xuan et al.’s [46] approach, which results in improvement in capacity.

Xuan et al. [48] published another reversible watermarking technique in which embedding is done on histogram of wavelet coefficients based on certain threshold value. In their approach, image is first transformed to IWT domain, which decomposed the image into four sub-bands HH, HL, LH, and LL. Again, only HH, HL, and LH sub-bands are selected for embedding. Coefficients of HH, HL, and LH sub-
bands are compared against a fixed threshold whose value is selected empirically. If a coefficient value is less than the threshold then watermark bit is embedded in it, else it is shifted away from zero by adding or subtracting the threshold value. At the end, inverse IWT is applied to get the final marked image. The process of watermarking may produce out of bound pixels; therefore, histogram preprocessing is applied to avoid this problem. In 2008, Khan et al. [49] modified the Xuan et al.’s [48] technique by introducing different threshold value for different wavelet sub-band. This process introduces some improvement in the results. In 2009, Ali et al. [50] introduces the concept of adaptive thresholding. In their technique, each sub-band is divided into blocks and each block has its own threshold value. The threshold for each block is computed adaptively, the rest of the embedding process is the same as proposed in [48]. However, their approach performs better compared to the previous approaches.

### 2.3 Quantization based Reversible Watermarking

Watermarking approaches that are based on quantization process are robust in general; however, to make the quantization approaches reversible, it turns to fragile nature. Cheung et al. [51] reported sequential quantization strategy (SQS). A weighted quantization method (WQM) is published by Saberian et al. [52] in 2008. In their approach, a set of linear convex functions are designed that satisfy the condition of blindness, reversibility, and imperceptibility. Then a function, which introduces least distortion, is selected. Quantization levels are adjusted to control the distortion. This technique is less computationally extensive and provides good performance compared to previous approaches. In 2011, Ko et al. [53] reported a nested quantization index modulation (QIM) approach. In general, conventional QIM based watermarking approaches are not reversible; because, the distortion produced in quantization process are not invertible. However, the technique presented by Ko et al. [53] ensures the recovery of the original image and hence reversible in nature. Due to nested QIM structure, the capacity of the proposed method is increased.

### 2.4 Expansion based Reversible Watermarking

Expansion based reversible watermarking is a newly evolving field. It is started in 2003, from the work of Tian [54]. Tian [54] introduced an innovative idea of difference expansion that opens a different course of research in the field of reversible watermarking. In his work, image is first split into pair of pixels then from these pair, difference and average values are calculated. Watermark is embedded by difference value expansion. From average value and watermarked difference value, pixel values are computed for the corresponding watermarked image. There are certain pixels that are not expandable. For those pixels,
location map is generated. This technique provides high embedding capacity and low computational complexity compared to preceding techniques.

In 2007, the work reported by Coltuc et al. [55] was based on contrast mapping, which is another form of difference expansion. Image is first divided into pair of pixels. A transformation is applied on each pair of pixels. This transformation is such that it is invertible even if least significant bits (LSBs) of transformed pair are lost. This technique provides high embedding rate at low mathematical complexity and is robust against cropping. Lu et al. [56] provided the improved version of Coltuc et al.’s [55] approach. In 2009, Hong et al. [57] reported another modification to Coltuc et al.’s [55] technique. In their technique, image is first split into non-overlapping blocks and then all blocks are sorted in ascending order of their variances. Embedding is performed in the sorted blocks according to the contrast mapping approach presented in [55].

In 2004, Thodi et al. [58] presented reversible watermarking approach using prediction error expansion instead of difference expansion. This technique utilizes the correlation of a pixel with their neighboring pixels for watermark embedding. Initially, a predicted model of an image is computed. There are different predictors reported in the literature. Thodi et al. [58] used median edge predictor (MED) for their work. Then, prediction error is computed. Watermark bit is inserted by prediction error expansion. To generate watermarked image, the modified prediction error is added to the predicted model of the image. There are certain locations for which prediction error expansion causes out of bound pixels; therefore, watermark is embedded only in the expandable locations. Location map for expandable locations is required for extraction and recovery process at the receiving side. This approach provides improved results compared to difference expansion technique [54] for moderate to high payload. Thodi et al. [59] reported improvement in difference expansion technique [54] and prediction error approach [58] by combining histogram shifting method with [54] and [58]. In 2009, Sachnev et al. [60] reported a technique that is based on sorted prediction error mechanism. In their approach, they combined rhombus pattern prediction scheme, sorting, and histogram shifting, which results in significant decrease in location map size and improvement in performance. Initially, the image is divided into two groups of alternate pixel locations. Then, by applying rhombus pattern prediction scheme on four neighboring pixels, prediction errors are computed. These prediction errors are sorted based on their local variance. Using sorted prediction errors and location map, watermark is embedded in the image, which produces low distortion. In 2009, Chen et al. [61] proposes a scheme in which full context is used for prediction. In their approach, they used the additive embedding method i.e. prediction error is changed at most by ‘1’ after embedding. Since, this approach utilizes full context for prediction; therefore, prediction error is
small, which results in increase in embedding capacity. In 2010, Ou et al. [62] proposed a reversible watermarking technique by employing a concept of linear predictor. For watermark embedding, prediction error expansion and histogram shifting is employed. In their approach, initially linear prediction is computed, which is modified at later stage by employing some variance of adjacent pixels to achieve prediction that is more accurate. Computation of accurate prediction generates more concentrated histogram of prediction error, which results in improved performance. Tudoroiu et al. [63] presents another reversible watermarking technique that is utilizing the concept of prediction error expansion. In their technique, block wise prediction error is computed by using MED predictor. Initially, image is split into overlapping blocks. Embedding is performed only in those blocks for which all of the pixels are suitable for embedding. Block map based embedding provides better results compared to previous approaches. In 2011, Luo et al. [64] published a reversible watermarking approach that employed prediction error expansion. In their approach, compensation concept has been introduced before computing prediction errors. “Compensation” is used to compensate the loss of spatial correlation due to modified pixels. This process improves the prediction accuracy that causes improved embedding performance. In 2011, Coltuc [65] reported a prediction expansion based reversible watermarking with reduced distortion. In his work, the main idea is to divide the distortion, produced due to error expansion, into current pixel and its prediction context. They have tested their technique by using three different predictors, MED, gradient-adjusted predictor, and simplified gradient-adjusted predictor. This technique provides improved performance; however, computational complexity increases compared to simple prediction-error expansion approach.

Interpolation error expansion [66–68] is the variant of the difference expansion and prediction error expansion techniques. In interpolation error expansion based reversible watermarking approaches; initially, a down-sampled image is computed which is half of the original image size. Then, the down-sampled image is expanded to the size of the original image by interpolating the missing pixels. Difference of the interpolated pixel and original pixel is called interpolation error, which is used for watermark embedding. Similarly, in 2010, Luo et al. [69] have reported a technique in which they have additively embedded the watermark in the interpolation error histogram.

2.5 Chapter Summary

In this chapter, reported literature in the field of the reversible watermarking is discussed. In the light of different reported techniques, field of reversible watermarking is divided into five groups. These groups are formed based on algorithm characteristics.

Next chapter will discuss the compression based reversible watermarking for medical images.
Chapter 3  COMPRESSION BASED REVERSIBLE WATERMARKING

In this chapter, compression based reversible watermarking and its application on medical imagery is discussed. In the field of medical science imagery, authentication and protection of medical images plays very important role. Some important constraints are required to handle while dealing with medical image watermarking. Watermarking process introduces distortion in the original image. These distortions are highly undesirable as they can affect image analysis and physician’s diagnosis. It is, therefore, very important to extract not only the embedded message from the received image but also to recover the original state of the image. This task can be accomplished through reversible watermarking [70–73].

Some researchers have introduced the idea of defining region of interest (ROI) to embed watermark in medical images. ROI is the part of an image that contains important information for physicians. Therefore, to preserve that information, these regions are skipped for embedding. In [74], Wakatani et al. proposed an application in which embedding is performed only in non-ROI regions. Similarly, in [75], a technique is reported in which watermark is embedded around ROI using GA.

In this chapter, block based intelligent reversible watermarking using companding technique is proposed. This GA based reversible watermarking (GA-revWM) technique intelligently computes threshold matrix (Tmap) that is used in companding process. To compute the optimum Tmap, GA exploits the information of wavelet coefficient structure in a block. Companding is performed in IWT domain and the watermark is embedded on compressed IWT coefficients.

3.1 Proposed GA-revWM Approach

The main architecture of GA-revWM is shown in Figure 3.1. It is based on two phases; namely, training and testing. In training phase, an optimal/near-optimal Tmap is computed through GA, which is used in testing phase during companding process.

3.2 Training Phase (GA based Optimization)

GA is a random search technique used for solving optimization problems. It is based on biological evolutionary phenomenon.
Chapter 4  
Reversible Watermarking for SFF Approach based 3D Images

Figure 3.1: Block diagram of GA-revWM Approach.

In this phase, binary GA is employed to generate optimal/near-optimal Tmap. Initially, GA generates a population randomly. Each individual of the population could be a possible solution of the problem. Population size determines the total number of individuals in the formulated problem. Each individual is evaluated using fitness function. From one generation to the next, GA evolves population depending upon the fitness value of individuals. Individuals with poor fitness value are discarded during evolution and to fill the space, new individuals are generated through crossover and mutation operations.

Optimal solution depends upon, how well the problem is formulated in the fitness function. In GA-revWM technique, fitness function is designed to optimize the tradeoff between capacity and imperceptibility. Initially, image is decomposed into IWT sub-bands and then, HL, LH, and HH sub-bands are divided into blocks. After this, the watermark embedding process is performed as explained in Section 3.3.1. Finally, fitness is computed from the marked image by using the following relation:

\[
\text{Fitness} = \begin{cases} 
10 \times \log_{10}\left(\frac{(255^2)}{\text{MSE}}\right), & \text{if } \text{effective\_payload} \geq \text{desired\_effective\_payload} \\
0, & \text{otherwise} 
\end{cases} 
\]  

(3.1)

where, \text{effective\_payload} is the total number of embedded watermark bits, \text{desired\_effective\_payload} is the amount of watermark bits that has to be embedded, and \text{MSE} represents mean square error of the marked image as shown in Equation (3.2):
\[ MSE = \frac{1}{X \times Y} \sum_{r=0}^{X-1} \sum_{s=0}^{Y-1} |I_o(r,s) - I_w(r,s)|^2 \]  \hspace{1cm} (3.2)

where \( I_w \) and \( I_o \) are the watermarked and original images, respectively. \( X \times Y \) represents the image size. The individual with the highest fitness value is considered as the best individual.

GA keeps on evolving the population towards the most optimal solution until the maximum number of generation reached. At the end, it returns the most optimal individual as the solution to the formulated problem.

### 3.3 Testing Phase

In this phase, the matrix \( T_{map} \) obtained in the training phase is utilized to perform embedding and extraction processes. While embedding, \( T_{map} \) is utilized to perform companding process, and during extraction process, \( T_{map} \) is used for image recovery.

![Block diagram of watermark embedding procedure.](image)

#### 3.3.1 Information Embedding Procedure

The main steps of watermark embedding process are shown in Figure 3.2. These steps are discussed in details in the upcoming sections.
3.3.1.1 Histogram Modification

After watermark embedding, there is a chance that the intensity value of some pixels may go outside the range of [0–255]. To tackle this problem, histogram modification is applied.

In histogram modification process, upper and lower grayscale values are shifted to provide the room for histogram expansion after embedding. Then, at receiving side, histogram recovery is performed after extraction procedure. The process is elaborated with an example. Consider an image of size 6×6 with \(2^8=256\) grayscale values as shown in Figure 3.3. From the figure, it can be observed that the range of the modified histogram is 1-254 instead of 0-255. After modification, grayscale 1 is merged into 2 and then, grayscale 0 is shifted to 1. Similarly, grayscale 254 is merged into 253 and then, grayscale 255 is shifted to 254.

![Figure 3.3: Histogram modification. (a) Original Image Data, (b) Modified Image Data.](image)

During histogram-modification, all 0’s are shifted to 1 and all 255 are shifted to 254. Therefore, there is no need to record these changes; because, actually all 1’s are 0’s and all 254’s are 255’s. However, 2’s in the histogram processed image are either original 2’s or the 1’s shifted to 2. Similarly, 253’s in the histogram modified image are either original 253’s or the 254’s shifted to 253. This is an ambiguous situation; therefore, for these grayscale values scan sequences are generated. For this, the histogram modified and the original images are scanned simultaneously. For scan sequence of 253, ‘1’ is recorded in the sequence, if ‘253’ is encountered in both the images. On the other hand, ‘0’ is stored in the sequence, if histogram modified image has value 253 for the pixel value 254 in the original image. Length of the sequence is equal to the number of 253’s in the histogram-modified image. By following the same steps, the scan sequence for grayscale 2 is computed [46].

Next step is to generate the bookkeeping data based on the scan sequences. The bookkeeping data store (say BDS) contains the following information.

\[
BDS = \{\text{Total Length of BDS} + \text{Number of Compressed Grayscales} + \text{First Right Hand Side Grayscale Histogram} + \text{Length of Right Hand Side Scan Sequence} + \text{Right Hand Side Scan Sequence} + \text{First Left} \} \]
Hand Side Grayscale Histogram + Length of Left Hand Side Scan Sequence + Left Hand Side Scan Sequence].

In the aforementioned example, to generate BDS, the number of compressed grayscales are 2 (i.e. ‘255’ and ‘0’), which can be stored in ‘4’ bits. First right hand side grayscale histogram is ‘254’ and this is stored in appropriate number of bits. Then, length of scan sequence ‘253’ stored in 16 or 32 bits, followed by scan sequence ‘253’. First left hand side grayscale histogram is ‘1’, which can also be stored in 4 bits. Length of Scan sequence ‘2’ is stored in 16 or 32 bits. At the end, scan sequence ‘2’ is appended. At the extraction side, BDS is required for image histogram recovery as shown in Figure 3.2; therefore, it is also embedded as an overhead data.

### 3.3.1.2 Integer Wavelet Transform

After histogram modification, next step is to decompose the image into wavelet sub-bands by employing IWT on the histogram-modified image. IWT does not involve any round off errors problem, therefore, we can recover the image without any loss of information. The wavelet family Cohen-Daubechies-Feauveau (CDF) (2,2) is used, as it provides higher capacity and better imperceptibility of the watermarked images [46].

![Image Wavelet Sub-bands](image)

**Figure 3.4:** Sub-bands split of an $X \times Y$ image.

After employing IWT, an image of size $X \times Y$ is transformed into four sub-bands, each of size $X/2 \times Y/2$ as shown in Figure 3.4. Beginning with upper left corner and moving in anti-clockwise manner, these bands are approximation sub-band ($LL$), vertical detail sub-band ($HL$), diagonal detail sub-band ($HH$), and horizontal detail sub-band ($LH$) [76]. $LL$ is highly sensitive band; it covers smooth area information of an image. Therefore, watermark embedding in this sub-band causes high perceptual distortion in the
overall image. Thus, this sub-band is skipped and watermark embedding procedure is applied on other three sub-bands only.

3.3.1.3 Companding Process and Data Embedding

The term “companding” is a successive process of compression and expansion operations. Mainly, it is used for data rate reduction in audio signals. This can be accomplished by selecting unequal quantization levels, which results in high signal to noise ratio. In 1988, Sklar [77] has presented the details of this process in his work.

In Compression, wider range of a signal is mapped to a narrower range and the original signal is recovered by applying expansion process. Ideal case of companding (i.e. without digitization) is shown in Equation (3.3)

\[
f_E(f_C(K)) = K
\]

(3.3)

where \( f_C \) and \( f_E \) are compression and expansion functions, respectively. \( K \) is a signal on which companding is performed.

Compression operation is applied mainly due to two reasons. First, to reduce the distortion in the marked signal so that underflow and overflow can be avoided. Secondly, the final watermarked signal should be close to the original signal so that noise in the signal is minimum.

In this work, Xuan et al.’s [46] companding expression with some modification is utilized. The modification is due to the block based nature of the proposed approach. In most of the images, the high frequency coefficients of IWT follow Laplacian-like distribution. Mainly, two types of features exist in the distribution [46]:

- After embedding process, the lower valued high frequency component coefficients are less likely to undergo underflow and overflow problem. Hence, there is no need to apply compression process on these coefficients. That is why, for this kind of coefficients, a linear function with unit slope is chosen, i.e. \( f_C(K) = K \).

- Higher valued high frequency IWT coefficients are more probable of introducing underflow or overflow problem after embedding. Therefore, these coefficients are compressed by employing a compression function with steep slope.

In view of the aforementioned situations, compression function chosen in this work is a piecewise linear function, as shown in Equation (3.4):
Chapter 4  
Reversible Watermarking for SFF Approach based 3D Images

\[ K_c = f_c(K) = \begin{cases} 
  K, & |K| < Tmap(i, j) \\
  \text{sign}(K) \left( \frac{|K| - Tmap(i, j)}{2} + Tmap(i, j) \right), & |K| \geq Tmap(i, j)
\end{cases} \tag{3.4} \]

where \( K_c \) and \( K \) are the compressed and original IWT coefficient, respectively. \( Tmap(i, j) \) is a threshold value for a particular block \((i, j)\) in which the coefficient \( K \) resides. If the total number of blocks in a sub-band is \( m \times n \) (as shown in Figure 3.5) then \( i = 1,2,\ldots,m \) and \( j = 1,2,\ldots,n \), depending upon the location of \( K \).

\[ T(1,1) \quad T(1,2) \quad T(1,n) \\
T(2,1) \quad T(2,2) \quad T(2,n) \\
T(m,1) \quad T(m,2) \quad T(m,n) \]

Threshold Matrix

\[ T(1,1) \quad T(1,2) \quad T(1,n) \\
T(2,1) \quad T(2,2) \quad T(2,n) \\
T(m,1) \quad T(m,2) \quad T(m,n) \]

\[ T(1,1) \quad T(1,2) \quad T(1,n) \\
T(2,1) \quad T(2,2) \quad T(2,n) \\
T(m,1) \quad T(m,2) \quad T(m,n) \]

\[ T(1,1) \quad T(1,2) \quad T(1,n) \\
T(2,1) \quad T(2,2) \quad T(2,n) \\
T(m,1) \quad T(m,2) \quad T(m,n) \]

\[ T(1,1) \quad T(1,2) \quad T(1,n) \\
T(2,1) \quad T(2,2) \quad T(2,n) \\
T(m,1) \quad T(m,2) \quad T(m,n) \]

\[ T(1,1) \quad T(1,2) \quad T(1,n) \\
T(2,1) \quad T(2,2) \quad T(2,n) \\
T(m,1) \quad T(m,2) \quad T(m,n) \]

Equation (3.4) cannot be applied to digital signals as it generates floating-point values. In order to apply it to the digital signals (e.g. images) quantized version of the companding function, \( f_{QC} \) and \( f_{QE} \) has to be utilized. Due to quantization, for some signal \( K \), we have

\[ f_{QE}(f_{QC}(K)) \neq K \tag{3.5} \]

namely, the error value is \( e = f_{QE}(f_{QC}(K)) - K \neq 0 \).
In order to recover the original IWT coefficient $K$ at receiving side, error values $e$ is required. The difference value $e$ combined with BDS is called overhead data. To make this technique blind and reversible watermarking, the overhead data is also embedded along with the actual watermark. Equation (3.6) and Equation (3.7) shows the quantized form of compression and expansion function, respectively.

$$\begin{align*}
K_c &= f_{qc}(K) = \begin{cases} 
K, & |K| < T_{map}(i, j) \\
\text{sign}(K) \left( \frac{|K| - T_{map}(i, j)}{2} \right) + T_{map}(i, j), & |K| \geq T_{map}(i, j)
\end{cases} \quad (3.6)
\end{align*}$$

$$\begin{align*}
K' &= f_{qe}(K_c) = \begin{cases} 
K_c, & |K_c| < T_{map}(i, j) \\
\text{sign}(K_c) \left( 2|K_c| - T_{map}(i, j) \right), & |K_c| \geq T_{map}(i, j)
\end{cases} \quad (3.7)
\end{align*}$$

Equation (3.8) is used to remove quantization error and to recover the original value of the coefficient.

$$K = K' + e \quad (3.8)$$

After companding function selection, next step is data embedding. Initially, IWT coefficients are compressed by employing compression function, i.e. $K_c = f_{qc}(K)$. In the embedding process, a bit $b \in \{0,1\}$ is appended after LSB of $K_c$, which produces $K_{cw}$. For example, let $K_c = 5 = (101)_2$ and $b = 1$ then $K_{cw} = (1011)_2 = 11$, it means $K_{cw} = 2 \times K_c + b$.

Value for $T_{map}$ has to be selected with great care. It effects on watermark capacity versus imperceptibility tradeoff. According to Equation (3.6), a coefficient is compressed, only if its value is greater than the corresponding threshold value. If, the threshold value is small; then, more number of coefficients are selected for compression. Therefore, coefficients alteration after embedding is small, which results in good watermark imperceptibility. However, corresponding watermark capacity suffers due to large number of compression error. For large threshold value, capacity is high due to less number of compression error but at the cost of low visual quality of the watermarked image.

### 3.3.2 Information Extraction Procedure

At the receiving side, watermarked image is fed to the watermark extraction procedure. This procedure performs extraction and recovery processes by following the same steps shown in Figure 3.6.

Initially, watermarked image is decomposed into wavelet sub-bands by applying IWT. Then, watermark bits are extracted from LSBs of the sub-bands coefficients by using expressions $b = \text{LSB}(K_{cw})$ and $K_c = (K_{cw} - b)/2$; where $K_{cw}$ is watermarked LH, HL, and HH sub-bands coefficient and $K_c$ is the
coefficient after watermark bit extraction. The extracted LSB bits are the collection of original watermark, \( T_{map} \), companding error \( e \), and \( BDS \); this information is separated. After extraction, the coefficients are still not in its original form.

![Block diagram of information extraction procedure.](image)

**Figure 3.6:** Block diagram of information extraction procedure.

### 3.3.3 Restoration of the Original Image

\( T_{map} \) obtained from extracted LSB bits is used to expand the compressed coefficient by using expression shown in Equation (3.7). After expansion, Equation (3.8) is employed to remove companding error from the coefficients. After this, inverse IWT is applied to transform the image back to the spatial domain. However, the image is still not in its original form due to histogram modification. To restore the image histogram, scan sequences are used from \( BDS \) in the same way as described in Section 3.3.1.1. At the end, the image attains its original state.
3.4 Experimental Results and Discussions

The grayscale images used in the experimental analysis of the proposed GA-revWM technique are shown in Figure 3.7. The size of each of these images is 512×512.

Figure 3.7: Images considered for accessing the performance of the GA-revWM.
Chapter 4  Reversible Watermarking for SFF Approach based 3D Images

GA-revWM is a block based technique and size of block is very important. Tmap is an overhead data, which affects the actual watermark capacity. Size of Tmap depends upon the size of the block; therefore, block size is an important parameter that can affect the system performance. Decreasing the block size, increases the size of Tmap that has to be sent to the extraction side in order to extract the watermark and restore the original image. Therefore, the number of blocks should be taken in a way to achieve good peak signal to noise ratio (PSNR) and effective payload. In this work, block size is taken as 16 thus producing a threshold matrix of size 16×16 (16×16=256 threshold values).

3.4.1 GA Related Implementation Details

In training phase, Matlab (Version: R2008b) based GA tool has been used to generate optimal/near-optimal Tmap. A binary GA simulation is carried out by keeping the number of generation 200 and population size 100. Rank function is used for fitness scaling, which utilizes the rank of each individual to scale the raw scores, rather than its absolute value. The rank of an individual is its position in the sorted scores. Rank fitness scaling removes the effect of the spread of the raw scores. Scattered crossover is employed for combining two individuals to form a new individual for the next generation. For mutation process, Gaussian mutation is used for providing genetic diversity. After crossover and mutation operations, Roulette selection function is used for choosing parents for the next generation.

3.4.2 GA based Tradeoff between Imperceptibility and Capacity

Figure 3.8 shows the GA simulations for Lena image. The simulation is carried out by varying the payload from 0.1 to 0.7. It can be observed that the value of PSNR increases after every generation, which implies that the performance improves from one generation to the next. Moreover, at the end of the simulation, GA produces the optimal/near-optimal Tmap, which gives the best results for a given payload. There exist certain individuals for which the companding error is too high that it decreases the effective payload enormously. Therefore, in order to overcome this problem, we give zero fitness to all those individuals that have to be discarded due to insufficient payload (as shown in Equation (3.1)). In Figure 3.8, zero values shows fitness values for these individuals. Tmap evolved using GA algorithm for 0.1 payload using Lena image, is shown in Figure 3.9. It is to be noted that each threshold of the matrix corresponds to a 16×16 block of the transformed image.
Chapter 4  
Reversible Watermarking for SFF Approach based 3D Images

Figure 3.8: GA simulation for Lena image

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>11</td>
<td>14</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>14</td>
<td>11</td>
<td>7</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>15</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>11</td>
<td>7</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>6</td>
<td>12</td>
<td>14</td>
<td>10</td>
<td>12</td>
<td>6</td>
<td>14</td>
<td>6</td>
<td>13</td>
<td>7</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>15</td>
<td>9</td>
<td>15</td>
<td>15</td>
<td>12</td>
<td>13</td>
<td>15</td>
<td>10</td>
<td>13</td>
<td>14</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>12</td>
<td>15</td>
<td>12</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>7</td>
<td>6</td>
<td>12</td>
<td>8</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>14</td>
<td>13</td>
<td>3</td>
<td>14</td>
<td>13</td>
<td>5</td>
<td>9</td>
<td>8</td>
<td>3</td>
<td>11</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>13</td>
<td>12</td>
<td>7</td>
<td>11</td>
<td>7</td>
<td>15</td>
<td>13</td>
<td>4</td>
<td>10</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>14</td>
<td>5</td>
<td>8</td>
<td>12</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>12</td>
<td>15</td>
<td>6</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>9</td>
<td>3</td>
<td>13</td>
<td>14</td>
<td>7</td>
<td>12</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>11</td>
<td>15</td>
<td>9</td>
<td>5</td>
<td>10</td>
<td>11</td>
<td>6</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>13</td>
<td>6</td>
<td>11</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>15</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>12</td>
<td>10</td>
<td>10</td>
<td>13</td>
<td>13</td>
<td>12</td>
<td>15</td>
<td>6</td>
<td>15</td>
<td>11</td>
<td>11</td>
<td>15</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>15</td>
<td>6</td>
<td>7</td>
<td>2</td>
<td>13</td>
<td>13</td>
<td>10</td>
<td>3</td>
<td>14</td>
<td>7</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>12</td>
<td>7</td>
<td>9</td>
<td>5</td>
<td>9</td>
<td>2</td>
<td>14</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>

Figure 3.9: Threshold Matrix for a 16×16 block of Lena image (payload=0.1)
Figure 3.10: Column (a) original input images, (b) watermarked images, (c) difference between original and watermarked images (difference of (a) and (b)), (d) restored images and (e) difference between original and restored images (difference of (a) and (d)).

Figure 3.10 shows the reversible capability of the proposed approach. In Figure 3.10, original images are shown in column (a), while column (b) shows the corresponding watermarked images. Difference of original and watermarked images is shown in column (c). It demonstrates the imperceptible embedding using the best-evolved Tmap. This difference is not visible with naked eye, so an enhancement technique is applied in order to make the difference visible. As the proposed GA-revWM is a reversible
watermarking technique; therefore, after watermark extraction at the receiving end original contents of the image are retrieved. The resultant restored images are shown in column (d) of Figure 3.10. In order to demonstrate the reversible capability of the proposed GA-_revWM technique, the difference of the original and restored images is computed. This difference is shown in Figure 3.10 column (e). It is to be noted that the difference column (e) is the set of black images, which indicate that the difference is zero. Reversibility performance is computed through the structured similarity index measure (SSIM). SSIM takes two images as a parameter and compute the quality or degradation in the second image with respect to the first image. It is a very effective measure to compute image quality. It is to be noted that, SSIM gives value ‘1’ for all the images shown in column (d), which proves the fact that both original and restored images are same. Therefore, the proposed technique can fully restore the image to its original state at the receiving side.

Figure 3.11: PSNR vs. Payload for block size 8×8 to 32×32 using Lena image.

Due to block-based nature of the proposed approach, the block size has enormous effect on the system performance. Figure 3.11 shows the system performance (capacity versus imperceptibility tradeoff) by varying the block size (from 8×8 to 32×32). It is clear from the figure that, as block size increases from 8×8 to 32×32 system performance decreases. As we decrease the block size, the number of coefficients to be handled by GA decreases. Therefore, the GA has better chances of learning the texture
related characteristics and coefficient distribution associated with that block. It can be observed that at 0.1 \( \text{bpp} \), our approach performs equally well for both the block sizes of 8×8 and 16×16. This might be due to the reason that at low payload, no compression is required for data embedding and hence the amount of header information is less and almost the same in both cases. However, at 0.2 \( \text{bpp} \), a sharp decrease in performance is observed for 16×16 block size. The reason for this behavior might be the higher number of compressed coefficients at 0.2 \( \text{bpp} \) compared to that of 0.1 \( \text{bpp} \) in case of 16×16 block size, consequently, increasing the amount of header information.

3.4.3 Performance Comparison with Existing Approaches

Figure 3.12 shows the comparison of proposed \( \text{GA-revWM} \) technique with that of Tian’s difference expansion approach [54], Xuan et al.’s distortion less data hiding approach [78], Xuan et al.’s wavelet spread spectrum approach [79], Xuan et al.’s companding approach [46], Usman et al.’s approach [80], and Lee et al.’s approach [81]. The performance evaluation is made in terms of imperceptibility versus capacity tradeoff using Lena image. Imperceptibility is computed in terms of \( \text{PSNR} \), and payload is represented in terms of bit per pixel (\( \text{bpp} \)). From the figure, it can be observed that, the proposed \( \text{GA-revWM} \) scheme gives better imperceptibility versus capacity tradeoff compared to the existing approaches. It is achieved due to the learning capability of \( \text{GA} \). From Figure 3.12 it can be seen that, payload of 0.7 is achieved against \( \text{PSNR} \) value 40.1 \( \text{dB} \), which is a significant improvement provided by the proposed approach. Figure 3.13 shows the performance analysis of proposed \( \text{GA-revWM} \) technique for 16-bit images. The effective payload for Cell image is set to 0.5; while for Gene image, it is 0.6.

The watermark embedding process is performed in \( HH, HL, \) and \( LH \) sub-bands only. Therefore, the maximum capacity that the proposed technique can offer (in a single run) is 0.75 \( \text{bpp} \). However, through multiple runs of watermark embedding process, we can achieve capacity more than 0.75 \( \text{bpp} \). In Figure 3.14, performance analysis by varying the payload from 0-1 for Lena image is presented. \( \text{SSIM} \) is another technique to compute the image imperceptibility/quality. Therefore, in Figure 3.14, results are computed by maximizing \( \text{SSIM} \) between original and watermarked image against a given payload. The fitness function used to compute these results is provided in Equation (3.9). In Figure 3.14, it is to be noted that the proposed \( \text{GA-revWM} \) gives improved results even in terms of \( \text{SSIM} \).

\[
\text{Fitness} = \begin{cases} 
-\text{SSIM} & \text{if effective\_payload } \geq \text{desired\_effective\_payload} \\
0 & \text{otherwise}
\end{cases}
\]

(3.9)
Chapter 4  Reversible Watermarking for SFF Approach based 3D Images

Figure 3.12: Performance comparison with existing approaches for Lena image.

Figure 3.13: PSNR versus payload for 16-bit grayscale images.
Figure 3.14: Performance comparison of GA-revWM approach with Xuan et al. approach [46] for payload 0-1.

The test time is computed for the proposed GA-revWM approach and Xuan et al. [46] technique. For this, simulation is run over the 512×512 Lena image. Xuan et al.’s [46] approach took 1.35 seconds while proposed GA-revWM needed 0.82 seconds to compute the results. However, the proposed GA-revWM approach consumes more time in training phase. It requires 45 minutes to compute optimal/near-optimal Tmap.

3.4.4 Performance on Medical Images

Table 3.1 and Table 3.2 present the performance evaluation of the proposed GA-revWM technique with Xuan et al.’s companding approach [46]. Table 3.1 and Table 3.2 show that the proposed GA-revWM is performing better compared to Xuan et al.’s [46] technique, in terms of PSNR and SSIM. The effective payload is set greater than 0.6, for X-ray and MRI images. Moreover, GA-revWM technique is reversible as it can recover the original state of the image successfully. Therefore, in medical imagery, ROI can also be chosen for watermark embedding.
### Table 3.1: PSNR comparison of Xuan et al. [46] with proposed technique, against different values of payload.

<table>
<thead>
<tr>
<th>Images</th>
<th>Effective Payload</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>Ref. [46]</td>
<td>49.8</td>
<td>46.2</td>
<td>43.4</td>
<td>42.2</td>
<td>41.2</td>
<td>40.3</td>
<td>39.0</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>55.8</td>
<td>48.2</td>
<td>45.8</td>
<td>43.7</td>
<td>42.4</td>
<td>41.0</td>
<td>40.1</td>
</tr>
<tr>
<td>Barbra</td>
<td>Ref. [46]</td>
<td>48.6</td>
<td>45.6</td>
<td>43.3</td>
<td>41.2</td>
<td>40.6</td>
<td>39.7</td>
<td>38.6</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>55.0</td>
<td>50.9</td>
<td>45.9</td>
<td>44.0</td>
<td>41.1</td>
<td>40.4</td>
<td>39.1</td>
</tr>
<tr>
<td>Baboon</td>
<td>Ref. [46]</td>
<td>44.4</td>
<td>41.9</td>
<td>39.9</td>
<td>38.8</td>
<td>38.0</td>
<td>37.2</td>
<td>36.5</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>43.9</td>
<td>47.5</td>
<td>49.3</td>
<td>44.6</td>
<td>42.4</td>
<td>39.1</td>
<td>37.9</td>
</tr>
<tr>
<td>Goldhill</td>
<td>Ref. [46]</td>
<td>50.0</td>
<td>46.6</td>
<td>44.4</td>
<td>42.9</td>
<td>41.8</td>
<td>40.9</td>
<td>40.2</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>54.0</td>
<td>51.8</td>
<td>49.2</td>
<td>46.5</td>
<td>43.4</td>
<td>41.7</td>
<td>40.6</td>
</tr>
<tr>
<td>MRI1</td>
<td>Ref. [46]</td>
<td>50.2</td>
<td>47.8</td>
<td>45.4</td>
<td>44.0</td>
<td>43.0</td>
<td>42.2</td>
<td>41.5</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>56.2</td>
<td>52.2</td>
<td>48.9</td>
<td>46.7</td>
<td>45.3</td>
<td>44.0</td>
<td>42.8</td>
</tr>
<tr>
<td>MRI2</td>
<td>Ref. [46]</td>
<td>52.3</td>
<td>49.7</td>
<td>48.1</td>
<td>46.6</td>
<td>45.2</td>
<td>44.3</td>
<td>43.2</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>56.3</td>
<td>52.2</td>
<td>50.1</td>
<td>48.3</td>
<td>46.5</td>
<td>45.4</td>
<td>44.9</td>
</tr>
<tr>
<td>SEM</td>
<td>Ref. [46]</td>
<td>52.4</td>
<td>49.0</td>
<td>47.2</td>
<td>45.5</td>
<td>44.8</td>
<td>44.1</td>
<td>43.5</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>55.9</td>
<td>52.4</td>
<td>49.6</td>
<td>48.1</td>
<td>46.8</td>
<td>45.7</td>
<td>44.9</td>
</tr>
<tr>
<td>X-Ray</td>
<td>Ref. [46]</td>
<td>53.8</td>
<td>50.0</td>
<td>48.1</td>
<td>46.0</td>
<td>45.6</td>
<td>44.8</td>
<td>44.1</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>56.6</td>
<td>52.7</td>
<td>49.8</td>
<td>48.3</td>
<td>47.1</td>
<td>46.2</td>
<td>44.6</td>
</tr>
</tbody>
</table>

### Table 3.2: SSIM comparison of Xuan et al. [46] with proposed technique, against different values of payload.

<table>
<thead>
<tr>
<th>Images</th>
<th>Effective Payload</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>Ref. [47]</td>
<td>0.994</td>
<td>0.988</td>
<td>0.983</td>
<td>0.978</td>
<td>0.973</td>
<td>0.968</td>
<td>0.963</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.998</td>
<td>0.992</td>
<td>0.987</td>
<td>0.981</td>
<td>0.962</td>
<td>0.968</td>
<td>0.963</td>
</tr>
<tr>
<td>Barbra</td>
<td>Ref. [47]</td>
<td>0.997</td>
<td>0.993</td>
<td>0.990</td>
<td>0.986</td>
<td>0.983</td>
<td>0.979</td>
<td>0.974</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.999</td>
<td>0.997</td>
<td>0.991</td>
<td>0.989</td>
<td>0.983</td>
<td>0.979</td>
<td>0.973</td>
</tr>
<tr>
<td></td>
<td>Effective Payload</td>
<td>SSIM</td>
<td>Ref. [47]</td>
<td>Proposed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------------</td>
<td>------------</td>
<td>-----------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baboon</td>
<td>0.33</td>
<td>0.998</td>
<td>0.991</td>
<td>0.986</td>
<td>0.982</td>
<td>0.977</td>
<td>0.972</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.998</td>
<td>0.998</td>
<td>0.993</td>
<td>0.987</td>
<td>0.978</td>
<td>0.970</td>
<td></td>
</tr>
<tr>
<td>Goldhill</td>
<td>0.4</td>
<td>0.996</td>
<td>0.992</td>
<td>0.989</td>
<td>0.985</td>
<td>0.981</td>
<td>0.979</td>
<td>0.975</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.998</td>
<td>0.998</td>
<td>0.996</td>
<td>0.992</td>
<td>0.986</td>
<td>0.981</td>
<td>0.973</td>
</tr>
<tr>
<td>MRI1</td>
<td>0.6</td>
<td>0.998</td>
<td>0.995</td>
<td>0.991</td>
<td>0.988</td>
<td>0.986</td>
<td>0.983</td>
<td>0.980</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.999</td>
<td>0.998</td>
<td>0.996</td>
<td>0.993</td>
<td>0.991</td>
<td>0.989</td>
<td>0.986</td>
</tr>
<tr>
<td>MRI2</td>
<td>0.6</td>
<td>0.945</td>
<td>0.944</td>
<td>0.943</td>
<td>0.942</td>
<td>0.940</td>
<td>0.939</td>
<td>0.930</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.999</td>
<td>0.996</td>
<td>0.993</td>
<td>0.990</td>
<td>0.987</td>
<td>0.985</td>
<td>0.982</td>
</tr>
<tr>
<td>SEM</td>
<td>0.6</td>
<td>0.992</td>
<td>0.989</td>
<td>0.985</td>
<td>0.980</td>
<td>0.976</td>
<td>0.971</td>
<td>0.967</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.994</td>
<td>0.992</td>
<td>0.987</td>
<td>0.984</td>
<td>0.980</td>
<td>0.975</td>
<td>0.971</td>
</tr>
<tr>
<td>X-Ray</td>
<td>0.6</td>
<td>0.998</td>
<td>0.995</td>
<td>0.992</td>
<td>0.989</td>
<td>0.986</td>
<td>0.983</td>
<td>0.981</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.999</td>
<td>0.998</td>
<td>0.996</td>
<td>0.994</td>
<td>0.992</td>
<td>0.989</td>
<td>0.986</td>
</tr>
</tbody>
</table>

### 3.5 Chapter Summary

In this Chapter, an intelligent reversible watermarking approach, \textit{GA-revWM}, for medical images is discussed. In this technique, companding technique is employed to attain higher PSNR as well as \textit{SSIM} values for the images and is controlled using a threshold matrix. The value of threshold has to be selected very carefully as it greatly effects the watermark capacity. Higher value of threshold decreases the companding process and the corresponding companding error, which results in increase in effective payload. Therefore, variation in threshold value affects the capacity versus imperceptibility tradeoff.

The learning capability of \textit{GA} is employed to compute optimum threshold value for each block of wavelet sub-band. Experimental analysis showed that \textit{GA-revWM} technique outperforms the other existing techniques. Since, it is reversible; therefore, sensitive regions of medical images can also be used for watermark embedding, which results in an increase in watermark capacity.

Next chapter discusses the reversible watermarking for images having \textit{3D} information. This chapter focuses on images obtained from shape from focus (\textit{SFF}) approach.
Chapter 4  REVERSIBLE WATERMARKING FOR IMAGES WITH 3D INFORMATION

3D images are usually investigated in terms of 2D image and its corresponding depth information called depth map. Images with their depth information are examined and utilized quite extensively in the field of computer vision. In image focus analysis, different approaches used to generate 3D shape or depth map of an object. In the paradigm of shape from X (defocus, shading, motion, stereo, etc.), shape from focus (SFF) is the most famous technique to be used for the computation of 3D information of an object. Secure and authentic communication of 2D images along with their corresponding depth map is essential in many scenarios like, military and medical imagery. For example, in case of a legal interpretation, classified 3D shape information may need to be transfer to a forensic pathologist for online analysis of injuries and skin condition. Recently, safe communication of medical images and videos between island and mainland hospitals for online discussion or tele-surgery is also becoming important [82]. Additionally, in DNA studies and microsurgery, analysis of 3D information is quite valuable.

In this chapter, our aim is to analyze and develop a technique that would provide security to 3D images obtained from microscopic system by employing SFF approach. For this purpose, depth information of an object is used to embed as a watermark in its corresponding 2D image. The developed watermarking system is fragile in nature and is unable to resist any type of attack or signal processing operations. The fragile property is used to incorporate the authentication capability of the system. To further enhance the security provided by the system, Rivest-Shamir-Adleman (RSA) encryption based strategy is used [83]. This work is based on implementation of two basic approaches, the generation of depth map through optical microscopic system, and the reversible watermarking.

From the analysis of any grayscale images, it is concluded that, the distribution of binary 1’s and 0’s are quite similar in the lower bit-plane compared to the higher bit-plane. In this respect, image transformed to the frequency domain is expected to deliver larger bias between 1’s and 0’s. Therefore, watermark is embedded in the transformed domain. To avoid the round off error, integer discrete wavelet transform (IDWT) is used with CDF as a lifting scheme. IDWT maps integers to integers and is employed by JPEG2000.

The proposed technique embeds the data block wise in level 2 and level 1 wavelet coefficients. In order to improve the imperceptibility of the system, embedding is performed in HH, HL, and LH wavelet
sub-bands only. These sub-bands composed of high and middle frequency coefficients that introduce less distortion after embedding.

GA is used for optimization of capacity versus imperceptibility tradeoff. The purpose of achieving high imperceptibility is that the embedded watermark is less probable of being detected. GA is an intelligent search approach that evolves randomly generated population based on fitness function. Recently, various interesting CI approaches have been established in the area of image processing and pattern recognition [10], [84–86]. In this work, GA is employed to evolve optimal/near-optimal threshold matrix $T_{map}$. $T_{map}$ is computed under the constraint that no overflow and underflow occurs after watermark embedding along with the optimum capacity versus imperceptibility tradeoff. For this reason, no histogram modification is required; consequently, size of auxiliary information is reduced. $T_{map}$ is the information that is essential for extraction and recovery processes at the receiving end. Therefore, it is also embedded in the cover work along with the watermark. $T_{map}$ is also utilized in authentication process.

4.1 Depth Map Computation

In literature, several algorithms in spatial as well as in frequency domain are reported to compute focus measure. For example, gray level variance, Tenenbaum focus measure, threshold absolute gradient, sum of modified Laplacian, and modified Laplacian. In order to enhance the depth map quality, some approximation method is employed after focus measure computation. For this, Mahmood et al. [87] published an approximation technique including focused image surface and Gaussian interpolation. Similarly, Mehmood et al. [88] reported another technique for depth map computation. In their approach, they utilized genetic programming to combine different focus measures.

Fourier transform (FT) is one of the fundamental algorithms used for information extraction and spectrum analysis of the images. However, FT works in frequency domain, which results in loss of some essential spacial information. Several other transforms (e.g. wavelet transform (WT) and short time frequency transformation (STFT)) are reported to circumvent this problem. S-transform is the extension of STFT and WT, it provides better time-frequency analysis. In this work, S-transform is used for focus quality measure. To measure the focus quality, the first step is to capture a sequence of images $I_z(x,y)$. For this, image detector of optical microscope is moved along the optical axis in small equidistance steps. The sequence consist of $Z$ images each of size $X\times Y$. Consider a window of size $N\times M$ around each pixel $(x,y)$ in the image sequence $Z$ [89]. Frequency component of S-Transform, for 2D discrete signal is computed as:
\[
S(p_{T_x}, q_{T_y}, \frac{n}{N_{T_x}}, \frac{m}{M_{T_y}}) = \sum_{n'=0}^{N-1} \sum_{m'=0}^{M-1} H \left(\frac{n'+n}{N_{T_x}}, \frac{m'+m}{M_{T_y}}\right) \exp\left(-\frac{2\pi^2 n'^2}{n^2}\right) \\
\exp\left(\frac{2\pi^2 m'^2}{m^2}\right) \exp\left(i\frac{2\pi\mu' q}{M}\right)
\]  
(4.1)

where \(H(\cdot)\) is FT of a window of size \(N \times M\). \(T_x\) and \(T_y\) are sampling intervals along \(x\) and \(y\) directions; \(N_{T_x}\) and \(M_{T_y}\) represent frequency coordinates. \(n=0,1,2,3,\ldots, N-I\) and \(m=0,1,2,3,\ldots, M-I\) are the transformed domain indices; while, \(p=0,1,2,3,\ldots, N-I\) and \(q=0,1,2,3,\ldots, M-I\) are the spatial indices along \(x\) and \(y\) directions.

The average value of the transform is measured as \(S(p_{T_x}, q_{T_y}, 0, 0)\). It has to be excluded from the sharpness/focus measure. Therefore, for a pixel \((x,y)\), focus measure is computed using relation shown in Equation (4.2).

\[
F_{ST} = \sum_{p=0}^{N-I} \sum_{q=0}^{M-I} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} S(p_{T_x}, q_{T_y}, \frac{n}{N_{T_x}}, \frac{m}{M_{T_y}}) ^2 ; (m,n) \neq (0,0)
\]  
(4.2)

After focus measure computation at each pixel in the sequence, the focus volume \(V_z(x,y)\) can be calculated through the relation presented in Equation (4.3).

\[
V_z(x,y) = F_{ST}(I_z(x,y)) , \quad x=0,1,\ldots,X-1; \quad y=0,1,\ldots,Y-1; \quad z=1,2,\ldots,Z
\]  
(4.3)

It is to be noted that, instead of using fixed size Gaussian window as used in case of \(STFT\); 2D Gaussian function whose width is a function of frequency is used to multiply with \(FT\). In order to improve the robustness, initial focus values are summed up within a small 3D window of size \((2W+1)\) [90], and a refined focus value is given as:

\[
V'_z(x,y) = \sum_{\alpha=x-W}^{x+W} \sum_{\beta=y-W}^{y+W} \sum_{\gamma=z-W}^{z+W} V_z(\alpha,\beta)
\]  
(4.4)

where \(\alpha, \beta, \gamma\) are the running indices used in summation, their values vary from \(x-W, y-W, z-W\) to \(x+W, y+W, z+W\), respectively. The window size is critical and affects the accuracy of the depth map. A small window is unable to suppress the noise while a larger window causes over smoothing of the object shape and more likely removes edges. Malik and Choi [90] studied the effect of the windows sizes ranging from \(3x3\) to \(15x15\). They concluded that larger window decreases the accuracy of the depth maps and suggested the use of smaller windows for good depth map generation. Therefore, in Equation (4.1) and Equation (4.2), window of size \(7x7\) (i.e. \(N=7\) and \(M=7\)) is chosen around each pixel to compute S-transform. While computing the results for S-transform, \(\lfloor N/2 \rfloor\) boundary pixels have been neglected to
avoid padding at the boundaries. In Equation (4.4), window size $W$ is empirically set as 5. To resolve the boundary pixel problem for the window operation, we have neglected 2 pixels from each side in the $xy$ plane and two image frames from top and bottom. A frame number ($Z_{\text{focus}}$), where a pixel position gives maximum focus measure (obtained using Equation (4.4)), is picked as the depth of that point, as shown below:

$$D(x, y) = Z_{\text{focus}} = \arg \max_{z} (V'_z(x, y))$$

(4.5)

Figure 4.1: Optical microscopic system for SFF approach.

Setup of optical microscopic system is shown in Figure 4.1. In case of SFF algorithm, a stack of images is acquired using a simple camera by varying the distance of the object along the optical axis. The images are taken at equidistance microscopic steps (represented by $\Delta s$) with respect to a reference plane. The frames are stacked just like computed tomography (CT) slices. An object of varying surface would have different valleys and peaks. Different points on the object will thus be focused in different frames. The focusing is measured through a focus measure (Equation (4.2) – Equation (4.4)), which returns a real value for a pixel position at each frame. The value of the focus measure increases as the image sharpness
increases and attains the maximum value for the best-focused pixel. The frame number (integer value), where a pixel position gives maximum focus measure is picked as the depth of that pixel.

i. **Depth Map Compression**

While analyzing the characteristics of depth maps, we have observed that there is certain amount of redundancy in depth maps. Therefore, in order to reduce the payload, depth map is compressed by employing lossless compression technique. In the proposed technique, Huffman coding is used. The basic principle behind Huffman coding is that, for more common characters use shorter bit patterns and for less common characters use longer bit patterns. This banishes the redundant data and reduces the payload, which in turns improves the visual quality of the host image.

ii. **Depth Map Encryption**

To make the system more secure, the compressed depth map is encrypted by employing RSA algorithm [83]. RSA is a famous technique for encryption. It works in three major steps; namely, key generation, encryption, and decryption. It is a public-key based cryptosystem, which is used to encrypt the message. Public key can be known to everyone. Messages encrypted with the public key can only be decrypted by using the private key, so only the authorized person knows the private key. Key length for RSA encryption is taken as 1024 bits, this is the minimum limit taken for the secure transmission of messages. The details of RSA algorithm is provided in Algorithm 4.1.

Algorithm 4.1: RSA Algorithm

```plaintext
// p, q = randomly generated prime numbers, they should be of the same length.
// n = modulus for both public and private keys.
// e = public key exponent, it should not be small.
// d = private key exponent.
// m = message to be encrypted, c = encrypted message.

Step I: Key Generation
01: n = p \times q
02: \phi(p,q) = (p-1) \times (q-1)
03: e = \{ e | 1 < e < \phi(p,q) \text{ AND } \text{greatest_common_divisor}(e, \phi(p,q)) = 1 \}
04: d = \{ d | \text{mod}(d \times e, 1 + \phi(p,q)) = 1 \} \text{ // i.e. } (de - 1) \text{ is evenly divided by } (p-1)(q-1)
05: \text{Public Key} = (n, e)
06: \text{Private Key} = (d)

Step II: Encryption
07: c = \text{mod}(\text{power}(m, e), n)

Step III: Decryption
08: m = \text{mod}(\text{power}(c, d), n)
```
4.2 **Tmap Generation**

The reversible watermarking algorithm presented here required a threshold matrix to perform embedding. Size of the matrix is equal to the number of blocks present in the image. Value of the matrix depends upon the maximum allowed threshold value and the maximum change allowed for that application. To compute threshold value for a block, IDWT is employed on the block upto 2nd level, as shown in Figure 4.2. Two approaches are proposed for the generation of Tmap.

a. Adaptive Threshold based Reversible watermarking (ATRW)
b. Genetic Algorithm based Reversible Watermarking (GARW)

![IDWT decomposition of an \(N_B \times N_B\) block showing different sub-bands.](image)

Figure 4.2: IDWT decomposition of an \(N_B \times N_B\) block showing different sub-bands.

### 4.2.1 Adaptive Threshold based Reversible Watermarking (ATRW)

ATRW is based on the concept of grid search based Tmap generation. Initially, Tmap is assigned zero values. Image is divided in blocks each of size \(N_B \times N_B\), then each block is decomposed in wavelet subband by employing IDWT upto 3rd level. For Tmap generation, bit ‘1’ is embedded at \(HH1, HL1, LH1, HH2, HL2, LH2, HH3, HL3\) and \(LH3\) sub-bands. Embedding is performed only for those coefficients whose values are less compared to the threshold calculated for that sub-band. Let, \(K\) represents the frequency domain coefficient, \(T_q\) is the threshold value, and \(b\) represents a watermark bit; then, expression for watermark embedding is shown in Equation (4.6),

\[
K' = \begin{cases} 
2 \times K + b, & \text{if } |K| < T_q \\
K + T_q, & \text{if } K \geq T_q \\
K - (T_q - 1), & \text{if } K \leq -T_q 
\end{cases}
\] (4.6)
where $K'$ is coefficient after watermark embedding and $q=1, 2, 3$ for $(LH1 \text{ and } LH2), (HL1 \text{ and } HL2)$ and $(HH1 \text{ and } HH2)$ sub-bands, respectively. For $LH3, HL3, \text{ and } HH3$ sub-bands, $T_q = \{T_1, T_2, T_3\} = \{2, 3, 3\}$, respectively. Values of threshold in $Tmap$ is selected according to the need of an application. If threshold value is increases then watermark embedding is also increases as more coefficients satisfy the embedding condition. Increasing embedding capacity results in quality degradation. The sub-bands containing large high frequency contents are more appropriate for embedding. The reason behind this fact is that, human visual system ($HVS$) is more sensitive to the low frequency distortion. Therefore, to improve the watermark imperceptibility, maximum embedding is performed in high frequency regions and higher value of threshold is selected for these regions. Moreover, secret key-based permutation is applied on the depth map, to keep the secrecy of the hidden information even after the algorithm is made public.

After bit embedding, inverse $IDWT$ is applied to transform the block from frequency domain to spatial domain. Minimum and maximum values of the block is checked to investigate whether the block has undergone overflow/underflow problem. If the minimum or maximum values are beyond $[0-255]$ bounds, then this block is skipped for further processing. If no overflow/underflow is encountered then change in error ($CIE$) is computed for that block by using the following relation:

$$CIE = |MSE - C_{ERROR}(i, j)|$$

(4.7)

where, $MSE$ represents the deviation between original and watermarked block using the relation

$$MSE = \left[\sum (original\_block - watermarked\_block)^2 / (block\_size)\right]$$

and $C_{ERROR}$ is initialized with zero for each block at the start. $CIE$ is the value that records the change in $MSE$. During iteration, when threshold value is increased then the embedding capacity also increases. To impose the limit on quality degradation of an image, $CIE$ value is compared with maximum allowable change in error ($CIE_{MAX}$). If $CIE$ is greater than $CIE_{MAX}$ then that block is skipped for further processing; otherwise, the threshold $T_i$ and $MSE$ for block $b(i, j)$ is recorded at $(i^{th}, j^{th})$ location of $Tmap$ and $C_{ERROR}$, respectively. At the end of each iteration, threshold value is incremented as:

$$T_i = T_i + 2, \; T_2 = T_i + 1, \; T_3 = T_i + 2$$

(4.8)

The algorithm continues till the threshold value $T_i$ exceeds maximum allowable threshold limit $T_{MAX}$. These steps are repeated for each block and threshold value for all image blocks are computed. At the end of the algorithm, $Tmap$ is obtained to be used in watermark embedding and extraction procedures. It is to be noted that zeros in $Tmap$ means, the corresponding blocks are not suitable for embedding; since it
produces values beyond the limit of [0-255] after embedding. Depending upon nature of applications, values of $T_{MAX}$ and $CIE_{MAX}$ are adjusted empirically between 2-20 and 10-100, respectively. Figure 4.3 shows the block diagram of this process, and Algorithm 4.2 contains high level description of the steps involved in $Tmap$ generation.

**Figure 4.3:** Block diagram of $Tmap$ generation using ATRW approach.

Algorithm 4.2: $Tmap$ Generation using ATRW.

01: // X,Y: Image size along x and y direction, $N_B$: Block size;
02: // $B_{MAX}$: maximum number of block, $B_i$: IDWT of $i$th block, $b_W^i$: ith Watermarked block in IDWT domain, $b_W$: ith Watermarked block in spatial domain
03: // $Tmap$: Threshold matrix, $T_{MAX}$: Maximum threshold value of a block
04: // $CIE$: Change in error, $CIE_{MAX}$: Maximum change in error
05: // $C_{ERROR}$: Mean square error of a block
06: // $T_1$, $T_2$, $T_3$: Threshold for LH, HL, HH band respectively, set $T_1 = 0$. 

---

```
01: // X,Y: Image size along x and y direction, $N_B$: Block size;
02: // $B_{MAX}$: maximum number of block, $B_i$: IDWT of $i$th block, $b_W^i$: ith Watermarked block in IDWT domain, $b_W$: ith Watermarked block in spatial domain
03: // $Tmap$: Threshold matrix, $T_{MAX}$: Maximum threshold value of a block
04: // $CIE$: Change in error, $CIE_{MAX}$: Maximum change in error
05: // $C_{ERROR}$: Mean square error of a block
06: // $T_1$, $T_2$, $T_3$: Threshold for LH, HL, HH band respectively, set $T_1 = 0$. 
```
for i ← 1 to $B_{\text{MAX}}$ do // select a different block

while ($T_1 \neq T_{\text{MAX}}$) do // Check for the threshold limit

$T_1 = T_1 + 2$; $T_2 = T_1 + 1$; $T_3 = T_1 + 2$; // update threshold for each block

$B_i^{\text{IDWT}}(b_i)$ \hspace{1cm} // Compute $N_B \times N_B$ block IDWT up-to level 3

$B_w^i$ = Embed_Watermark() \hspace{1cm} // perform watermark embedding using Equation (4.6)

$b_w^i = \text{invIDWT}(B_w^i)$ \hspace{1cm} // Compute inverse IDWT

if ($b_w^i$ is within bound [0-255]) then // Check for the bounds after embedding

Compute CIE and $MSE$ \hspace{1cm} // Calculate change in error and mean square error

if ($CIE < CIE_{\text{MAX}}$) then//Check if change in error doesn’t exceeds the maximum allowable change in error

$Tmap^i = T_1$; $C_{\text{ERROR}}^i = MSE$; // update $Tmap$ and $C_{\text{ERROR}}$

end if

end if

end while

end for

4.2.2 Genetic Algorithm based Reversible Watermarking (GARW)

In this technique, intelligent search of GA is employed to compute optimal/near-optimal $Tmap$. GA evolves randomly generated population to search for the optimal solution. Each individual of a population is a potential solution of the problem. In each generation, population is sorted based upon their fitness values. Individuals having poor fitness values are discarded and to fill that space, new individuals are generated by employing crossover and mutation operations. The new population is processed in the next generation and this process continues until maximum number of generation is reached. At the end of the algorithm, an individual with best fitness value is returned as a solution to the formulated problem.

Accuracy of the solution depends upon, how well the problem is formulated in the fitness function. For the current situation, fitness function is developed to optimize the imperceptibility versus capacity tradeoff, without arousing overflow/underflow problem. Initially, GA generates a random population. Each individual of the population is $Tmap$ and the fitness value for each individual is computed. In fitness function, initially the image is split into non-overlapping blocks, each of size $N_B \times N_B$. Then, each block is transformed into wavelet sub-bands by applying $IDWT$ upto $2^{nd}$ level. Compressed and encrypted depth map is embedded in block’s $LH_2$, $HL_2$, $HH_2$, $LH_1$, $HL_1$ and $HH_1$ sub-bands using Equation (4.6). After bit embedding, inverse $IDWT$ is taken to transform the block from frequency domain to spatial domain. After processing all blocks, a watermarked image $I'$ is obtained. $Tmap$ is also required at the receiving side to perform extraction and recovery processes. Therefore, next step is to embed $Tmap$ in the
Chapter 4  
Reversible Watermarking for SFF Approach based 3D Images

watermarked image. Before embedding, $Tmap$ is compressed by using Huffman coding. Then watermarked image $I'$ is decomposed into wavelet sub-band by employing $IDWT$ upto third level and $Tmap$ is embedded in $HH3$, $HL3$, and $LH3$ sub-bands using threshold $T_z$ to be $\{2, 3, 3\}$, respectively.

![Figure 4.4: Block diagram of $Tmap$ generation using GARW approach.](image)

After $Tmap$ embedding, inverse $IDWT$ is applied to get the final watermarked image $I''$. Depth map is an important information required at the receiving side. Therefore, it is checked that whether the $Tmap$ allow enough capacity to embed the full length of the compressed and encrypted depth map. If not, then that individual is discarded by setting zero fitness value. Otherwise, imperceptibility of the image $I''$ is calculated by computing $PSNR$ between the original and $I''$ images. Since, the objective is to maximize the imperceptibility for a given capacity. Therefore, $PSNR$ value is assigned to the fitness parameter of an individual. The process is continued for each individual from generation to generation until maximum number of generation is met. At the end of $GA$ iterations, individual with best fitness value is returned as a
Chapter 4  
Reversible Watermarking for SFF Approach based 3D Images

\textit{Tmap} to be used in our technique. The steps involved in \textit{Tmap} computation using \textit{GARW} is shown in Figure 4.4.

4.3 Information Embedding

Once \textit{Tmap} is generated, it is used in the depth map embedding process. The process of information embedding is executed in two phases, as shown in Figure 4.5.

In the first stage, compressed and encrypted depth map is embedded in the images by using \textit{Tmap}. For this, image is divided into non-overlapping blocks each of size $N_b \times N_b$. Each block is decomposed in wavelet sub-band by employing IDWT upto 2nd level. Then, compressed and encrypted depth map is embedded at $HH1$, $HL1$, $LH1$, $HH2$, $HL2$ and $LH2$ sub-bands using Equation (4.6). Embedding is performed only for those coefficients whose values are less compared to the threshold computed for that particular sub-band of the particular block. After embedding, inverse IDWT is taken to get watermarked block $b_w(i,j)$. After processing all the blocks to embed depth map, watermarked image $I'$ is obtained.

As different threshold values are used for embedding in different blocks. Therefore, \textit{Tmap} is also required at the receiving side to perform extraction and recovery processes. In second stage of the algorithm, \textit{Tmap} is embedded in the watermarked image $I'$. Before embedding, \textit{Tmap} is compressed by using Huffman coding. Then, the watermarked image $I'$ is decomposed into wavelet sub-band by employing IDWT upto third level and \textit{Tmap} is embedded in $HH3$, $HL3$, and $LH3$ sub-bands using threshold $T_Z$ to be \{2, 3, 3\}, respectively. The block size $N_b$ is also embedded along with the header information. Value of $T_Z$ is empirically selected by considering the lower coefficient values of 3rd level sub-bands. Threshold values are selected low to reduce the distortion produced by this embedding. After \textit{Tmap} embedding, inverse IDWT is applied to get final watermarked image $I''$. In this work, the header information is embedded in 3rd level coefficients, separate from the actual watermark. Therefore, it does not affect the effective payload of the system.

It is worth noting that, different threshold values are used for different sub-bands. It is due to the fact that distribution of high frequency coefficients are different in different sub-bands. Moreover, large number of high frequency coefficients exist in $HH$ and $HL$ sub-bands, compared to $LH$ sub-bands. Therefore, the threshold value for $HH$ and $HL$ sub-bands are selected slightly high compared to $LH$ sub-band, so that, more coefficients are selected for embedding in $HH$ and $HL$ sub-band compared to $LH$ sub-band.

It is to be noted that the values recorded in \textit{Tmap} are the values of threshold $T_1$; while, $T_3$ and $T_2$ are derived from $T_1$ using relation shown in Equation (4.8). This is the essence of the proposed technique that
helps in improving the watermarked image quality. Algorithm 4.3 contains high level description of the watermark embedding process.

**Algorithm 4.3: Reversible Hiding of Depth Information (Embedding Procedure)**

01: // IMGw: Watermarked image in IDWT domain, img_w: Watermarked image in spatial domain
02: // X,Y: Image size along x and y direction, NB: Block size;
03: // B_MAX: maximum number of block, B: IDWT of ith block, B_w: ith Watermarked block in IDWT domain, b_w: ith Watermarked block in spatial domain
04: // D: Depth information of 2D image
05: // Tmap: Threshold matrix,
06: D=Generate_Depth_Map() // Generate depth map of 2D image, compress and encrypt it before embedding

![Figure 4.5: The basic block diagram of the embedding process.](image)
07: for $i \leftarrow 1$ to $B_{\text{MAX}}$ do // select a different block
08: \hspace{1em} $B' = \text{IDWT}(b')$ // Compute $N_B \times N_B$ block IDWT up-to level 2
09: \hspace{1em} $B_w = \text{Embed}_\text{Watermark}(D)$ // perform watermark embedding using Equation (4.6)
10: \hspace{1em} $b_w = \text{InvIDWT}(B_w)$ // Compute inverse IDWT
11: end for
12: $C_{\text{Tmap}} = \text{Compress}(\text{Tmap})$ //Compress $\text{Tmap}$ using Huffman Coding
13: $\text{IMG}_w = \text{IDWT}(\text{img}_w)$ // Compute IDWT of Image up-to level 3
14: $\text{IMG}_w = \text{Embed}_\text{Watermark}(C_{\text{Tmap}})$ // perform watermark embedding using Equation (4.6)
15: $\text{img}_w = \text{InvIDWT}(\text{IMG}_w)$ // Compute inverse IDWT

### 4.4 Information Extraction

The extraction process is also based on two stages, as shown in Figure 4.6. In the first stage, $\text{Tmap}$ is extracted by using fixed threshold $T_Z = \{2, 3, 3\}$. For this, image $I''$ is decomposed into wavelet sub-bands by employing $\text{IDWT}$ upto 3rd level using CDF filter. Header information is extracted from LH3, HL3, and HH3 sub-bands using $T_Z$. From header information, block size $N_B$ and compressed $\text{Tmap}$ are separated. $\text{Tmap}$ are decompressed by employing Huffman decoding; then, inverse $\text{IDWT}$ is applied to get image $I'$. In second stage, $\text{Tmap}$ obtained from the previous stage, is used to extract depth map. For this, image is divided in blocks each of size $N_B \times N_B$. Each block is decomposed in wavelet sub-band by employing $\text{IDWT}$ upto 2nd level. Then, compressed and encrypted depth map is extracted from HH2, HL2, LH2, HH1, HL1 and LH1 sub-bands using Equation (4.9),

\[
K = \begin{cases} 
\left\lfloor \frac{K'}{2} \right\rfloor, & \text{if } (-2T_q + 1) < K' < 2T_q \\
K' - T_q, & \text{if } K' \geq 2T_q \\
K' + (T_q - 1), & \text{if } K' \leq (-2T_q + 1)
\end{cases} \tag{4.9}
\]

where, symbol $\left\lfloor a \right\rfloor$ stands for the largest integer value smaller than $a$. With the help of the formula presented in Equation (4.9), depth map is extracted as well as original values are restored. After extraction, inverse $\text{IDWT}$ is applied to get block $b (i, j)$. The process of depth map extraction is repeated for all blocks. At the end, original image $I$ and compressed and encrypted depth map are obtained. The depth map is decompressed and decrypted to get its original form. Algorithm 4.4 shows the high level discription of the extraction algorithm.
Chapter 4  Reversible Watermarking for SFF Approach based 3D Images

Figure 4.6: The basic block diagram of the extraction process.

Algorithm 4.4: Reversible Hiding of Depth Information (Extraction Procedure)

01: // IMG_w: Watermarked image in IDWT domain, img_w: Watermarked image in spatial domain
02: // X, Y: Image size along x and y direction, N_B: Block size;
03: // B_MAX: maximum number of block, B_i: IDWT of ith block, B_w_i: ith Watermarked block in IDWT domain,
    b_w_i: ith Watermarked block in spatial domain
04: // D: Depth information of 2D image
05: // Tmap: Threshold matrix,
06: IMG_w = IDWT(img_w)  // Compute IDWT of Image up-to level 3
07: Tmap = Extract_Watermark(IMG_w)  // perform watermark extraction using Equation (4.9)
08: img_w = InvIDWT(IMG_w)  // Compute inverse IDWT
09: for i ← 1 to B_MAX do
10:    B_i = IDWT(b_w_i)  // Compute N_B×N_B block IDWT up-to level 2
11:    D = Extract_Watermark(B_i)  // perform watermark extraction using Equation (4.9)
12:    b_i = InvIDWT(B_i)  // Compute inverse IDWT
13: end for
Figure 4.7: Column (a) original input images, (b) watermarked images, (c) difference between original and watermarked images (difference of (a) and (b)), (d) restored images, and (e) difference between original and restored images (difference of (a) and (d)).
Figure 4.7 depicts the reversible capability of the proposed approach. In Figure 4.7, original images are shown in column (a), while column (b) shows the corresponding watermarked images. Difference of original and watermarked images is shown in column (c). It demonstrates the imperceptible embedding using the best-evolved $Tmap$. As the proposed technique is a reversible watermarking technique; therefore, after watermark extraction at the receiving end original contents of the image are retrieved. The resultant restored images are shown in column (d) of Figure 4.7. In order to demonstrate the reversible capability of the proposed technique, the difference of the original and restored images is computed. This difference is shown in column (e). It is to be noted that the difference column (e) shows the black images, which indicate that the difference is zero, which proves the fact that both original and restored images are the same. Therefore, the proposed technique can fully restore the image to its original state at the receiving side.

### 4.5 Authentication of the Host Image through $Tmap$

In this work, $Tmap$ is an information that is required at the receiving side for extraction and recovery procedures. To make this technique a blind one, $Tmap$ has to be embedded along with the depth map. This is an extra overhead to our system. However, a subsidiary advantage has been taken from this overhead by using $Tmap$ in authentication process. With the help of authentication, one can analyse that whether the received work has undergone any type of attack or not. In this work, two types of attacks are considered; namely, collage and manipulation attacks. Manipulation attack are easy to detect, but detecting a collage attack is problematic [26]. In order to detect collage attack, certain operations have to be performed on $Tmap$ at the embedding side. Figure 4.8 displays the steps of operations that are applied on $Tmap$. In the first step, correlation operation is performed on $Tmap$ with second level approximation sub-band $LL2$. Then, correlated $Tmap$ is compressed by applying Huffman coding. The process of correlating $Tmap$ with approximation sub-band helps in detecting collage attack. At the end, $BCH$ coding is employed on the correlated and compressed $Tmap$. $BCH$ coding is applied for error correction.

At the receiving side, first step is to extract $Tmap$. Then, $BCH$ decoding and Huffman decompression is applied on the extracted $Tmap$. To get original values of $Tmap$, it is decorrelated using $LL2$ sub-band. After recovering $Tmap$ it is used for depth map extraction. Since, extraction is reversible; therefore, after extraction the recovered image is the same as that of the original image. Then, $Tmap$ is computed for the recovered image. The two $Tmaps$, one generated using the recovered image and second extracted from the watermarked image are compared. If both $Tmaps$ are same, it implies that the received image is authentic.
Otherwise, it is concluded that image has undergone some kind of adversary attacks (either manipulation or collage).

4.6 Results and Discussion

4.6.1 Preparation of Data for Experimental Analysis

First, with the help of a microscopic control system, a sequence of 60 images of TFT-LCD color filter is captured. Microscope used in this setup is utilizing a CCD camera for image capturing and a step-motor with step size 2.5nm. Step-motor is used to control the distance of the object along the optical axis. The depth is then generated by applying SFF method based on S-transform discussed in Section 4.1. Generally, overlapping windows are used in SFF based approaches. However, in order to reduce the size of the depth map for subsequent embedding, we have used non-overlapping windows with the assumption that the depth remains the same for all the pixels in that window. Figure 4.9 shows the images and their corresponding depth maps. Due to limited resources, few 3D images obtained from 3D cameras have also been used in the experimental analysis.
Figure 4.9: (a-e) Sample images of TFT-LCD color filter, human face, human body, and simulated cone respectively. (f-j) show their corresponding depth maps.

In Figure 4.9, glass and human images and their corresponding depth maps are acquired using the 3D camera; while, depth maps of cone and 3D color filter are obtained using SFF approach. For experimental analysis, images of size 512×512 are used.
4.6.2 **Reversible Hiding of Depth Map using ATRW**

In this work, to make detailed comparisons with the proposed GARW approach, some new experiments are performed using our previous approach ATRW [50]. In previous ATRW work, it is reported that ATRW technique performs better than the conventional watermarking techniques. From the experimental analysis presented in [50], the value of $T_{\text{MAX}}$ is set to 16, while the appropriate value for parameter $CIE_{\text{MAX}}$ is 20.

![Figure 4.10: Effect of varying the block size from 8x8 to 64x64 on face image shown in Figure 4.9 (b), using ATRW approach.](image)

Block size has considerable effect in system’s performance. Therefore, it is necessary to choose a suitable value for the block size. In [50], results are computed by varying the block size from 16x16 to 32x32. In this work, system performance (in terms of imperceptibility versus capacity tradeoff) is analyzed by changing block size from 8x8 to 64x64 is presented, as shown in Figure 4.10. It can be observed from the figure that for larger block size, performance at lower bpp is good. However, as the payload increases, smaller block size outperforms the larger block size. From Figure 4.10, it can be observed that the optimum value for the block size is 32.
Figure 4.11: Performance comparison of proposed GARW and ATRW techniques with other existing approaches, for the image shown in Figure 4.9 (b).

In [50], comparison of ATRW technique is performed with Tian’s [54] DE, Xuan’s [78] distortion less embedding, and Xuan’s Fixed Threshold [48] approach. However, in this work the performance comparison with that of Lee et al.’s [81] technique is also included, as shown in Figure 4.11 and Figure 4.12. In Figure 4.12, results are obtained with compression; while in Figure 4.11, results are computed without compression.

It is clear from the figures that, ATRW provides good capacity versus imperceptibility tradeoff. Moreover, it also provides maximum embedding capacity compared to the other approaches. It is to be noted that, at lower payload the performance of ATRW is lower compared to Lee et al.’s [81] approach. The reason behind this fact is that, for low payload, lower values for threshold are selected, as less number of coefficients is required to embed the full length of the watermark. Then, embedding of Tmap is performed at 3rd level of IDWT. However, ATRW technique first simulates the message embedding procedure (at 1st and 2nd level of IDWT) and constructs Tmap while keeping the pixel value within the bound i.e. [0-255]. During this simulation step, a block is discarded as a whole even if a single coefficient is not embeddable. This results in loss of some useful coefficients that satisfy the embedding conditions, because that block as a whole cannot be selected for embedding. Whereas, in case of Lee et al.’s [81]
work, the embedding of the location map as well as the watermark takes place at the first level of decomposition, which allows it to use almost all available embeddable coefficients at smaller thresholds. Nevertheless, this affect diminishes quickly with an increase in the number of available coefficients for embedding in ATRW technique (1\textsuperscript{st} and 2\textsuperscript{nd} decomposition levels), as we increase the threshold values. Consequently, with an increase in the threshold, blocks those were not previously utilizable due to threshold-limitation, will now become available and thus allowing enhanced capacity. In short, at smaller thresholds, ATRW technique is not able to use all the available coefficients compared to Lee's approach but the effect is temporary.

Figure 4.12: Performance comparison of GARW and ATRW techniques (after employing lossless compression) with other existing approaches, for the image shown in Figure 4.9 (b).

Experimental results of ATRW approach is presented by testing its performance on an image dataset. It consists of total 255 images and is used by Usman et al. [91] in the experimental analysis of their work. It contains out-door and in-door images of varying frequency contents. The resultant PSNR value against a constant payload of 0.21bpp is shown in Figure 4.13.
4.6.3 Lossless Compression of Depth Maps

Table 4.1 shows the reduction in depth map size of various images after employing Huffman compression. It is to be noted that more than 50% size reduction is achieved for TFT-LCD color filter image.

Table 4.1: Compression results for different depth maps of various images.

<table>
<thead>
<tr>
<th>Image Name</th>
<th>Depth Map Dimension</th>
<th>Original Bit Length</th>
<th>Compressed Bit Length</th>
<th>Compression Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFT-LCD color filter</td>
<td>80 × 80</td>
<td>51,200</td>
<td>25,784</td>
<td>50.35</td>
</tr>
<tr>
<td></td>
<td>100 × 100</td>
<td>80,000</td>
<td>40,291</td>
<td>50.36</td>
</tr>
<tr>
<td></td>
<td>120 × 120</td>
<td>1,15,200</td>
<td>58,164</td>
<td>50.48</td>
</tr>
<tr>
<td>Simulated Cone</td>
<td>80 × 80</td>
<td>51,200</td>
<td>32,544</td>
<td>63.56</td>
</tr>
<tr>
<td></td>
<td>100 × 100</td>
<td>80,000</td>
<td>51,888</td>
<td>64.86</td>
</tr>
<tr>
<td></td>
<td>120 × 120</td>
<td>1,15,200</td>
<td>74,233</td>
<td>64.43</td>
</tr>
<tr>
<td>Face 1</td>
<td>80 × 80</td>
<td>51,200</td>
<td>44,996</td>
<td>87.88</td>
</tr>
<tr>
<td></td>
<td>100 × 100</td>
<td>80,000</td>
<td>70,396</td>
<td>87.99</td>
</tr>
<tr>
<td></td>
<td>120 × 120</td>
<td>1,15,200</td>
<td>1,01,461</td>
<td>88.07</td>
</tr>
<tr>
<td>Face 2</td>
<td>80 × 80</td>
<td>51,200</td>
<td>38,146</td>
<td>74.50</td>
</tr>
<tr>
<td></td>
<td>100 × 100</td>
<td>80,000</td>
<td>59,685</td>
<td>74.60</td>
</tr>
<tr>
<td></td>
<td>120 × 120</td>
<td>1,15,200</td>
<td>86,034</td>
<td>74.68</td>
</tr>
</tbody>
</table>
Figure 4.12 shows the performance comparison of uncompressed and compressed ATRW approach with other approaches. It is clear from the figure that, at a particular value of the payload, significant improvement in image visual quality is achieved on compressed ATRW technique. It is to be noted that, the reduction in size due to compression does not compromise on significance of payload. The reduced depth map size in turn can be effective in achieving a high $PSNR$ value without any loss of information in the depth map. Hence, equally significant payload is delivered along with an increase in image quality.

4.6.4 GARW Results

In this section, performance comparison of GARW technique with that of ATRW is performed. For this purpose, different grayscale images of size 512×512 are used for the experimental analysis of ATRW and the proposed GARW approaches. The results are displayed in Table 4.2, whereby the same depth map image is used as a watermark. Comparison is made in terms of imperceptibility versus capacity tradeoff. Therefore, the imperceptibility of an image in terms of $PSNR$ and $SSIM$ is computed for a constant effective payload. $SSIM$ and $PSNR$ are calculated between original and watermarked images. For good imperceptibility of an image, the value of $PSNR$ should be high, while in case of $SSIM$, the value should be close to ‘1’. From Table 4.2, it can be easily observed that GARW technique yields improved results compared to ATRW technique.

![Graph](image.png)

Figure 4.14: Performance comparison of GARW and ATRW approaches using database of 75 images.
Table 4.2: Performance comparison of proposed GARW technique with that of ATRW approach at fixed effective payload.

<table>
<thead>
<tr>
<th>Images</th>
<th>ATRW</th>
<th>Proposed GARW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>SSIM</td>
</tr>
<tr>
<td>Face &amp; Hand</td>
<td>49.23</td>
<td>0.993</td>
</tr>
<tr>
<td>House</td>
<td>48.18</td>
<td>0.994</td>
</tr>
<tr>
<td>Pepper</td>
<td>45.88</td>
<td>0.992</td>
</tr>
<tr>
<td>Circuit</td>
<td>44.70</td>
<td>0.989</td>
</tr>
<tr>
<td>Baboon</td>
<td>37.40</td>
<td>0.988</td>
</tr>
<tr>
<td>Boat</td>
<td>42.31</td>
<td>0.985</td>
</tr>
<tr>
<td>Lena</td>
<td>41.34</td>
<td>0.986</td>
</tr>
</tbody>
</table>

Figure 4.11 and Figure 4.12 show the plot of watermark imperceptibility against different values of \( bpp \) (without and with compression, respectively). It can be observed that, GARW outperforms ATRW technique. It is to be noted that, in case of ATRW and Lee et al.’s [81] approach, multi-run embedding has been performed. That is, the watermarked image is again watermarked and that is why both the curves stretch beyond 0.85 \( bpp \).

Both ATRW and GARW techniques are applied on a smaller subset (first 75 images) of the 255 image dataset. This is because GARW approach is computationally extensive in the training phase, but in the test phase, it consumes a few seconds to watermark the image. Figure 4.14 displays the average PSNR against a constant payload 0.21 \( bpp \), which shows that GARW approach provides better results compared to ATRW technique. Table 4.3 shows details of the experimental results.

In the experimental analysis, the test time is computed for ATRW and GARW techniques for a 512×512 Lena image. It is observed that ATRW technique takes 6 seconds while GARW needs 5 seconds for watermark embedding process. However, GARW technique consumes more time in computing \( Tmap \). It requires 25 minutes for \( Tmap \) computation, while ATRW technique needs only 1.05 minutes for this process. For this purpose, Pentium D machine with 3.40 \( GHz \) processor and 1 \( GB RAM \) is used. During \( Tmap \) generation phase, population size is taken as 100, while number of generations is set to 50. Due to high computational time in training phase, parallel GA is becoming more and more popular compared to simple conventional GAs. In parallel GA, evolutionary process is run over the parallel processors. Therefore, computational time could be reduced, considerably.
Table 4.3: PSNR results for proposed GARW and ATRW approaches using the dataset of 75 images.

<table>
<thead>
<tr>
<th>Images</th>
<th>GARW</th>
<th>ATRW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR(dB)</td>
<td>Average PSNR</td>
</tr>
<tr>
<td>1</td>
<td>51.553</td>
<td>52.462</td>
</tr>
<tr>
<td>2</td>
<td>48.773</td>
<td>51.229</td>
</tr>
<tr>
<td>3</td>
<td>47.982</td>
<td>50.578</td>
</tr>
<tr>
<td>4</td>
<td>61.318</td>
<td>59.678</td>
</tr>
<tr>
<td>5</td>
<td>52.682</td>
<td>59.892</td>
</tr>
<tr>
<td>6</td>
<td>55.286</td>
<td>53.902</td>
</tr>
<tr>
<td>7</td>
<td>47.507</td>
<td>49.907</td>
</tr>
<tr>
<td>8</td>
<td>50.940</td>
<td>54.607</td>
</tr>
<tr>
<td>9</td>
<td>54.155</td>
<td>54.607</td>
</tr>
<tr>
<td>10</td>
<td>48.258</td>
<td>48.258</td>
</tr>
<tr>
<td>11</td>
<td>47.037</td>
<td>47.037</td>
</tr>
<tr>
<td>12</td>
<td>49.673</td>
<td>49.673</td>
</tr>
<tr>
<td>13</td>
<td>56.784</td>
<td>56.784</td>
</tr>
<tr>
<td>14</td>
<td>46.795</td>
<td>46.795</td>
</tr>
<tr>
<td>15</td>
<td>52.598</td>
<td>52.598</td>
</tr>
<tr>
<td>16</td>
<td>64.328</td>
<td>64.328</td>
</tr>
<tr>
<td>17</td>
<td>57.569</td>
<td>57.569</td>
</tr>
<tr>
<td>18</td>
<td>59.203</td>
<td>59.203</td>
</tr>
<tr>
<td>19</td>
<td>61.714</td>
<td>61.714</td>
</tr>
<tr>
<td>20</td>
<td>55.576</td>
<td>55.576</td>
</tr>
<tr>
<td>21</td>
<td>59.590</td>
<td>59.590</td>
</tr>
<tr>
<td>22</td>
<td>48.277</td>
<td>48.277</td>
</tr>
<tr>
<td>23</td>
<td>47.438</td>
<td>47.438</td>
</tr>
<tr>
<td>24</td>
<td>59.537</td>
<td>59.537</td>
</tr>
<tr>
<td>25</td>
<td>54.669</td>
<td>54.669</td>
</tr>
<tr>
<td>26</td>
<td>46.802</td>
<td>46.802</td>
</tr>
<tr>
<td>27</td>
<td>57.999</td>
<td>57.999</td>
</tr>
<tr>
<td>28</td>
<td>51.961</td>
<td>51.961</td>
</tr>
<tr>
<td>29</td>
<td>58.576</td>
<td>58.576</td>
</tr>
<tr>
<td>30</td>
<td>57.697</td>
<td>57.697</td>
</tr>
<tr>
<td>31</td>
<td>61.904</td>
<td>61.904</td>
</tr>
<tr>
<td>32</td>
<td>61.221</td>
<td>61.221</td>
</tr>
<tr>
<td>33</td>
<td>39.803</td>
<td>39.803</td>
</tr>
<tr>
<td>34</td>
<td>57.513</td>
<td>57.513</td>
</tr>
<tr>
<td>35</td>
<td>59.044</td>
<td>59.044</td>
</tr>
<tr>
<td>36</td>
<td>47.837</td>
<td>47.837</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>37</td>
<td>46.860</td>
<td>50.186</td>
</tr>
<tr>
<td>38</td>
<td>54.072</td>
<td>53.584</td>
</tr>
<tr>
<td>39</td>
<td>51.293</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>50.868</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>52.819</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>56.375</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>54.833</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>50.781</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>53.110</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>51.200</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>48.190</td>
<td>49.245</td>
</tr>
<tr>
<td>48</td>
<td>52.815</td>
<td>46.120</td>
</tr>
<tr>
<td>49</td>
<td>45.635</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>48.383</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>48.310</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>45.010</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>41.218</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>44.913</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>51.146</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>52.198</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>45.088</td>
<td>51.474</td>
</tr>
<tr>
<td>58</td>
<td>57.243</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>52.133</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>50.708</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>56.088</td>
<td>51.421</td>
</tr>
<tr>
<td>62</td>
<td>44.004</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>51.522</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>48.875</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>56.615</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>57.356</td>
<td>50.736</td>
</tr>
<tr>
<td>67</td>
<td>43.425</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>50.289</td>
<td>51.153</td>
</tr>
<tr>
<td>69</td>
<td>49.643</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>52.968</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>48.345</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>49.246</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>50.429</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>59.591</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>48.155</td>
<td></td>
</tr>
</tbody>
</table>
4.6.5 Authentication Analysis of a Host Image

Figure 4.15 shows the results related to the authentication process. It is an additional capability of the proposed technique, which is not provided by the aforementioned existing approaches. Different watermarked images with collage and manipulation attacks are used for the analysis of authentication capability.

![Figure 4.15](image)

Figure 4.15: Column (a) watermarked images, (b) tampered watermarked images, (c) difference of $T_{maps}$ for non-tampered form of watermarked image (column (a)), (d) difference of $T_{maps}$ for tampered form of watermarked image (column (b)).
In Figure 4.15, watermarked images are shown in column (a), while; column (b) is the tampered form of column (a). In the first row, the watermarked Cameraman image is tampered by removing tower. In the second row, the watermarked Obaid image is changed by adding salt and pepper noise. In row three, the watermarked Barbara image is undergone with manipulation attack by removal of a tool from the table. In row four, JPEG compression with quality factor 90 is introduced in the watermarked Couple image. In the fifth row, the watermarked Student image is tampered by removal of air-conditioning duck from the ceiling. In row six, the watermarked Baboon image is tampered with collage/counterfeiting attack.

In Figure 4.15, column (c) is the difference of the two Tmaps (as discussed in Section 4.5) using non-tampered watermarked image (column (a)). The difference maps are black, which implies that the two Tmaps are identical. Therefore, it is concluded that the watermarked image is authentic and has not undergone any kind of adversary attacks. Column (d) of Figure 4.15 depicts the difference of two Tmaps (as discussed in Section 4.5) using tampered watermarked images (column (b)). This column displays images that are not completely black, which implies that the two Tmaps are not identical. Hence, it is concluded that the watermarked image has undergone some kind of adversary attacks. It is to be noted that the difference column (d) of the sixth row is also non-zero, which implies that the proposed technique has successfully detected the collage attack. As the proposed technique is fragile, hence, the noisy image cannot be recovered completely and therefore be considered as non-authentic.

### 4.7 Chapter Summary

In this chapter, an application for images having 3D information is discussed, by reversibly embedding the 3D information as a watermark in its corresponding 2D image. The 3D information is obtained from SFF technique. Watermark is embedded block wise in IDWT domain, depending upon certain threshold value. The learning capability of GA is employed to compute optimum threshold value for each block of the wavelet sub-band. Threshold map is evolved by maximizing the tradeoff between watermark capacity and imperceptibility without arousing any underflow/overflow problem. Experimental analysis showed that the proposed GARW technique outperforms the other existing techniques in terms of capacity versus imperceptibility tradeoff. An additional capability of the proposed approach is the authentication property. From the analysis of the authentication results, it is concluded that the proposed technique is successfully able to indicate the presence of collage and manipulation attacks.

The next chapter is about reversible watermarking for 3D cameras.
Chapter 5  **REVERSIBLE WATERMARKING FOR 3D CAMERAS**

Devices having the capability of computing 3D information of an object are very important for controlling and routing applications. These applications required fast and accurate computation of depth information of an object. Due to depth sensing capability of 3D cameras, they are becoming more and more popular and have wide range of applications in computer systems and software community. 3D gaming, web-conferencing, automotive, object tracking, mobile phones, medical devices, and robotics are the potential areas of 3D cameras. Depth sensing capability makes 3D camera superior from traditional image capturing devices. In literature, depth estimation methods are classified into two categories; namely, passive and active. In passive techniques, phase delay of simple light rays reflections are recorded. While in active methods, such as stereo light, time of flight (TOF), and triangulation light, laser or X-ray is projected for depth estimation. Currently, the active devices are widely used because they achieve high resolution and accuracy [92]. However, these devices are expensive and have high computational requirement compared to passive devices [90].

Our aim in this chapter is to analyze and design a technique for 3D cameras. In this technique, a reversible watermarking technique with authentication capability is developed. It enables a 3D camera to incorporate image protection by embedding the depth map of an object in its corresponding 2D cover image. Hence, not only the depth map can be secretly communicated but the corresponding 2D image can also be protected. The technique utilizes CI approaches to improve the tradeoff between watermark capacity and imperceptibility. In this work, we considered 3D camera that uses time of flight (TOF) principle for depth map calculation of an object. Before going into further details of the proposed scheme, a brief discussion of TOF technique is provided in the next section.

5.1  **Time of Flight (TOF) Principle**

Mostly, 3D cameras utilize TOF principle for computation of 3D information of an object. Light source of the camera project a light on the target object. The light reflected from the target object is captured by the camera sensors. The beam detected at the camera detectors is delayed in phase compared to the light emitted by the camera source. To compute 3D information of an object, the whole surface is illuminated by the light and the phase delay is computed pixel by pixel [93], [94]. The phase delay is computed in...
pico-seconds to achieve at least millimeter accuracy. In these cameras, radio frequency modulator light sources are used, so that the outgoing beam is modulated by the radio frequency carrier. To measure the phase shift, phase shift detectors are used at the receiving side of the camera.

![Figure 5.1: Basic principle of TOF.](image)

Figure 5.1: Basic principle of TOF.

![Figure 5.2: Column (a) 2D Images, Column (b) Depth map images of Column (a). Row (1) man Image, Row (2) Glass Image, Row (3) Toy Image, Row (4) Face Image.](image)
Figure 5.1 shows the basic principle of TOF. While, Figure 5.2 (a) shows the 2D images of a man (Row 1), Glass (Row 2), Toy (Row 3), and Face (Row 4) and Figure 5.2 (b) displays their corresponding depth maps.

5.2 Watermark Embedding Procedure

The proposed work is utilizing depth map as a watermark to be embedded in its corresponding 2D image. Here, the depth map is obtained from 3D camera. Before embedding, depth map is compressed and encrypted by employing using Huffman coding and RSA algorithm [83], respectively. Use of RSA algorithm [83] enhances the security of the system. Now, the compressed and encrypted depth map is embedded in its corresponding 2D image, using a threshold matrix $T_{map}$.

For $T_{map}$ computation, evolutionary algorithms (EAs) are utilized. In numerous field of science and engineering, EAs are employed quite extensively in solving constrained optimization problems [84], [86]. EAs include genetic programming, GA, differential evolution (DE), etc. These algorithms are based upon natural selection and recombination under certain fitness criteria without violating the problem constraints to search for the possible solution of the given problem. In this work, EA’s are employed to search for the optimal/near-optimal values for $T_{map}$ in a way to improve the system performance without any overflow/underflow. System performance is computed in terms of maximizing the capacity versus imperceptibility tradeoff and preventing overflow/underflow of the system means that after watermark embedding the pixel values are within the bound of [0-255].

For $T_{map}$ generation, two approaches are proposed here.

1. Differential evolution based reversible watermarking (DERW)
2. Hybrid PSO-DE based reversible watermarking (HPDRW)

5.2.1 Threshold Matrix ($T_{map}$) Generation using DERW Approach

DE is a kind of optimization algorithm that is utilized special form of differential operator. To search the solution in search space it utilizes the information about distance and direction of the individuals. In this algorithm, as shown in Equations (5.1) and (5.2), a trial vector $u_i(t)$ is generated through mutation operation then crossover is employed on $u_i(t)$ to generate an offspring denoted as $z'$.

$$u_i(t) = z_{i1}(t) + \beta(z_{i2}(t) - z_{i3}(t))$$  \hspace{1cm} (5.1)
where \( z_i(t) \) is a target vector, \( z_{i2}(t) \) and \( z_{i3}(t) \) are randomly selected individuals, \( i \) is the index of the individual, \( t \) is iteration, \( \beta (0, \infty) \) is a scale factor, and \( \tau \) is the set of crossover points.

In this work, \( DE \) is applied to generate optimum/near-optimum \( Tmap \). For this, capacity versus imperceptibility tradeoff is maximized without creating any overflow/underflow pixels. The algorithm \( DE/rand/1 \) algorithm developed by Buehren [95] is used in the simulation study.

![Block diagram of computing Tmap using DE.](image)

Figure 5.3: Block diagram of computing \( Tmap \) using \( DE \).

Figure 5.3 displays the steps involved in \( Tmap \) generation. After parameter settings, \( DE \) generates an initial population randomly. Each individual of the population could be the possible solution of the given problem (i.e. \( Tmap \)). Fitness of each individual is evaluated in each generation. For this, a fitness function is designed in a way that it optimizes the system performance without arousing any overflow/underflow problem. To compute individual fitness, complete watermark embedding procedure (see Section 5.2.3) is performed by using that particular individual as a \( Tmap \). After getting watermarked image, fitness is computed by using relation shown in Equation (5.3).
5.2.2 Threshold Matrix \( T_{map} \) Generation using HPDRW Approach

In 2010, Liu et al. [96] proposed a technique to solve multi-objective constrained optimization problem. In this technique, they have hybridized particle swarm optimization (PSO) algorithm with DE for performance improvement. At the end, they have concluded that hybrid PSO-DE outperform other existing approaches. In this work, hybrid PSO-DE algorithm is modified so that it can be used in 3D image watermarking. With the help of hybrid PSO-DE optimal/near-optimal values for \( T_{map} \) is computed.

Before discussing the details of the algorithm, the notations used in this algorithm are presented. The algorithm evolves two types of populations each of same sizes; namely, Population and LocalBest. Population is the set of individuals that are randomly generated at the start of the algorithm and then evolved in each iteration while, LocalBest stores the best results for each individual of Population. An individual in the set LocalBest is denoted by \( ibest \). Global best is the best individual in LocalBest population and is denoted by \( gbest \). In this work, \( T_{map} \) is a parameter that is optimized through hybrid PSO-DE; therefore, the size of each individual of a population depends upon the block size \( N_B \). If an image of size 512x512 is divided into blocks of size \( N_B = 32 \), then, the individual size is 16x16.

Hybrid PSO-DE works in two phases. In the first phase, half of the Population is evolved using PSO algorithm. In this phase, PSO algorithm reported by Krohling and dos Santos Coelho [97] is employed. Difference of Krohling and dos Santos Coelho PSO algorithm with other PSO algorithm is that, they have introduced a coefficient in the PSO velocity equation. The value of the coefficient is computed through

\[
\text{Fitness} = \begin{cases} 
-\left(\frac{\text{PSNR}}{50}\right) + \left(\frac{\text{Total No. of Out of Bound Pixels}}{\text{ImageSize}}\right) \times 100 \quad \text{if} \quad \text{payload} \geq \text{desired} - \text{payload} \\
0 \quad \text{otherwise}
\end{cases}
\]  

As the depth map is a critical information and it should be embedded fully to be available at the receiving side. Therefore, fitness function is designed in a way that whether the \( T_{map} \) allowing the enough capacity to embed the full string of compressed and encrypted depth map. If not, then that individual is discarded by setting zero fitness value. Otherwise, \( T_{map} \) is evolved to improve imperceptibility for a given payload and to avoid underflow/overflow problem as well. Imperceptibility is calculated through PSNR between original and watermarked images. The process is continued for each individual from generation to generation until the maximum number of generation reached. At the end of the last generation, the individual with best fitness value is returned as a \( T_{map} \) to be used in embedding and extraction stages.
absolute value of Gaussian probability distribution with unit variance and zero mean. The expression of velocity is shown in Equation (5.4),

$$velocity_i^j = C_1 (ibest_i^j - p_i^j) + C_2 (gbest_i^j - p_i^j) \quad ; i = 1, 2, ..., pop\_size, \ j = 1, 2, ..., max\_gen \quad (5.4)$$

where $C_1$ and $C_2$ are the coefficients obtained using $\text{abs}(N(0,1))$, $p_i^j$ is the $i^{th}$ individual of Population at the $j^{th}$ generation, $ibest_i^j$ is the $i^{th}$ individual of LocalBest at the $j^{th}$ generation, population size is represented by $pop\_size$, and $max\_gen$ is the maximum number of generations. Based upon the value of $velocity_i^j$, $p_i^j$ is updated by using Equation (5.5).

$$p_i^{j+1} = p_i^j + velocity_i^j \quad ; i = 1, 2, ..., pop\_size, \ j = 1, 2, ..., max\_gen \quad (5.5)$$

There exists a possibility that after updating the value of an individual of Population, the individual takes value that is not permissible for that particular application, i.e. individual may goes out of bound. To cope with this problem, the updated individual is checked for the boundary condition violation. In case of out of bound, the updated individual value is adjusted according to the following rule [98]:

$$p_i^{j+1} = \begin{cases} 
0.5(l + p_i^{j+1}) & \text{if} \ p_i^{j+1} < l \\
0.5(u + p_i^{j+1}) & \text{if} \ p_i^{j+1} > u \\
p_i^{j+1} & \text{otherwise}
\end{cases} \quad ; i = 1, 2, ..., pop\_size, \ j = 1, 2, ..., max\_gen \quad (5.6)$$

where, $l$ and $u$ are the minimum and maximum allowed threshold value, respectively. In this technique, the lower and upper bounds of each individual is taken as [0-8], the range is selected empirically. In this work, the main objective is to compute the optimum $Tmap$ that can maximize the system performance without violating any constraint. Watermark embedding procedure is performed by utilizing the updated individual as a $Tmap$ (Section 5.2.3 discusses the steps involved in watermark embedding process). The system performance is computed in terms of maximizing the imperceptibility after achieving certain capacity and constraint is that the pixels in the marked image should not exceed the bound of [0-255]. Since, the depth map is inserted as a watermark; therefore, capacity should be large enough to accommodate the full depth map. The next step is to compute fitness function $(f)$ and degree of constraint violation $(G)$ using relation shown in Equation (5.7) and Equation (5.8), respectively.

$$f = \begin{cases} 
-\text{SSIM} & \text{if} \ \text{payload} \geq \text{desired payload} \\
0 & \text{otherwise}
\end{cases} \quad (5.7)$$

$$G = \text{Total No. of Out of Bound Pixels} \quad (5.8)$$
For watermark imperceptibility, SSIM is used. The fitness function \( f \) is designed to maximize imperceptibility for the desired payload. \( G \) is the boundary limitation, i.e. watermarked image should be within the bounds [0 255]. After computing \( f \) and \( G \) for the updated individual, these values are compared with its corresponding \( ibest \) in the set LocalBest. The updated individual is replaced with its corresponding \( ibest \) in the set LocalBest, if it meets the selection criteria proposed by Deb [99].

In the second stage, LocalBest is evolved using DE algorithm. Before processing, each individual is transformed from matrix to array format. For each entry \( ibest \) in LocalBest, three offspring \((s_1, s_2, \text{and } s_3)\) are generated by employing three mutation strategies, namely current_to_Best/1, rand/1, and rand/2 as shown in Equation (5.9), Equation (5.10), and Equation (5.11), respectively.

\[
s_1 = ibest_{(i1)} + F(\text{ibest}_{(i2)} - \text{ibest}_{(i3)}) + F(\text{ibest}_{(i4)} - \text{ibest}_{(i5)})
\]

\[
s_2 = ibest_{(i1)} + F(\text{ibest}_{(i2)} - \text{ibest}_{(i3)})
\]

\[
s_3 = ibest_{(i1)} + F(gbest' - \text{ibest}_{(i1)}) + F(\text{ibest}_{(i1)} - \text{ibest}_{(i2)})
\]

where \( ibest_{(i)} \) is the \( i^{th} \) individual of LocalBest at \( i^{th} \) generation, \( r[m] = \{ \text{where } m \in \{1,2,3,4,5\} \} \) is a uniformly distributed random number in the range \([1 \text{ pop_size}]\), and \( F(s.t. F \in [0,2]) \) stands for an amplification factor. Each offspring \( s_k \) is checked against boundary violation. If the value of any \( s_k \) goes beyond the bounds then it is adjusted by using the following expression [100].

\[
s_k = \begin{cases} 
2l - s_k & \text{if } s_k < l \\
2u + s_k & \text{if } s_k > u \\
\hat{s}_k & \text{otherwise} 
\end{cases} ; k = 1, 2, 3
\]

For each offspring \( s_k \), \( f \) and \( G \) is computed. At the end of the second stage, three offspring’s \( s_k \) are compared to their corresponding \( ibest_{(i)} \), the offspring \( s_k \) replaces \( ibest_{(i)} \) only if it has lower degree of constraint violation and better fitness value. The steps involved in stage two are repeated for each entry of LocalBest. At the end of each generation, the value of \( gbest \) is also updated. The algorithm continues until the maximum number of generation is reached. At the end, the algorithm returns \( gbest \) as a Tmap. Figure 5.4 is a block diagram for hybrid PSO-DE algorithm.
Figure 5.4: Block diagram of computing $T_{map}$ using hybrid PSO-DE.
5.2.3 Information Embedding

*Tmap* generated in the previous sections is meant to deliver high imperceptibility for the particular depth map under the constraint of no overflow/underflow problem. *Tmap* is used in depth map embedding process. In this regards, the 2D image is divided in blocks each of size $N_b \times N_b$. In Chapter 4 Section 4.6.2, effect of change in block size (from 8×8 to 64×64) on the system performance is analyzed. It is observed that for the payload ranging from 0.18 to 0.67, the block size 32 provides better capacity versus imperceptibility tradeoff. In this work, the payload required by depth maps lies in the same range; therefore, the block size $N_b$ is set to 32 in all simulation run.

After block creation, each block is decomposed in wavelet sub-band by employing *IDWT* upto 2nd level using *CDF* filter. Then, the depth map is embedded, at LH, HL, and HH sub-bands of level 1 and level 2, using Equation (4.6). Embedding is performed only for those coefficients that are less than the threshold value computed for that sub-band of the particular block. After embedding, inverse *IDWT* is applied to get watermarked block. Each block is processed to embed depth map. At the end, watermarked image $I'$ is obtained.

As different threshold values is used for embedding in different block. Therefore, *Tmap* is also required at the receiving side to perform extraction and recovery processes. In second stage of the algorithm, *Tmap* is embedded in watermarked image $I'$. Before embedding, *Tmap* is compressed by using Huffman coding. Then watermarked image $I'$ is decomposed into wavelet sub-band by employing *IDWT* upto third level and *Tmap* is embedded in $HH3$, $HL3$, and $LH3$ sub-bands using threshold $T_z$ to be {2, 3, 3}, respectively. The block size $N_b$ is also embedded as a header information. Value of $T_z$ is empirically selected by considering the lower coefficient values of 3rd level sub-bands. After *Tmap* embedding, inverse *IDWT* is applied to get final watermarked image $I''$. In this work, the header information is embedded seperately in level 3 coefficients; therefore, it does not affect the actual payload of the system. It is worth noting that, different threshold values are used for different sub-bands. It can be observed that, distribution of high frequency coefficients are different in different sub-bands. Moreover, the number of high frequency coefficients are large in $HH$ and $HL$ sub-band, compared to $LH$ sub-bands. Therefore, the threshold value for $HL$ and $HH$ sub-bands are selected slightly high compared to $LH$ sub-band, so that more coefficients are selected for embedding in $HL$ and $HH$ sub-band compared to $LH$ sub-band.

Value of threshold $T_i$ is assigned to *Tmap*. Block diagram of watermark embedding process is shown in Figure 5.5.
5.3 Watermark Extraction Procedure

In the extraction process, $T_{map}$ is extracted by using fixed threshold $T_z = \{2, 3, 3\}$. For this, image $I''$ is decomposed into wavelet sub-bands by employing $IDWT$ upto 3rd level using $CDF$ filter. Header information is extracted from $HH3, HL3,$ and $LH3$ sub-bands using $T_z$. From header information block size $N_B$ and compressed $T_{map}$ are separated. $T_{map}$ are decompressed by employing Huffman decoding. Then, inverse $IDWT$ is applied to obtain image $I'$.

$T_{map}$ obtained in previous stage, is used to extract depth map. For this, image is divided in blocks each of size $N_B \times N_B$. Each block is decomposed in wavelet sub-band by applying $IDWT$ upto 2nd level. Then, compressed and encrypted depth map is extracted from $HH1, HL1, LH1, HH2, HL1,$ and $LH1$ sub-bands. With the help of the formula presented in Equation (4.9), depth map is extracted as well as original values are restored. After extraction, inverse $IDWT$ is applied to get block $b(i, j)$. Each block is processed to extract depth map from the coefficients $K'$ using threshold from $T_{map}$ for that block and original coefficients value is restored. At the end, original image $I$ and its corresponding depth map are obtained. The depth map is decompressed and decrypted to transform it to original form. The block diagram for the extraction procedure is shown in Figure 5.6.

Figure 5.5: Block diagram of watermark embedding procedure.
5.4 Image Authentication through Auxiliary Information

*Tmap* is required at the receiving side to perform extraction and recovery processes. Therefore, in addition to depth map, *Tmap* is also embedded in the 2D cover image so that it can be utilized in extraction and recovery processes, at the receiving side. In order to incorporate the authentication capability of the proposed approach, *Tmap* is utilized. In this way, a subsidiary advantage is taken from this auxiliary data. Authentication process ensures that whether the received image has undergone any type of malicious activity or not. In this regard, two types of attacks are considered; namely, collage and manipulation attacks. Manipulation attacks are easy to detect; while, detecting a collage attack is problematic [26]. In order to detect collage attack, certain operations have to be performed on *Tmap* at the
embedding side. Initially, correlation operation is performed on $T_{map}$ with second level approximation sub-band $LL2$. Then, correlated $T_{map}$ is compressed by applying Huffman coding. The resultant $T_{map}$ is embedded 3rd level $HH$, $HL$ and $LH$ sub-bands. The process of correlating $T_{map}$ with approximation sub-band helps in detecting collage attack.

For authentication at receiving end, initially, $T_{map}$ is extracted. Then, Huffman decompression is applied to the extracted $T_{map}$. To get the original values of $T_{map}$, it is decorrelated using $LL2$ sub-band. After getting $T_{map}$, it is used for depth map extraction. Since, extraction is reversible; therefore, after extraction the recovered image is the same as that of the original image. Then, $T_{map}$ is computed for the recovered image. The two $T_{maps}$, one generated using the recovered image and second extracted from the watermarked image are compared. If both $T_{maps}$ are the same, it implies that image is authentic. Otherwise, it is concluded that image has undergone some kind of adversary attacks (either manipulation or collage).

### 5.5 Results and Discussion

#### 5.5.1 Performance Analysis in terms of Watermark Imperceptibility

To highlight the effectiveness of the proposed techniques, performance comparison with some of the existing techniques is presented in this section.

Table 5.1: Image quality (PSNR and SSIM) based comparison with existing approaches.

<table>
<thead>
<tr>
<th></th>
<th>Lena</th>
<th>Barbra</th>
<th>Man</th>
<th>Tree</th>
<th>Glass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xuan et al. [48]</td>
<td>PSNR</td>
<td>44.460</td>
<td>40.803</td>
<td>48.577</td>
<td>40.002</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.989</td>
<td>0.982</td>
<td>0.993</td>
<td>0.977</td>
</tr>
<tr>
<td>Khan et al. [49]</td>
<td>PSNR</td>
<td>44.708</td>
<td>41.216</td>
<td>49.655</td>
<td>40.637</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.991</td>
<td>0.985</td>
<td>0.994</td>
<td>0.975</td>
</tr>
<tr>
<td>Ali et al. [50]</td>
<td>PSNR</td>
<td>46.663</td>
<td>43.315</td>
<td>49.662</td>
<td>41.764</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.992</td>
<td>0.988</td>
<td>0.994</td>
<td>0.981</td>
</tr>
<tr>
<td>Proposed DERW</td>
<td>PSNR</td>
<td>47.389</td>
<td>44.192</td>
<td>49.864</td>
<td>41.600</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.993</td>
<td>0.989</td>
<td>0.994</td>
<td>0.979</td>
</tr>
<tr>
<td>Proposed HPDRW</td>
<td>PSNR</td>
<td>48.985</td>
<td>49.644</td>
<td>53.530</td>
<td>42.763</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.995</td>
<td>0.997</td>
<td>0.997</td>
<td>0.984</td>
</tr>
</tbody>
</table>

* Note that $bpp$ is fixed at 0.20784, as the aim is to embed the whole depth map shown in Figure 5.2 (b).
Table 5.1 shows the comparison of the proposed DERW and HPDRW approaches with that of Ali et al.’s [50] adaptive threshold based lossless watermarking approaches, Xuan et al.’s [48] fixed threshold based reversible watermarking, and Khan’s [49] variable threshold based reversible watermarking. The comparison is performed in terms of watermarked image imperceptibility against a fixed payload. The terms PSNR and SSIM are used for imperceptibility measure. Different images of size 512×512 are used for experimental analysis. It is to be noted that the proposed DERW and HPDRW approaches perform better than the other approaches, both in terms of PSNR and SSIM.

Figure 5.7: Performance comparison of proposed HPDRW and DERW with adaptive threshold [50], fixed threshold [48] and variable threshold [49] approaches, against different values of payload (for Lena image).

Graphical comparison of the proposed DERW and HPDRW approaches with that of adaptive threshold technique [50], Xuan et al.’s [48] fixed threshold based reversible watermarking and Khan’s [49] reversible watermarking using variable threshold approach is presented in Figure 5.7. The results are obtained using Lena image of size 512×512, by varying the payload. Imperceptibility is computed in terms of PSNR and payload is represented in terms of bpp. From the figure, it is concluded that DERW
and HPDRW approaches perform well compared to other approaches in terms of capacity versus imperceptibility tradeoff. This improvement in performance is mainly due to the integration of the intelligence of CI approaches with the reversible watermarking technique.

5.5.2 Performance Analysis of the proposed Techniques on Image Database

In this section, performance analysis of both DERW and HPDRW is performed using a database of 100 images. Results obtained is compared with adaptive threshold [50], fixed threshold [48] and variable threshold [49] approaches. The image dataset used in this section is the same that is used by Usman et al. [91] for the performance analysis of their approach. It is the collection of some outdoor and indoor images with different overall frequency contents.

![Figure 5.8: Performance comparison of proposed HPDRW and DERW techniques with that of Adaptive Threshold [50], Fixed Threshold [48] and Variable Threshold [49] approaches using a database of 100 images.](image-url)
Figure 5.8 displays the performance comparison graphically on image dataset. The results are computed on the depth map of Glass image (Row 2 of Figure 5.2 (b)) as a watermark. It provides a constant payload of 0.21 bpp. As the number of images are large; therefore, to make the comparison more clear the PSNR is displayed by averaging the PSNR of every 5 images. Figure 5.8 shows that, compared to the other existing approaches, the proposed DERW and HPDRW approaches performed well in terms of imperceptibility for a given payload.

While comparing the results of DERW and HPDRW approaches, it can be concluded from Table 5.1, Figure 5.7 and Figure 5.8 that HPDRW outperforms DERW. Improvement is due to the hybridizing PSO with DE algorithm. Stagnation problem is the only weakness in PSO algorithm, due to which it may quickly converge to a local minima. In HPDRW approach, DE integration with PSO resolves this problem, by making HPDRW to escape from stagnation and quickly converge to global optima.

5.5.3 Authentication Analysis of a Host Image

Figure 5.9 displays the results of authentication process, which is the additional capability of the proposed approach, not provided by the aforementioned existing approaches. For analysis of authentication process, some attacks (manipulation and collage) and noise are introduced on different watermarked images.

In the first row, Gaussian noise with unit variance and zero mean is added in the watermarked Lena image. In second row, JPEG compression with quality factor taken as 85 is applied to the watermarked Barbra image. In third row, watermarked Man image is undergone with manipulation attack by removal of a switch board from the right side wall. In fourth row, the watermarked Tree image is tampered by collage/counterfeiting attack. In row five, picture is changed on the right side screen behind the glass, which is again a manipulation attack.

In Figure 5.9, column (c) is the difference of the two Tmaps (as discussed in Section 5.4) using non-tampered watermarked image (column (a)). It can be observed that the difference maps in column (c) are black, which implies that the two Tmaps are identical. Therefore, it is concluded that the watermarked images shown in column (a) are authentic and has not undergone any kind of adversary attacks. Column (d) of Figure 5.9 depicts the difference of two Tmaps (as discussed in Section 5.4) using tampered watermarked images (column (b)). This Column displays images that are not completely black, which implies that the two Tmaps are not identical. Since, the two Tmaps are not the same; therefore, it is concluded that the watermarked images shown in column (b) have undergone some kind of adversary attacks. It is to be noted that the difference column (d) of the fourth row is non-zero, which implies that the proposed technique has successfully detected the collage attack.
Chapter 5

Reversible Watermarking for 3D Cameras

5.5.4 Temporal Analysis of DERW and HPDRW

In the experimental analysis, the test time is computed for Ali et al.’s [50], DERW and HPDRW techniques for a 512×512 Lena image. It is observed that, Ali et al.’s [50] technique takes 6 seconds while DERW and HPDRW need 5 seconds during embedding process. However, DERW and HPDRW approaches consume more time in computing Tmap. DERW has taken 15 minutes while HPDRW has taken 30 minutes to compute optimal/near-optimal Tmap, while, Ali et al.’s [50] technique needed 0.05 minutes for Tmap generation. For this purpose, Pentium D machine with 3.40 GHz processor and 1 GB RAM is used. During Tmap generation phase, population size is taken as 100 and number of generations is kept as 50.
5.5.5 System Related Discussion

Keeping in view the intended application in 3D cameras, the proposed technique is simple and independent of complex floating-point manipulations. Hence, an embedded implementation in DSP (digital signal processing) chip or FPGA (field-programmable gate array) can easily be visualized. However, DSP chip must possess a combination of fast memory access operations to the bit-map pixel memory and processing horsepower to handle the volume of matrix arithmetic. Some of DSP chips that are designed to handle this sort of processing are already available in open market. These devices satisfy the needs of advanced graphics and imaging applications including data throughput rate of more than 50 MFLOPS (mega floating point operations per seconds). In addition, these DSP chips are capable of applying multiprocessor resources and accessing large banks of inexpensive memory. Several features of recently developed DSP chips make them ideally suited to graphics and imaging applications.

Some of DSP chips designed to handle this sort of processing include TMS320C30 and the recently introduced TMS320C40 (which comprises C30 and has six ports for communicating with other C40s) from Texas Instruments. Similarly, i860XP from Intel Corporation, 32C from AT&T, 21020 from Analog Devices, and DSP96002 from Motorola may also be good candidates for this purpose. Several features of TMS320C40, TMS320C30, and DSP96002 make them ideally suited to graphics and imaging applications.

5.5.6 Security Analysis

The technique discussed in this chapter, provides security to 3D cameras images through embedding their corresponding depth map as a watermark. In this way, the image along with its corresponding 3D information can be communicated securely to the receiving side. At the receiving side, the original contents of the image as well as its depth map are recovered completely after extraction and recovery procedures. The security of the algorithm is enhanced by employing RSA encryption algorithm.

The proposed technique is a block wise algorithm. Due to limited length of depth map all blocks of the image are not utilized in embedding procedure. Therefore, another layer of security can be incorporated by selecting blocks randomly for embedding procedure. The selection is based upon a private key, which is known only to the authorized person.

5.6 Chapter Summary

In this chapter, an application for images from 3D cameras is discussed, by reversibly embedding the 3D information as a watermark in its corresponding 2D image. Watermark is embedded block wise in IDWT
domain, depending upon certain threshold value. CI algorithm such as DE or hybrid PSO-DE is employed to compute optimum threshold value for each block of wavelet sub-band. Threshold map is evolved by maximizing the tradeoff between watermark capacity and imperceptibility without arousing any underflow/overflow problem. Experimental analysis showed that the proposed DERW and HPDRW technique outperforms the other existing techniques in terms of capacity versus imperceptibility tradeoff. An additional capability of the proposed approach is the authentication property. From the analysis of the authentication results, it is concluded that, the proposed technique is successfully able to indicate the presence of collage and manipulation attacks. While investigating the performance of DERW and HPDRW, it is observed that, HPDRW outperforms DERW in terms of capacity versus imperceptibility tradeoff. Improvement is due to the integration of PSO with DE algorithm.

In the next chapter, histogram based reversible watermarking is discussed. This type of algorithms is easy to implement and consume less computation time.
Chapter 6  **HISTOGRAM BASED REVERSIBLE WATERMARKING**

In the previous chapters, the strength of CI algorithms is exploited to enhance watermark capacity versus imperceptibility tradeoff. However, the CI approaches are more time consuming, which is undesirable in some real time applications. Therefore, in this chapter, a histogram based reversible watermarking technique is presented.

Reversible watermark approaches based on histogram technique are mostly robust or semi-fragile in nature. These techniques are easy to implement and require less computational time. Histogram based reversible watermarking approach developed by Vleeschouwer et al. [37], is a robust invertible data hiding technique. In [38], Vleeschouwer et al. published an improved version of their previous technique. However, their approach suffers from salt and pepper noise due to modulo-256 addition. Ni et al. [41], highlighted this problem in their work and also introduces another histogram based reversible watermarking approach. In their approach, they introduce a difference value $\alpha$ which enhances the robustness of their approach. However, Ni et al.’s [41] method is not able to recover original image and watermark at the extraction side in some cases. Gao et al. [42] addressed this shortcoming of Ni et al.’s [41] technique and remove it in their proposed reversible data hiding using block skipping (RDHBS) scheme. According to their approach, a bit is embedded in an image block according to block category. There is a possibility of wrong message bit embedding for certain block category, which introduces error in extracted message at the receiving side. To cope with this problem, BCH coding and permutation scheme is employed on the watermarked message before embedding.

In this chapter, Gao et al.’s [42] approach has been investigated thoroughly and few weaknesses have been found in it. In RDHBS, each block of the image is categorized in one of 10 cases depending on its statistics. The actual watermark bit is embedded in the blocks that fall in the cases 1, 3, and 6. While, in the rest of the blocks, bit 1 or bit 0 is embedded depending on the case they belong to, irrespective of what the actual watermark bit is. Therefore, there are chances that incorrect message bits are embedded in some of these blocks. Gao et al. [10] employed BCH coding to remove the errors that occurred due to embedding of incorrect message bits in some blocks of the image. BCH codes are one of the several error-correcting codes, and they are generally represented in a form $BCH (n, k, t)$; where, $k$ represents the number of information bits, $n$ is the total length of code word, and $t$ is number of error bits it can correct.
Therefore, BCH coding has limited capability of correcting the erroneous bits, if the number of erroneous bits is greater than $t$ then it will fail to correct all the erroneous bits. In Gao et al.’s [10] approach, the number of incorrect message bits may be greater than $t$. Consequently, both message and image cannot be recovered completely. Another disadvantage using BCH coding in this approach is that it converts $k$-bit message to $n$-bit code word, where $n>k$. This in turn increases the amount of auxiliary data and decreases the number of information bits to-be-embedded (effective-payload). Secondly, the category identification equation is somewhat ambiguous. A block may become a candidate for two categories at a time. Another problem of this technique is that it does not discuss how the location map is provided at the receiving side. Solution to these problems will be discussed in this chapter. As a solution to these problems, improved RDHBS is presented.

In addition to this, a new reversible watermarking based on histogram processing and block selection (RW-HPBS) is proposed by extending the improved RDHBS scheme and exploiting the concept of modified down sampling. It results in enhancing the effective payload.

Before going into further details of the proposed schemes, a brief discussion of RDHBS [42] scheme is provided in the next section.

### 6.1 Reversible Data Hiding using Block Skipping Scheme (RDHBS)

![Pixel gray scale values distribution and conditions for block classification in different categories](image)

Figure 6.1: Pixel gray scale values distribution and conditions for block classification in different categories [42].
Gao et al. [42] in their work (RDHBS) have improved the shortcomings of the approach proposed by Ni et al. [41]. In RDHBS technique, cover image is first decomposed into non-overlapping blocks of size $m \times n$. Category for each block is identified based upon its histogram conditions as shown in Figure 6.1. In Figure 6.1, $\beta$ is the embedding level, $dl$ and $dr$ are computed through Equation (6.1).

\[
\begin{align*}
    dl &= \min(B) - 0 \\
    dr &= 255 - \max(B)
\end{align*}
\]  

(6.1)

where, $B$ is the set of block pixels.

After category identification, block is divided into two sets of same size i.e. $S_1$ and $S_2$, as shown in Figure 6.2. Then, the robust statistical quantity $\alpha$ is computed through the relation shown in Equation (6.2),

\[
    \alpha = \frac{1}{w} \sum_{i=1}^{w} (s_{1,i} - s_{2,i})
\]  

(6.2)

where, $w$ is the number of pixel pairs in each block. Depending upon $\alpha$ value, block category case is identified (as shown in Figure 6.3). Embedding level ($\beta$) is the amount by which pixels of set $S_1$ are shifted in order to embed bit “1”. $\beta$ is obtained by the relation [42]:

\[
    \beta = t \times T + \varepsilon
\]  

(6.3)

where, $t$ is an integer greater than 1, $\varepsilon$ is a small quantity and $T$ is a threshold that varies from 1 to 5. Watermark is embedded in the block as shown in Figure 6.3.

(a) Image block showing distribution of set $S_1$ pixels.  

(b) Image block showing distribution of set $S_2$ pixels.

Figure 6.2: Distribution of pixels in sets $S_1$ and $S_2$. 
Gao et al. [42] highlighted that, there exist some blocks that are changing their category case after watermark embedding. When a block falls either in case 5, 7 or 10; then, after watermark embedding it changes its category (as shown in Figure 6.4) and it is not possible to extract correct message bit and complete block recovery at the extraction side. In that particular situation, block is considered as unstable.
Therefore, before watermark embedding, each block is checked for its stability using relation shown in Table 6.1. In Table 6.1, \( \text{min} \) and \( \text{max} \) represent the minimum and maximum pixel value of a block, whereas, \( \bar{\text{max}} \) and \( \bar{\text{min}} \) are the maximum and minimum pixel value in the set \( S_i \). In order to check the stability of the block at extraction side a location map for complete image is generated. For this, image blocks are traversed row wise, “0” is recorded for unstable blocks and “1” is saved for stable block in the map. Only stable blocks are used for watermark embedding using the strategy shown in Figure 6.3.

Table 6.1: Block category changing conditions [42].

<table>
<thead>
<tr>
<th>Category</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category 1: ( \alpha &gt; 0 )</td>
<td>[ \exists \text{max} \in A \land \text{max} &gt; (255 - 2\beta) \lor \left( \forall \text{max} \notin A \land (\text{max} + \beta) &gt; \text{max} \cap \text{max} &gt; (255 - 2\beta) \right) ]</td>
</tr>
<tr>
<td>Category 1: ( \alpha \leq 0 )</td>
<td>[ \exists \text{min} \in A \land \text{min} &lt; 2\beta \lor \left( \forall \text{min} \notin A \land (\text{min} - \beta) &lt; \text{min} \cap \text{min} &lt; 2\beta \right) ]</td>
</tr>
<tr>
<td>Category 2: ( \alpha &gt; 0 )</td>
<td>[ \exists \text{max} \in A \land \text{max} &gt; (255 - 2\beta) \lor \left( \forall \text{max} \notin A \land (\text{max} + \beta) &gt; \text{max} \cap \text{max} &gt; (255 - 2\beta) \right) ]</td>
</tr>
<tr>
<td>Category 3: ( \alpha \leq 0 )</td>
<td>[ \exists \text{min} \in A \land \text{min} &lt; 2\beta \lor \left( \forall \text{min} \notin A \land (\text{min} - \beta) &lt; \text{min} \cap \text{min} &lt; 2\beta \right) ]</td>
</tr>
</tbody>
</table>

6.2 Proposed Improved RDHBS Approach

In RDHBS [42] approach, if a block is stable, then, a watermark bit is embedded in it regardless of its suitability for embedding the correct message bit. In certain situations, a block may not be suitable for embedding the correct message bit; therefore, wrong message bit is embedded in the block. This thing results in extraction of erroneous watermarked message at the extraction side. To rectify the presence of wrong message bits in the extracted watermarked message, BCH coding is applied on the watermark message before embedding. There may be a case of burst of errors in the watermark after extraction; BCH is unable to rectify the burst of errors. For this, some permutation scheme is used along with the BCH code. Use of BCH coding and permutation scheme can rectify almost all error bits in the extracted watermark but at the extent of lots of extra bits in the original message. A solution to above problem is proposed by embedding the correct message bit only. According to the proposed improved RDHBS, a block will become unstable, either, it satisfies the category changing condition (shown in Table 6.1), or it is not suitable for embedding the correct watermark bit.

Another problem in RDHBS scheme is in the category classification conditions (mentioned in Figure 6.1). Category identification conditions are such that, a block may satisfy conditions for two different categories at a time, which is very confusing. For example, from the conditions mentioned in Figure 6.1, it can be seen that for \( dl = \beta \) and \( dr = \beta \), block satisfied all four conditions of category classification.
algorithm. Second proposed improvement to RDHBS scheme is the modification in the category classification conditions. The improved version of the category classification condition is shown in Equation (6.4).

\[
\begin{align*}
    dl \geq \beta \land dr \geq \beta & \ (\text{Category 1}) \\
    dl < \beta \land dr \geq \beta & \ (\text{Category 2}) \\
    dl \geq \beta \land dr < \beta & \ (\text{Category 3}) \\
    dl < \beta \land dr < \beta & \ (\text{Category 4})
\end{align*}
\]  

(6.4)

Figure 6.5: Images used in simulation.
Table 6.2: Comparison of block classification by RDHBS [42] and proposed improved RDHBS method.

<table>
<thead>
<tr>
<th>Images</th>
<th>Method Name</th>
<th>Category 1</th>
<th>Category 2</th>
<th>Category 3</th>
<th>Category 4</th>
<th>Total</th>
<th>Image Blocks</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>RDHBS method</td>
<td>4075</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>4109</td>
<td>4096</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Improved RDHBS</td>
<td>4062</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>4096</td>
<td>4096</td>
<td>0</td>
</tr>
<tr>
<td>Baboon</td>
<td>RDHBS method</td>
<td>1764</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1764</td>
<td>1764</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Improved RDHBS</td>
<td>1764</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1764</td>
<td>1764</td>
<td>0</td>
</tr>
<tr>
<td>Boat</td>
<td>RDHBS method</td>
<td>117</td>
<td>495</td>
<td>51</td>
<td>163</td>
<td>826</td>
<td>784</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>Improved RDHBS</td>
<td>98</td>
<td>477</td>
<td>48</td>
<td>161</td>
<td>784</td>
<td>784</td>
<td>0</td>
</tr>
<tr>
<td>Mpic1</td>
<td>RDHBS method</td>
<td>3428</td>
<td>664</td>
<td>136</td>
<td>136</td>
<td>4364</td>
<td>4096</td>
<td>268</td>
</tr>
<tr>
<td></td>
<td>Improved RDHBS</td>
<td>3208</td>
<td>661</td>
<td>92</td>
<td>135</td>
<td>4096</td>
<td>4096</td>
<td>0</td>
</tr>
<tr>
<td>Mpic2</td>
<td>RDHBS method</td>
<td>3952</td>
<td>64</td>
<td>10</td>
<td>73</td>
<td>4099</td>
<td>4096</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Improved RDHBS</td>
<td>3951</td>
<td>64</td>
<td>10</td>
<td>71</td>
<td>4096</td>
<td>4096</td>
<td>0</td>
</tr>
<tr>
<td>Cpic1</td>
<td>RDHBS method</td>
<td>6935</td>
<td>249</td>
<td>230</td>
<td>1</td>
<td>7415</td>
<td>7296</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>Improved RDHBS</td>
<td>6870</td>
<td>249</td>
<td>176</td>
<td>1</td>
<td>7296</td>
<td>7296</td>
<td>0</td>
</tr>
<tr>
<td>Cpic2</td>
<td>RDHBS method</td>
<td>6787</td>
<td>72</td>
<td>474</td>
<td>7</td>
<td>7430</td>
<td>7296</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>Improved RDHBS</td>
<td>6779</td>
<td>71</td>
<td>439</td>
<td>7</td>
<td>7296</td>
<td>7296</td>
<td>0</td>
</tr>
<tr>
<td>Cpic3</td>
<td>RDHBS method</td>
<td>6561</td>
<td>66</td>
<td>708</td>
<td>12</td>
<td>7347</td>
<td>7296</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Improved RDHBS</td>
<td>6553</td>
<td>64</td>
<td>667</td>
<td>12</td>
<td>7296</td>
<td>7296</td>
<td>0</td>
</tr>
</tbody>
</table>

The proposed category classification conditions classify each block in only one category. To verify this argument simulation results are obtained using images shown in Figure 6.5. These results are shown in Table 6.2. Difference between total number of blocks and summation of blocks classified into categories 1, 2, 3, and 4 is shown in the last column of Table 6.2. It can be observed that the difference is zero in case of the proposed improved method and it is non-zero for RDHBS approach. Hence, the revised set of category classification conditions classifies each block in a unique category only.

### 6.3 Proposed RW-HPBS Approach

In the previous section, it is to be noted that the proposed improved RDHBS provides better effective payload. In this section, a novel reversible watermarking approach using histogram processing and block skipping (RW-HPBS) techniques, is discussed. This approach employs the concept of down-sampling, as is adopted in [101], [102], to achieve better capacity versus imperceptibility tradeoff compared to improved RDHBS technique. The rationale behind using the down-sampling approach is to increase the effective-payload of the proposed system for a given imperceptibility. The down-sampling scheme split the image in two sub-sampled versions. By utilizing these two versions, we proposed the new way to generate blocks for watermark embedding. The blocks generated in this way increases the effective
payload of the system. However, the number of pixels changed due to embedding is the same as used in RDHBS or improved-RDHBS. Therefore, overall imperceptibility of the system is not disturbed due to increase in watermark capacity.

6.3.1 Embedding using RW-HPBS

Initially, the original image $I$ of size $X \times Y$, is down sampled that results in two sub-sampled versions of $I$. One is the reference sub-sampled version $I_{\text{ref}}$, of size $X \times \left(\frac{Y}{3}\right)$, while the other is the data hiding sub-sampled version $I_{\text{hd}}$, of size $X \times \left(\frac{Y}{3} \times 2\right)$. $I_{\text{ref}}$ is generated by starting from second column and selecting every third column of the original image. $I_{\text{hd}}$ is the remaining columns of the original image, starting from first column. Algorithm 6.1 shows the steps involved in generation of $I_{\text{ref}}$ and $I_{\text{hd}}$. Input of the algorithm is an image $I$ of size $X \times Y$, whereas; output is two sub-sampled version of $I$, that are $I_{\text{ref}}$ and $I_{\text{hd}}$.

Algorithm 6.1: Generation of sub-sampled versions ($I_{\text{ref}}$ and $I_{\text{hd}}$) of image

```
01: // X, Y: Image size along x and y direction;
02: // I: Original image
03: // I_ref, I_hd: Sub-sampled image versions
04: I_hd(:,1) = I(:,1);
05: j=2, k=1;
06: for i ← 2 to Y do
07:     if (mod((i-2),3) == 0) then
08:         I_ref(:,j) = I(:,i);
09:         j = j+1;
10:     else
11:         I_hd(:,k)=I(:,i);
12:         k = k+1;
13: end
14: End
```

Figure 6.6: (a) Original image (b) Reference sub-sampled image ($I_{\text{ref}}$) (c) Data hiding sub-sampled version ($I_{\text{hd}}$)

Figure 6.7: Selection of pixels from reference sub-sampled verion and data hiding version (a) First block pixels (b) Second block pixels
Figure 6.8: Flow chart of embedding process using RW-HPBS technique.

The size of $I_{dh}$ is twice as that of $I_{ref}$. These two steps are explained in Figure 6.6. Figure 6.6 (a) is a matrix with 3 rows and 6 columns. Figure 6.6 (b) and Figure 6.6 (c) shows the reference and data hiding sub-sampled versions of Figure 6.6 (a), respectively. Blocks of size $2 \times 1$ are created by combining the pixels of $I_{ref}$ and $I_{hd}$. For this, first pixel from $I_{ref}$ and $I_{hd}$ are combined to generate the first block. By using the same pixel of $I_{ref}$ another block is created by taking the $2^{nd}$ pixels of $I_{hd}$ in the same row as shown in
Figure 6.7. This procedure is repeated by combining 2\textsuperscript{nd} column of \(I_{ref}\) with 3\textsuperscript{rd} and 4\textsuperscript{th} column of \(I_{hd}\) and so on. Therefore, by using 3 pixels of the \(I_{ref}\) and 6 pixels of the \(I_{hd}\), total 6 blocks are created and we can embed a maximum of six bits, if all the blocks are stable.

After block creation watermark is embedded using the following steps:

1. \(dl\) and \(dr\) are calculated using Equation (6.1).
2. Category of block is determined through the relation given in Equation (6.4).
3. A block is considered as stable if it embeds correct message bit and after embedding, it does not change its category. If a block is unstable, then its location is stored in location map.
4. After stability check, \(\alpha\) is calculated using the following relation:
   
   \[ \alpha = \frac{x_{ref} - x_{hd}}{2} \]  
   
   where \(x_{ref}\) and \(x_{hd}\) are the block bits that come from \(I_{ref}\) and \(I_{hd}\), respectively.
5. Embedding has been performed through strategy shown in Figure 6.3.
6. To obtain the watermarked image \((H')\), perform inverse of the modified down sampling.

Figure 6.8 shows the flow chart for embedding process.

### 6.3.2 Extraction using RW-HPBS

Extraction process is simply the reverse of the embedding procedure. Let \(I'\) represents the watermarked image of size \(X \times Y\). Down-sampled \(I'\) to get two sub-sampled versions of the original image, reference sub-sampled version \(I'_{ref}\) of size \(X \times (Y/3)\), and data hiding sub-sampled version \(I'_{hd}\) of size \(X \times ((Y/3) \times 2)\). Down sampling of \(I'\) has been performed in the same way as mentioned in Algorithm 6.1. The size of \(I'_{dh}\) is twice as that of \(I'_{ref}\) as is discussed in embedding process. Create a block by taking first pixel of \(I'_{ref}\) and \(I'_{hd}\); then, stability of the block has been checked with the help of the location map. If the block is unstable, then leave the block unchanged and no message bit will be extracted. For stable block, message bit is extracted and the block is recovered. To recover the block, \(\alpha\) is computed using Equation (6.5). For \(|\alpha| > T\), if the block belongs to either of case 5, 7 or 10 then bit “0” will be extracted and pixel gray scale values will remain intact. Otherwise, bit “1” will be extracted and \(\alpha\) will be shifted back towards the zero by adding (or subtracting) \(\beta\) for positive (or negative) value of \(\alpha\). Therefore, by shifting \(\alpha\) back to its original value, all the pixels in set \(I_{hd}\) will be restored and block will be recovered. For \(|\alpha| < T\), bit “0” will be extracted and no change will occur in pixel grayscale value of block. Second block is created by combining the same pixel of \(I'_{ref}\) with 2\textsuperscript{nd} pixel of \(I'_{hd}\) as shown in Figure 6.7. This process is repeated for
all the blocks. To obtain the original image, the modified down-sampling is performed at the end of the algorithm. Figure 6.8 shows the flow chart for extraction process.

Figure 6.8: Flow chart for extraction process.

6.3.3 Embedding of Location Map as an Auxiliary Information

A location map is generated in the proposed improved RDHBS and proposed RW-HPBS approach, which is required at the extraction side to perform watermark extraction and image recovery. Gao et al. [42] have not addressed the issue of whether the location map is communicated as side information or not and thus, their approach can be considered as a semi-blind watermarking approach. To make the proposed technique a blind one, the location map is embedded as an auxiliary information in the watermarked

---

**Figure 6.9: Flow chart for message extraction in the RW-HPBS method.**

**6.3.3 Embedding of Location Map as an Auxiliary Information**

A location map is generated in the proposed improved RDHBS and proposed RW-HPBS approach, which is required at the extraction side to perform watermark extraction and image recovery. Gao et al. [42] have not addressed the issue of whether the location map is communicated as side information or not and thus, their approach can be considered as a semi-blind watermarking approach. To make the proposed technique a blind one, the location map is embedded as an auxiliary information in the watermarked
image. For this purpose, the generated location map is embedded by using a modification of the techniques reported in [103] and [30].

To embed location map, watermarked image is first decomposed into wavelet sub-bands up to 3rd level. IDWT using CDF lifting scheme is employed for this purpose. Location map is first compressed using Huffman coding. Third level approximation sub-band $LL3$ has been used to embed the compressed location map. This is because in case of common signal processing operations (that usually fall in high frequency range such as JPEG Compression, noise addition, etc.), the approximation sub-band of wavelet transform is not that much affected relative to the high frequency sub-bands, and especially, if we go to the higher resolution levels [39]. However, it is to be noted that the proposed technique is of fragile nature. Therefore, watermark will be already destroyed due to any signal processing operation. Hence, even when the location map is fully extracted, the watermark may not be extracted correctly.

Compressed location map is embedded by LSB replacement in $LL3$, whereby the original LSBs are saved before replacement. After embedding the location map, LSBs are embedded in the first level horizontal, vertical, and diagonal sub-bands ($HH$, $HL$, and $LH$, respectively). For this purpose, before embedding, we empty the coefficients having ±2 value in each sub-band by using Equation (6.6).

$$K'_q(i, j) = \begin{cases} K_q(i, j) + 1 & \text{if } K_q(i, j) \geq 2 \\ K_q(i, j) - 1 & \text{if } K_q(i, j) \leq -2 \end{cases}$$

(6.6)

where, $K$ is the $(i^{th}, j^{th})$ coefficient of sub-band $q$ and $q = \{2, 3, 4\}$ for $LH$, $HL$ and $HH$ sub-bands, respectively. Then LSBs are embedded using the following relation:

$$K'_q(i, j) = \begin{cases} K_q(i, j) + 1 & \text{if } \{K_q(i, j) = 1 \land LSB(k) = 1\} \\ K_q(i, j) & \text{elseif } \{K_q(i, j) = 1 \land LSB(k) = 0\} \\ K_q(i, j) - 1 & \text{elseif } \{K_q(i, j) = -1 \land LSB(k) = 1\} \\ K_q(i, j) & \text{elseif } \{K_q(i, j) = -1 \land LSB(k) = 0\} \end{cases}$$

(6.7)

At the end, inverse IDWT is performed to get final marked image.

At the receiving side, first the location map is extracted. For this purpose, the watermarked image is decomposed into wavelet sub-bands up to 3rd level. Then, LSBs are collected (from $HH$, $HL$ and $LH$ sub-bands) and coefficient values are restored by using Equation (6.8).

$$K_q(i, j) = \begin{cases} K'_q(i, j); LSB(k) = 0 & \text{if } \{K'_q(i, j) = 1 \lor K'_q(i, j) = -1\} \\ K'_q(i, j) - 1; LSB(k) = 1 & \text{elseif } K'_q(i, j) = 2 \\ K'_q(i, j) + 1; LSB(k) = 1 & \text{elseif } K'_q(i, j) = -2 \\ K'_q(i, j) - 1 & \text{elseif } K'_q(i, j) > 2 \\ K'_q(i, j) + 1 & \text{elseif } K'_q(i, j) < -2 \end{cases}$$

(6.8)
After this, location map is extracted from the LSBs of LL3 sub-band and original LSBs are restored. Inverse IDWT is employed to get actual watermarked image. Location map is decompressed and with the help of this location map, extraction and restoration process is performed as discussed in Section 6.3.2.

### 6.3.4 Additional Use of Location Map for Authentication

Location map is an extra information that is required at receiving side for extraction and recovery processes. Therefore, it is also embedded along with the actual payload. In this technique, subsidiary advantage has been taken with this overhead data by utilizing it in authentication process. This process ensures that, whether the received image has undergone any type of attack or not. In this work, we have concentrated on only manipulation and collage attacks. Manipulation attacks are easy to detect; while, detecting a collage attack is problematic. In order to detect college attack, location map is correlated with third level approximation sub-band LL3 then it is embedded in watermarked image, as discussed in Section 6.3.3.

For authentication at receiving end, first location map is extracted from the received image as explained in Section 6.3.3. To get original values of location map, it is correlated with third level approximation sub-band LL3 and then Huffman decoding employed for decompression process. The resultant location map is used for watermark extraction and image restoration processes. Since, extraction is reversible; therefore, after extraction and recovery processes watermarked image is restored to its original form. Then, location map is computed for recovered image. The two location maps, one extracted from the watermarked image and second generated from recovered image, are compared. If both location maps are the same, then their difference will be zero and the received image is authentic. Otherwise, it is concluded that image has undergone some kind of adversary attacks (either manipulation or collage).

### 6.4 Results and Discussion

To show the effectiveness of the proposed improved RDHBS technique, simulation results are obtained on the same images with same embedding level and block size as adopted in [42]. Image and block sizes are shown in Table 6.3.

<table>
<thead>
<tr>
<th>Image</th>
<th>Lena</th>
<th>Baboon</th>
<th>Boat</th>
<th>Mpic1</th>
<th>Mpic2</th>
<th>Cpic1</th>
<th>Cpic2</th>
<th>Cpic3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of image</td>
<td>512×512</td>
<td>512×512</td>
<td>512×512</td>
<td>512×512</td>
<td>512×512</td>
<td>1536×1920</td>
<td>1536×1920</td>
<td>1536×1920</td>
</tr>
<tr>
<td>Block</td>
<td>8</td>
<td>12</td>
<td>18</td>
<td>8</td>
<td>8</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>
MSE and PSNR are well-known distortion measures for accessing distortion in the watermarked image. MSE is given by the relation shown in Equation (6.9):

$$MSE = \frac{1}{M \times N} \sum_{r=1}^{M} \sum_{c=1}^{N} (I' - I)$$  \hspace{1cm} (6.9)

while, PSNR is given in Equation (6.10).

$$PSNR = 10 \times \log_{10} \left( \frac{255^2}{MSE} \right)$$  \hspace{1cm} (6.10)

Effective payload and PSNR based comparisons of the proposed improved RDHBS and RDHBS [42] techniques are shown in Table 6.4. From the table, it is to be noted that, the proposed improved RDHBS technique provides significant improvement in capacity versus imperceptibility tradeoff. Improvement is due to the removal of BCH coding and permutation scheme from RDHBS [42] approach. As proposed improved version embeds correct message bit only, therefore, the uncertainty of complete error correction is not present in this case.

Simulation results for the proposed RW-HPBS technique are displayed in Figure 6.10. Figure 6.10, column (d) proved that the proposed RW-HPBS method results in complete recovery of the original image after extraction process.

Table 6.4: PSNR (dB) and payload comparison of RDHBS [42] and proposed improved RDHBS method.

<table>
<thead>
<tr>
<th>Images</th>
<th>Embedding level (β)</th>
<th>RDHBS [42] method</th>
<th>Proposed Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>Payload (bits)</td>
<td>PSNR (dB)</td>
</tr>
<tr>
<td>Lena</td>
<td>2</td>
<td>44.19</td>
<td>778</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>38.23</td>
<td>768</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>34.82</td>
<td>744</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>32.46</td>
<td>726</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>30.53</td>
<td>726</td>
</tr>
<tr>
<td>Baboon</td>
<td>2</td>
<td>43.84</td>
<td>560</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>37.83</td>
<td>560</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>34.31</td>
<td>560</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>31.81</td>
<td>560</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>29.87</td>
<td>560</td>
</tr>
<tr>
<td>Boat</td>
<td>2</td>
<td>48.87</td>
<td>246</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>42.64</td>
<td>239</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>38.95</td>
<td>257</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>36.08</td>
<td>261</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>34.04</td>
<td>274</td>
</tr>
<tr>
<td></td>
<td>Mpic1</td>
<td></td>
<td>Mpic2</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>51.91</td>
<td>45.88</td>
<td>51.47</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>95</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>53.22</td>
<td>47.35</td>
<td>52.20</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>600</td>
<td>800</td>
</tr>
</tbody>
</table>
6.4.1 Effect of Reducing the Block Size

As the proposed technique is a block based watermarking approach; therefore, changing block size affects the watermark embedding capacity. This fact can be observed clearly from the analysis of the results presented in Table 6.5. If we are using the block size $8 \times 8$, then the total number of pixels which will be shifted to embed one bit are 32. However, if we reduce the block size to $2 \times 2$ then, with a shift of two pixels we embed one bit. As a result, with block size $2 \times 2$, embedding capacity is increased by 16 times compared to that of $8 \times 8$. However, it is to be noted that by reducing block size, the size of the location
map is also increases. Consequently, one has to make a tradeoff. PSNR and payload comparison of proposed improved method with block size 8×8 and 2×2 is shown in Table 6.5. Image size for all the experiments is 512×512.

Table 6.5: Imperceptibility and payload comparison of proposed improved RDHBS method for different block sizes.

<table>
<thead>
<tr>
<th>Images</th>
<th>Embedding level (β)</th>
<th>Block size 8×8</th>
<th>Block size 2×2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>Payload(bits)</td>
<td>PSNR (dB)</td>
</tr>
<tr>
<td>Lena</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>46.46</td>
<td>2400</td>
<td>46.47</td>
</tr>
<tr>
<td>4</td>
<td>40.44</td>
<td>2400</td>
<td>40.45</td>
</tr>
<tr>
<td>6</td>
<td>36.92</td>
<td>2400</td>
<td>36.93</td>
</tr>
<tr>
<td>8</td>
<td>34.42</td>
<td>2400</td>
<td>34.43</td>
</tr>
<tr>
<td>10</td>
<td>32.48</td>
<td>2400</td>
<td>32.49</td>
</tr>
<tr>
<td>Baboon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>45.99</td>
<td>1100</td>
<td>46.00</td>
</tr>
<tr>
<td>4</td>
<td>39.69</td>
<td>1100</td>
<td>39.70</td>
</tr>
<tr>
<td>6</td>
<td>36.17</td>
<td>1100</td>
<td>36.18</td>
</tr>
<tr>
<td>8</td>
<td>33.67</td>
<td>1100</td>
<td>33.68</td>
</tr>
<tr>
<td>10</td>
<td>31.73</td>
<td>1100</td>
<td>31.74</td>
</tr>
<tr>
<td>Boat</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>48.92</td>
<td>324</td>
<td>48.92</td>
</tr>
<tr>
<td>4</td>
<td>42.89</td>
<td>324</td>
<td>42.90</td>
</tr>
<tr>
<td>6</td>
<td>39.38</td>
<td>324</td>
<td>39.39</td>
</tr>
<tr>
<td>8</td>
<td>36.88</td>
<td>324</td>
<td>36.89</td>
</tr>
<tr>
<td>10</td>
<td>34.94</td>
<td>324</td>
<td>34.95</td>
</tr>
<tr>
<td>Mpic1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>53.22</td>
<td>600</td>
<td>53.23</td>
</tr>
<tr>
<td>4</td>
<td>47.35</td>
<td>600</td>
<td>47.36</td>
</tr>
<tr>
<td>6</td>
<td>43.86</td>
<td>600</td>
<td>43.87</td>
</tr>
<tr>
<td>8</td>
<td>41.38</td>
<td>600</td>
<td>41.39</td>
</tr>
<tr>
<td>10</td>
<td>39.44</td>
<td>600</td>
<td>39.45</td>
</tr>
<tr>
<td>Mpic2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>52.20</td>
<td>800</td>
<td>52.21</td>
</tr>
<tr>
<td>4</td>
<td>46.18</td>
<td>800</td>
<td>46.19</td>
</tr>
<tr>
<td>6</td>
<td>42.65</td>
<td>800</td>
<td>42.66</td>
</tr>
<tr>
<td>8</td>
<td>40.16</td>
<td>800</td>
<td>40.17</td>
</tr>
<tr>
<td>10</td>
<td>38.22</td>
<td>800</td>
<td>38.23</td>
</tr>
<tr>
<td>Cpic1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>45.71</td>
<td>4200</td>
<td>45.72</td>
</tr>
<tr>
<td>4</td>
<td>39.69</td>
<td>4200</td>
<td>39.70</td>
</tr>
<tr>
<td>6</td>
<td>36.17</td>
<td>4200</td>
<td>36.18</td>
</tr>
<tr>
<td>8</td>
<td>33.67</td>
<td>4200</td>
<td>33.68</td>
</tr>
<tr>
<td>10</td>
<td>31.73</td>
<td>4200</td>
<td>31.74</td>
</tr>
<tr>
<td>Cpic2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>47.51</td>
<td>4200</td>
<td>47.52</td>
</tr>
<tr>
<td>4</td>
<td>41.94</td>
<td>4200</td>
<td>41.95</td>
</tr>
<tr>
<td>6</td>
<td>37.97</td>
<td>4200</td>
<td>37.96</td>
</tr>
<tr>
<td>8</td>
<td>35.47</td>
<td>4200</td>
<td>35.48</td>
</tr>
<tr>
<td>10</td>
<td>33.54</td>
<td>4200</td>
<td>33.55</td>
</tr>
</tbody>
</table>
6.4.2 Effect of Modified Down-Sampling

In RW-HPBS method, down-sampling is performed to obtain two sub-sampled versions of the original image ($I_{ref}$ and $I_{hd}$). Then, blocks are generated by utilizing the pixels from the sets $I_{ref}$ and $I_{hd}$ in a way that consecutive pixels fall in one block. Since, the grayscale values of the consecutive pixels are close to each other and their difference is small. According to the embedding procedure of the proposed RW-HPBS, one can embed correct message bit if the difference between pixels is small. Hence, probability of a block to become unstable due to the second condition of instability is very minimized, which increases the watermark capacity.

RW-HPBS technique is compared with the proposed improved RDHBS approach and Luo et al. [102] technique. Number of experiments is performed using different images on different embedding levels. As the Images are of size 512x512, therefore, the pixels of last two columns are ignored to make them multiple of three. Simulation results are compared with Luo et al. [28] approach. Comparison shows that the RW-HPBS method gives better results compared to both methods in terms of embedding capacity. $\beta$ is selected as twice the threshold $T$. Effective payload and PSNR based comparison of the RW-HPBS and improved RDHBS method is made in Table 6.6 and Table 6.7, while comparison with Luo et al. [102] approach is made in Table 6.8 and Table 6.9. Simulation results make it clear that the complete recovery of image is possible with RW-HPBS method with high watermark bits.

Table 6.6: Imperceptibility and payload comparison of the proposed improved RDHBS method (block size 8x8) and the proposed RW-HPBS method.

<table>
<thead>
<tr>
<th>Images</th>
<th>Watermarked bits</th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>3500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena Image</td>
<td>$\beta=2$ PSNR</td>
<td>Improved RDHBS</td>
<td>50.25</td>
<td>47.25</td>
<td>45.49</td>
</tr>
<tr>
<td></td>
<td>RW-HPBS</td>
<td>65.31</td>
<td>62.32</td>
<td>60.56</td>
<td>59.89</td>
</tr>
<tr>
<td></td>
<td>$\beta=4$ PSNR</td>
<td>Improved RDHBS</td>
<td>44.22</td>
<td>41.23</td>
<td>39.47</td>
</tr>
<tr>
<td></td>
<td>RW-HPBS</td>
<td>59.29</td>
<td>56.30</td>
<td>54.54</td>
<td>53.87</td>
</tr>
<tr>
<td></td>
<td>$\beta=6$ PSNR</td>
<td>Improved RDHBS</td>
<td>40.70</td>
<td>37.71</td>
<td>35.95</td>
</tr>
<tr>
<td></td>
<td>RW-HPBS</td>
<td>55.77</td>
<td>52.77</td>
<td>51.02</td>
<td>50.35</td>
</tr>
<tr>
<td></td>
<td>$\beta=8$ PSNR</td>
<td>Improved RDHBS</td>
<td>38.20</td>
<td>35.21</td>
<td>33.45</td>
</tr>
<tr>
<td></td>
<td>RW-HPBS</td>
<td>53.27</td>
<td>50.28</td>
<td>48.52</td>
<td>47.85</td>
</tr>
<tr>
<td></td>
<td>$\beta=10$ PSNR</td>
<td>Improved RDHBS</td>
<td>36.27</td>
<td>33.27</td>
<td>31.51</td>
</tr>
<tr>
<td></td>
<td>RW-HPBS</td>
<td>51.34</td>
<td>48.34</td>
<td>46.58</td>
<td>45.91</td>
</tr>
<tr>
<td>Image</td>
<td>$\beta$</td>
<td>$\text{PSNR}$</td>
<td>Improved RDHBS</td>
<td>RW-HPBS</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>---------------</td>
<td>----------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Barbara</strong></td>
<td>$2$</td>
<td>$50.91$</td>
<td>$47.91$</td>
<td>$46.15$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$4$</td>
<td>$66.01$</td>
<td>$63.01$</td>
<td>$61.25$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$6$</td>
<td>$44.89$</td>
<td>$41.89$</td>
<td>$40.13$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$8$</td>
<td>$59.99$</td>
<td>$56.99$</td>
<td>$55.23$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$10$</td>
<td>$41.36$</td>
<td>$38.37$</td>
<td>$36.61$</td>
<td></td>
</tr>
<tr>
<td><strong>Airplane</strong></td>
<td>$2$</td>
<td>$53.97$</td>
<td>$50.97$</td>
<td>$49.21$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$4$</td>
<td>$41.15$</td>
<td>$38.47$</td>
<td>$36.74$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$6$</td>
<td>$66.01$</td>
<td>$63.01$</td>
<td>$61.25$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$8$</td>
<td>$44.89$</td>
<td>$41.89$</td>
<td>$40.13$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$10$</td>
<td>$59.99$</td>
<td>$56.99$</td>
<td>$55.23$</td>
<td></td>
</tr>
<tr>
<td><strong>Aerial</strong></td>
<td>$2$</td>
<td>$41.36$</td>
<td>$38.37$</td>
<td>$36.61$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$4$</td>
<td>$55.89$</td>
<td>$52.89$</td>
<td>$51.13$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$6$</td>
<td>$38.32$</td>
<td>$35.32$</td>
<td>$33.56$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$8$</td>
<td>$53.39$</td>
<td>$50.39$</td>
<td>$48.63$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$10$</td>
<td>$36.38$</td>
<td>$33.38$</td>
<td>$31.62$</td>
<td></td>
</tr>
<tr>
<td><strong>Truck</strong></td>
<td>$2$</td>
<td>$51.45$</td>
<td>$48.45$</td>
<td>$46.69$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$4$</td>
<td>$40.82$</td>
<td>$37.82$</td>
<td>$36.06$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$6$</td>
<td>$55.89$</td>
<td>$52.89$</td>
<td>$51.13$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$8$</td>
<td>$38.32$</td>
<td>$35.32$</td>
<td>$33.56$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$10$</td>
<td>$53.39$</td>
<td>$50.39$</td>
<td>$48.63$</td>
<td></td>
</tr>
</tbody>
</table>

**Page 94**
Table 6.7: Imperceptibility and payload comparison of the proposed improved $RDHBS$ method (block size $2\times2$) and the proposed $RW$-$HPBS$ method.

<table>
<thead>
<tr>
<th>Images</th>
<th>Watermarked bits</th>
<th>16000</th>
<th>32000</th>
<th>48000</th>
<th>53000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena Image</td>
<td>$\beta=2$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>50.27</td>
<td>47.26</td>
<td>45.50</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>53.30</td>
<td>50.29</td>
<td>48.53</td>
<td>48.10</td>
</tr>
<tr>
<td></td>
<td>$\beta=4$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>44.24</td>
<td>41.24</td>
<td>39.48</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>47.28</td>
<td>44.27</td>
<td>42.51</td>
<td>42.08</td>
</tr>
<tr>
<td></td>
<td>$\beta=6$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>40.72</td>
<td>37.72</td>
<td>35.96</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>43.76</td>
<td>40.75</td>
<td>38.99</td>
<td>38.55</td>
</tr>
<tr>
<td></td>
<td>$\beta=8$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>38.22</td>
<td>35.22</td>
<td>33.46</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>41.26</td>
<td>38.25</td>
<td>36.49</td>
<td>36.06</td>
</tr>
<tr>
<td></td>
<td>$\beta=10$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>36.30</td>
<td>33.28</td>
<td>31.52</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>39.32</td>
<td>36.31</td>
<td>34.55</td>
<td>34.12</td>
</tr>
<tr>
<td>Barbara Image</td>
<td>$\beta=2$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>50.93</td>
<td>47.92</td>
<td>46.16</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>53.99</td>
<td>50.98</td>
<td>49.22</td>
<td>48.79</td>
</tr>
<tr>
<td></td>
<td>$\beta=4$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>44.91</td>
<td>41.90</td>
<td>40.14</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>47.97</td>
<td>44.96</td>
<td>43.20</td>
<td>42.77</td>
</tr>
<tr>
<td></td>
<td>$\beta=6$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>41.38</td>
<td>38.38</td>
<td>36.62</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>44.45</td>
<td>41.44</td>
<td>39.68</td>
<td>39.25</td>
</tr>
<tr>
<td></td>
<td>$\beta=8$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>36.88</td>
<td>35.88</td>
<td>34.12</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>41.95</td>
<td>38.94</td>
<td>37.18</td>
<td>36.75</td>
</tr>
<tr>
<td></td>
<td>$\beta=10$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>36.95</td>
<td>33.94</td>
<td>32.18</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>40.01</td>
<td>37.00</td>
<td>35.24</td>
<td>34.81</td>
</tr>
<tr>
<td>Airplane Image</td>
<td>$\beta=2$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>50.38</td>
<td>47.37</td>
<td>45.61</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>53.41</td>
<td>50.40</td>
<td>48.64</td>
<td>48.21</td>
</tr>
<tr>
<td></td>
<td>$\beta=4$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>44.36</td>
<td>41.35</td>
<td>39.59</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>47.39</td>
<td>44.38</td>
<td>42.62</td>
<td>42.19</td>
</tr>
<tr>
<td></td>
<td>$\beta=6$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>40.84</td>
<td>37.83</td>
<td>36.07</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>43.87</td>
<td>40.86</td>
<td>39.10</td>
<td>38.67</td>
</tr>
<tr>
<td></td>
<td>$\beta=8$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>38.34</td>
<td>35.33</td>
<td>33.57</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>41.37</td>
<td>38.36</td>
<td>36.60</td>
<td>36.17</td>
</tr>
<tr>
<td></td>
<td>$\beta=10$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>36.40</td>
<td>33.39</td>
<td>31.63</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>39.43</td>
<td>36.42</td>
<td>34.66</td>
<td>34.23</td>
</tr>
<tr>
<td>Aerial Image</td>
<td>$\beta=2$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>50.93</td>
<td>47.92</td>
<td>46.16</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>53.96</td>
<td>50.95</td>
<td>49.19</td>
<td>48.76</td>
</tr>
<tr>
<td></td>
<td>$\beta=4$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>44.88</td>
<td>41.90</td>
<td>40.14</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>47.94</td>
<td>44.93</td>
<td>43.17</td>
<td>42.74</td>
</tr>
<tr>
<td></td>
<td>$\beta=6$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>41.38</td>
<td>38.38</td>
<td>36.62</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>44.42</td>
<td>41.41</td>
<td>39.65</td>
<td>39.21</td>
</tr>
<tr>
<td></td>
<td>$\beta=8$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>38.88</td>
<td>35.88</td>
<td>34.12</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>41.92</td>
<td>38.91</td>
<td>37.15</td>
<td>36.72</td>
</tr>
<tr>
<td></td>
<td>$\beta=10$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>36.95</td>
<td>33.94</td>
<td>32.18</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>39.98</td>
<td>36.97</td>
<td>35.21</td>
<td>34.78</td>
</tr>
<tr>
<td></td>
<td>$\beta=2$ PSNR</td>
<td>Improved $RDHBS$</td>
<td>49.47</td>
<td>46.46</td>
<td>44.70</td>
</tr>
<tr>
<td></td>
<td>$RW$-$HPBS$</td>
<td>52.54</td>
<td>49.53</td>
<td>47.77</td>
<td>47.34</td>
</tr>
</tbody>
</table>
Modified down sampling provides a larger size of the data hiding sub-sampled version of the original image, than the reference sub-sampled version. Therefore, the embedding capacity of RW-HPBS method is very large compared to the proposed improved method. In Table 6.6 and Table 6.7, different block sizes for proposed improved RDHBS method are taken and the results are compared with RW-HPBS method. Simulation results show that RW-HPBS method provides much higher embedding capacity than proposed improved method. In Table 6.8 and Table 6.9, comparison with Luo et al. [102] approach is also made, which clearly shows that RW-HPBS is better.

Table 6.8: PSNR comparison of Luo’s approach [102] and the proposed RW-HPBS method.

<table>
<thead>
<tr>
<th>Track Image</th>
<th>β=4 PSNR</th>
<th>Improved RDHBS</th>
<th>RW-HPBS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>43.45</td>
<td>46.52</td>
</tr>
<tr>
<td>β=6 PSNR</td>
<td></td>
<td>39.93</td>
<td>42.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>37.43</td>
<td>40.50</td>
</tr>
<tr>
<td>β=10 PSNR</td>
<td></td>
<td>35.49</td>
<td>38.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.85</td>
<td>41.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>38.68</td>
<td>41.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.85</td>
<td>38.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33.30</td>
<td>35.73</td>
</tr>
</tbody>
</table>

Table 6.8: PSNR comparison of Luo’s approach [102] and the proposed RW-HPBS method.

(a) Lena Image

<table>
<thead>
<tr>
<th>Watermarked bits</th>
<th>26465</th>
<th>71769</th>
<th>105421</th>
<th>28860</th>
<th>79787</th>
<th>118372</th>
<th>27723</th>
<th>77861</th>
<th>117160</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luo’s PSNR</td>
<td>49.68</td>
<td>44.35</td>
<td>41.65</td>
<td>48.92</td>
<td>43.55</td>
<td>40.79</td>
<td>48.66</td>
<td>43.22</td>
<td>40.41</td>
</tr>
<tr>
<td>RW-HPBS’s PSNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(β=2)</td>
<td>51.11</td>
<td>46.78</td>
<td>45.11</td>
<td>50.74</td>
<td>46.32</td>
<td>44.61</td>
<td>50.91</td>
<td>46.43</td>
<td>44.65</td>
</tr>
<tr>
<td>(β=1)</td>
<td>57.13</td>
<td>52.80</td>
<td>51.13</td>
<td>56.76</td>
<td>52.34</td>
<td>50.63</td>
<td>56.93</td>
<td>52.45</td>
<td>50.67</td>
</tr>
<tr>
<td>β=1 [Maximum capacity = 175243 PSNR = 48.92], β=2 [Maximum capacity = 175245 PSNR = 42.90]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) Barbra Image

<table>
<thead>
<tr>
<th>Watermarked bits</th>
<th>18281</th>
<th>50718</th>
<th>76420</th>
<th>19788</th>
<th>56054</th>
<th>85033</th>
<th>18849</th>
<th>54558</th>
<th>83928</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luo’s PSNR</td>
<td>49.59</td>
<td>44.03</td>
<td>41.05</td>
<td>48.83</td>
<td>43.25</td>
<td>40.23</td>
<td>48.58</td>
<td>42.95</td>
<td>39.90</td>
</tr>
<tr>
<td>RW-HPBS’s PSNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(β=2)</td>
<td>53.41</td>
<td>48.98</td>
<td>47.20</td>
<td>53.07</td>
<td>48.55</td>
<td>46.74</td>
<td>53.28</td>
<td>48.67</td>
<td>46.80</td>
</tr>
<tr>
<td>(β=1)</td>
<td>59.43</td>
<td>55.00</td>
<td>53.22</td>
<td>59.09</td>
<td>54.57</td>
<td>52.76</td>
<td>59.30</td>
<td>54.69</td>
<td>52.82</td>
</tr>
<tr>
<td>β=1 [Maximum capacity = 175341 PSNR = 49.62], β=2 [Maximum capacity = 175336 PSNR = 43.60]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(c) Airplane Image

<table>
<thead>
<tr>
<th>Watermarked bits</th>
<th>36221</th>
<th>86036</th>
<th>115262</th>
<th>39853</th>
<th>96501</th>
<th>130636</th>
<th>38976</th>
<th>95580</th>
<th>130948</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luo’s PSNR</td>
<td>49.80</td>
<td>44.64</td>
<td>42.04</td>
<td>49.03</td>
<td>43.83</td>
<td>41.18</td>
<td>48.77</td>
<td>43.50</td>
<td>40.78</td>
</tr>
<tr>
<td>RW-HPBS’s PSNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(β=2)</td>
<td>49.86</td>
<td>46.11</td>
<td>44.84</td>
<td>49.45</td>
<td>45.61</td>
<td>44.29</td>
<td>49.55</td>
<td>45.65</td>
<td>44.28</td>
</tr>
<tr>
<td>(β=1)</td>
<td>55.88</td>
<td>52.13</td>
<td>50.86</td>
<td>55.47</td>
<td>51.63</td>
<td>50.31</td>
<td>55.57</td>
<td>51.67</td>
<td>50.30</td>
</tr>
<tr>
<td>β=1 [Maximum capacity = 175392 PSNR = 49.62], β=2 [Maximum capacity = 175414 PSNR = 43.01]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(d) Aerial Image

<table>
<thead>
<tr>
<th>Watermarked bits</th>
<th>8909</th>
<th>25886</th>
<th>41342</th>
<th>9156</th>
<th>26789</th>
<th>43650</th>
<th>8677</th>
<th>25767</th>
<th>42124</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luo’s PSNR</td>
<td>49.48</td>
<td>43.68</td>
<td>40.43</td>
<td>48.73</td>
<td>42.90</td>
<td>39.62</td>
<td>48.49</td>
<td>42.65</td>
<td>39.34</td>
</tr>
<tr>
<td>RW-HPBS’s PSNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(β=2)</td>
<td>56.50</td>
<td>51.87</td>
<td>49.84</td>
<td>56.38</td>
<td>51.72</td>
<td>49.60</td>
<td>56.61</td>
<td>51.89</td>
<td>49.75</td>
</tr>
<tr>
<td>(β=1)</td>
<td>62.52</td>
<td>57.89</td>
<td>55.86</td>
<td>62.40</td>
<td>57.74</td>
<td>55.62</td>
<td>62.63</td>
<td>57.91</td>
<td>55.78</td>
</tr>
<tr>
<td>(e) Truck Image</td>
<td>Watermarked bits</td>
<td>Luo's PSNR</td>
<td>RW-HPBS's PSNR</td>
<td>beta=1 [Maximum capacity =175193 PSNR=48.17], beta=2 [Maximum capacity =175221 PSNR=42.15]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------</td>
<td>------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16423</td>
<td>30577</td>
<td>51574</td>
<td>18184 33674 57340 16900 32023 54774</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>49.57</td>
<td>43.82</td>
<td>40.60</td>
<td>48.82 43.05 39.81 48.56 42.77 39.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(beta=2)</td>
<td>54.43</td>
<td>49.73</td>
<td>47.46 51.99 49.31 46.99 52.30 49.53 47.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>58.45</td>
<td>55.75</td>
<td>53.48 58.00 55.33 53.02 58.33 55.55 53.22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.9: PSNR comparison of Luo’s approach [102] and proposed RW-HPBS method.
6.4.3 **Effect of Changing the Embedding Level**

Embedding level ($\beta$) decides the amount of distortion produced in the original image due to embedding. For smaller $\beta$, data hiding causes small variations in pixels intensity, hence, imperceptibility improves but at the cost of low robustness. On the other hand, for large value of $\beta$, robustness will generally be high; but visual quality might reduce. In Table 6.6, Table 6.7, Table 6.8, and Table 6.9, PSNR comparison of RW-HPBS technique and other methods is made with different embedding levels on different grayscale images. These tables show that, higher PSNR is obtained with smaller embedding level and vice versa. In Table 6.6 and Table 6.7, block sizes for RW-HPBS method are different, which ensure that, smaller the block size, larger will be the embedding capacity with same PSNR.

6.4.4 **Performance Analysis when the Location Map is embedded as an Auxiliary Information**

Figure 6.11 shows the performance comparison (in terms of capacity versus imperceptibility tradeoff) of the proposed RW-HPBS technique with that of Xuan et al.’s [46] (companding based reversible watermarking) approach, Tian’s [54] (difference expansion based reversible watermarking) approach, Xuan et al.’s [48] (fixed threshold based reversible watermarking) approach, Thodi et al.’s [58] (prediction error based lossless data hiding) approach, and Usman et al.’s [80] (genetic programming based reversible watermarking) approach. It can be observed that the proposed RW-HPBS approach performs well compared to the existing approaches in terms of capacity versus imperceptibility tradeoff. To achieve ‘1’ bit per pixel (bpp), multi-run watermarking is employed.

To show the effectiveness of the proposed RW-HPBS approach, simulations results are computed on an image database. It consists of total 300 images and is used by Usman et al. [91] in the experimental analysis of their work. It contains out-door and in-door images of varying frequency. Figure 6.12 displays the comparison of the proposed RW-HPBS approach with that of Thodi et al.’s [58], Xuan et al.’s [48], and Gao et al.’s [42] approaches, using the image dataset. The plot is taken at constant capacity of 1000 bits. Similarly, Figure 6.13, Figure 6.14, and Figure 6.15 are the comparison of the proposed RW-HPBS approach with that of Thodi et al.’s [58] and Xuan et al.’s [48] techniques for payload of 0.4, 0.5, and 0.7 bpp, respectively (which is quite high capacity compared to that shown in Figure 6.12 i.e. 1000 bits). In Figure 6.12, Figure 6.13, Figure 6.14, and Figure 6.15 a single data point shows an average PSNR value at an interval of 10 images. It can be observed that, the proposed RW-HPBS approach shows better performance compared to that of Thodi et al. [58], Xuan et al. [48], and Gao et al. [42] techniques.
Figure 6.11: Performance comparison of RW-HPBS technique with other existing approaches, using Lena image.

Figure 6.12: Performance comparison of proposed RW-HPBS technique with Thodi et al. [58], Xuan et al. [48], and Gao et al. [42] approaches using database of 300 images (Capacity is taken as 1000 bits).
Figure 6.13: Performance comparison of proposed RW-HPBS technique with Thodi et al. [58] and Xuan et al. [48] approaches using database of 300 images (Capacity is taken as 0.4 bpp).

Figure 6.14: Performance comparison of proposed RW-HPBS technique with Thodi et al. [58] and Xuan et al. [48] approaches using database of 300 images (Capacity is taken as 0.5 bpp).
Figure 6.15: Performance comparison of proposed RW-HPBS technique with Thodi et al. [58] and Xuan et al. [48] approaches using database of 300 images (Capacity is taken as 0.7 bpp).

Similarly, in Figure 6.12, Figure 6.13, Figure 6.14, and Figure 6.15 it can be observed that, on x-axis, when we move from one point to another, Gao et al. [42] and proposed RW-HPBS approaches show small variations in PSNR. However, this variation is high in case of Xuan et al.’s [48] and Thodi et al. [58] approaches. This is mainly due to the fact that, Gao et al. [42] and RW-HPBS techniques changes the pixel intensity by a constant value $\beta$ for embedding the watermark bit and thus, the strength of embedding does not depend upon the pixel intensity or image structure. However, in case of Thodi et al.’s [58] approach, prediction error is computed, which depends upon the current as well as the neighboring pixels intensities. Therefore, for highly textured images, most of the prediction errors are large compared to the smooth images, which results in decrease in PSNR. In case of Xuan et al. [48] technique, watermark embedding is performed in wavelet domain by using a threshold. The value of this threshold is set according to the frequency distribution of an image. However, we have chosen same threshold (i.e. 6) for all the images in the database. This might result in variation of PSNR values.

6.4.5 Image Authentication related Analysis

Authentication of images has been analyzed by using different type of attacks and image processing. Figure 6.16 shows the results related to the authentication.
Column (a) of Figure 6.16 is the collection of some watermarked images; column (b) displays the tampered form of column (a). Column (c) is the difference of the two location maps (as discussed in Section 6.3.3) using non-tampered form of watermarked images (column (a)). Column (d) of Figure 6.16 depicts the difference of the two location maps (as discussed in Section 6.3.3) using tampered form of watermarked images (column (b)). It can be observed that, the difference maps in column (c) are black, which implies that the two location maps are identical. Since, the two location maps are identical; therefore, it is concluded that the watermarked images in column (a) are authentic and have not undergone any kind of adversary attacks.

I. Noise and Manipulation Attack related Analysis

In first row of Figure 6.16, watermarked Pebbles image is changed by adding a salt and pepper noise. In second row, Gaussian noise with 0.01 variance and zero mean is added in watermarked Lena image. In third row, JPEG compression with quality factor taken as 75 is applied on watermarked Baboon image. From column (d) of Figure 6.16, it can be observed that image consists of black and white pixels for the first three rows, which means that according to our technique, images are not authentic. Furthermore, our proposed technique is fragile, as the watermark is destroyed and image cannot be restored due to the presence of noise.

In the fourth row of Figure 6.16, the watermarked Cameraman image is tampered by removing tower. In the fifth row, the watermarked Barbara image is undergone with manipulation attack by removal of a tool from the table. In row six, a portion of image has been cut from the watermarked Aerial image. In the seventh row, a man has been copied and replaced in the watermarked Hill image. In row eight, watermarked Peppers image has been 180° rotated. Column (d) of these rows shows the images of black and white portion, which means that difference of the two location maps contain non-zero regions and images have undergone some kind of malicious attacks. Therefore, our approach can detect the manipulation attack successfully.
II. Collage Attack related Analysis

Row nine of Figure 6.16 contains watermarked Baby image, which has undergone collage/counterfeiting attack. In this type of attack, an attacker has one or more copies of the same image with different watermark. He may combine different portion of these images and construct a forged image. This type of attack is usually difficult to detect. Column (d) of row nine again shows an image of black and white portion, which means that the difference of two location maps contains non-zero regions and thus the image is not authentic. Hence, our approach is able to detect collage attack, which is primarily achieved by correlating the location map with the approximation sub-band before embedding.

6.5 Chapter Summary

This chapter has presented and analyzed two reversible watermarking approaches based upon histogram modification concept. First, improved-RDHBS approach is presented, which makes it possible to recover not only the message but also the image. In improved-RDHBS, first we look at the block conditions before embedding to check, whether it is suitable for watermarking or not. If it is suitable, then bit is embedded, otherwise the block is marked as unstable block. Therefore, BCH coding plus permutation of the message before embedding is not required for correcting any erroneous bits and hence, PSNR and embedding capacity is increased. Furthermore, the conditions for block classification are modified, so that a block may become a candidate for only one category and will not satisfy the conditions for two different categories.

For further improvement in the imperceptibility versus capacity tradeoff, another watermarking approach named as RW-HPBS is discussed. RW-HPBS utilizes the concept of down-sampling to achieve higher PSNR and embedding capacity than the improved-RDHBS method and other existing techniques. Additionally, it is experimentally shown that the proposed RW-HPBS method has extra ability of performing authentication and detecting collage attack.

In the next chapter, the conclusion and the analysis of the work presented in this dissertation is presented.
Chapter 7  CONCLUSIONS AND FUTURE DIRECTIONS

This chapter summarizes the research work presented in this dissertation. In the end, future directions are also presented.

7.1 Research Summary

Watermarking is one of the effective solutions to secure transmission of the digital content over the communication channel. In order to remove the degradation caused by watermark embedding and thus recover the original work, reversible watermarking is applied. Reversible watermarking is very beneficial in applications such as medical imagery, 3D reconstruction, and military imagery.

In this thesis, four different reversible watermarking techniques have been developed. These techniques are providing good capacity versus imperceptibility tradeoff in addition to authentication against manipulation and collage attacks.

The work done in Chapter 3 utilizes the companding concept integrated with GA. Companding is performed block wise based on certain threshold. GA is employed to evolve an optimal/near-optimal threshold matrix by exploiting the distribution of the wavelet coefficients in a block. The threshold map is evolved to improve the performance in terms of imperceptibility versus capacity tradeoff. Simulation results are computed for various medical images as well as some standard images. Experimental analysis proved that the developed technique outperforms the other existing techniques. Since, the proposed technique is reversible; therefore, sensitive regions of medical images can also be used for watermark embedding, which results in an increase in watermark capacity. However, the approach has the limitation of requiring high training time being time consuming.

In the next chapter, i.e. Chapter 4, a reversible watermarking approach for images with depth information is developed. These images are acquired by employing shape from focus approach. The techniques presented in this chapter perform reversible watermarking on images by embedding their corresponding 3D information as a watermark. For this purpose, wavelet coefficients are selected based on a suitable threshold value. GA is utilized to evolve an optimal threshold matrix. The technique provides good imperceptibility versus capacity tradeoff compared to existing techniques with the additional capability of image authentication against manipulation and collage attacks. Furthermore, to
highlight the effectiveness of the developed technique, it is tested on a dataset of 75 images. From the results, it is concluded that the proposed technique outperform the other existing approaches. However, the improvement is due to the high computational time (in training phase). It is inferred from the analysis of the authentication capability that the developed technique can successfully detect the collage and manipulation attacks.

Chapter 5 is about reversible watermarking for 3D camera. In this approach, again the depth information of an image is embedded as a watermark. Additionally, CI approaches (i.e. DE and hybrid PSO-DE) are used for improving capacity versus imperceptibility tradeoff. Experiments are performed on standard images as well as on a dataset of hundred images. From the analysis of the Experimental results, it is concluded that the proposed technique provides good imperceptibility versus capacity tradeoff compared to existing approaches. Additionally, image authentication property has also been achieved against manipulation and collage attack. However, the cost of high computational time at the training phase is the only issue of this approach.

Proposed techniques, presented in chapters 3, 4 and 5 are based on CI methods that involve high computational time in training phase. Therefore, in Chapter 6, a reversible watermarking technique is devised based on histogram shifting and down sampling. This technique is easy to implement and consumes less computational time as it does not involve any CI algorithm. This technique is a suitable choice for real time applications. Simulation results proved that the developed technique provides a good capacity versus imperceptibility tradeoff compared to the existing approaches. It also performed well on the dataset of 300 images. Additionally, it is proved experimentally that the proposed technique provides the authentication capability (against manipulation and collage attacks).

However, it is to be noted that the additional capability of authentication of developed approaches is merely for detecting the existence of any manipulations. The developed approaches are unable to perform localization of the tampered region in the image, which is also considered a potential extension of this current research work.

### 7.2 Future Directions

The research work presented in this thesis can be extended into the following areas:

#### 7.2.1 Improving computational Time

The work presented in this thesis is mostly relying on CI approaches, namely GA, DE, hybrid PSO-DE. Use of CI approaches helps in significantly improving the watermark capacity versus imperceptibility
However, CI approaches are more time consuming, which is undesirable in some real time applications. In future, one may intend to reduce this computational time by employing parallel CI approaches.

7.2.2 Detection of Tampered Region

In this research, authentication capability is provided in the proposed techniques that can detect the presence of manipulation or collage attack in the received image. In future, we intend to develop algorithms that can detect the tampered regions in the received image as well as try to recover these regions.

7.2.3 Developing Tracing Capabilities

In some applications, in addition to authentication, it is very important to identify the source modifying the digital content. Monitoring the usage of digital content and its tracing are some of the potential fields that are used extensively to control movie release and bank transactions records. Some researchers have developed the codes to provide traceability to the pirated data [104], [105]. However, there still exist margin of improvement.

7.2.4 Multiple Watermarking

Multiple watermarking can be a very useful extension of the work presented in this dissertation. It can be effectively used in medical imagery. In the field of medical imagery, one can embed and extract information related to diagnosis of different physicians. Different researchers have presented their work on multiple watermarking [106], [107].
REFERENCES


