Design and Implementation of Transform Domain Watermarking Techniques for Color Image Authentication

Thesis submitted for the Degree of Doctor of Philosophy
(Engineering) in the Department of Computer Science and Engineering,
Faculty of Engineering, Technology and Management,
University of Kalyani

By

Sudipta Kr Ghosal

Under the Supervision of

Prof (Dr.) Jyotsna Kumar Mandal

Department of Computer Science and Engineering
University of Kalyani,
Kalyani, West Bengal, India

November, 2015

To
My Parents
&
All Beloved
Teachers

University of Kalyani

FACULTY OF ENGINEERING, TECHNOLOGY & MANAGEMENT

Prof. J. K. Mandal
DEPT. OF CSE, FACULTY OF ENGINEERING,
TECHNOLOGY & MANAGEMENT
UNIVERSITY OF KALYANI

Kalyani-741235, Nadia
West Bengal, India
Phone: (033) 25809617 (O)
Mobile: +91- 9434352214
e-mail: jkm.cse@gmail.com

Certificate

This is to certify that the thesis entitled **“Design and Implementation of Transform Domain Watermarking Techniques for Color Image Authentication”** submitted by Sudipta Kr Ghosal, who got his research proposal registration with effect from 07.11.2012 (No: KU/Ph.D.-0053 of 2015) for the award of Ph.D. (Engineering) degree of the University of Kalyani is absolutely based upon his own work under my supervision and that neither thesis nor any part of the thesis has been submitted for any degree or diploma or any other academic award anywhere before. I recommend that Mr. Sudipta Kr Ghosal has fulfilled all the requirements according to rules of this University regarding the work embodied in this thesis.

Date:
Place:

Prof. Dr. J. K. Mandal
Department of Computer Science and Engineering
University of Kalyani
Kalyani, West Bengal, India

Acknowledgment

This research work has been carried out as a university research scholar without fellowship basis in the department of Computer Science and Engineering of the University of Kalyani, India. First, I would like to express my sincere gratitude to Prof (Dr.) J. K. Mandal of Computer Science and Engineering, University of Kalyani for his careful supervision of my doctoral thesis and for all his support. He was always available with new ideas, advises and suggestions during the difficult phases of the thesis work. I would also thankful and express my indebtedness to all the reporters, examiners and jury members of the thesis for accepting to review this report and for their careful evaluation of the thesis presentation.

I would like to express my deep regards to Ms. Saswati Mukherjee, Assistant Professor, School of Education Technology, Jadavpur University and Dr. Matangini Chattopadhyay, Director, School of Education Technology, Jadavpur University for their valuable suggestions.

This auspicious occasion gives me a wonderful opportunity and great pleasure to express my thanks to Mr. Arnab Nath, Ms. Sanghamitra De, Ms. Sayani Dhar and Mr. Souvik Dutta Chowdhury for their technical support at different stages of this research work.

I am thankful and have pleasure to express my deep friendship to all the scholars of the department of Computer Science and Engineering that supported my work and entertained me with love; that helped me a lot to progress rapidly in my work on daily basis. I would like to express my thanks to all my present and ex-colleagues Prof. (Dr.) Alok Kumar Ghosh, Prof. (Dr.) Amit Kumar Aditya, Mr. R. V. Pai, Mr. Manoj Kumar Sur, Mr. Santanu Dam, Mr. Tridib Chakraborty, Mr. Avijeet Kundu, Mrs. Amrita Khamrui, Mr. Avijeet Saha, Mr. R. K. Yadav, Mr. Shibaprasad Sen, Mr. Avik Majumdar, Dr. Rupa Bhattachariya, Mr. Rajib Das, Mr. Niladri Sekhar Pal and many more for their support.

I would like to express my gratitude to Mr. Bijoy Roy, Mr. Nirendra Narayan Banerjee, Mr. Pradip Mukherjee and Mr. Amitava Ghosh for being my inspirations toward higher studies.

I would also like to thank my closest friends Mr. Shibdas Mukherjee, Md. Kaisuzzaman, Mr. Taradas Mukherjee and Mr. Sitangshu Pal for supporting and motivating me a lot throughout the research work since journey started.

I would like to express all my sincere love to my parents, my ritual mother, my sisters “Priyanka” and “Pragati”, my brother-in-law and all my family members who always have supported me in every possible way.

Finally, I beg my apologies to those whose names I have missed out to mention.

Sudipta Kr Ghosal
Department of Computer Science and Engineering
University of Kalyani
Kalyani, West Bengal, India

List of Abbreviations

Acronyms	*Descriptions*
2DKLT	Two-Dimensional Karhunen-Loeve Transform
ADE	Adaptive Differential Evolution
BT	Binomial Transform
CAT	Cellular Automata Transform
CDMA	Code Division Multiple Access
CIFT	Cascaded Iterative Fourier Transform
CSF	Contrast Sensitivity Function
CWT	Complex Wavelet Transform
dB	Decibel
DCT	Discrete Cosine Transformation
DFT	Discrete Fourier Transformation
DGT	Discrete Gould Transform
DGTDHS	Discrete Gould Transform based Data Hiding Scheme
DHT	Discrete Hartley Transform
DMST	Deformable Multi-scale Transform
DNWT	Discrete Non-separable Wavelet Transform
DPT	Discrete Pascal Transform, Deformable Pyramid Transform
DPTHDI	Discrete Pascal Transform based Data Hiding Scheme
DRPE	Double Random Phase Encoding
DSWT	Discrete Separable Wavelet Transform
DT-CWT	Dual-Tree Complex Wavelet Transform
DTT	Discrete Trigonometric Transform
DWT	Discrete Wavelet Transformation
DYWT	Dyadic Wavelet Transform
EMD	Exploiting Modification Direction
FCM	Fuzzy c-means Clustering
FFT	Fast Fourier Transformation
FHT	Fast Hadamard Transformation
FMW	Fourier-Mellin Transform based Watermarking

FRAT	Finite Radon Transform
FRFT	Fractional Fourier Transform
FTM	Full-Tree Matrix
GA	Genetic Algorithm
GAO	Genetic Algorithm based Optimization
GBS	Genetic Band Selection
GEMD	Generalized Exploiting Modification Direction
G-LETS D4	Group of Linear Transformations for Dihedral Group of Order 4
HVS	Human Visual System
ICA	Independent Component Analysis
IDHT	Inverse Discrete Hartley Transform
IF	Image Fidelity
IPR	Intellectual Property Rights
IWT	Integer Wavelet Transform
JND	Just Noticeable Distortion
JPEG	Joint Photographers Expert Group
LPM	Log-Polar Map
LSB	Least Significant Bit
LT	Legendre Transform
LWT	Lifting Wavelet Transform
MD	Message Digests
MSB	Most Significant Bit
MSDT	Multi-Scale Derivative Transform
MSE	Mean Squared Error
NC	Normalized Cross-correlation
NMF	Non-negative Matrix Factorization
NPT	Naturalness Preserving Transform
NVF	Noise Visibility Function
OT	Orthogonal Transformation
PHT	Polar Harmonic Transform
PSC	Perceptually Significant Coefficient
PSNR	Peak Signal to Noise Ratio

PSNR	Peak Signal to Noise Ratio
PST	Pinned Sine Transform
PVD	Pixel Value Differencing
QE	Quality Enhancement
QSWT	Qualified Significant Wavelet Tree
RDWT	Redundant Discrete Wavelet Transform
RST	Rotation, Scale and Translation
SCDFT	Spatio-chromatic Discrete Fourier Transform
SD	Standard Deviation
SDE	Standard Deviation Error
SDHT	Separable Discrete Hartley Transform
SF	Spatio-frequency
SIFT	Scale-invariant Feature Transform
SPT	Steerable Pyramid Transform
SSIM	Structural Similarity index
ST	Spread Transform, Stirling Transform
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TPVD	Tri-way Pixel Value Differencing (TPVD)
UIQ	Universal Image Quality index
VC	Visual Cryptography
WBT	Watermarking based on Binomial Transform
WDHT	Watermarking based on Discrete Hartley Transform
WGTD4	Watermarking based on G-lets D4 domain
WHT	Walsh-Hadamard Transform
WLT	Watermarking based on Legendre Transform (WLT)
WPT	Wavelet Packet Transform
WST	Watermarking based on Stirling Transform
XVSS	XOR-based Visual Secret Sharing

List of Symbols

Symbols	*Significance*
σ	Standard Deviation
σ_{Δ}	Standard Deviation Error
$\sum$	Summation
&	Bitwise AND
λ	Number of Secret Bits
$\neq$	Not equal to
$+$	Addition
$-$	Subtraction
$\times$	Multiplication
$/$	Division
$<$	Less than
$\leq$	Less than Equal to
$>$	Greater than
$\geq$	Greater than equal to

List of Tables

List of Figures

Fig. No. *Descriptions* *Page*

Fig. No. *Descriptions* *Page*

Fig. No. *Descriptions* *Page*

List of Publications

Journals

1. Ghosal S.K., Mandal J.K., “Authentication based on Fragile Watermarking in Stirling Transform Domain (AFWSTD)”, Security and Communication Networks, Wiley Online Library, 2015 (Impact Factor: 0.72) (ACCEPTED).
2. Ghosal S.K., Mandal J.K., “Binomial transform based fragile watermarking for image authentication”, Journal of Information Security and Applications, Elsevier B.V., 19(4-5), pp. 272-281, 2014.
3. Ghosal, S.K, Mandal J.K, “Color Image Authentication based on Two-Dimensional Separable Discrete Hartley Transform (CIA2D-SDHT)”, Association for the Advancement of Modelling and Simulation Techniques in Enterprises (AMSE) Journals – 2014-Series: Advances B, 57(1), pp. 68-87, 2014.
4. Ghosal, S.K, Mandal J.K, “A Fragile Watermarking based on Legendre Transform for Color Images (FWLTCI)”, Signal & Image Processing : An International Journal (SIPIJ), DOI: 10.5121/sipij.2013.4410, ISSN : 0976 - 710X (Online) ; 2229 - 3922 (print), 4(4), pp. 119-127, August 2013.
5. Ghosal, S.K, Mandal J.K, "Binomial Transform based Image Authentication (BTIA)", International Journal of Multimedia & Its Applications (IJMA), DOI: 10.5121/ijma.2013.5405, ISSN: 0975 - 5578[Online]; 0975 - 5934 [Print], 5(4), pp. 67-74, August 2013, 2013.
6. Mandal J.K, Ghosal, S.K, "A Fragile Watermarking based on Separable Discrete Hartley Transform for Color Image Authentication (FWSDHTCIA)", Signal & Image Processing : An International Journal (SIPIJ), DOI : 10.5121/sipij.2012.3603, ISSN : 0976 - 710X (Online) ; 2229 - 3922 (print), 3(6), pp. 23-33, 2012.
7. Mandal J.K, Ghosal, S.K, “A Two Dimensional Discrete Fourier Transform Based Secret Data Embedding for Color Image Authentication (2D-DFTSDECIA)”, Signal & Image Processing : An International Journal (SIPIJ), DOI : 10.5121/sipij.2012.3608, ISSN : 0976 - 710X (Online) ; 2229 - 3922 (print), 3(6), pp 87-97, 2012.

Book Chapters

8. Ghosal S.K., “Genetic Algorithm based Optimization of Fragile Watermarking in Discrete Hartley Transform Domain”, (Handbook of Research on Natural Computing for Optimization Problems, Advances in Computational Intelligence and Robotics (ACIR), IGI Global, 701 E. Chocolate Ave., Hershey, PA 17033, USA) (ACCEPTED).

International Conferences

9. Ghosal, S.K, Mandal J.K, “Stirling Transform based Color Image Authentication (STCIA)”, Proceedings of International Conference on Computational Intelligence: Modeling, Techniques and Applications (CIMTA- 2013), Procedia Technology, Elsevier B.V., ISSN: 2212-0173, Vol. 10, pp. 95-104, Kalyani, 2013.
10. Mandal, J. K., Ghosal S. K., “Legendre Transformation based Color Image Authentication (LTCIA)”, Proceedings of Computer Science & Information Technology (CS & IT), DOI : 10.5121/csit.2013.3630, ISSN : 2231 - 5403, AIRCC, Feb.18-20, 2013, pp. 265–272, Bangalore, 2013.
11. Mandal, J. K., Ghosal S. K., “A Fragile Watermarking based on Binomial Transform in Color Images”, SIPP- 2013, Proceedings of Computer Science & Information Technology (CS & IT), DOI : 10.5121/csit.2013.3632, ISSN : 2231 - 5403, AIRCC, Feb.18-20, 2013, pp. 281–288, Bangalore, 2013
12. Mandal J.K, Ghosal, S.K, “Separable Discrete Hartley Transform based Invisible Watermarking for Color Image Authentication (SDHTIWCIA)”, Second International Conference on Digital Image Processing and Pattern Recognition (DPPR-2012), Advances in Intelligent and Soft Computing, Springer Berlin Heidelberg, DOI: 10.1007/978-3-642-30111-7_73, Vol. 177, pp. 763-772, Chennai, 2012.
13. Mandal J.K, Ghosal, S.K, “A Novel DFT Based Information Embedding for Color Image Authentication (DFTIECIA)”, Third International Conference on Communication Security and Information Assurance (CSIA 2012), Advances in Computing and Information Technology Advances in Intelligent Systems and Computing, Springer Berlin Heidelberg, DOI: 10.1007/978-3-642-31552-7_78, Vol. 177, pp. 767-776, Delhi, 2012.

Other Publications

14. Mandal J.K, Ghosal, S.K, “A High Payload Steganographic Technique for Color Images (HPST)”, International Conference on Information’s System Design and Intelligent Applications, ISBN- 978-981-07-1158-0, pp. 65-70, Vizag, 2012.
15. Sudipta Kr Ghosal, “A New Pair Wise Bit Based Data Hiding Approach on 24 Bit Color Image using Steganographic Technique”, International Conference on Scientific Paradigm Shift in Information Technology & Management (SPSITM 2011) in collaboration with IEEE, Kolkata, 2011.
16. Ghosal Sudipta Kr, Mukherjee Saswati, “An Enhanced Secure and Comprehensive Data Hiding Approach Using 24 Bit Color Images”, International Conference on Advanced Computing & Communication (ICACC 2010), pp. 61-65, Kerala, 2010.
17. Mandal J.K, Ghosal, S.K, “Separable Discrete Hartley Transform based Embedding for Color Image Authentication (SDHTECIA)”, Second National Conference on Computing and Systems (NACCS-2012), ISBN-978-93-80813-18-9, pp. 178-183, Burdwan, 2012.
18. Sudipta Kr Ghosal, Saswati Mukherjee, Matangini Chattopadhyay, ”A Novel Approach: Data Hiding and Security of Multimedia Content using Steganography”, National Conference on Emerging Trends in Computer Science and Information Technology 2010 (ETCSIT 2010) ,pp. 133-137, Nashik, 2010.
19. Sudipta Kr Ghosal, Arnab Nath, Saswati Mukherjee, Arunashish Acharya, ” A New Data Hiding Application Tool on ASP.NET Framework using Steganography”, UGC Sponsored National Conference on Image Processing 2010 (NCIMP 2010), ISBN-978-81-8424-574-5, pp. 249-252, Gandhigram,2010.
20. Sudipta Kr Ghosal, Arnab Nath, Saswati Mukherjee, Arunashish Acharya,” A New Approach: Data Hiding and Security of Multimedia Content using Steganography”, UGC Sponsored National Conference on Image Processing 2010 (NCIMP 2010), ISBN-978-81-8424-574-5, pp. 253-256, Gandhigram, 2010.
21. Arnab Nath, Sudipta Kr Ghosal, Saswati Mukherjee, Arunashish Acharya,” An Efficient Approach of Watermarking to Protect Multimedia Content”, UGC Sponsored National Conference on Image Processing 2010 (NCIMP 2010), ISBN-978-81-8424-574-5, pp. 259-264, Gandhigram, 2010.

22. Ghosal Sudipta Kr, Dutta Chowdhury Souvik, “A DCT based Watermarking on Grayscale Images against Copy-right Infringement (DCTWGICI)”, JIS Educational initiative sponsored one day National seminar on Intellectual Property Rights and Patent Laws”, Kolkata, 2012.

List of Papers Presented

International Conferences

1. Ghosal, S.K, Mandal J.K, “Stirling Transform based Color Image Authentication (STCIA)”, Proceedings of International Conference on Computational Intelligence: Modeling, Techniques and Applications (CIMTA- 2013), Procedia Technology, ELSEVIER, ISSN: 2212-0173, Kalyani, 2013.
2. Mandal, J. K., Ghosal S. K., “Legendre Transformation based Color Image Authentication(LTCIA)”, Proceedings of Computer Science & Information Technology (CS & IT), DOI : 10.5121/csit.2013.3630, ISSN : 2231 - 5403, AIRCC, Feb.18-20, 2013, pp. 265–272, Bangalore, 2013.
3. Mandal, J. K., Ghosal S. K., “A Fragile Watermarking based on Binomial Transform in Color Images”,SIPP- 2013, Proceedings of Computer Science & Information Technology (CS & IT), DOI : 10.5121/csit.2013.3632, ISSN : 2231 - 5403, AIRCC, Feb.18-20, 2013, pp. 281–288, Bangalore, 2013
4. Mandal J.K, Ghosal, S.K, “A Novel DFT Based Information Embedding for Color Image Authentication (DFTIECIA)”, Third International Conference on Communication Security and Information Assurance (CSIA 2012), Advances in Computing and Information Technology Advances in Intelligent Systems and Computing, 2013, Volume 177, 767-776, DOI: 10.1007/978-3-642-31552-7_78, Delhi, 2012.
5. Mandal J.K, Ghosal, S.K, “A High Payload Steganographic Technique for Color Images (HPST)”, International Conference on Information’s System Design and Intelligent Applications 2012, ISBN- 978-981-07-1158-0, Vizag, 2012.
6. Sudipta Kr Ghosal, “A New Pair Wise Bit Based Data Hiding Approach on 24 Bit Color Image using Steganographic Technique”, International Conference on Scientific Paradigm Shift in Information Technology & Management (SPSITM 2011) in collaboration with IEEE, Kolkata, 2011.

National Conferences

9. Mandal J.K, Ghosal, S.K, ”Separable Discrete Hartley Transform based Embedding for Color Image Authentication (SDHTECIA)”, Second National Conference on Computing and Systems (NACCS-2012), page no. 178-183,ISBN-978-93-80813-18-9, Burdwan, 2012.

10. Sudipta Kr Ghosal, Saswati Mukherjee, Matangini Chattopadhyay, ”A Novel Approach: Data Hiding and Security of Multimedia Content using Steganography”, National Conference on Emerging Trends in Computer Science and Information Technology 2010 (ETCSIT 2010), pp. 133-137, Nashik, 2010.

Publication Indexing Database

The lists of publications relevant to the thesis work are indexed / abstracted in the following databases which are mentioned in the following table serially.

Publication Serial No.	*Database*
1	SCIE, SCOPUS, DBLP, Web of Knowledge (Thomson Reuters), Web of Science (Thomson Reuters), COMPENDEX (Elsevier), Computer & Information Systems Abstracts (ProQuest),CSA Technology Research Database (ProQuest), Current Contents: Engineering, Computing & Technology (Thomson Reuters), INSPEC (IET), PASCAL Database (INIST/CNRS) etc.
2	SCOPUS, DBLP, Engineering Index, INFONA, Zentralblatt Math, Elsevier B. V., Google Scholar etc.
3	SCOPUS, EBSCO, IEEE-INSPECT, Elsevier B. V., Institute for Scientific Information etc.
4	EBSCO, Scribd, CiteSeer, Pubget, Google Scholar, DOAJ, ProQuest etc.
5	EBSCO, Scribd, CiteSeer, Pubget, Google Scholar, DOAJ, ProQuest etc.
6	EBSCO, Scribd, CiteSeer, Pubget, Google Scholar, DOAJ, ProQuest etc.
7	EBSCO, Scribd, CiteSeer, Pubget, Google Scholar, DOAJ, ProQuest etc.
8	Thomson Reuters Book Citation Index, DBLP Computer Science Bibliography, ERIC - Education Resources Information Center, and ACM Digital Library, and CrossReflinking network
9	SCOPUS, DBLP, INFONA, Elsevier B. V., Google Scholar etc.
10	Open J-Gate, ArXiv.org, EBSCO, Scribd, CiteSeer, Google Scholar, DOAJ etc.
11	Open J-Gate, ArXiv.org, EBSCO, Scribd, CiteSeer, Google Scholar, DOAJ etc.
12	SCOPUS, ISI Proceedings, EI-Compendex, DBLP, Google Scholar and Springerlink etc.
13	SCOPUS, ISI Proceedings, DBLP. Ulrich's, EI-Compendex, Zentralblatt Math, MetaPress, Springerlink etc.

Abstract

Digital watermarking is the mechanism of concealing secret information into digital contents in an imperceptible way. Copy-right protection and the authenticity verification are two prominent applications of digital watermarking in this modern era. In general, the issues related to copy-right protection is managed by the robust watermarking however, the authenticity verification conforms the usage of a fragile or semi-fragile watermarking scheme. Any sort of alterations, even a single bit alteration, occurred to the watermarked image reveals the violation of fragile nature and hence, a cryptographic message digest is adopted to verify the authenticity. Digital watermarking in transform domain offers improved security and high robustness against common image processing attacks. In this study, Discrete Hartley Transform (DHT), Binomial Transform (BT), Legendre Transform (LT), Stirling Transform (ST) and group of linear transformations for dihedral group of order 4 (G-lets D4) based watermarking has been proposed to overcome the limitations of the existing techniques.

The schemes transform the cover image into the transform domain using the specified transformation. Each 2 x 2 or 1 x 2 sub-matrices of pixel components are pre-adjusted to avoid overflow and underflow, if needed. Watermark size, contents and the message digests obtained from the watermark are inserted into the transformed components for authentication. The authenticating data is embedded in varying proportions to achieve minimal quality degradation. If the pixel components are not initially adjusted a delicate re-adjustment on the embedded components may be incorporated to avoid overflow/underflow. A post-embedding quality adjustment has also been incorporated to keep the embedded components closest to the original without hampering the fabricated watermark bits. Few important parameters of watermarking include imperceptibility, payload, statistical undetectability and robustness to attacks. But some of the factors conflict with one another, such as, increasing payload might reduce the imperceptibility, etc. Thus watermarking is considered as an optimization problem. Genetic Algorithm (GA) has also been used as a tool to optimize the embedded transformed components. Inverse transform is done to produce the watermarked image in spatial domain. The recipient extracts the whole watermark through reverse procedure which in turn may be verified for authentication by ensuring the integrity of the re-generated message digest. So, any kind of intentional or unintentional attacks on the watermarked images are easily detectable.

Contents

Chapter 8: Genetic Algorithm based Optimization (GAO) ……………………….323-368

Chapter 1

Introduction

1.1. Introduction

Digital watermarking plays an important role in the field of information security. The secret message is fabricated into a carrier media in such a way that no one except the sender/recipient is aware of its existence. The basic characteristics of watermarking are payload, imperceptibility and robustness. The payload is nothing but the measure of the embedded information into the cover media. Imperceptibility concerns the visual inability to detect the secret information, and robustness refers to the ability to resist visual/geometrical attacks. However, there is a tradeoff among these characteristics. Section 1.2 and section 1.3 deals with the essence of watermarking and the scrambling of watermark using Arnold's cat map respectively that of the literature survey has been presented in section 1.4. Section 1.5 and section 1.6 deals with the problem domain and the objectives respectively. The methodologies have been discussed in section 1.7. Section 1.8 and section 1.9 describes the organization of the thesis and the metrics of evaluations respectively. Some salient features of the thesis are given in section 1.10.

1.2. Essence of Watermarking

The rapid advancement of communication system enables easier transmission of materials such as text, image, audio and video over the public network. The private information might be destroyed, tampered, copied or altered by the impostor in the unrestricted domain. In effect, important information such as secret message, corporate data, and private details are to be protected from any such illegal manipulation or malicious attacks. Therefore, security has becomes as much as important in today's world. Over the years, several security approaches have been adopted by the researchers among of which Cryptography and Information hiding are two prominent area of research. Cryptography is the process of protecting secret message by transforming it into a non-readable form. In contrast, information hiding is the art of fabricating secret information into the cover media by keeping the fidelity near equal to original one.

Two prominent areas of information hiding processes are steganography and watermarking respectively. Steganography reveals the mechanism of fabricating the secret messages into the carrier media in which the fabricated secret information is the object of communication. Digital watermarking also fabricate secret information into the digital media however, digital watermarking focuses mainly on the copy-right protection and authentication of digital content. The digital watermarking can further be classified into fragile, semi-fragile and robust categories. The major application of fragile watermarking is authentication and integrity verification of digital images, wherein the fabricated watermark is expected to be destroyed when the attacks are mounted on the host media. Cryptographic techniques such as message digest or digital signature can also be used for authentication. Semi-fragile watermarking techniques aim at detecting malicious alterations on an image, while allowing tolerable manipulations such as lossy-compression. A digital watermark is said to be robust, if it resists certain class of transformations. Robust watermarking may be used in copy-right protection applications to carry copy and no access control information. For a “fragile” image authentication, a single bit error in the hidden watermark leads to a totally different authenticator, however, for a “semi-fragile” image authentication the authenticator does not altered at all. Due to the high sensitivity against malicious attacks, “fragile” watermarking is one of the suggestive solutions to verify the authenticity however, “semi-fragile” watermarking based authentication reveal the sensitive nature to content modification and serious image quality distortion. The latter one is ideally independent on the logical content-based, non-variant relation among image pixels. Watermark can be inserted into a carrier media (also referred to as cover) in two different domains: Spatial and Transform.

Spatial-domain techniques are implemented easily but suffer from the lack of security and robustness against image processing attacks such as filtering, blurring, noise addition and compression etc. On the contrary, transform domain techniques offer improved security and high robustness against such attacks. However, proposed watermarking schemes are basically designed for authentication purpose and hence, the payload and image quality is considered as the primary issues.

1.3. Arnold’s cat map

Arnold transform is a kind of image scrambling methods which was proposed by Vladimir Arnold in the year 1960 [150]. The transform was first applied on the image of a cat and thus it is also known as Arnold’s cat map. It is a chaos based scheme which is widely used to enhance the security of an image by shuffling the pixel positions. The generalized Arnold’s cat map is a two-dimensional invertible chaotic map described by,

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod\ N \xrightarrow{yields} A \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod\ N \qquad (1.1)$$

The map possesses the property of area-preserving because the determinant of its linear transformation matrix is equal to 1. Here, a and b are positive integers, det(A) = 1 and the phase space is generalized to [0, 1, 2, . . ., N −1] × [0, 1, 2, . . ., N−1], i.e., only positive integers from 0 to N−1 has been considered.

The phase space is restricted to positive integers which conform the generalized cat map becomes periodic in nature. By applying the transformation T times, if the pixel at location (x, y) returns to its original position then T is considered as the period of the cat map. The period T is dependent on the parameters a, b and size N of the original image. By considering the parameters a = b = 1, the simplified form of equation (1.1) can be derived as given in equation (1.2).

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod\ N \qquad (1.2)$$

Instead of shuffling the positions of pixel components, Arnold’s cat map has been exploited on 1 x 2 block based watermark fabrication in Stirling Transform (ST) domain at section 5.2.2 of chapter 5 to transform the values of each pair of pixel components into the chaotic domain. Therefore, the equation (1.2) has been redefined and re-written as given in equation (1.3).

$$\begin{bmatrix} p_{n+1} \\ p'_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} p_n \\ p'_n \end{bmatrix} mod\ \Phi \qquad (1.3)$$

The scrambling is done by considering (p, p') as a pair of pixel components of the u x v secret watermark where, the parameter Φ (as computed 2^B) represents the decimal equivalent of B bits binary information of a pixel component. In general, it may be claimed that as the value of Φ increases, the period tends to increases. However, for a fixed value of Φ, the period (T) remains constant irrespective of changing dimensions of the watermark. Therefore, the variation of the parameter T depends on the variation of the parameter Φ which is again dependent on the parameter B.

Since the pixel components are represented using 8 bits (i.e., B = 8), Φ is considered as 2^8 or, 256. Mathematical analysis and program implementation ensures that the period (T) becomes 192 while subsequent values of B and Φ are 8 and 256 respectively. As a consequence, on execution of 192 iterations, the actual values of each pair of pixel components are re-generated. Therefore, each pair of pixel components is scrambled through any arbitrary number suppose, n iterations ($0 \leq n < 192$) whereas, the unscrambling is achieved by transforming the scrambled pair of components through (192 – n) iterations.

1.4. Literature Survey

In the field of information hiding, digital watermarking plays a vital role where, useful information is incorporated into various digital media like image, audio and video etc. to verify the ownership, authenticity and integrity respectively. Most of the researches in this field revealed the authentication mechanism through embedding secret information into the still images, both in spatial and transform domain.

A simplest approach for embedding secret information in a cover image was proposed by Bender et al. [1] in 1996 which is widely known as LSB technique. The cover image is responsible for concealing the secret data without creating any attention to the third-party observer. Each carrier image is consists of thousands of pixels where, each pixel is consists of one byte in case of gray-scale images and three bytes for color images, respectively. The embedding is nothing but the insertion of one bit of secret data at the least significant bit's position of each pixel where only the emphasized bits are modified. Subsequent embedding conforms the generation of watermarked image which is well perceptible and almost identical to the original image.

In 1998, a public key cryptography based image watermarking technique has been proposed by Wong et al. [2]. The technique used the private key to sign the watermark and embed it into the carrier image. The method introduced a watermark detection and localization property where, the place of modification is easily identified.

Nikoladis [3] projected a robust watermarking scheme in spatial domain where, the pixel components are randomly selected and then are modified accordingly. The pixels are modified by minimizing the energy content of the watermark signal at higher frequencies. The fabricated watermark is highly resilient to common visual and geometrical distortions such as low pass filtering, noise attacks and JPEG compression etc.

In 2001, Wang et al. [4] devised the optimal LSB substitution scheme by incorporating the Genetic Algorithm (GA) based optimization. The scheme introduced GA to solve the problem of hiding important data in the rightmost k-LSBs of the host image. The optimization takes a huge computational time but, ensured an improved quality of embedding results.

The dynamic programming strategy based embedding was a novel idea of obtaining optimal result. The technique utilizes a simple least-significant-bit (LSB) substitution which has been developed by Chang et al. [5] in the year 2003. The secret image is embedded in the least significant bits of the pixels in the cover image. In contrast to GA, the dynamic programming strategy ensured less computation time to generate the optimal result.

Wu et al. [6] proposed an efficient approach for embedding secret messages into a gray-valued cover image. The cover image is partitioned into a pair of pixels to embed the secret message. A difference value is calculated from the values of the two pixels in each block where, each difference value is belongs to a pre-defined range. Based on the human perceptibility of gray-valued pixel, the range intervals are obtained. As a consequence, a new value replaces the difference value to fabricate the secret information. The number of bits to be embedded is also decided based on the width of the selected range. The scheme also ensured that the pixel value never falls outside the range interval. This scheme offers better result over simple least-significant-bit replacement scheme in terms of imperceptibility.

Tri-way pixel-value differencing (TPVD) technique is based on three different directional edges to remove the capacity limitation of the original PVD method. In 2008, Chang et al. [7] proposed an optimal selection based approach for the reference point and adaptive rules to reduce distortion. Simulations results demonstrated that the scheme can provide an improved embedding capacity.

Wu et al. [8] proposed a novel watermarking scheme based on chaotic maps in spatial domain. Two chaotic maps have been used in this scheme out of which 1-D Logistic map is used to scramble the watermark whereas generalized 2-D Arnold cat map has been utilized to shuffle the embedding position of the host image. Experimental results demonstrate that the proposed scheme is robust against commonly used image processing attacks.

In 2010, Sathisha et al. [9] proposed an embedding technique based on 1-Bit Most Significant Bit (MSB) with chaotic behavior. Cover image is partitioned into 8 x 8 blocks however, the initial block is fabricated with 8 bits of upper bound and lower bound values. The mean of median values and difference between consecutive pixels is determined to fabricate three bits of Least Significant Bit (LSB) and one bit of MSB in chaotic manner. Simulation results ensured the effectiveness over existing schemes in terms of capacity, security and PSNR respectively.

Rawat et al. [10] devised a chaos based fragile watermarking for image authentication and tamper detection in the year 2011. In this scheme, any kind of alteration made to the image as well as the tampered location is easily detectable. The security of the technique is improved tremendously by incorporating two chaotic maps where, initial values of the chaotic maps are used as the secret key. This scheme is not only secure but also achieves superior tamper detection and localization accuracy has been achieved under different attacks such as copy-and-paste attack and collage attack.

In 2006, Zhang et al. [11] proposed a novel embedding method based on exploiting modification direction (EMD). In this scheme, each secret digit in a (2n+1)-ary notational system is carried by n cover pixels and, at most, only one pixel is increased or decreased by 1. In other words, (2n + 1) different ways of modification to the cover pixels correspond to (2n+1) possible values of a secret digit. In contrast to traditional LSB technique, the method ensured less degradation of quality in the embedded images.

The embedding capacity of the traditional EMD scheme [11] has been improved in Lee et al.'s [12] IEMD based embedding in the year 2007. In this scheme, binary messages are converted into a sequence of digits in an 8-ary notational system where, the least significant bits (LSBs) of a pair of cover pixels fabricate a secret digit. Simulation results ensured that the embedding bit-rate of this scheme is 1.5 times better than the Zhang et al.'s [11] embedding scheme.

In 2013, Kuo et al. [13] proposed a generalized exploiting modification direction (GEMD) method to obtain increased embedding capacity over EMD [11] and IEMD [12] methods. The GEMD scheme does not require the sender to transform the secret message into a special number form before embedding, so the method has improved capacity and reduced quality degradation than that of EMD [11] and IEMD [12] schemes, respectively.

In 2014, a hybrid GEMD (generalized exploiting modification direction) data hiding method has been proposed by Kuo et al. [14]. The technique offers up to 4bpp of embedding capacity along with retaining good fidelity stego-images. Moreover, it does not change to the variable codes that eliminated additional communication. The number of projection vectors used no longer excessive and no overflow/underflow problem exists. This method not only provides the improved embedding capacity but also maintain good stego-image quality.

In transform domain, most of the watermarking/steganographic techniques are based on Discrete Cosine Transformation (DCT), Discrete Wavelet Transformation (DWT), or discrete Fourier transform (DFT). These transformations convert the carrier image from spatial domain into transform domain in a sliding window manner. Each M x N block contains a set of transformed components which can be slightly modified to embed the watermark data.

In 1996, Bors et al. [15] devised a Gaussian network classifier based watermarking in which the pixel values of the selected blocks are modified by embedding a linear constraint among selected DCT coefficients. It also defines circular detection regions in the DCT domain. The technique offers resistant to JPEG compression.

Discrete Wavelet Transform (DWT) based robust watermarking has been devised by Xia et al. [16] in 1998. In this scheme, DWT is used to convert the image into transform domain where, pseudo-random codes are added to the large coefficients at the high and middle frequency bands of the DWT of an image. The scheme is robust against attacks such as the wavelet transform based image compression, image rescaling/stretching and image half-toning etc.

In 1999, a robust watermarking scheme has been proposed by Lee et al. [17] to protect the copyright of digital content. The watermark is embedded into the JPEG images based on variable quality level of JPEG which ensured the watermark extraction in JPEG compression domain.

Kang et al. [18] proposed a data embedding method using the Fresnel Transform which converts the watermark into a transformed pattern. The intensity of the pattern is fabricated into the cover image. However, the watermark such as the shape, photo or letter can be embedded into various observable planes with a choice of distance parameters of the Fresnel transform.

Discrete Cosine Transform (DCT) is an excellent choice for embedding watermark information into the transformed components. To enhance the robustness, Huang et al. [19] made a quantitative analysis on magnitude of DCT components and devised a novel watermarking scheme where, the watermark bits are embedded into the DC components to achieve high perceptual capacity.

In 1999, Kim et al. [20] presented a robust digital watermarking algorithm based on wavelet transformation. In order to maintain the uniformity, the coefficients of all sub-bands including LL sub-band are utilized for watermark embedding. A level-adaptive threshold scheme ensured the selection of perceptually significant coefficients for each sub-band. Based on the varying scale factors and the level of decomposition, watermark bits are fabricated into the selected coefficients. The vector projection method reveals the detection strategy. The technique is robust to various attacks, such as image compression, image filtering, geometric transformations and noises etc.

Premaratne et al. [21] proposed a novel watermark embedding and detection scheme based on Discrete Fourier Transform (DFT). The technique facilitates the detection of a watermark without any need for the original image. Thus, speedy watermarking is possible for still images as well as for video streaming.

Solachidis et al. [22] fabricated a rotation and scale invariant blind image watermarking scheme in discrete Fourier transform (DFT) domain. Watermark bits are embedded in magnitude of the DFT domain. The technique is robust to compression, filtering, cropping, translation and rotation. Experimental results demonstrate that the DFT based scheme offers high imperceptibility.

Pereira et al. [23] devised a digital watermarking scheme based on the Fast Fourier Transform (FFT). This method works by embedding watermark in template form which makes the technique robust against rotations and scaling, or aspect ratio changes. Based on a log-polar or log-log maps, the accurate template is recovered from a rotated and scaled image. Experimental results demonstrated the robustness of the method against some common image processing operations such as compression, rotation, scaling and aspect ratio changes.

In 2000, Wang et al. [24] proposed a novel watermarking based on the wavelet packet transform and spread spectrum. The host image is converted into transform domain based on wavelet packet transform followed by embedding the watermark bits into the significant coefficients in the wavelet packet domain. A pseudo-noise sequence is utilized to decide which sub-band of the wavelet packet transform is performed such that only the authorized people who have the pattern of the pseudo-noise sequence can retrieve the watermark data. The wavelet packet transform ensured high robustness against attacks such as compression, translation etc. where the feature spread spectrum introduces high security.

Zhicheng et al. [25] presented a digital image watermarking technique based on a fractal transform. The advantage of this technique is its ability to resist certain geometric attacks such as rotation, translation, flip, stretch, and zoom as well as normal signal processing operations such as JPEG compression.

Based on the human visual system (HVS) characteristics and the statistical behavior of the edge structures, a digital image watermarking technique has been devised by Sutharam et al. [26]. The watermark is generated using human visual system parameters, so that the watermark does not alter the perceived quality (transparency requirement) of a watermarked image and a sequence of random numbers generated from a bounded normal distribution so, that the technique is robust to malicious attacks. The watermarking technique inserts the watermark into the perceptually most significant Discrete Cosine Transform (DCT) coefficients, so that the technique is robust to image processing operations, robust to malicious watermark attacks and perceptually invisible.

Zhang et al.'s [27] robust watermarking scheme becomes popular due to the presence of the spatial-frequency feature of wavelet transformation. A completed tree is extracted from a coefficient array in the LL sub-band to construct a full-tree matrix (FTM), which represents both of the spatial localizability and multi-resolution decomposition of an overlapped local region in an original image. The local spatial luminance sensitivity, the frequency sensitivity and the texture activity are exploited to determine the existence of the watermark into the FTM. Simulations results demonstrated that the watermarking sequence is inserted in those regions that are of high texture activity with the maximum strength of JND tolerance by compromising the robustness and transparency.

In 2001, Fotopoulos et al. [28] introduced a novel watermarking scheme based on Gabor Transform. A casting scheme and the detection results are comprehensively inspected. Simulations results demonstrated that the scheme offers high embedding capacity and less quality degradation.

Fresnel transform can generate many embedding patters from a single watermark based on the varying distance parameters. Seto et al. [29] converts the watermark image into Fresnel-transform domain and then the values of difference between coefficients of a transformed pattern as well as their average are embedded into an original image. Image models, such as shapes, letters and photos, has been used as the watermark data. The experiment shows the validity of the watermarking technique and its robustness in resisting attacks.

According to Pereira et al. [30], watermarking is a linear programming problem in which one can maximize the strength of the watermark subject to a set of linear constraints on the pixel distortions as determined by a masking function. The special case has been considered for DCT domain and wavelet domain using the Haar wavelet and Daubechies 4-tap filter in conjunction with a masking function based on a non-stationary Gaussian model. This approach is applicable to any combination of transform and masking functions. Simulation results are computed, analyzed and compared to existing schemes to demonstrate that the scheme performs well against lossy compression such as JPEG and other types of filtering which do not change the geometry of the image.

In 2002, Yang et al. [31] presented a multi-resolution watermarking scheme, in which mixed chaotic sequences are embedded in the wavelet transform domain. The advantage of the multi-resolution feature of the wavelet transform and the non-relevant feature of chaotic signals has been used for watermarking purpose. The security is enhanced using a chaotic sequence generated by Chebyshev map which is mixed with the watermark bits followed by embedding in important coefficients located in the high-pass sub-bands. A group of adaptive thresholds based on each sub-band is selected and watermarks are embedded by modifying the coefficients whose amplitude is larger than these thresholds. Inverse wavelet transform yield the generation of the watermarked image. The experimental results demonstrated that the technique is robust against commonly used image processing attacks.

Ping et al. [32] introduced a public watermarking technique which is normalized with respect to affine transformation representation of the image based on the image moments. A CDMA scheme is used to embed a multi-bit watermark in the Discrete Cosine Transform (DCT) domain of the normalized image. Simulations results ensured that the technique is very robust to wide range of geometrical attacks.

In 2003, Ho et al. [33] devised a robust digital image watermarking scheme based on Fast Hadamard Transform (FHT) which offers a significant advantage in shorter processing time and ease of hardware implementation than other orthogonal transforms. The technique can embed or hide an entire image or pattern as a watermark such as a company's logo or trademark directly into the original image. A total of ninety attacks have been introduced into the watermarked image and the performance is evaluated based on Stirmark 3.1. These attacks were tested on 512×512 gray-scale images that can fabricate a gray-scale watermark image of size 64×64. Experimental results demonstrated that the algorithm is very robust and can survive most of the Stirmark attacks.

Ashourian et al. [34] suggested a watermarking scheme based on dithered quantization where, the original signal is used by the detector for fingerprinting. The analysis is done in the wavelet transform or equivalently in filter bank domain. The nonlinear effect of dithered quantization in the filter bank analysis using the time-domain formulation of the filter bank is involved for this purpose. A general and compact form for distortion can be observed in the host image due to the encoding and embedding process.

The composition of Discrete Wavelet Transform-Discrete Fourier Transform (DWT-DFT) makes the Kang et al.’s [35] image watermarking scheme as robust against both affine transformation and JPEG compression. The robustness has been improved by using a new embedding strategy, watermark structure, 2D interleaving, and synchronization technique, respectively. A spread-spectrum-based informative watermark with a training sequence is embedded in the coefficients of the LL sub-band in the DWT domain while a template is embedded in the middle frequency components in the DFT domain. In watermark extraction, the template is detected from a possibly corrupted watermarked image to obtain the parameters of an affine transform and convert the image back to its original shape. Then, translation registration is performed using the training sequence embedded in the DWT domain, and, finally, extract the informative watermark.

Content adaptive watermark embedding using a stochastic image model in the multi-wavelet transform domain is a novel concept presented by Kwon et al. [36]. Watermark bits are inserted into the cover image with a uniform embedding strength regardless of the local properties; as a consequence, the visible artifacts are taken placed at flat regions. A watermark is embedded into the perceptually significant coefficients (PSCs) of each sub-band using multi-wavelet transform. The PSCs in high frequency sub-band are selected by SSQ, that is, by setting the thresholds as the one half of the largest coefficient in each sub-band. A noise visibility function (NVF) based stochastic approach along with a perceptual model is applied for watermark fabrication. The stationary Generalized Gaussian model characteristic is used since the watermark has noise properties. The watermark estimation use shape parameter and variance of sub-band region, it is derive content adaptive criteria according to edge and texture, and flat region.

In 2004, Ho et al. [37] introduced a copyright protection scheme in Slant transform domain where, the watermark is embedded into the complex textured images such as the satellite images. The watermark is embedded especially into the mid-frequency bands of the Slant transform coefficients of relevant sub-blocks of the host image. Simulation results ensured that the robustness of the scheme is improved tremendously against various Stirmark and Checkmark attacks along with a good visual clarity in the embedded image.

Shiba et al. [38] devised a data embedding method where, the original image is cellular automata-transformed and the watermark information is fabricated into the coefficients of CA-transformed pattern. A number of watermark patterns can be embedded in varying CAT planes with dissimilar rule number parameters and CA bases class of CAT. Different image models such as shape, photo and letter can be used as the watermark data. Experimental results demonstrated that the scheme can provide multi frequency domains for embedding watermark.

A novel digital image-watermarking scheme based on Radon transform and 2-D Fourier transform has been proposed by Cai et al. [39]. An invariant centroid is used as the origin of Radon transform to fabricate the watermark into the selected circular area of the image in Radon transform domain, and then extracting 3-D Fourier magnitude of the Radon transformed image. The scheme is invariant to rotation, scale and translation (RST) attacks. The severe degradation in quality of the watermarked images are avoided by performing the inverse Radon transform on the watermark signal only, which in turn is added to the original image.

The blind watermarking scheme based on wavelet lifting transform and human visual system (HVS) is the novel concept of Liu et al. [40]. The pseudo-random sequence is responsible for the encryption of the watermark sequence. The high frequency coefficient of image multi-resolution enables the digital watermark to be embedded into the low frequency domain based on a reference template mapping of the low frequency coefficient of image multi-resolution. Experimental results demonstrated that the scheme is feasible for the copyright protection.

Safabakhsh et al. [41] presented a non-blind entropy based watermarking of still gray level images in the wavelet domain. The original image is decomposed in DWT domain into three hierarchical levels and embeds a logo image, which is scrambled through a well-known PN-sequence. The entropy-based scheme selects DWT coefficients in an adaptive way for determining the number of watermarked coefficients and watermarking factor at different level of DWT decomposition. Experimental results ensured that the watermarking scheme is resilient against most of the common attacks.

Jeon et al. [42] devised a new scheme for 3-D model watermarking in the DCT domain. The watermark embedding and extraction is achieved by traversing the 3-D mesh model to generate triangle strips. The vertex coordinates of the triangle strips are then converted from the spatial domain into the transform domain through DCT. To accomplish high robustness and imperceptibility, the mid-frequency band of AC coefficients are chosen for embedding. The scheme is resilient to additive random noise, affine transformation, geometry compression by the MPEG-4 SNHC standards and multiple watermarking.

The application of Fourier-Mellin transform in watermarking makes the scheme robust to RST attacks. Kim et al. [43] proposed a novel watermarking scheme which reorders and modifies function blocks of FMW for improvement of realization and performance. In contrast to FMW, the method uses Log-Polar Map (LPM) in the spatial domain for scaling invariance, while translation invariance is provided by the use of an invariant centroid as the origin of LPM. An invariant centroid is used as the origin of a central area on gray-scale image which is invariant to RST attacks. However, the rotation of Cartesian coordinates system into a cyclic shift is a significant property of LPM, the 2D-DFT is applied on the LPM image and the magnitude spectrum retrieved to provide a domain that is rotation invariant. Experimental results demonstrated that the scheme is robust to RST attacks.

Jin et al. [44] proposed a robust watermarking technique for 3D mesh. The technique is based on spherical wavelet transform. Spherical wavelet transform decomposes the original mesh into a set of levels of details through varying scale factors that embeds the watermark data. The embedding steps includes: global sphere parameterization, spherical uniform sampling, spherical wavelet forward transform, embedding of watermark, spherical wavelet inverse transform, and finally the re-sampling of the mesh watermarked to recover the topological connectivity of the original model. Simulation results demonstrated that the capacity of the watermark and the robustness of watermarking against attacks are significantly improved.

Watermarking based on genetic algorithms (GA) in the transform domain is a novel idea devised by Shieh et al. [45]. The effectiveness of this scheme is evaluated by checking the fitness function in GA, which includes both factors related to robustness and invisibility. Simulation results demonstrated the robustness under different kinds of attacks, and the improvement in watermarked image quality with GA.

Ahmed et al. [46] introduced an image watermarking technique using the naturalness preserving transform (NPT) in frequency domain. The method inserts and extracts the watermark via a special orthogonal transform class in between the spatial and frequency representations of the host image data. To improve the quality of the watermarked image, two different forms of NPT such as the Hartley and Discrete Cosine Transforms have been utilized. However, the quality enhancement is achieved via reduction to the artifacts which occur to transformed images via the classical Hadamard-based NPT. Simulation results ensured robustness against several image attacks viz. JPEG compression, low pass filtering, cropping of images, and noise effects.

In 2005, Areef et al. [47] presented DWT based watermarking scheme that employs the genetic algorithm (GA) in transform domain. It is observed that embedding the watermark into the higher frequency coefficients is not robust, although the watermarked image quality is assured. Based on genetic algorithm (GA), the optimal frequency bands for watermark embedding into a DWT-based watermarking system are found. The technique is improved in terms of security, robustness, and image quality of the watermarked image.

Feng et al. [48] suggested a digital watermarking scheme based on fractional Fourier transform (FRFT) and the energy distribution of two-dimensional signal at different FRFT domain. Multiple chirps are treated as the watermark and the same is embedded in the spatial domain however, the detection is done in the FRFT domain. Simulation results demonstrate that the watermarking technique have good image quality and are robust to some common image processing attacks.

Ho et al. [49] formulated a novel semi-fragile watermarking scheme for the content authentication of satellite images using the pinned sine transform (PST). In the PST domain, the image field is decomposed into two mutually orthogonal sub-fields such as the boundary field and the pinned field respectively. The fabrication of watermark is done into the pinned field of PST that preserves the texture data of the original image. The PST based scheme offers special sensitivity to any texture modification to the watermarked image. The texture characters are really important semantic understanding to authenticate the satellite images. The localization property can identify the tampered region with high accuracy and recover it for a high degree of image integrity. The inter-block relationship of PST renders the watermarking scheme resilient to content cutting-and-pasting attacks. Experimental results demonstrated that authenticating system provides the probability of tamper detection of higher than 98%.

The watermark insertion into the phase components of a transformed image by Dual-Tree Complex Wavelet Transform (DT-CWT) is a novel concept of Lee et al. [50]. The scheme is robust for affine transformation and time-variable according to the degree of distortion. The extraction is done by incrementally comparing the extracted data with the original watermark based on correlation from lowest to highest level. The performance evaluation ensured the higher robustness against geometric distortions.

Wu et al. [51] presented a semi-fragile watermarking scheme for digital content authentication where, the watermark is obtained from the Zernike moments of the low frequency sub-band in IWT (Integer Wavelet Transform) domain of the original image. Watermark is embedded in the IWT domain of an image by exploiting the features of human visual system (HVS) which produces high-fidelity and high-quality watermarked images. The difference of Zernike moments before and after attack ensured mild distortion. Simulations results demonstrated that the scheme can locate the tampered area accurately while is robust to content-preserved manipulation, such as JPEG lossy compression.

In 2006, Yang et al. [52] proposed a novel watermarking method in which the high frequency sub-bands of discrete non-separable wavelet transform (DNWT) can reveal more features than that of the common used separable wavelet transform. Unlike discrete separable wavelet transform (DSWT), the discrete non-separable wavelet transform (DNWT) can fabricate more watermark bits into the transformed coefficients. Simulations results ensured that the DNWT watermarking scheme is robust to noising, JPEG compression, and cropping.

Tsui et al. [53] proposed a color image watermarking scheme which encodes the chromatic content of a color image as CIE a*b* chromaticity coordinates and the achromatic content is encoded as CIE L tri-stimulus value, respectively. The Spatio-Chromatic Discrete Fourier Transform (SCDFT) enables the fabrication of color watermark into the chromatic channel in frequency domain. It first encodes a* and b* as complex values, followed by a single discrete Fourier transform. The casting of watermark is performed by measuring the just-noticeable distortion (JND) of the images to ensure the imperceptibility of the hidden watermark.

Yu et al. [54] proposed a novel digital watermarking and detection technique, which uses the chirp signal as a watermark and embeds it in the Fractional Fourier Transform (FRFT) domain. The position of the watermark as well as the order of transform is used as the encryption keys. The impulse characteristic in the FRFT domain for chirp signal ensured the ability of watermark detection. The scheme is of good imperceptibility, security and is very robust to JPEG compression, noise attacks, cropping and the filtering.

Wang et al. [55] embedded the digital watermark into the significant coefficients of wavelet domain which can be identified by the human visual system (HVS) characteristics. The watermark is retrieved back through the trained neural networks. The learning and adaptive capabilities of neural networks ensured the exact watermark retrieval from the trained neural networks of the watermarked image. Simulation results demonstrated that the scheme is effective in terms of reduced degradation and robustness against common attacks.

Yin et al. [56] utilized a hyper-chaotic sequence for watermarking in Discrete Wavelet Transform (DWT) domain. The host image is scaled with respect to a scale factor to construct cosets for inserting the digital watermark. A hyper-chaotic sequence has been used to overcome the problem of small key space and improves the watermark security, respectively. Experimental results demonstrated that the scheme offers a high robustness and imperceptibility.

Gui et al. [57] proposed a novel asymmetric watermarking scheme based on a secret real fractional DCT-I transform. A primitive watermark is employed to generate an asymmetric watermark. The secret watermark for embedding is derived from the primitive watermark, and is embedded in the large fractional DCT-I transformation coefficients of the cover image. The correlation test has been used for asymmetric detection. Experimental results ensured that the asymmetric detection is reliable.

In 2007, Agarwal et al. [58] embeds the watermark in Discrete Hartley transform (DHT) domain or in Discrete Cosine Transform (DCT) domain based on the number of edges in the blocks of the original image. The threshold number of edges in the original image has been used as the key to perform the block by block embedding operation. The reverse approach is followed to extract the watermark. This scheme is robust to visual attacks such as addition of Salt & Pepper noise, JPEG compression etc.

Hien et al. [59] utilized curvelet transform for watermarking as it represent edges along curves much more efficiently than the traditional transforms. The watermark is embedded in curvelet coefficients which are selected by a criterion whether they contain as much edge information as possible. Experimental results demonstrated that the scheme is robust to most prominent attacks.

Circular Hough Transform possesses circular features that are invariant to geometric distortions. Lee et al. [60] used these circular features to synchronize the location for watermark insertion and detection in an additive way on the spatial domain to devise a robust watermarking scheme. Simulations results demonstrated that the method is useful and considerably robust against both geometrical and signal processing attacks as listed in Stirmark 3.1.

Guo et al. [61] proposed a novel watermarking algorithm based on Discrete Fractional Random Transform. The imperceptibility of the watermarking is improved by adjusting the intensity of phase shift keying whereas, the random block selection and high amplitude selection techniques ensured higher robustness against the attacks of cropping, noising and low-pass filtering.

In 2008, Kumaran et al. [62] presented a Contourlet Transform based watermarking where, Contourlet Transform is a two-dimensional extension of the wavelet transforms using multi-scale and directional filter banks. The genetic algorithm based optimization has been applied to the Contourlet transform to improvise the quality of the watermarked images. The genetic algorithm based embedding schemes namely surrounding mean and zero-tree embedding approach has been employed. Experimental results demonstrated that the scheme is blind which makes the watermark robust to several attacks.

A novel digital watermarking scheme based on Radon transform with 2D-wavelet transform has been devised by Li et al. [63]. Radon transform converts the point singularity into the line singularity; however, the 2D-wavelet transform is effective to deal with the point singularity. Experimental results demonstrated that the scheme has good performance in invisibility and robustness, especially in brightness enhancement and weakening.

Xian [64] presented a semi-fragile digital watermarking scheme. The self-adaptive secret information is scrambled through Arnold Transform and then embedded into the host image based on calculating the adjacent wavelet coefficients by exploiting the characteristics of HVS (Human Visual System). The various attacks are applied to the watermarked images which ensured robustness against conventional compression attacks such as JPEG and hostility attacks and falsification.

Korohoda et al. [65] devised a discrete trigonometric transforms (DTTs) based image watermarking where, the main study is based on the correlation coefficient. Performance analysis deals with the distorted watermarked images due to the Gaussian and JPEG attacks however, the results indicate that the DTTs are equally effective as the discrete cosine transform II-even (DCT II e) and might be considered as a set of alternative solutions.

Ozturk et al. [66] hides the secret watermark into the spatio-frequency (SF) domain. The discrete evolutionary transform is calculated by the Gabor expansion to represent an image in the SF domain. The watermark is fabricated into the selected cells in the joint SF domain. The mixed features of spatial and spectral domain make the watermarking scheme robust.

Tsai et al. [67] utilized principal component analysis and discrete wavelet transform as a solution of robust image watermarking. A scale invariant feature transform ensured the direction of the feature points of an image through which the orientation histogram is obtained. The highest peak of the orientation histogram is a start index to decompose the image into bins covering 360 degree range of orientation. By applying principal component analysis, the principal components of image coefficients within each bin are obtained. Watermark bits are fabricated into the wavelet coefficients of these components in quantization steps. Simulation results demonstrated that the scheme can tolerate high robustness against median filtering, Gaussian filtering, JPEG compression, sharpening, and rotation attacks.

Falkowski [68] devised a robust phase watermarking scheme where, the multi-polarity Walsh-Hadamard and Complex Hadamard transform in the phase spectrum is used to convey the useful watermark. The robustness has been tested against certain visual as well as geometrical attacks such as JPEG encoding, image resizing, dithering noise distortions, sharpening and cropping.

In 2009, Rawat et al. [69] presented a watermarking scheme in which the pre-processed host image is decomposed through wavelet packet transform (WPT) to obtain the best tree by entropy based algorithm. All frequency bands of the entropy based tree are used to hide the watermark. Simulation results demonstrated that the WPT based scheme is robust against variety of attacks.

The morphological wavelet transform, introduced by Zhuang et al. [70] is useful for digital watermarking due to important reasons which are to notify the copyright owner with a visible gray image watermark and to protect the copyright with an invisible binary image watermark. Different approaches have been utilized to embed both the watermarks in different blocks. Simulations results demonstrated that the morphological wavelet has state-of-the-art performance to make the visible watermark hard to remove and the invisible watermark robust.

Jianzhong et al. [71] proposed a new adaptive blind watermarking scheme with the watermarks encrypted by Fresnel transform. The watermark is adaptively embedded based on the classified blocks of fuzzy c-means clustering (FCM) technique and human visual system (HVS). The scheme modifies the pixel-values with reduced distortion and higher robustness against different kinds of attacks.

Kumar et al. [72] embeds an encrypted watermark based on fractional Fourier transform where, the watermark is encrypted using double random fractional order Fourier domain encoding scheme. The fractional orders and random phase masks yields the recovery of the fabricated watermark from the watermarked image in fractional Fourier transform domain. The security is enhanced by enlarging the size of the considered key in fractional Fourier transform domain.

The robust method of non-blind image watermarking in which the modification is made on singular value decomposition (SVD) of images in Complex Wavelet Transform (CWT) domain is first formulated by Mansouri et al. [73]. The CWT offers higher capacity than the real wavelet domain and the alteration of the appropriate sub-bands leads to a good quality watermarking scheme. Simulation results ensured that the scheme is highly robust against the usual attacks and shows better performance over the pure SVD-based as well as hybrid methods (e.g. DWT-SVD scheme).

Wang et al. [74] presented a watermarking scheme based on discrete cosine transform (DCT) which transforms the host image from spatial domain into frequency domain. The user specified key selects a number of DCT bands which in turn are modified to fabricate the watermark bits. However, the scheme can be improvised by introducing a genetic band selection (GBS) based training procedure which uses genetic algorithm (GA) to identify the suitable DCT bands for embedding. The trained result is then used in the mentioned DCT-based watermarking to get better performance.

Cintra et al. [75] hides secret watermark into a host image where, the host image is transformed through the block-wise application of two-dimensional finite field cosine or Hartley transforms. The finite field transforms are suitable for error-free calculation as it is adjusted to be number theoretic transforms. The finite field characteristics ensured tamper location capability and good imperceptibility. The watermarked image is treated as the signature of authenticity in public domain.

Ouhsain et al. [76] suggested a watermarking scheme in which the host image is decomposed into four wavelet sub-bands. A non-negative matrix factorization (NMF) is applied to the blocks of each sub-band followed by an Eigen decomposition distortion. Simulations results demonstrated the imperceptibility and resiliency of the watermarking scheme against intentional and geometric attacks.

Bohra et al. [77] devised a self-authentication scheme based on second-generation wavelets (lifting-based integer wavelet transforms). The integer-to-integer transform ensures that the watermarked image obtains reduced perpetual degradation and higher robustness against attacks such as filtering, compression and rotation.

Discrete Wavelet Transform (DWT) is useful in watermarking for the properties such as its spatial localization, frequency spread and multi-resolution analysis. In 2010, Kang et al. [78] introduced a scheme in which the visual watermark (grayscale image) is transformed through Discrete Cosine Transform (DCT) for getting low frequency coefficients in frequency domain. The majority of the visual information is contained in low frequency coefficients of the watermarked image in transform domain. Discrete Wavelet Transform (DWT) converts the original image from spatial domain into transform domain to fabricate the secret bits into the coefficients of LL bands by maintaining the imperceptibility and robustness. The embedding and extraction locations of the multi-energy watermarking technique are identified based on the qualified significant wavelet tree (QSWT). The adaptive casting energy in different resolutions available from the wavelet transform of QSWT provides higher robustness against sharpening, smoothening, image cropping and JPEG compression.

Peng et al. [79] performed single-level multi-wavelet decomposition on each image block of an image where, a mean value modulation method has been used to modulates mean value relationship of multi-wavelet coefficients in two approximation sub-bands for carrying watermark information. Support Vector Machines (SVMs) has been used to learn the mean value relationship and extract the watermark under different kinds of attacks. Simulations results ensured that the method offered a high imperceptibility and is robust to common image processing attacks such as JPEG, low-pass filtering, noise addition, rotation and scaling, etc.

Xi’an et al. [80] projected a digital watermarking technique in which the Logistic or Chebyshev chaos model has been used to determine the order of embedding. The watermark is scrambled through Arnold transform and the same is embedded into carrier image in wavelet transform domain. The scheme has a good visual clarity and robustness against common attacks.

Huawei et al. [81] presented a novel multi-bit image watermarking scheme in Discrete Cosine Transform (DCT) domain which is resilient to rotation and scaling attacks. The scaling-invariant spread spectrum-based watermarking scheme is used to embed and extract the watermark message in the DCT domain where, the Radon transform is used to ensured the correct orientation of the image. Simulation results revealed that the scheme possesses good robustness against rotation, scaling attacks and considerable robustness against typical image processing.

Wang et al. [82] utilized a deformable multi-scale transform (DMST) in watermarking that possesses the properties such as joint shift-ability in position, orientation, and scale. The performance of watermark extraction can be enhanced by extending the hidden Markov model in the standard wavelet domain to the steerable wavelet domain. Experimental results demonstrated that the watermarking scheme is quite robust to the common signal processing and geometrical attacks.

Sujatha et al. [83] introduced a watermarking scheme in which each 2x2 block of the host image is converted into transform domain through Discrete Wavelet Transform (DWT). Arnold transform has been applied thrice into the secret watermark to enhance the security. The scheme has reduced distortion and good robustness against different image processing attacks.

Huawei et al. [84] utilized scale-invariant feature transform (SIFT) for watermarking as it is invariant to rotation, scaling, translation, partial affine distortion and addition of noise. SIFT features are clustered through k-means clustering and the watermark bits are fabricated in each cluster. In presence of many feature points, only one watermark bit is embedded in each cluster. Performance analysis ensured that the scheme offers robustness against attacks such as scaling, rotation, cropping, affine transforms, JPEG compression, image filtering etc.

The image watermarking scheme based on the phase retrieval algorithm in gyrator domain is designed by Liu et al. [85]. Arnold transform has been used to obtain a noise like image which is regarded as the amplitude of gyrator spectrum. The Gerchberg–Saxton algorithm is used to compute the unknown phase function in gyrator pair where, the host image is treated as the amplitude of input function. The scheme derives the key from the phase information and the parameters of the two transforms.

Singh et al. [86] introduced a novel watermarking scheme in which the gyrator transforms and two chaotic random phase masks are used to encode the input image. Four chaotic maps such as the logistic map, the tent map, the Kaplan–Yorke map and the Ikeda map are used to generate the chaotic random phase masks. Two chaotic maps with different seed values have been used to generate the single chaotic random phase mask. Simulations results demonstrated that the scheme is robust and the watermarked images retain a good visual clarity.

Lin et al. [87] projected a Discrete Cosine Transform (DCT) based embedding scheme in which six sub-areas of transformed coefficients located in the middle frequency are chosen for embedding the secret bits. The secret information is transformed through a notational system. This scheme offered an enhanced capacity and reduced distortion as compared to the existing DCT based schemes.

Varsaki et al. [88] proposed a data hiding scheme based on Discrete Pascal Transform (DPT). Secret information is embedded into each even-sized block by monitoring the lower-right corner of the DPT coefficient matrix. This particular coefficient suffers the highest change for small pixel modifications and the embedding affects the coefficient's sign. If the sign is negative for a message bit value of '0' and positive for a message bit value of '1', it is changed by repeatedly adding to the block or subtracting from the block. The scheme is evaluated in terms of capacity and image distortion which ensured the superiority over existing schemes.

In 2011, Yan and Yang [89] suggested a robust digital image watermarking scheme based on singular value decomposition (SVD) and discrete wavelet transform (DWT). The method is based on modifying the singular values of small blocks of the low-frequency approximation sub-band (LL) of the DWT domain to embed watermark into host image. Experimental results demonstrated the robustness against common image-processing operations.

Li et al. [90] devised a novel digital watermarking scheme in which the carrier image is disassembled by Cellular Automata Transform (CAT) based on the gateway values. The CAT coefficients ckl fall into four distinct groups of which the group I is further transformed by level-1 CAT which is also known as the "low-low frequency". Watermark fabrication is done into the group (low-low Frequency) of the CAT coefficients ckl. Simulations results revealed that the method is more robust against different watermarking attacks.

Manoochehri et al. [91] suggested a watermarking scheme based on the combination of Discrete Wavelet Transform (DWT) and Fourier-Mellin Transform (FMT). The horizontal and vertical sub-bands are selected from the DWT of watermark and original images which are then transformed through FMT for embedding. The inverse transformations i.e., IFMT and IDWT ensured the generation of the watermarked image. The DWT and FMT have been combined to ensure the high robustness against the noise attacks and the geometrical transformations, respectively.

The robust watermarking scheme based on discrete fractional random transform and a generalization of the discrete fractional Fourier transform with intrinsic randomness is the novel concept formulated by Lee et al. [92]. The randomness ensured high security for embedded information into the watermarked image. The scheme is robust against most of the frequency and geometric attacks.

Kumar et al. [93] projected a color image watermarking scheme based on combination of discrete cosine transform, contourlet transform and singular value decomposition. The color image is transformed into two basic components: luminance (Y) and chrominance (Cb and Cr). Discrete cosine transform is applied on the components to obtain DCT coefficients which are decomposed into directional sub bands using 2 level contourlet transform. Contourlet transform decomposed a gray-scale watermark into different sub-bands. Laplacian Pyramid decomposition is applied on both, fourth directional sub-band from watermark and cover image to obtain Low pass sub-band. The singular values of low-pass sub-band of watermark are embedded into singular values of low-pass sub-band of cover image. Simulations results demonstrated that the watermarking scheme is robust against common image processing attacks.

Cancellaro et al. [94] utilized the commutative property to cipher a watermarked image without affecting the concealed data or to watermark a scrambled image still considering a perfect deciphering on Tree Structured Haar transform domain. The key dependency of the mentioned transform enhances the security of the scheme. For extraction purpose, the key has to be re-generated from the ciphered watermarked image. Experimental results ensured the effectiveness of this scheme.

Saeed et al. [95] introduced a hybrid robust digital watermarking method based on finite Radon transform (FRAT) and Singular Value Decomposition (SVD). Higher values of normalized correlation (NC) can be obtained by altering the singular values. Tradeoff between robustness and transparency of digital watermarking is maintained by using middle frequencies of HL3 and LH3 for embedding purpose. Experimental results demonstrated the robustness of this scheme against several attacks such as: noise addition, filtering, histogram equalization, gamma correction, JPEG compression, scaling and rotation.

Hao et al. [96] projected a blind watermarking scheme based on discrete fractional random transform. The sequence of bits, the gray-scale image or set of decimal fraction obtained from a source signal are treated as the watermark. The host image is sub-sampled into four sub-images, and the high correlations among their discrete fractional random transform coefficients are exploited for watermark embedding. Experimental results demonstrated that the scheme offers high payload, enhanced security and reduced quality degradation.

Maity and Kundu [97] proposed a digital image watermarking scheme using the characteristics of the human visual system (HVS), spread transform technique and statistical information measure. Both the cover and watermark images utilized the transformed coefficient in Spread transform (ST) domain. The frequency sensitivity, contrast, luminance and entropy masking of HVS model adjusts the embedding strength. The choice of Hadamard transform as watermark embedding domain offers reduced quality degradation, greater reliability of watermark detection and a large payload at high degree of compression.

In 2012, Hao et al. [98] presented a watermarking scheme based on chaos and fast curvelet transform. The watermark is scrambled through the Arnold transform and the same is fabricated into the transformed components. The embedding order is chosen based on a chebyshev chaotic sequence. The inverse operation is applied to extract the fabricated watermark.

Elshazly et al. [99] proposed an efficient approach for digital image watermarking based on the Fractional Fourier Transform (FRFT). The watermark is embedded into the transformed components in the form of a PN sequence. The Discrete Wavelet Transform - Fractional Fourier Transform (DWT-FRFT) based watermarking gives high robustness and improved security.

Surekha and Swamy [100] proposed a combined watermarking and secret sharing scheme in frequency domain. The XOR-based Visual Secret Sharing (XVSS) scheme splits a watermark into public and private watermark, respectively. The private watermark is extracted from the features of host image and original watermark, whereas, the public watermark is extracted from the controversial image at any time without the need for original host image and/or watermark. The security of the scheme is improved by processing LL sub-band coefficients of Discrete Wavelet Transform (DWT) of the host image to satisfy central limit theorem. Experimental results revealed that the scheme can resist to a variety of common image processing manipulations.

Zhao et al. [101] introduced a novel digital watermarking scheme featuring centroid-based sectoring where, a delicate synchronization mechanism has been incorporated to get higher robustness resilient to rotation, scaling, and translation (RST) attacks. Synchronization information as well as the message bits is then embedded into the centroid-based sectors. The approach is capable of restoring the correct sectoring even if it has experienced severe geometric distortion. Simulation results demonstrated that the scheme possesses good robustness against RST attacks and considerable robustness against other common image processing attacks.

The security of 3D geometry watermarking using the independent component analysis (ICA) de-mixing matrix for transformation is improved by Wu et al. [102]. The inclusion of orthogonal transformation (OT) can generate another watermarking scheme. The same embedding rule is applied over two watermarking schemes on the test models. Experimental results ensured that they have similar performance in imperceptibility and capacity.

Rawat and Balasubramanian [103] presented a robust watermarking method based on Fractional Fourier Transform (FRFT) and visual cryptography (VC). The visual secret sharing scheme has been used to construct a master share and an ownership share respectively. SVD is used to extract the features of the original image from which the master share to be generated; however, the ownership share is generated from the watermark and the master share respectively. The ownership identification is possible by stacking the master share and the ownership share. The security and robustness is achieved by using the following properties: VC, FRFT and SVD.

Huand and Zhao [104] proposed an adaptive digital image watermarking based on Morphological Haar Wavelet Transform which is a non-linear wavelet transform. The multi-scale morphological wavelet transform is used to decompose the original image and the watermark, however, the watermark is adaptively fabricated into the original image in varying resolutions. Experimental results demonstrated that the technique offers higher robustness than the ordinary wavelet transform based techniques.

Martino et al. [105] projected a fuzzy transform based fragile watermarking process where, the pre-processing phase is considered to determine the best compression rate for the coding process. The scheme is tested for tamper detection on a sample of images by simulating various types of computer attack. Performance analysis reveals the high accuracy of tamper detection with compressed images.

According to Li et al. [106], Polar Harmonic Transform (PHT) is defined on a circular domain where, the magnitudes of PHTs are invariant to image rotation and scaling. The PHTs are free of numerical instability and also possesses invariant properties which make it an ideal choice for watermarking. The embedding is done by modifying a subset of the accurate PHTs based on the binary watermark sequence. The reconstruction of the modified PHT vector yields the formatting of compensation image however, by combining the compensation image to the original image, the watermarked image is produced. Watermark bits are extracted from the magnitudes of the PHTs. Experimental results illustrated that the scheme outperforms ZM/PZM based schemes in terms of embedding capacity and watermark robustness.

Hamidreza et al. [107] presented a robust method of spread spectrum based image watermarking where, the carrier image is partitioned into non-overlapping blocks and ridgelet transform is applied to each single block. Thus, a curved edge is divided into some straight edges and shows optimal performance even for complicated images with curve edges. The variance intensity identifies the best directions of embedding into the ridgelet coefficients. An efficient detection method is used for detecting watermark logo blindly from distorted watermarked image. The robustness is enhanced by finding the best place to insert the watermark which is followed by scrambling the watermark bits based on a pseudo random sequence with an authentication key.

A key-dependent, content based and non-deterministic neighborhood-dependent fragile watermarking scheme in wavelet domain has been introduced by Betancourth [108]. The scheme is sensitive to all kinds of modifications and has the ability to localize the tampered regions. Moreover, few familiar attacks such as the transplantation attack and vector quantization attack can be resisted. The most remarkable feature of the scheme is that only a small number of wavelet coefficients are watermarked but all are implicitly protected. Consequently, the embedding distortion is dramatically reduced without compromising the authentication resolution.

Yang et al. [109] projected an integer wavelet transform based reversible data hiding scheme in which the secret information is embedded into the low-high (LH) and high-low (HL) sub-bands of the IWT domain. The embedding into each IWT block is nothing but the adjustment of the transformed coefficients which ensured reduced quality degradation. The scheme also offered certain degree of robustness against attacks and supports reversibility.

In 2013, Tsougenis et al. [110] introduced an image watermarking method based on Polar Harmonic Transforms (PHTs). The significance of PHTs parameters such as order and magnitude along with the use of a generalized embedding strength calculation process are easily applied to circularly orthogonal transformations which leads to a promising solution of the adaptive issue. The scheme is improved in terms of robustness, capacity and complexity.

It also promotes the traditional schemes to a next generation of moment-based image watermarking.

Makbol et al. [111] devised a novel image watermarking scheme based on the Redundant Discrete Wavelet Transform (RDWT) and the Singular Value Decomposition (SVD). Watermark information is embedded into the singular values of the RDWT sub-bands of the host image. The security issue can be overcome by embedding the watermarking pixel's values without any modification inside the wavelet coefficient of the host image. Simulation results demonstrated that the scheme achieved a large capacity due to the redundancy in the RDWT domain, high imperceptibility for SVD properties and high level of robustness against image processing and geometrical attacks.

Sang et al. [112] presented a double random phase encoding (DRPE) technique where, the image watermark is generated based on Cascaded Iterative Fourier Transform (CIFT). Two phase masks were obtained by applying the CIFT algorithm on the original reference image and the final embedded image of which one of them is used as the secret key. The detection of watermark is achieved by extracting the embedded watermark from the watermarked host image. The recovery of the watermarked was done based on the extracted watermark and the secret key. The presence of watermark can be tested by comparing the content of the recovered watermark and the original reference watermark. The scheme is useful as only one phase mask is needed to recover the watermark. Simulation results show that the scheme has good imperceptibility and robustness.

A novel hologram authentication watermarking scheme has been designed and implemented in Discrete Cosine Transform (DCT) by Cheng et al. [113]. The fabricated hologram is obtained in spatial domain with finite level of precision. The distortion can be minimized by enhancing the precision for storing the watermarked hologram pixels. High perceptual transparency also ensured high performance detection to delivery errors and malicious tampering. Experimental results reveal that the scheme used as an effective filter for blocking polluted or tampered holograms from 3D magnitude and/or phase reconstruction.

Han et al. [114] proposed a dual watermarking technique where, the watermark is embedded into the high-frequency part of Discrete Wavelet Transform (DWT) followed by generating two shares based on visual cryptography. One share is embedded into the low-frequency part of DWT and another is protected by the copyright. Simulation results ensured that the technique offered enhanced security, large embedding capacity and improved robustness.

Benyoussef et al. [115] offered a robust image watermarking scheme based on Dual Tree Complex Wavelet Transform (DT-CWT) and Visual Cryptography concept (VC). The watermark is embedded into the complex wavelet domain to generate a secret and a public share respectively, using LL sub-band features and a VC codebook. The watermark is extracted by stacking the secret and public shares together. The visual quality of the extracted watermark is improved based on a post-extracting reduction procedure. Simulation results demonstrated that the method is resilient to several image processing attacks such as cropping, filtering and compression etc.

Zheng et al. [116] proposed a Walsh-Hadamard Transform (WHT) and its fast algorithm based watermarking in the encrypted domain, which is particularly suitable for its transform matrix consists of only integers. The technique also supports a blind extraction of the watermark both in the decrypted domain and the encrypted domain. The experiments demonstrated the validity and the advantages of the method in terms of security as the secret information is embedded in encrypted domain.

Wang et al. [117] proposed a deformable pyramid transform (DPT) based robust watermarking technique. The DPT is expanded from a closed-form polar-separable steerable pyramid transform (SPT) where, the radial component of the SPT’s basis filters is used for the kernel of the scalable basis filters, and the angular component is considered for the steerable basis filters. The high-pass and band-pass sub-bands are retained by inheriting the shift-invariance from the SPT. The interpolation function is derived for steer ability, scalability and synchronization mechanisms to make it resilient to translation, rotation, and scaling, respectively. Simulations results demonstrated that the scheme is highly robust to geometrical attacks, such as RST, cropping, and row/column line removal, as well as common signal processing attacks such as JPEG compression, additive white Gaussian noise, and median filtering.

To restrict the unauthorized access, a potential solution based on combined Lifting Wavelet Transform (LWT) and Discrete Cosine Transform (DCT) has been designed and implemented by Amy and Yadana [118]. The original image is decomposed into four sub-band images based on the lifting wavelet transform. The DCT is computed on the selected sub-band of the LWT coefficients. The watermark is fabricated in the DCT transformed of the selected LWT sub-band of the carrier image. The performance of the scheme ensured reduced degradation by embedding the secret information.

In 2014, Kakkirala and Chalamala [119] introduced a blind image watermarking scheme in frequency domain based on discrete wavelet transform, singular value decomposition and torus automorphism techniques. Experimental results ensured that the method is robust against different signal and non-signal processing attacks.

Minamoto and Yamaaguchi [120] proposed a chaos-based watermarking method for digital images using the dyadic wavelet transform (DYWT). In order to utilize fast interval arithmetic techniques for fabrication purpose, the wavelet filters values are expressed in interval representation. The watermark is a ternary-valued logo that is embedded into the high-frequency components, and the Rossler system and the Lorentz system are employed as chaos models. The security has been increased by using the chaotic maps which generates a key, and the watermark image is reprocessed with this key. Simulations results demonstrated that the method obtains an improved quality and higher robustness against attacks such as addition of Gaussian white noise, addition of salt & pepper noise, marking and clipping.

Idrissi and Roukhe [121] presented a contourlet transform based digital watermarking technique which inserts a digital document, containing a code robust against any attack that can affect the watermarked data. Simulation results demonstrated the effectiveness of the technique against common image processing attacks.

A novel Discrete Fourier Transform (DFT) based watermarking has been devised by Urvoy et al. [122] in which the watermark is embedded as a noise-like square patch of coefficients. Watermark strength is adjusted by the amplitude whereas, the information is concealed by the phase component. The optimal strength i.e., the visibility threshold is obtained by two important factors: contrast sensitivity function (CSF) of the human visual system (HVS) and a local contrast pooling.

Song et al. [123] presented a novel watermarking scheme for quantum images based on Hadamard transform. The embedding process is controlled by utilizing a dynamic vector which is decided by both the carrier quantum image and the watermark image authorized by the owner. Simulation results demonstrated that the scheme has better visual quality under a higher embedding capacity and lower complexity compared with existing schemes.

Wang and Li [124] projected a novel image watermarking scheme based on contourlet transform. The carrier image has been transformed using contourlet transform and four directions of the second level sub-band were marked. The scrambled digital watermarking is embedded into them. Simulations results demonstrated that the watermarking robust against signals processing attacks such as noise addition, rotation, cropping, compression, brightness variations and mosaic, etc.

Forczmański [125] presented a novel watermarking scheme based on two-dimensional Karhunen-Loeve Transform (2DKLT). The information embedding is performed in the two-dimensional spectrum of KLT which results a lower noise and better adaptation to the image characteristics. A set of experiments related to the color-space, embedding variants and their parameters have been done which ensured the advantages in comparison to certain standard algorithms, such as Discrete Cosine Transformation (DCT), Fast Fourier Transformation (FFT) and Discrete Wavelet Transformation (DWT) respectively.

Lei et al. [126] proposed an adaptive differential evolution (ADE) based optimized watermarking where, the watermark is first scrambled by a piecewise linear chaotic map to increase security and then inserted into the real QWT amplitude coefficients intelligently. The scaling factor is automatically optimized based on the ADE algorithm. Experimental results ensured high robustness against common signal processing operations, especially geometrical distortions.

Nezhadarya and Ward [127] devised a novel image watermarking scheme based on Multi-Scale Derivative Transform (MSDT) where, the detail wavelet coefficients of the image has been used to calculate the first and higher-order image derivatives. The transform maps the diagonal as well as the horizontal and vertical wavelet coefficients to the horizontal and vertical derivatives of the image for embedding purpose. The inverse transform is designed in such a way that any change made in the image derivative domain results effects minimum possible change in the wavelet coefficients.

Lang and Zhang [128] proposed a generalized two-dimensional Fractional Fourier Transform (FRFT) based watermarking scheme where, the output has the mixed time and frequency components of the signal. The original image is partitioned into non-overlapping blocks where, each block is transformed by the two dimensional Fractional Fourier Transforms with two fractional orders. Watermark bits are embedded by modifying the back-diagonal FRFT coefficients of each image block at the same location with a random array. Inverse two-dimensional Fractional Fourier Transform is used to obtain the watermarked image and the transform orders can be considered as the encryption keys. Simulations results demonstrated that the scheme is very robust to JPEG compression, noise attacks and image manipulation operations.

According to Varsaki et al. [129], Discrete Gould Transform (DGT) is an excellent idea to devise a novel data hiding scheme for authentication and tamper detection. The Gould coefficients represent the differences between neighboring pixels where, the small changes of the coefficients generate even smaller changes to the pixels. A minor manipulation can destroy the fabricated watermark, and therefore, its absence can prove the alteration.

1.5. Problem Domain

Digital watermarking is a technique that can fabricate useful information into the carrier images to verify the integrity and authenticity. Security, payload, quality and robustness are four important characteristics of watermarking to measure the effectiveness of a proposed algorithm. Based on the literature survey, limitations have been observed in existing techniques mainly most of the existing watermarking techniques are focused on high robustness against different visual and geometrical attacks for copy-right protection. It can be noted that there is a trade-off among robustness, payload and the quality. Therefore, most of the techniques available in the literature survey achieved a high robustness by compromising the payload and the quality. In addition, very few of the researchers have chosen digital watermarking as a solution of authentication. A large number of watermarking techniques are available but, among them very few are applicable for color images. Watermarking in transform domain offers better security and higher robustness over spatial domain techniques. In recent days, frequently used transformations for watermarking include Slant Transform (ST), Contourlet Transformation (CT), Discrete Cosine Transformation (DCT), Discrete Pascal Transform (DPT), Discrete Wavelet Transform (DWT), Fourier-Mellin Transform (FMT) and Discrete Gould Transform (DGT) respectively. Performance of the existing watermarking schemes are analyzed in terms of payload (as expressed in “bits per Byte” or in short “bpB”) and the quality metric peak signal to noise ratio (PSNR) which is expressed in dB respectively.

Ho et al. [37] embeds the watermark bits into the selected coefficients of the 8 × 8 Slant transformed matrix of the textured images. It is seen that only 16 embedding locations are identified from each 8 x 8 sub-blocks of the transformed coefficients to maintain the trade-off between robustness and security. The PSNR for “Singapore” and “Dolomites” images are

37.43 dB and 40.65 dB respectively, which are considered as significant distortion with respect to 0.25 bpB of payload.

Kumaran and Thangavel [62] used Contourlet Transform (CT) for watermarking due to its multi-scale and directional filter bank properties. They introduced surrounding mean and zero-tree embedding schemes, respectively. Both of the schemes offers high robustness against attacks such as low-pass filtering (LPF), media filtering (MF) and JPEG compression but the schemes lacks in terms of payload and image quality. For a 256 x 256 carrier image, the payload is 1024 bits i.e., 0.015625 bpB since only one coefficient is modified in each 8 x 8 block of a sub-band to fabricate the secret information. The PSNR is around 40 dB for both surrounding mean and zero-tree embedding schemes which ensures significant degradation in the quality of the watermarked images with respect to the specified (low) payload.

Wang et al. [74] partitions the carrier images into 8 x 8 non-overlapping blocks which are transformed through Discrete Cosine Transformation (DCT). In general, embedding bits in lower frequency bands results high distortion whereas, embedding in high frequency bands suffers from robustness against attacks. Thus, mid-frequency bands are preferable for embedding. Based on a generic band selection (GBS) scheme, watermark bits are embedded into the selected bands. It has been observed that for a cover images of dimension 512 x 512, only 4 bits are embedded in each 8 x 8 block which ensures the payload of 0.0625 bpB. Consequently, the PSNR for “Lena” is 34.78 dB and that of for “Baboon” is 28.10 dB which results high distortion relative to the achieved payload.

Varsaki et al. [88] used Discrete Pascal Transform (DPT) for embedding using two important features i.e., the high sensitivity against noise and the weighted difference of neighboring data values. The scheme embeds secret bits at the lower right corner of the DPT co-efficient matrix. It is seen that the payload is only 0.25 bpB for each 2 x 2 transformed matrix which is significantly low. The average PSNR for “Lena”, “Baboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images is 37.40 dB at 0.25 bpB of fabrication density. Simulation results demonstrate low imperceptibility at the fixed (as well as low) payload.

Manoochehri et al. [91] used Discrete Wavelet Transform (DWT) and Fourier-Mellin Transform (FMT) for embedding watermark into the carrier images to achieve higher robustness against visual and geometrical attacks. But, the scheme ensures low payload values of 0.0538 and 0.1558 bpB by separately fabricating two logos of sizes 41 x 31 and 69 x 74 into the “Plane” image of dimension 512 x 512. The watermarked image retains an average PSNR of around 35 dB by which is considered as significant distortion with respect to the specified payload.

Varsaki et al. [129] adopted Discrete Gould Transform (DGT) for embedding secret bits into the transformed coefficients by determining the difference between neighboring pixels to alter the DGT coefficients for embedding secret bits. It has been observed that one bit of the secret data is embedded into each DGT co-efficient of the 2 x 2 block. Simulation results demonstrate the average PSNR of 48.70 dB as obtained from “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” images at 1 bpB of fixed as well as low payload.

1.6. Objectives

To overcome the shortcomings of the existing techniques, some novel algorithms have been devised based on Discrete Hartley Transform (DHT), Binomial Transform (BT), Legendre Transform (LT), Stirling Transform (ST) and group of linear transformations for dihedral group of order 4 (G-lets D4) respectively. It is seen that most of the existing watermarking techniques are applicable for gray-scale images. On the contrary, embedding in color images could be an effective choice since the use of color images as cover media ensured better transparency and higher payload. In addition, it has been observed from the literature survey that the major applications of watermarking are copy-right protection. In contrast, very few of the researches have chosen watermarking as an authenticating tool for tamper detection. Watermarking based on Slant Transform (ST), Contourlet Transformation (CT) and Discrete Cosine Transformation (DCT) [37, 62 and 74] used selective components for secret bits insertion which effects high robustness, low payload and significant quality distortion. The trade-off between payload and image quality can be maintained by choosing the block size as 2 x 2 or 1 x 2 which in turn provides the payload values in the range [0.5 – 3 bpB] with acceptable quality distortion in the output watermarked images [148]. It is seen that DCT based watermarking scheme [74] deals with fractional coefficients which ensures floating-point calculation and makes an operation slower. It could be avoided by introducing such transforms which ensured real transformed matrices as output with respect to real pixel matrices as input. Most of the watermarking schemes in transform domain suffer from overflow and underflow which could be avoided by using pixel adjustment process prior to embedding. Sometimes, an additional re-adjustment operation may also be incorporated to avoid overflow and underflow situations. The re-computed pixel components become non-negative and fall into the range [0, 255].

The main objectives of this thesis are:

i) To introduce color image based authentication which will ensure the integrity and authenticity of color images.

ii) To investigate the limitations of existing watermarking schemes and to overcome the limitations in terms of image quality and payload.

iii) To design and develop new schemes to improve the enhancement of watermarked images quality.

iv) To enhance the security of embedded information by fabricating the watermark in transform domain.

v) To enhance payload by maintaining the tradeoff between quality and payload.

vi) To improve the quality based on genetic algorithm based optimization/quality enhancement scheme.

vii) To avoid overflow and underflow conditions

viii) To ensure only real valued calculations.

1.7. Methodologies

The major limitations of the existing transforms based watermarking schemes [37, 62, 74, 88, 91 and 129] as discussed in the problem domain have been resolved by exploiting watermarking based on Discrete Hartley Transform (DHT), Binomial Transform (BT), Legendre Transform (LT), Stirling Transform (ST) and group of linear transformations for dihedral group of order 4 (G-lets D4) respectively. In contrast to existing schemes [37, 62, 74, 88, 91 and 129], the proposed transform uses 2 x 2 and 1 x 2 block as the window size because it has been examined that the decrease in block size effects enhancement in fidelity. So, the carrier images are partitioned into 2 x 2 or 1 x 2 sub-matrices of pixel components in a sliding window manner. The proposed transforms are applied on each 2 x 2 or 1 x 2 sub-matrices of pixel components to obtain the transformed components of identical block sizes. In contrast to the traditional transforms based watermarking schemes, no overflow or underflow conditions are arises as the pixel components are adjusted prior to embedding. However, if still overflow/underflow occurs then an additional re-adjustment is applied in transform domain. Watermark size, content and the message digest (MD) obtained from the watermark are inserted into the transformed components to achieve variable payload that offers a spread from 0.5 to 3 bpB. The security of the watermark has also been improved in Stirling Transform (ST) based watermarking scheme where, the watermark is scrambled through Arnold’s cat map prior to embedding. A post-embedding quality enhancement / genetic algorithm (GA) based optimization has also been used to minimize the degradation of quality in the watermarked image. The quality enhancement and GA optimization have been applied in such a way that the fabricated bits of each transformed component are kept unaltered. Inverse transform converts each 2 x 2 or 1 x 2 sub-matrices of quality enhanced or optimized components into the spatial domain. Successive block embedding operations produces the watermarked image. At the recipient end, the message digests (MD), size and the content of the watermark bits are extracted from the transformed components. For each 8 (eight) bits extraction of the watermark, construct one alphabet/one pixel component. The recipient operate the authentication process by matching the extracted message digest (MD) with the re-computed message digest (MD') as obtained from the extracted watermark image. If both message digests MD and MD' are identical then the authentication process is said to be successful, otherwise, it is said to be unsuccessful. Therefore, any kind of attack on the watermarked image is easily detectable. Unlike DCT based scheme [74], above mentioned transforms produces a real transformed matrix in transform domain which ensures less computational time and faster execution.

1.8. Organization of the Thesis

The thesis consists of ten chapters. Chapter 1 deals with the introductory discussions of the watermarking, essence of watermarking, scrambling of watermark using Arnold’s cat map, literature survey, problem domain, objectives, methodologies, organization of the thesis, metrics for evaluation and the salient features of the proposed thesis, respectively.

Chapter 2 deals with two novel watermarking schemes in Discrete Hartley Transform (DHT) domain which are 2 x 2 block based watermark fabrication using two dimensional Discrete Hartley Transform (2D-DHT) or more specifically Separable Discrete Hartley Transform (SDHT) i.e., WDHT_2x2 and 1 x 2 block based watermark fabrication using one dimensional Discrete Hartley Transform (1D-DHT) i.e., WDHT_1x2, respectively.

In chapter 3, watermarking schemes based on Legendre Transform (LT) has been presented. It has been classified into two categories: 2 x 2 block based watermark fabrication using Legendre Transform (LT) (WLT_2x2) and 1 x 2 block based watermark fabrication using Legendre Transform (LT) (WLT_1x2) respectively.

Watermarking based on Binomial Transform (BT) has been proposed in chapter 4. This chapter is further been classified into two sections namely, the 2 x 2 block based watermark fabrication using Binomial Transform (BT) i.e., WBT_2x2 and 1 x 2 block based watermark fabrication using Binomial Transform (BT) i.e., WBT_1x2, respectively.

Watermarking based on Stirling Transform (ST) has been proposed in chapter 5 which in turn is classified into 2 x 2 block based watermark fabrication using Stirling Transform (ST) (WST_2x2) and 1 x 2 block based watermark fabrication using Stirling Transform (ST) (WST_1x2) respectively.

In chapter 6, fragile watermarking based on group of linear transformations for dihedral group of order 4 (G-lets D4) has been presented. It has been classified into two classes: 2 x 2 block based watermark fabrication using group of linear transformations for dihedral group of order 4 (G-lets D4) i.e., WGD4_2x2 and 1 x 2 block based watermark fabrication using group of linear transformations for dihedral group of order 4 (G-lets D4) i.e., WGD4_1x2 respectively.

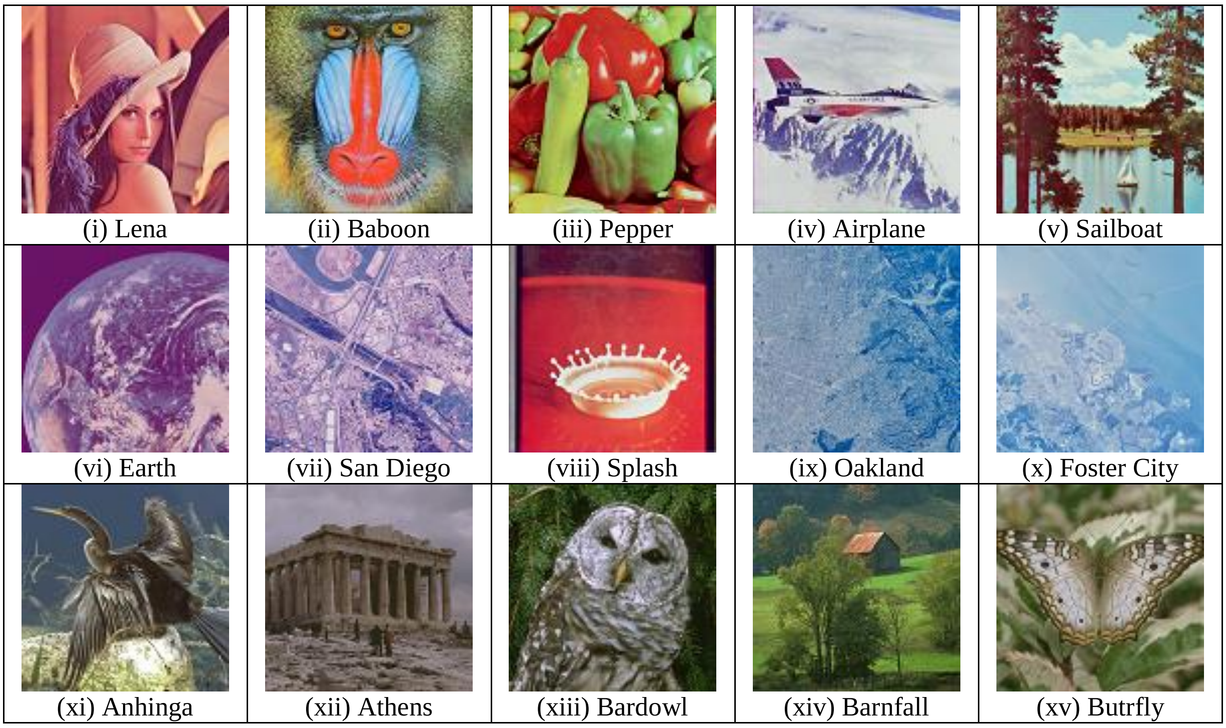
Chapter 7 deals with quality enhancement of Discrete Hartley Transform (DHT) based watermarking of chapter 2, Legendre Transform (LT) based watermarking of chapter 3, Binomial Transform (BT) based watermarking of chapter 4, Stirling Transform (ST) based watermarking of chapter 5 and the group of linear transformations for dihedral group of order 4 (G-lets D4) based watermarking of chapter 6, respectively. The quality enhancement method has been incorporated to reduce the degradation in quality of the watermarked images by keeping the embedded secret data intact.

In chapter 8, the fragile watermarking schemes based on Discrete Hartley Transform (DHT) (chapter 2), Legendre Transform (LT) (chapter 3), Binomial Transform (BT) (chapter 4), Stirling Transform (ST) (chapter 5) and group of linear transformations for dihedral group of order 4 (G-lets D4) (chapter 6) respectively have been extended to improvise the visual quality of the watermarked images. Since, watermarking is also treated as an optimization problem, genetic algorithm (GA) has been used to find the optimized solution corresponding to the problem. The optimization enhances the quality of the watermarked images and also maintains the trade-off between the quality and the payload by keeping the embedded information intact.

Analysis and discussions are done in chapter 9 and the conclusions are outlined in chapter 10. References are given at the end of the thesis.

1.9. Metrics for Evaluations

Color images have been considered as the cover media. The performance of the watermarked images can be measured in terms of widely acceptable features such as payload, quality, robustness and imperceptibility, respectively. As the research work is based on fragile watermarking, the major emphasis has been given on payload and quality. Twenty different benchmark (BMP) images [130, 131] of dimension 512 × 512 are taken to compute results which are shown in fig. 1.1. Images are labeled as: (i) Lena, (ii) Baboon, (iii) Pepper, (iv) Airplane, (v) Sailboat, (vi) Earth, (vii) San Diego, (viii) Splash, (ix) Oakland, (x) Foster City, (xi) Anhinga, (xii) Athens, (xiii) Bardowl, (xiv) Barnfall, (xv) Butrfly, (xvi) Bobcat, (xvii) Bodie, (xviii) Bluheron, (xix) Colomtn and (xx) Desert. The variable sizes of (xi) Gold-Coin (i.e., the authenticating watermark data) are embedded into the benchmark images based on the offered payload [0.5 – 3 bpB]. The degradation of quality for the watermarked images is lowest at 0.5 bpB and highest at 3 bpB, respectively. In general, the quality of watermarked images with respect to the original cover images is measured in terms of peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD) and standard deviation error (SDE), respectively. For comparative analysis, five cover images such as Lena, Baboon, Pepper, Airplane and Sailboat has been considered as these are widely used in existing schemes. In this section, the payload as well as the quality metrics is briefly described.



(i) Lena (ii) Baboon (iii) Pepper (iv) Airplane (v) Sailboat

(vi) Earth (vii) San Diego (viii) Splash (ix) Oakland (x) Foster City

(xi) Anhinga (xii) Athens (xiii) Bardowl (xiv) Barnfall (xv) Butrfly

(xvi) Bobcat (xvii) Bodie (xviii) Bluheron (xix) Colomtn (xx) Desert

(xxi) Gold Coin

Fig 1.1. Different cover images [130, 131] of dimension 512 × 512 along with the authenticating watermark image

1.9.1. Payload

The amount of concealed information into the watermarked image is considered to be the payload [144]. It can be measured in bits, Bytes, Kilo Bytes or Mega Bytes. In recent days, a new unit is emerged i.e., bits per Byte (bpB) which is nothing but the number of embedded bits per byte. For example, the payload of 2 bpB for the 512 x 512 cover image corresponding to RGB color channels yields (2 × 3 × 512 × 512) = 15,72,864 bits of fabricated secret information.

1.9.2. Mean Squared Error (MSE)

The mean squared error (MSE) is defined as the difference between the estimator and what is estimated however, an estimator measures the average of the squares of the “errors”. It is computed as the average squared difference of pixels between an original image and its distorted version. The computation is done by summing up the squared differences of all the pixels followed by dividing it with the total number of pixels. Considering the original image (I) and the watermarked image (K) of dimension m x n, the MSE [143] is defined as:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{1.4}$$

1.9.3. Peak Signal to Noise Ratio (PSNR)

Peak signal-to-noise ratio (PSNR) is the ratio between the power of the noisy signal and the power of the original signal. The original signal is the cover image and the noisy signal is nothing but the watermarked image in the context of digital image watermarking. The wide dynamic range of PSNR intends to express it in terms of the logarithmic decibel scale. The PSNR (in dB) [143] is defined as:

$$PSNR = 20log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{1.5}$$

Here, MAX_I is is 2^B-1 as the samples are represented using linear PCM with B bits per sample. For gray-scale images, MAX_I is 255 since the pixels are represented using 8 bits per sample. The PSNR for color images is as similar as the definition of PSNR for gray-scale images except the MSE is the sum over all squared value differences divided by the dimension of the image and by three.

1.9.4. Image Fidelity (IF)

Considering the original image (I_1) and the watermarked image (I_2) of dimension m x n, the image fidelity (IF) [143] is defined as:

$$IF = 1 - \sum_{m,n}\left(I_1(m,n) - I_2(m,n)\right)^2 \Big/ \sum_{m,n} {I_1}^2(m,n) \tag{1.6}$$

The image fidelity (IF) lies in the ranges from [0, 1], the closer the IF to 1, watermarked image is closer to the original image.

1.9.5. Universal Image Quality index (UIQ)

Universal image quality index (Q) was investigated by Wang et al. (2002) [145] for human visual system model. It is expresses as follows:

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}((\bar{x})^2 + (\bar{y})^2)^{-1}}{(\sigma_x^2 + \sigma_y^2)} \tag{1.7}$$

where, $\bar{x}$, $\bar{y}$ are mean values of original image (I) and watermarked image (I'), σ_x, σ_y and σ_{xy} are variances of I and I', respectively. The dynamic range of UIQ is [0, 1] however, the value closest to one signifies higher similarity between the original and the watermarked images with less structural distortion.

1.9.6. Structural Similarity index (SSIM)

The structural similarity index (SSIM) [146] is used to measure the similarity in the structure of two images. The visual quality of the watermarked image with respect to the original image can be measured in terms of structural information to ensure stronger pixel's inter-dependencies in particular they are spatially close. The SSIM value lies between -1 and 1 however, for watermarking purpose the SSIM lie between 0 and 1. It is mathematically expressed as follows:

$$SSIM(I,I') = \frac{(2\mu_i\mu_{i_s} + c_1)(2\sigma_{ii_s} + c_2)}{(\mu_i^2 + \mu_{i_s}^2 + c_1)(\sigma_i^2 + \sigma_{i_s}^2 + c_2)} \tag{1.8}$$

where, μ_i, μ_{i_s}, σ_i, $\sigma_{i_s}$and σ_{ii_s} are mean of i and i_s, variance of i and i_s, the covariance of i and i_s, respectively. Two variables c_1 and c_2 are constants which are used to stabilize the division with weak denominator.

1.9.7. Standard Deviation (SD)

Standard deviation (SD) [149] is used to measure the amount of variation or dispersion from the average. A low standard deviation ensures that the data points tend to be very close to the mean (also termed as the expected value); a high standard deviation reveals that the data points are spread out over a large range of values. Considering the original image (I) and the watermarked image (I') of dimension U x V, the standard deviation (σ) is mathematically expressed as follows:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{1.9}$$

where, N is the number of pixels obtained from the dimension U x V, x_i is the value of each pixel and μ is the mean of N pixels. For color images, the standard deviation (SD) is obtained by taking the average of standard deviation values corresponding to red, green and blue channels.

In general, as the size of the images increases, the smooth area within the image is also increasing therefore, the error/distortion is distributed among more pixels in large smooth area with same group of pixels which conform the decreasing the density of the errors and as a result the standard deviation is also decreasing. Unlikely, in case of images consisting of huge edge area, the opposite trends has been observed. The numbers of edges are enormous as well as the large numbers of smooth area with smaller size is observed; therefore, the rate of increase of smooth area are less with increase of size of the images compared to other benchmark images.

1.9.8. Standard Deviation Error (SDE)

As the standard deviation is a measure of how far the signal fluctuates from the mean, a better analysis of standard deviation has been made in terms of standard deviation error (SDE) where, the error is nothing but the absolute difference of standard deviation between the original and the watermarked images. The standard deviation error (σ_Δ) is mathematically expressed as follows:

$$\sigma_\Delta = |\sigma_I - \sigma_{I'}| \tag{1.10}$$

where, $\sigma_I$and $\sigma_{I'}$ are the standard deviations of the original (I) and the watermarked (I') images, respectively.

1.10. Salient Features of the Thesis

The major limitations of the existing methods presented in the literature consist of low payload and high distortion. Most of the schemes were designed to achieve high robustness against visual/geometrical attacks which made watermarking as an effective tool for copy-right protection. Unlikely, very few of the researchers had chosen digital watermarking as a solution of authentication. The main motivation of the thesis is to design and implementation of fragile watermarking schemes in transform domain which can verify the authenticity of color images. It is seen that most of the existing schemes as discussed in the literature are facing the following issues. Firstly, the larger window size is used to convert each sub-image block of the carrier image into the transform domain because these schemes were focused to achieve high robustness by compromising the payload and quality. Secondly, most of the existing schemes had chosen gray-scale images as the cover image which might reduce the payload and transparency over the color images. Thirdly, some transformations of the existing watermarking schemes also suffered from high computational cost as the computation is based on the floating-point values. Fourthly, the content of the watermarked image may be altered during transmission by some intentional/unintentional attacks of the intruder. The attacks may be mounted in such way that it leads to a slight modification in the watermarked images but are kept undetected. In general, it is seen that very few of the watermarking schemes deal with the authenticity of color images. In addition, the majority of these schemes supports fixed payload. However, these problems have been addressed carefully and efficiently in the proposed methodologies of the thesis by introducing variable payload (i.e., 0.5 to 3 bpB) with acceptable visual imperceptibility.

The literature surveys reveal that most of the watermarking techniques used gray-scale images as the cover image. In this thesis, a set of fragile watermarking schemes based on Discrete Hartley Transform (DHT), Binomial Transform (BT), Legendre Transform (LT), Stirling Transform (ST) and group of linear transformations based on dihedral group of order 4 (G-lets D4) have been proposed to verify the authenticity of color images. The color images as the cover image ensured better transparency and high payload as the watermark information are embedded separately into the red, green and blue channels. The techniques outperform the existing methods in terms of payload, fidelity and computational overhead.

It is seen that Ho et al.'s [37] Discrete Cosine Transform (DCT) based watermarking scheme and Kumaran and Thangavel's [62] Contourlet Transform (CT) based watermarking scheme used mid-frequency regions of each 8 x 8 sub-matrix for embedding. The primary concern of both these schemes is to achieve higher robustness by compromising the payload and quality. In contrast, proposed watermarking schemes used 2 x 2 and 1 x 2 transformed blocks for watermark embedding which results variable payload (i.e., 0.5 to 3 bpB) with acceptable visual clarity.

Unlike DCT based watermarking scheme [74], Discrete Hartley Transform (DHT), Binomial Transform (BT), Legendre Transform (LT), Stirling Transform (ST) and group of linear transformations based on dihedral group of order 4 (G-lets D4) produces a real transformed matrix as output from a real input pixel matrix. Therefore, it avoids floating-point calculations and offers less computational overhead.

For a given authentication system, the role of the intruder is to alter the image in such a way that perceptually the image is not considerably degraded, however, the frequency distribution is transformed in such a manner that the decoder fails to extract the fabricated data. The recipient extracts the fabricated watermark size, content and a message digest (MD) from the watermarked image. The extracted message digest (MD) was actually obtained through MD5 algorithm during embedding. This message digest (MD) is compared with respect to another message digest (MD'), where MD' has been obtained from the extracted watermark at the recipient end. If the extracted message digest (MD) matches with the re-computed message digest (MD'), then the authentication process is said to be successful, otherwise, it is said to be unsuccessful.

Two basic characteristics of digital watermarking are payload and imperceptibility respectively. Watermarking is considered as an optimization problem since the above mentioned metrics are conflicting between themselves. For instance, if payload increases then the imperceptibility (quality) decreases and vice-versa. Hence, Genetic Algorithm (GA) has been utilized to search the user parameters in a wide range to obtain the optimal results that reflects improvement in terms of imperceptibility with respect to varying payload. In contrast, the imperceptibility of the watermarked images has also been improved by applying a post-embedding quality enhancement scheme. The optimization or quality enhancement scheme has been applied in such a way that the fabricated watermark information is kept unaffected. However, the optimization or quality enhancement scheme results a high computational overhead which may slow down the execution of the program.

Section 5.2.1 deals with scrambling of watermark through Arnold's cat map prior to embedding. Arnold's cat map is a chaos based scheme which is utilized for transforming the values of each pair of pixel components into a new pair based on a hash function. The scrambled bits are fabricated in Stirling Transform (ST) domain to improve security.

Chapter 2

Watermarking based on Discrete Hartley Transform (WDHT)

2.1 Introduction

In this chapter, fragile watermarking based on Discrete Hartley Transform (DHT) has been proposed for color image authentication. The carrier image is decomposed into non-overlapping blocks which in succession are converted into transform domain based on Discrete Hartley Transform (DHT). Secret information is fabricated into the transformed components to achieve variable payload with minimum degradation in fidelity. Inverse Discrete Hartley Transform (IDHT) is applied on each block of embedded components to re-compute the pixel components in spatial domain. However, carrying out the process repeatedly, the watermarked image is produced. The recipient extract the fabricated secret information based on the reverse operation and the authenticity of the secret information is verified through a pair of message digests. Simulation results demonstrate that the proposed techniques obtain variable payload and less quality degradation over the existing techniques.

2.2 The Technique

Two novel fragile watermarking techniques based on Discrete Hartley Transform (DHT) have been proposed for color image authentication. The carrier image is divided into P x Q non-overlapping blocks (where, $1 \leq P \leq 2$ and $Q = 2$) corresponding to red, green and blue channels. A pre-embedding adjustment is applied on pixel components to avoid the occurrence of overflow and underflow. Each P x Q sub-image block of the carrier image is converted into transform domain based on Discrete Hartley Transform (DHT) in a sliding window manner. Secret bits corresponding to the message digest (MD), size and the content of the watermark are subsequently fabricated into the transformed components without hampering the P least significant bits. An inverse Discrete Hartley Transform (IDHT) is applied on each P x Q sub-block of embedded transformed components which yields P x Q sub-image block in spatial domain. The unaltered P least significant bits of each embedded component guarantee the non-occurrence of fractional value subsequent to embedding. The repetition of the above process yields the watermarked image in spatial domain. At the recipient end, the watermarked image is partitioned into P x Q non-overlapping blocks (where, $1 \leq P \leq 2$ and $Q = 2$) for red, green and blue channels. The secret bits corresponding to the message digest (MD), size and the content of the watermark are subsequently extracted from the transformed components. An inverse Discrete Hartley Transform (IDHT) is applied on each P x Q sub-block of embedded components to get back the pixel components in spatial domain. Each 8 (eight) bits extraction of the watermark, construct one alphabet/one primary (R/G/B) color component. The process is repeated until and unless the extraction of secret bits is completed.

Watermarked image may be altered as a result of deliberate attacks such as filtering, blurring etc. or inadvertent attacks viz. lossy compression, noise addition, and filtering etc. Due to various kinds of attacks, an altered version of the watermarked image is available to the recipient. For a given authentication system, the role of the intruder is to alter the image in such a way that perceptually the image is not considerably degraded, however, the frequency distribution is transformed in such a manner that the decoder fails to extract the fabricated data. The recipient of the proposed authenticating system extracts the fabricated watermark size, content and the message digest (MD) from the watermarked image and then operate the authentication process by matching the extracted message digest MD with respect to another message digest (MD'), where MD' has been obtained from the extracted watermark at the recipient end. The identical contents of the extracted message digest MD and the re-computed message digest MD' makes the authentication process as successful. The dissimilar values of MD and MD' yields an unsuccessful authentication. As a consequence, any kind of alteration made to the watermarked image is easily detectable.

Section 2.2.1 of this chapter deals with 2 x 2 block based watermark fabrication that of section 2.2.2 describes 1 x 2 block based watermark fabrication.

2.2.1 2 x 2 Block based Watermark Fabrication

The two dimensional Discrete Hartley Transform (2D-DHT) which is also widely known as the Separable Discrete Hartley Transform (SDHT) [132] produces real output sequence for a given real input sequence. Separable Discrete Hartley Transform (SDHT) is applied over M x N block of a carrier image to obtain transformed components $H_s(\delta, \omega)$ in frequency domain as given in equation (2.1).

$$H_S(\delta,\omega) = \sum_{\alpha=0}^{N-1}\sum_{\beta=0}^{M-1} p(\alpha,\beta)cas\left(\frac{2\pi\delta\alpha}{N}\right)cas\left(\frac{2\pi\omega\beta}{M}\right) \qquad (2.1)$$

where, δ varies from 0 to M-1 and ω varies from 0 to N-1.

The variable δ and ω are the frequency variables corresponding to the spatial domain variables α and β where, p(α, β) represents the intensity values of the pixel components in spatial domain. The sequence *cas* is defined in equation (2.2) and (2.3).

$$cas\left(\frac{2\pi\delta\alpha}{N}\right) = \cos\left(\frac{2\pi\delta\alpha}{N}\right) + \sin\left(\frac{2\pi\delta\alpha}{N}\right) \tag{2.2}$$

$$cas\left(\frac{2\pi\omega\beta}{M}\right) = \cos\left(\frac{2\pi\omega\beta}{M}\right) + \sin\left(\frac{2\pi\omega\beta}{M}\right) \tag{2.3}$$

Similarly, the inverse transform in Separable Discrete Hartley Transform (SDHT) domain is used to convert the M x N block of frequency components into the spatial domain as given in equation (2.4).

$$p(\alpha,\beta) = \frac{1}{NM}\sum_{\delta=0}^{N-1}\sum_{\omega=0}^{M-1} H_S(\delta,\omega)cas\left(\frac{2\pi\delta\alpha}{N}\right)cas\left(\frac{2\pi\omega\beta}{M}\right) \tag{2.4}$$

where, α varies from 0 to M-1 and β from 0 to N-1.

Equations (2.1) and (2.4) ensures that both transforms are separable in α and β, and can be executed by applying 1D-DHT to each row and then a 1D-IDHT to each column (or vice-versa).

The formulation of Separable Discrete Hartley Transform (SDHT) for image sub-block of size 2 x 2 has been expressed in equation (2.5).

$$H_S(\delta,\omega) = \sum_{\alpha=0}^{1}\sum_{\beta=0}^{1}(-1)^{\delta\alpha}(-1)^{\omega\beta}p(i,j) = f_{\delta,\omega}(Say) \tag{2.5}$$

where, α and β varies from 0 to 1 and the variable δ and ω are the frequency variables corresponding to the frequency components $f_{\delta,\omega}$ respectively.

Similarly, by applying the inverse Separable Discrete Hartley Transform (ISDHT) over the 2 x 2 block of embedded components in transform domain, the re-computed pixel components of the 2 x 2 mask can be obtained as given in equation (2.6).

$$p(\alpha,\beta) = \frac{1}{4}\sum_{\delta=0}^{1}\sum_{\omega=0}^{1}(-1)^{\delta\alpha}(-1)^{\omega\beta}H_S\,(\delta,\omega) = p_{\alpha,\beta}(Say) \tag{2.6}$$

where, δ and ω varies from 0 to 1 and the variable α and β are the spatial domain variables corresponding to the pixel intensity values $p_{\alpha,\beta}$ in spatial domain.

The proposed technique has been described in the following sections. The algorithms for insertion as well as extraction along with an example are described in sections 2.2.1.1, 2.2.1.2 and 2.2.1.3, respectively. Results and discussions have been elaborated in section 2.2.1.4.

2.2.1.1 Insertion

The cover image of dimension U x V is partitioned into 2 x 2 non-overlapping blocks in a sliding window manner. A pre-embedding adjustment has been applied on each pixel component to avoid overflow and underflow. Separable Discrete Hartley Transform (SDHT) is applied on each 2 x 2 sub-image block of red/green/blue channel to convert the pixel components into transformed components. Variable payload with minimum degradation in fidelity can be achieved by successively embedding the message digest, size and the content of the watermark into the transformed components. The bit insertion is started from the second bit position of the least significant part (i.e., LSB-2) toward higher order bit position to restrict the generation of fractional pixel components during inverse transform. Inverse Separable Discrete Hartley Transform (ISDHT) is applied on each 2 x 2 embedded block to obtain the pixel components in spatial domain. The process is repeated until and unless the watermarked image is obtained in spatial domain.

Algorithm 2.1:

Input: Carrier/cover image (I) and authenticating watermark image (W).

Output: Watermarked image (I′).

Method: Separable Discrete Hartley Transform (2D-SDHT) converts the carrier images from spatial domain into transform domain in a block-wise manner. The watermark (along with a message digest) is embedded in transform domain to achieve varying payload, less distortion and improved security. The embedding steps are as follows:

Step 1: Obtain a message digest (MD) from the authenticating watermark.

Step 2: The authenticating watermark size (in bits) is obtained by embedding the watermark into the three channels of the U x V color image as given in equation (2.7).

$$W_{size} = B \times \big(3 \times (U \times V)\big) - (MD + L) \qquad (2.7)$$

where, B, MD and L represents the embedding payload in terms of bits per Byte, the message digest computed from the watermark and the header information corresponding to the size of the watermark, respectively. The MD and L are consisting of 128 and 32 bits while the usual values of B are 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively.

Step 3: The cover image (I) is partitioned into 2 x 2 non-overlapping blocks in row major order. Each 2 x 2 block is consisting of four pixel components $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$ of red/green/blue channel where, for all i and j, $0 \leq i, j \leq 1$.

Step 4: A pre-embedding adjustment is applied on each pixel component (p) to avoid overflow and underflow based on the payload value of B bits per Byte as given in equation (2.8).

$$p = \begin{cases} \left(2^8 - 2^{\lceil B \rceil + 2}\right) : p \geq \left(2^8 - 2^{\lceil B \rceil + 2}\right) \\ 2^{\lceil B \rceil + 2} : p \leq 2^{\lceil B \rceil + 2} \end{cases} \qquad (2.8)$$

Step 5: Apply Separable Discrete Hartley Transform (SDHT) on 2 x 2 sub-matrices of pixel components to obtain the frequency components consisting of $f_{i,j}$, $f_{i,j+1}$, $f_{i+1,j}$ and $f_{i+1,j+1}$ respectively.

Step 6: λ_1/ λ_2/ λ_3/ λ_4 bits from the secret bit-stream (corresponding to the message digest (MD), size (L) and the content (W) of the watermark) are subsequently fabricated on first/second/third/fourth transformed component starting from the 2nd bit position of the least significant part (LSB-2) toward higher order bit position. The generalized form of λ bits of secret information fabrication on each transformed component for the payload value of B bits per Byte (bpB) can be derived by the given equation (2.9).

$$\lambda = \begin{cases} \lfloor B \rfloor : \lambda = \lambda_1 | \lambda_2 \\ \lceil B \rceil : \lambda = \lambda_3 | \lambda_4 \end{cases} \qquad (2.9)$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5 and for all λ, $0 \leq \lambda \leq 3$.

Step 7: Inverse Separable Discrete Hartley Transform (ISDHT) is applied on each 2 x 2 sub-matrix of embedded components to re-compute the pixel components.

Step 8: Repeat steps 3 to 7 to embed the entire watermark size, content and the message digest MD, respectively. The block embedding operation in succession generates the watermarked image (I') in spatial domain.

Step 9: Stop.

2.2.1.2 Extraction

At the recipient end, the watermarked image is partitioned into 2 x 2 non-overlapping blocks in a sliding window manner. Separable Discrete Hartley Transform (SDHT) is applied on each 2 x 2 sub-image block of red/green/blue channel to convert the pixel components into transform domain. The reverse procedure is applied to extract the watermark size, content and the message digest (MD) from the transformed components. Inverse Separable Discrete Hartley Transform (ISDHT) is applied on each 2 x 2 sub-matrix of embedded components. The process is repeated to re-construct the message digest and the watermark from the retrieved information. Another message digest MD′ is obtained from the extracted watermark and the same is compared against the extracted message digest MD. If the respective message digests MD and MD′ are identical, then the watermarked image is considered to be authenticated; otherwise, it is to be treated as tampered. The extraction procedure is described in algorithm 2.2.

Algorithm 2.2:

Input: Watermarked image (I′).

Output: The authenticating watermark (W) and the 128 bits message digest (MD).

Method: Separable Discrete Hartley Transform (SDHT) is used to extract the watermark (along with the message digest MD) from the watermarked image (I′) by converting the image from spatial domain into transform domain. Successive extracted bits constitute the secret watermark from which another message digest (MD′) is re-computed. Both message digests MD and MD′ are compared to verify the authenticity. The detailed steps of extraction are as follows:

Step 1: The watermarked image (I') is partitioned into 2 x 2 non-overlapping blocks in row major order. Each block is consisting of four pixel components $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$ of red/green/blue channel where, for all i and j, $0 \leq i, j \leq 1$..

Step 2: Apply Separable Discrete Hartley Transform (SDHT) on 2 x 2 sub-matrices of pixel components to obtain the following frequency components: $f_{i,j}$, $f_{i,j+1}$, $f_{i+1,j}$ and $f_{i+1,j+1}$ respectively.

Step 3: λ_1/ λ_2/ λ_3/ λ_4 bits of the fabricated secret bit-stream are successively extracted from first/second/third/fourth transformed component. The bit extraction is started from the 2nd bit position of the least significant part (i.e., LSB-2) toward *higher order bit position. Considering the payload value of B bits per Byte, λ bits of secret information is extracted from each embedded component based on the generalized extraction rule as given in equation (2.10).*

$$\lambda = \begin{cases} \lfloor B \rfloor : \lambda = \lambda_1 | \lambda_2 \\ \lceil B \rceil : \lambda = \lambda_3 | \lambda_4 \end{cases} \qquad (2.10)$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5 and for all λ, $0 \leq \lambda \leq 3$.

Step 4: For each 8 (eight) bits extraction, construct one alphabet/one primary (R/G/B) color component.

Step 5: Inverse Separable Discrete Hartley Transform (ISDHT) is applied on each 2 x 2 sub-matrix of embedded components to convert it back into spatial domain.

Step 6: Repeat steps 1 to 5 to complete the extraction of the watermark size (L), content (W) and the fabricated message digest (MD).

Step 7: Obtain 128 bits message digest MD' from the extracted watermark.

Step 8: Compare MD' with the extracted MD. If both are matches then the image is considered as authorized, else it is unauthorized.

Step 9: Stop.

2.2.1.3 Example

The carrier image is partitioned into 2 x 2 non-overlapping blocks in row major order. The 2 x 2 sub-matrices namely R_1, G_1 and B_1 corresponding to red, green and blue channels are considered for watermark fabrication. The 2 x 2 sub-matrices of pixel components are obtained as follows:

$$R_1 = \begin{bmatrix} 245 & 69 \\ 21 & 112 \end{bmatrix} G_1 = \begin{bmatrix} 92 & 202 \\ 7 & 51 \end{bmatrix} B_1 = \begin{bmatrix} 25 & 119 \\ 220 & 245 \end{bmatrix}$$

The pixel adjustment process adjusts the upper and lower bounds of the pixel components into the specified range as given in equation (2.8). In this example, the payload value is 3 bpB (B = 3) and the modified upper bound (UB) and the lower bound (LB) are derived as follows:

$$UB = \left(2^8 - 2^{\lceil 3 \rceil + 2}\right) = (2^8 - 2^5) = (256 - 32) = 224$$

$$LB = 2^{\lceil 3 \rceil + 2} = 2^5 = 32$$

Consequently, the pixel components outside the modified range are immediately adjusted. The 2 x 2 sub-matrices of pixel components are obtained as given below:

$$R_1 = \begin{bmatrix} 224 & 69 \\ 32 & 112 \end{bmatrix} \; G_1 = \begin{bmatrix} 92 & 202 \\ 32 & 51 \end{bmatrix} \; B_1 = \begin{bmatrix} 32 & 119 \\ 220 & 224 \end{bmatrix}$$

Separable Discrete Hartley Transform (SDHT) is applied on 2 x 2 sub-matrices of pixel components which yields the 2 x 2 sub-matrices in transform domain. The 2 x 2 sub-matrices i.e., T(R1), T(G1) and T(B1) are obtained as follows:

$$T(R_1) = \begin{bmatrix} 437 & 75 \\ 149 & 235 \end{bmatrix} \; T(G_1) = \begin{bmatrix} 377 & -129 \\ 211 & -91 \end{bmatrix} \; T(B_1) = \begin{bmatrix} 595 & -91 \\ -293 & -83 \end{bmatrix}$$

Considering the secret bit-stream as *"10100001011000001100111101101 1000001"*, three bits are successively fabricated (λ = 3) on each frequency/transformed component starting from LSB-2 toward higher order bit position. To avoid fractional pixel components, the two least significant bits (LSB-1 and LSB-0) of each transformed component are kept unaltered. Hence, the 2 x 2 sub-matrices of embedded components are obtained as follows:

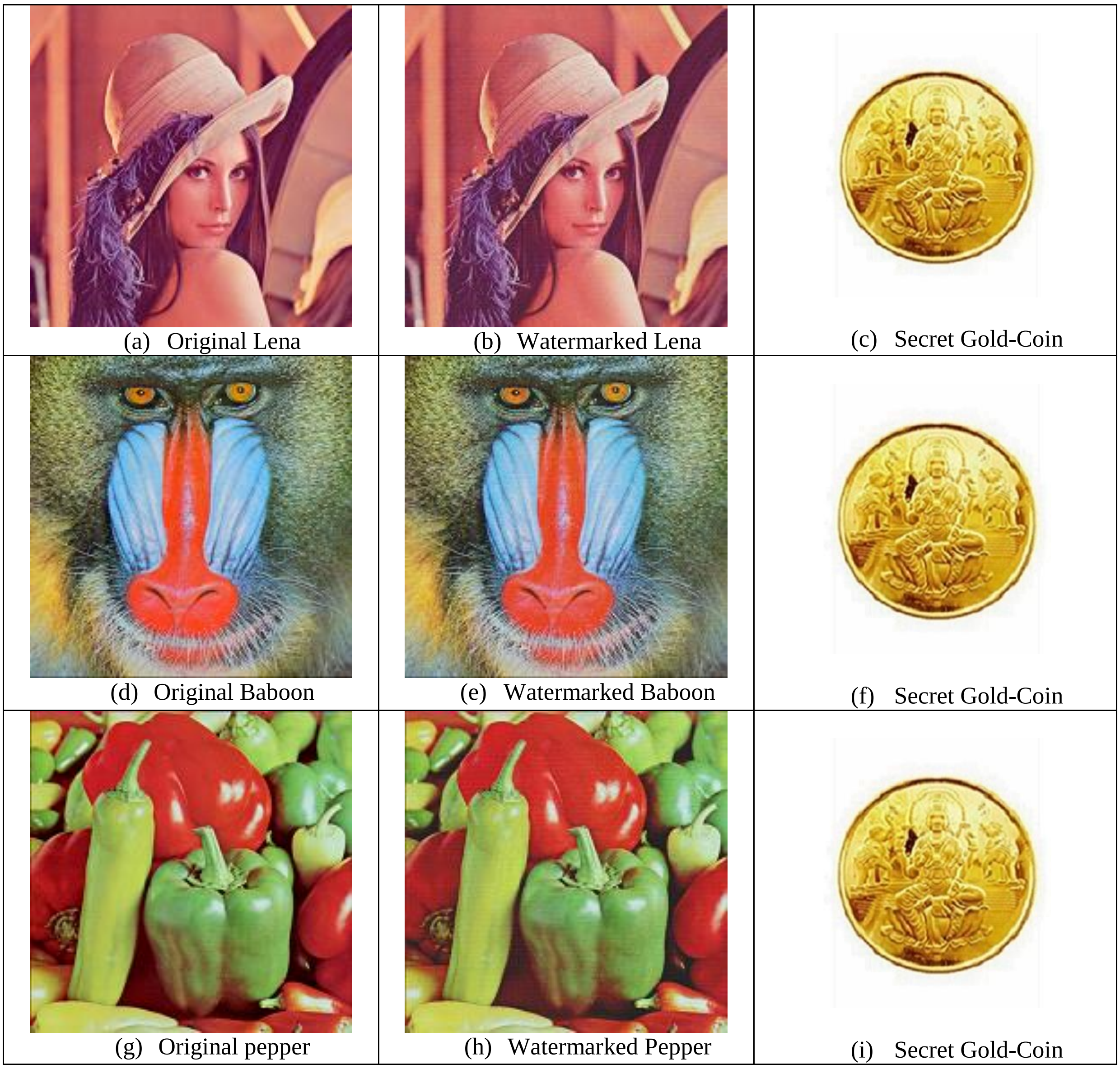
$$T'(R_1) = \begin{bmatrix} 437 & 67 \\ 137 & 239 \end{bmatrix} \; T'(G_1) = \begin{bmatrix} 353 & -153 \\ 207 & -95 \end{bmatrix} \; T'(B_1) = \begin{bmatrix} 603 & -91 \\ -289 & -83 \end{bmatrix}$$

On application of inverse Separable Discrete Hartley Transform (ISDHT) over the embedded components of each 2 x 2 sub-matrix, the pixel components of same dimension in spatial domain are obtained as follows:

$$R'_1 = \begin{bmatrix} 220 & 67 \\ 32 & 118 \end{bmatrix} \; G'_1 = \begin{bmatrix} 78 & 202 \\ 22 & 51 \end{bmatrix} \; B'_1 = \begin{bmatrix} 35 & 122 \\ 221 & 225 \end{bmatrix}$$

2.2.1.4 Results and Discussions

This section represents the results and discussions of the proposed 2 x 2 block based watermark fabrication using Separable Discrete Hartley Transform (SDHT) (WDHT_2x2). Analysis has been made in terms of payload and visual interpretation where, the interpretation of visual quality is analyzed in terms of peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD) and standard deviation error (SDE) respectively. Benchmark (BMP) images [130, 131] and the variable sizes of the secret watermark as given in fig. 1.1 have been taken to compute results. The 2 x 2 block based schemes such as WDHT_2x2, DPTHDI [88] and DGTDHS [129] are compared among themselves to validate the results. Fig. 2.1 depicts the fabrication of the "Gold-Coin" image into the carrier images such as "Lena", "Baboon" and "Pepper" respectively.



(a) Original Lena

(b) Watermarked Lena

(c) Secret Gold-Coin

(d) Original Baboon

(e) Watermarked Baboon

(f) Secret Gold-Coin

(g) Original pepper

(h) Watermarked Pepper

(i) Secret Gold-Coin

Fig 2.1. Cover, watermarked and the authenticating watermark images in the proposed WDHT_2x2 technique

The experiment is carried out for twenty different carrier images where, the perceptual distortion of the proposed technique (WDHT_2x2) is evaluated by means of the peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) with respect to 0.5, 1.0, 1.5, 2.0, 2.5 and 3 bpB of payloads. It is seen from table 2.1 that the quality of the watermarked images sharply decreases as payload value increases. In proposed WDHT_2x2, the minimum value of PSNR is 23.91 dB at 3 bpB of payload for the “Bobcat” image whereas, the maximum value of PSNR is 47.89 dB at 0.5 bpB of payload for the “San Diego” image. Since, the minimum value of PSNR is falling below the acceptable level (i.e., < 30 dB) [148], the PSNR obtained at 3 bpB for “Bobcat” (as well as majority of the benchmark images) ensured severe quality degradation. However, the variable payload feature that offers a spread from 0.5 to 3 bpB draws the key attention. The metric MSE is inversely proportional to the most popularly used metric PSNR and hence, lower MSE values ensured higher PSNR and vice-versa. The lowest MSE obtained for “San Diego” at a payload of 0.5 bpB is 1.05 and that of highest MSE is 263.77 for “Bobcat” at 3 bpB of payload. Analysis of other metrics such as the IF, SSIM and UIQ have been made to measures the rate of quality degradation, pixel’s independence and structural distortion applied over watermarked image with respect to the original image. Ideally, the IF, SSIM and UIQ are ranges from [0, 1] however, the WDHT_2x2 offered the following ranges of values for the payload variation of 0.5 to 3 bpB: [0.999962 (Airplane) - 0.962685 (Bobcat)], [0.999999 (Earth) - 0.830725 (Bobcat)] and [0.996779 (San Diego) - 0.399069 (Splash)] respectively. The closer the IF, SSIM and UIQ to one, watermarked image is more similar to the original one. The average values are also computed from twenty color images for the above mentioned metrics at variable payload to summarize the experimental results.

Table 2.1. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to variable payload in WDHT_2x2 technique

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	47.407641	1.181180	0.999927	0.999973	0.973613
	1.0	44.734959	2.185668	0.999863	0.996808	0.957305
	1.5	38.593962	8.988436	0.999454	0.996503	0.875848
	2.0	36.541658	14.418314	0.999131	0.984516	0.834402
	2.5	30.030791	64.565605	0.996077	0.973861	0.669147
	3.0	28.324232	95.643864	0.994025	0.929782	0.606651
Baboon	0.5	47.729109	1.096906	0.999942	0.999925	0.994269
	1.0	44.913212	2.097775	0.999889	0.998832	0.990719
	1.5	39.148333	7.911284	0.999583	0.997845	0.966640
	2.0	37.088897	12.711320	0.999329	0.993460	0.956270
	2.5	31.148233	49.917981	0.997376	0.986383	0.895912
	3.0	29.531476	72.432373	0.996174	0.970545	0.870364
Pepper	0.5	42.305850	3.823801	0.999629	0.983520	0.958071
	1.0	40.936259	5.241485	0.999503	0.979477	0.945116
	1.5	34.869948	21.187783	0.997973	0.966646	0.874000
	2.0	33.613212	28.298221	0.997345	0.953105	0.836292
	2.5	27.482119	116.109673	0.988896	0.925483	0.684764
	3.0	26.053328	161.342042	0.984791	0.878421	0.617658

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	46.941701	1.314952	0.999962	0.999998	0.928282
	1.0	44.438064	2.340312	0.999933	0.996223	0.897211
	1.5	38.219724	9.797334	0.999719	0.996396	0.765586
	2.0	36.145822	15.794218	0.999547	0.982159	0.717653
	2.5	30.690636	55.464707	0.998412	0.983815	0.578533
	3.0	28.645985	88.814070	0.997458	0.933969	0.526136
Sailboat	0.5	46.959658	1.309527	0.999934	0.999891	0.978710
	1.0	44.411404	2.354723	0.999881	0.997320	0.965363
	1.5	38.471228	9.246077	0.999535	0.996295	0.909655
	2.0	36.481162	14.620563	0.999263	0.986293	0.880747
	2.5	30.122051	63.223021	0.996841	0.971462	0.755196
	3.0	28.287314	96.460371	0.995167	0.935053	0.702769
Earth	0.5	47.743824	1.093195	0.999935	0.999999	0.988634
	1.0	44.900632	2.103861	0.999875	0.997560	0.979644
	1.5	39.233386	7.757855	0.999540	0.997468	0.933201
	2.0	36.985664	13.017089	0.999235	0.988533	0.902510
	2.5	30.612022	56.477839	0.996350	0.982323	0.770621
	3.0	28.613499	89.480921	0.994296	0.947542	0.714423
San Diego	0.5	47.897953	1.055079	0.999960	0.999999	0.996779
	1.0	44.978641	2.066408	0.999923	0.999086	0.994316
	1.5	39.515092	7.270614	0.999729	0.999129	0.981999
	2.0	37.253537	12.238454	0.999545	0.995656	0.973094
	2.5	32.373704	37.645224	0.998601	0.995920	0.934611
	3.0	30.229065	61.684179	0.997710	0.981338	0.909768
Splash	0.5	43.288726	3.049353	0.999687	0.986118	0.900533
	1.0	41.783047	4.312961	0.999566	0.980952	0.867927
	1.5	35.672954	17.611038	0.998262	0.976398	0.718510
	2.0	34.141955	25.054448	0.997635	0.957432	0.656771
	2.5	28.184595	98.769025	0.990366	0.944266	0.460136
	3.0	26.629736	141.288475	0.986530	0.881166	0.399069
Oakland	0.5	45.339798	1.901518	0.999877	0.999119	0.992304
	1.0	43.355342	3.002936	0.999816	0.997173	0.987753
	1.5	37.460817	11.668059	0.999273	0.995401	0.961088
	2.0	35.783624	17.167929	0.998966	0.987666	0.944141
	2.5	30.129916	63.108623	0.996090	0.982228	0.865241
	3.0	28.458099	92.740718	0.994436	0.952029	0.818944

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	47.654231	1.115982	0.999960	0.999999	0.967908
	1.0	44.776217	2.165003	0.999923	0.995607	0.945088
	1.5	38.947063	8.286554	0.999710	0.995872	0.845115
	2.0	36.742839	13.765640	0.999516	0.979483	0.795921
	2.5	31.456160	46.501211	0.998387	0.981572	0.657673
	3.0	29.342450	75.654600	0.997366	0.921404	0.592592
Anhinga	0.5	43.903242	2.647013	0.999797	0.999711	0.860142
	1.0	42.401148	3.740809	0.999713	0.998247	0.849703
	1.5	36.367019	15.009919	0.998851	0.997407	0.797650
	2.0	34.920894	20.940687	0.998395	0.990992	0.776635
	2.5	29.400718	74.646339	0.994279	0.984039	0.698034
	3.0	28.019087	102.605733	0.992125	0.953631	0.663011
Athens	0.5	45.872814	1.681894	0.999865	0.999999	0.944283
	1.0	43.364844	2.996373	0.999760	0.998599	0.929877
	1.5	38.219453	9.797945	0.999216	0.998829	0.853154
	2.0	35.987621	16.380162	0.998694	0.990649	0.818981
	2.5	30.953193	52.210884	0.995816	0.989149	0.707675
	3.0	28.784404	86.028015	0.993113	0.950024	0.653699
Bardowl	0.5	44.362844	2.381200	0.999755	0.998586	0.989803
	1.0	42.578614	3.591028	0.999633	0.997484	0.986911
	1.5	36.061477	16.10395	0.998336	0.987355	0.941708
	2.0	34.510640	23.015270	0.997638	0.981555	0.927946
	2.5	27.656020	111.552092	0.988462	0.936955	0.813320
	3.0	26.201594	155.926850	0.983959	0.914404	0.776026
Barnfall	0.5	46.665185	1.401398	0.999773	0.999921	0.991770
	1.0	44.139956	2.506598	0.999598	0.997916	0.986069
	1.5	38.594400	8.987528	0.998505	0.996648	0.954522
	2.0	36.559674	14.358627	0.997646	0.988851	0.934212
	2.5	30.884473	53.043607	0.990762	0.972873	0.828512
	3.0	28.906076	83.651308	0.985594	0.938496	0.777198
Butrfly	0.5	46.534886	1.444081	0.999897	0.999999	0.985351
	1.0	43.914079	2.640416	0.999812	0.998612	0.978703
	1.5	38.809115	8.553990	0.999392	0.998731	0.941048
	2.0	36.568079	14.330866	0.998981	0.991571	0.913953
	2.5	31.483835	46.205838	0.996654	0.989769	0.809569
	3.0	29.225577	77.718185	0.994378	0.960493	0.749024

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	46.235915	1.546994	0.999782	0.999991	0.734985
	1.0	43.641806	2.811251	0.999604	0.997918	0.727944
	1.5	37.892193	10.564792	0.998510	0.996908	0.687432
	2.0	35.717730	17.430398	0.997543	0.986606	0.669947
	2.5	25.366667	188.979043	0.973304	0.877017	0.543118
	3.0	23.918387	263.779832	0.962685	0.830725	0.513546
Bodie	0.5	44.134201	2.509922	0.999555	0.998801	0.967046
	1.0	42.586546	3.584476	0.999373	0.997278	0.962158
	1.5	35.832709	16.974985	0.996961	0.984118	0.917990
	2.0	34.429148	23.451222	0.995864	0.976460	0.902649
	2.5	26.951858	131.188110	0.976116	0.896976	0.755709
	3.0	25.633188	177.730313	0.967844	0.864174	0.719035
Bluheron	0.5	46.036184	1.619801	0.999801	0.999999	0.982729
	1.0	43.674114	2.790415	0.999658	0.998055	0.974714
	1.5	38.129906	10.002065	0.998775	0.997906	0.920053
	2.0	36.059434	16.111534	0.998023	0.988812	0.884807
	2.5	30.723297	55.049148	0.993257	0.984524	0.741818
	3.0	28.707744	87.560035	0.989271	0.949512	0.661762
Colomtn	0.5	45.203325	1.962220	0.999846	0.999846	0.973598
	1.0	43.393759	2.976490	0.999767	0.997948	0.965955
	1.5	37.577063	11.359887	0.999112	0.997345	0.927238
	2.0	35.914944	16.656584	0.998698	0.990165	0.907529
	2.5	30.366386	59.764277	0.995338	0.980833	0.823635
	3.0	28.757450	86.563596	0.993247	0.949064	0.784551
Desert	0.5	40.323197	6.036158	0.999055	0.993480	0.983388
	1.0	39.293657	7.650937	0.998812	0.991600	0.980269
	1.5	32.925454	33.153925	0.994908	0.973915	0.934260
	2.0	32.160639	39.538157	0.993994	0.967982	0.923934
	2.5	25.277376	192.904670	0.971089	0.898195	0.807483
	3.0	24.299964	241.592773	0.963904	0.870905	0.786353
Average case	0.5	45.627015	2.008608	0.999797	0.997944	0.954610
	1.0	43.410815	3.157996	0.999695	0.995635	0.943637
	1.5	37.527065	12.511700	0.998767	0.992356	0.885335
	2.0	35.680360	18.464985	0.998249	0.983097	0.857920
	2.5	29.564705	75.060850	0.991623	0.961882	0.740035
	3.0	27.828435	116.934900	0.988203	0.925634	0.692129

Fig. 2.2 illustrates the analysis of peak signal to noise ratio (PSNR) (in terms of dB) between proposed WDHT_2x2 and Varsaki et al.'s (Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] as well as Discrete Gould Transform based data hiding scheme (DGTDHS) [129]) schemes, respectively. Selection of block size as 2 x 2 yields the fixed payload values of 0.25 bpB and 1 bpB for DPTHDI [88] and DGTDHS [129], respectively. As a consequence, the PSNR values of DPTHDI [88] and DGTDHS [129] are also obtained with respect to the specified payload values which are fixed as well as considerably low. The WDHT_2x2 is proposed to support the variable nature in payload values for a spread from 0.5 to 3 bpB. The PSNR (dB) comparison has been made for five color images such as "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat", respectively. Compared to DPTHDI [88], the WDHT_2x2 ensured higher PSNR (dB) at 0.5 and 1 bpB of payloads for "Lena", 0.5, 1, 1.5, 2 and 2.5 bpB for "Baboon", 0.5 and 1 bpB of payloads for "Pepper", 0.5 and 1 bpB of payloads for "Airplane" and 0.5, 1, 1.5 and 2 bpB of payloads for "Sailboat" respectively. In comparison with DGTDHS [129], the WDHT_2x2 gives less PSNR (dB) at 1 bpB; however, it offers variation in payload values for the range [0.5 – 3 bpB]. The quality of the obtained watermarked images in WDHT_2x2 are well perceived for the spread of 0.5 to 2.5 bpB, however, the quality is severely degraded at 3 bpB of payload since the PSNR values of all the tested watermarked images lies below the acceptable level (i.e., < 30 dB) [148]. Therefore, in comparison with the original images, the diffusion of quality for the watermarked images decreases with respect to increasing values of payload.

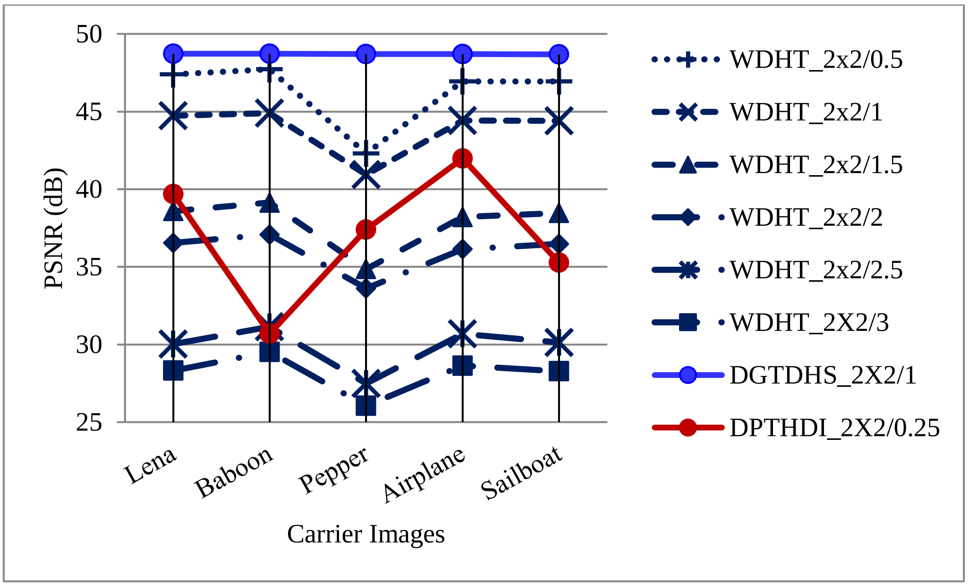


Fig. 2.2. Performance analysis of PSNR (dB) for variable payload based WDHT_2x2 and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Table 2.1 demonstrates that the maximum and minimum values of average peak signal to noise ratio (PSNR) for proposed WDHT_2x2 are 45.62 dB and 27.82 dB, respectively. The WDHT_2x2 is the variable payload based scheme that computes the average PSNR (dB) from the obtained PSNR (dB) values of twenty color images as shown in fig. 1.1. In general, the PSNR decreases while the embedding payload increases (and vice-versa). It is seen that the average PSNR (dB) values of WDHT_2x2 are greater than or equal to 30 dB [148] for the payload range [0.5 - 2 bpB]. On the contrary, the average PSNR (dB) values obtained at 2.5 and 3 bpB become lesser than the acceptable quality value (i.e., < 30 dB) [148]. On application of the quality improvement, the quality of the watermarked images can be improved as evident from the results summarized in chapter 7 and chapter 8. Fig. 2.3 depicts the comparative analysis of average PSNR (dB) for the variable payload based WDHT_2x2 and the fixed payload based DPTHDI [88] as well as DGTDHS [129], respectively. The DPTHDI [88] computes the average PSNR (dB) of 37.40 dB as the average of PSNR (dB) values of “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images at 0.25 bpB of payload. In contrast to DPTHDI [88], the average PSNR of WDHT_2x2 ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads. The DGTDHS [129] computes the average PSNR (dB) of 48.70 dB as the average of PSNR (dB) values of “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” images at 1 bpB of payload. Compared to DGTDHS [129], the WDHT_2x2 is lacking with the average PSNR (dB) at 1 bpB of payload however, the payload variation in the range of 0.5 to 3 bpB specifies the significance of the technique.

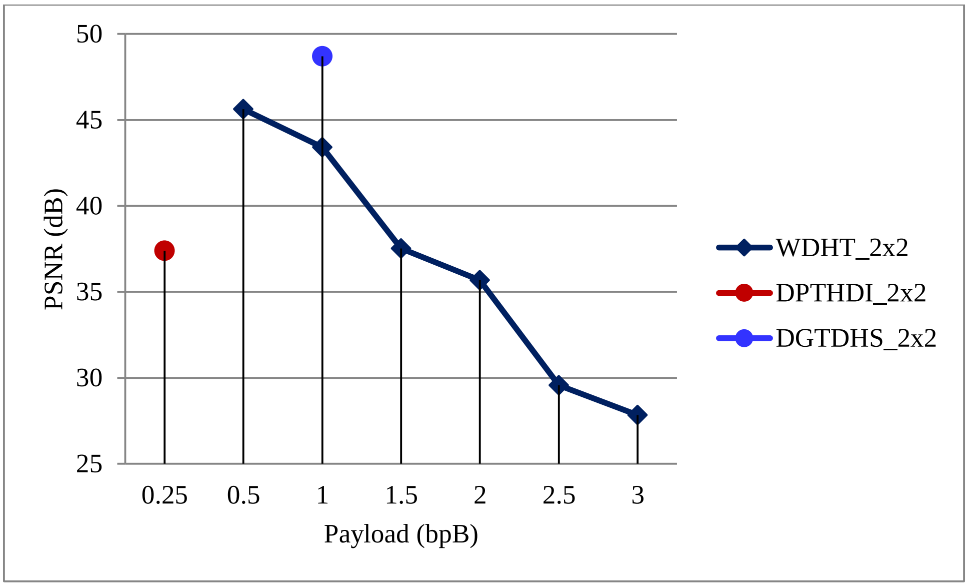


Fig. 2.3. Graphical representation of variation of average PSNR (dB) with respect to payload for WDHT_2x2 and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

In fig. 2.4, a standard deviation (SD) analysis has been made to ensure that the proposed WDHT_2x2 embeds data in such a way that one cannot easily detect the differences between U x V source and watermarked images. To identify the change in the watermarked image with respect to the original image, the standard deviation (SD) analysis has been made for five color images such as "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat", respectively. In this pretext, the standard deviation (SD) values for a 24-bit watermarked image is nothing but the average of standard deviation values corresponding to red, green and blue channels, respectively. As the payload of 0 bpB designates the original image prior to embedding, the standard deviation (SD) at that payload specifies the standard deviation (SD) of the original image. It is also seen from fig. 2.4 that the standard deviation (SD) values remains constant up to an average payload of 2 bits per Byte (bpB) except for the "Pepper" image since the standard deviation (SD) of "Pepper" image is gradually decreasing starting from 0.5 bpB of payload. It is also observed that while the payload increases from 2 bpB, the standard deviation (SD) decreases in usual case such as for "Lena", "Baboon", "Pepper" and "Sailboat" images and the standard deviation (SD) increases for "Airplane" image.

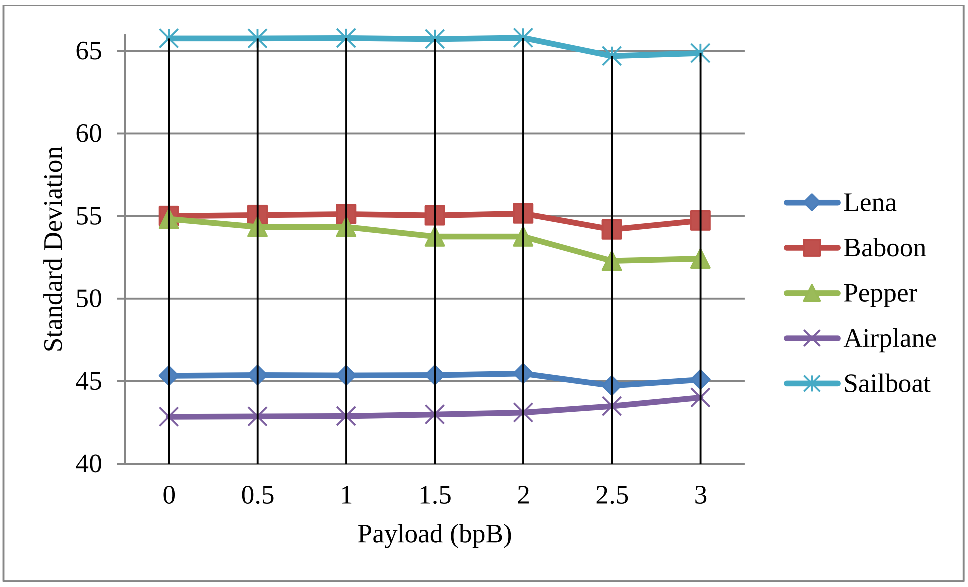


Fig. 2.4. Graphical representation of standard deviation (SD) for WDHT_2x2 with respect to 0, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

Fig. 2.5 illustrates that the standard deviation error (SDE) sharply increases as the payload increases. In effect, the watermarked images are gradually deviating from the original image with respect to increasing payload. The error is very low up to 2 bpB and afterward it is increasing. The error is comparatively high for "Pepper" image and it produces significant deviation in the watermarked image with respect to the original image.

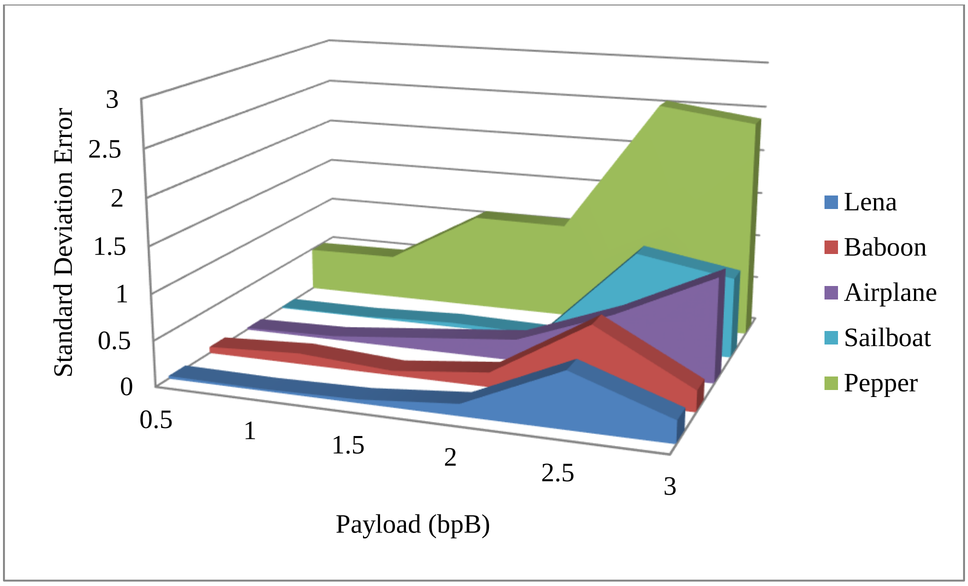


Fig. 2.5. Graphical representation of standard deviation error (SDE) for WDHT_2x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

2.2.2 1 x 2 Block based Watermark Fabrication

One dimensional Discrete Hartley Transform (1D-DHT) [132] takes a real sequence as input and generates a real sequence as output. The 1D-DHT is applied over 1 x N block of a carrier image (which is actually considered as the 1D-sequence of N pixel components) to convert the pixel components into the transformed components as given in equation (2.11).

$$P_H(\delta) = \sum_{\alpha=0}^{N-1} p(\alpha) cas\left(\frac{2\pi\delta\alpha}{N}\right) \tag{2.11}$$

where, δ varies from 0 to N-1.

The variable u is the frequency variable corresponding to the spatial domain variable α where, p(α) represents the intensity value of the pixel component in spatial domain. The sequence *cas* is defined in equation (2.12).

$$cas\left(\frac{2\pi\delta\alpha}{N}\right) = \cos\left(\frac{2\pi\delta\alpha}{N}\right) + \sin\left(\frac{2\pi\delta\alpha}{N}\right) \tag{2.12}$$

Similarly, the inverse transformation is applied over 1 x N block of transformed components i.e., the 1D-sequence of N transformed components to convert it back into the spatial domain as given in equation (2.13).

$$p(\alpha) = \frac{1}{N} \sum_{\delta=0}^{N-1} P_H(\delta) cas\left(\frac{2\pi\delta\alpha}{N}\right) \tag{2.13}$$

where, α varies from 0 to N-1.

The formulations of one dimensional Discrete Hartley Transform (1D-DHT) for a 1 x 2 sub-image block (or pair of pixel components) have been expressed in equation (2.14).

$$P_H(\delta) = \sum_{\alpha=0}^{1} p(\alpha)(-1)^{\delta\alpha} \tag{2.14}$$

where, the variable δ is the frequency variable corresponding to the frequency component in transformed domain that varies from 0 to 1.

One dimensional Inverse Discrete Hartley Transform (1D-IDHT) is applied on each pair of transformed components to obtain the pair of pixel components in spatial domain as given in equation (2.15).

$$p(\alpha) = \frac{1}{2}\sum_{\delta=0}^{1} P_H(\delta)(-1)^{\delta\alpha} \tag{2.15}$$

where, the variable α is the spatial domain variable corresponding to the pixel intensity value in spatial domain that varies from 0 to 1.

In contrast to Separable Discrete Hartley Transformation (SDHT), the one-dimensional Discrete Hartley Transformation (2D-DHT) takes 1 x 2 as the window size for embedding watermark information. The reduction of window size from 2 x 2 to 1 x 2 ensured the fabrication of secret bits up to the LSB-3 (instead of LSB-4) of the transformed component. As a consequence, the quality distortion of the watermarked image is significantly reduced for the 1 x 2 window based watermarking by keeping the embedding payload unchanged.

Proposed WDHT_1x2 has been discussed in the following sections. The algorithm for insertion, the algorithm for extraction and an example are described in section 2.2.2.1, 2.2.2.2 and 2.2.2.3 respectively. Simulation results and discussions are given in section 2.2.2.4.

2.2.2.1 Insertion

The carrier image is decomposed into 1 x 2 non-overlapping blocks where, each block is nothing but the pair of pixel components considered in a row major order. Pixel components are adjusted prior to embedding to avoid overflow and underflow. Each 1 x 2 sub-image block or pair of pixel components of red/green/blue channel are subsequently converted into transform domain through one dimensional Discrete Hartley Transform (1D-DHT). Based on the different perceptibility of human eye on red/green/blue channel, variable numbers of bits are embedded into the transformed components by maintaining the trade-off between payload and image quality. The message digest computed from the watermark, the size and the content of the watermark constitutes a secret bit-stream from which requisite bits are successively fabricated into the transformed components starting from the 1st bit position of the least significant part (i.e., LSB-1) toward higher order bit position. The embedding is done for 0.5 to 3 bpB of payload which ensured that a transformed component is capable of fabricating with an average of three watermark bits. The unaltered least significant bit (LSB-0) of each transformed component yields the non-fractional pixel component during inverse transform. One dimensional Inverse Discrete Hartley Transform (1D-IDHT) is applied on embedded 1 x 2 blocks or pair of embedded components to obtain the pixel components. This process is repeated till the message digest, size and content of the watermark are embedded into the transformed components and generates the watermarked image in spatial domain.

Algorithm 2.3:

Input: Carrier/cover image (I) and authenticating watermark image (W).

Output: Watermarked image (I′).

Method: One dimensional Discrete Hartley Transform (1D-DHT) is applied over each 1 x 2 sub-block of the cover image to convert the pixel components into the transformed components. Watermark (along with a message digest) are fabricated in transform domain. The scheme offers variable payload, less quality distortion and improved security. The complete steps of embedding are as follows:

Step 1: Obtain a message digest (MD) from the authenticating watermark.

Step 2: The authenticating watermark size (in bits) is obtained by embedding the watermark bits into the three sub-matrices of the U x V color image as given in equation (2.16).

$$W_{size} = B \times \left(3 \times (U \times V)\right) - (MD + L) \qquad (2.16)$$

where, B, MD and L represents the embedding payload in terms of bits per Byte, the message digest computed from the watermark and the header information corresponding to the size of the watermark respectively. The MD and L are consisting of 128 and 32 bits while the usual values of B are 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively.

Step 3: The cover image (I) is partitioned into 1 x 2 non-overlapping blocks in row major order. Each 1 x 2 block is consisting of a pair of pixel components p_i and p_{i+1} of red/green/blue channel where, the values of i lies in the range $0 \leq i \leq 1$.

Step 4: To avoid the occurrence of overflow and underflow, the pixel components are adjusted by re-defining the upper and lower bounds of pixel component (p) for a payload value of B bits per byte as given in equation (2.17).

$$p = \begin{cases} \left(2^8 - 2^{\lceil B \rceil + 1}\right) : p \geq \left(2^8 - 2^{\lceil B \rceil + 1}\right) \\ 2^{\lceil B \rceil + 1} : p \leq 2^{\lceil B \rceil + 1} \end{cases} \tag{2.17}$$

Step 5: Apply one dimensional Discrete Hartley Transform (1D-DHT) on 1 x 2 sub-matrices of pixel components of red/green/blue channel to obtain the pair of frequency components f_i and f_{i+1} respectively.

Step 6: λ_1 /λ_2 bits from the secret bit-stream (corresponding to the message digest (MD), size (L) and the content (W) of the watermark) are fabricated on first/second transformed component starting from the 1st bit position of the least significant part (i.e., LSB-1) toward higher order bit position. Based on the perceptibility of human eye at different channel (C), more bits are embedded in blue (b) channel; whereas, fewer bits are fabricated on red (r) and green (g) channels. The generalized form of λ_1 /λ_2 bits of secret information fabrication on each pair of transformed components for the payload of B bits per Byte (bpB) are derived from equation (2.18).

$$(\lambda_1, \lambda_2) = \begin{cases} \lfloor B \rfloor, \lceil B \rceil : C = r \\ \lfloor B \rfloor, \lfloor B \rfloor : C = g \\ \lceil B \rceil, \lceil B \rceil : C = b \end{cases} \tag{2.18}$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5; for all (λ_1, λ_2), $0 \leq \lambda_1, \lambda_2 \leq 4$.

Step 7: One dimensional Inverse Discrete Hartley Transform (1D-IDHT) is applied on each 1 x 2 sub-matrix / pair of embedded *components to re-obtain the modified pair of pixel components in spatial domain.*

Step 8: Repeat steps 3 to 7 till the watermark size, content and the message digest (MD) are embedded. The repetitive actions of the embedding process yield the watermarked image (I').

Step 9: Stop.

2.2.2.2 Extraction

The recipient partitions the watermarked image into 1 x 2 non-overlapping blocks in a sliding window manner. One dimensional Discrete Hartley Transform (1D-DHT) is applied over each 1 x 2 sub-image block or pair of pixel components of red/green/blue channel which yields the 1 x 2 sub-block of transformed components. Secret bits corresponding to the watermark size, content and message digest (MD) are extracted from the transformed components in variable proportion. The process is repeated until and unless the fabricated watermark and the message digest (MD) is re-constructed. Message digest MD′ is re-computed from the extracted watermark which in turn is compared with the extracted message digest MD for authentication. The extraction procedure is described in algorithm 2.4.

Algorithm 2.4:

Input: Watermarked image (I′).

Output: The authenticating watermark (W) and the 128 bits message digest (MD).

Method: The watermark (along with a message digest) is extracted from the 1 x 2 transformed blocks of the watermarked image (I′) in one dimensional Discrete Hartley Transform (1D-DHT) domain. Another message digest (MD') is re-computed which in turn is compared with the extracted message digest for authentication. The extraction process is elaborated as follows:

Step 1: The watermarked image (I') is decomposed into 1 x 2 non-overlapping blocks in row major order. Each 1 x 2 block is consisting of pair of pixel components p_i and p_{i+1} of red/green/blue channel where, for all i, $0 \leq i \leq 1$.

Step 2: Apply one dimensional Discrete Hartley Transform (1D-DHT) on 1 x 2 sub-matrices of red/green/blue channel to obtain the pair of transformed components f_i and f_{i+1} respectively.

Step 3: λ_1 / λ_2 bits of the secret bit-stream are extracted from first/second embedded component starting from the 1st bit position of the *least significant part (i.e., LSB-1) toward higher order bit position based on the extraction rule given in equation (2.19).*

$$(\lambda_1, \lambda_2) = \begin{cases} \lfloor B \rfloor, \lceil B \rceil : C = r \\ \lfloor B \rfloor, \lfloor B \rfloor : C = g \\ \lceil B \rceil, \lceil B \rceil : C = b \end{cases} \tag{2.19}$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5; for all (λ_1, λ_2), $0 \leq \lambda_1, \lambda_2 \leq 4$.

Step 4: For each 8 (eight) bits extraction, construct one alphabet/one primary (r/g/b) color component.

Step 5: One dimensional Inverse Discrete Hartley Transform (1D-IDHT) is applied on each 1 x 2 sub-matrix or pair of embedded components to re-obtain the pixel components in spatial domain.

Step 6: Repeat steps 1 to 5 to complete the extraction of the watermark size, content and the fabricated message digest (MD) respectively.

Step 7: Obtain 128 bits message digest MD' from the extracted watermark.

Step 8: Compare MD' with the extracted MD. If both are matches then the image is authorized, else unauthorized.

Step 9: Stop.

2.2.2.3 Example

Since, the carrier image is partitioned into 1 x 2 non-overlapping blocks, three 1 x 2 sub-blocks of red (R_1), green (G_1) and blue (B_1) channels are considered for embedding secret bits. The 1 x 2 sub-matrices of pixel components are as follows:

$$R_1 = [245\ 69]\quad G_1 = [92\ 202]\quad B_1 = [11\ 110]$$

A pre-embedding pixel adjustment has been incorporated to handle the overflow and underflow. It is done by re-defining the upper and lower bounds of the pixel components as given in equation (2.17). In this example, the payload value is three (i.e., B = 3) and the modified upper bound (UB) and the lower bound (LB) are computed as follows:

$$UB = \left(2^8 - 2^{\lceil 3 \rceil + 1}\right) = (2^8 - 2^4) = (256 - 16) = 240$$

$$LB = 2^{\lceil 3 \rceil + 1} = 2^4 = 16$$

Therefore, pixel components, falling outside this range, are immediately adjusted to reset the upper and lower bounds based on the pixel adjustment method. The 1 x 2 sub-matrices of pixel components are as here under:

$$R_1 = [240\ 69] \quad G_1 = [92\ 202] \quad B_1 = [16\ 110]$$

One dimensional Discrete Hartley Transform (1D-DHT) is applied on 1 x 2 sub-matrices or pairs of pixel components to generate the transformed components of identical block size as follows:

$$T(R_1) = [309\ 171] \quad T(G_1) = [294\ -110] \quad T(B_1) = [126\ -94]$$

Embedding of authenticating watermark bit-stream *"101000010110000011"* into the transformed components is accomplished by the embedding rule given in equation (2.18). In this example, three bits are fabricated (i.e., $\lambda_1 = 3$, $\lambda_2 = 3$) on each transformed component starting from LSB-1 toward the higher order bit position. Hence, the 1 x 2 sub-matrices of embedded components are as follows:

$$T'(R_1) = [315\ 161] \quad T'(G_1) = [292\ -102] \quad T'(B_1) = [112\ -92]$$

One dimensional inverse Discrete Hartley Transform (1D-IDHT) is responsible for converting the 1 x 2 sub-matrices of embedded components into the 1 x 2 sub-matrices of pixel components in spatial domain. The 1 x 2 sub-matrices of pixel components are re-computed as follows:

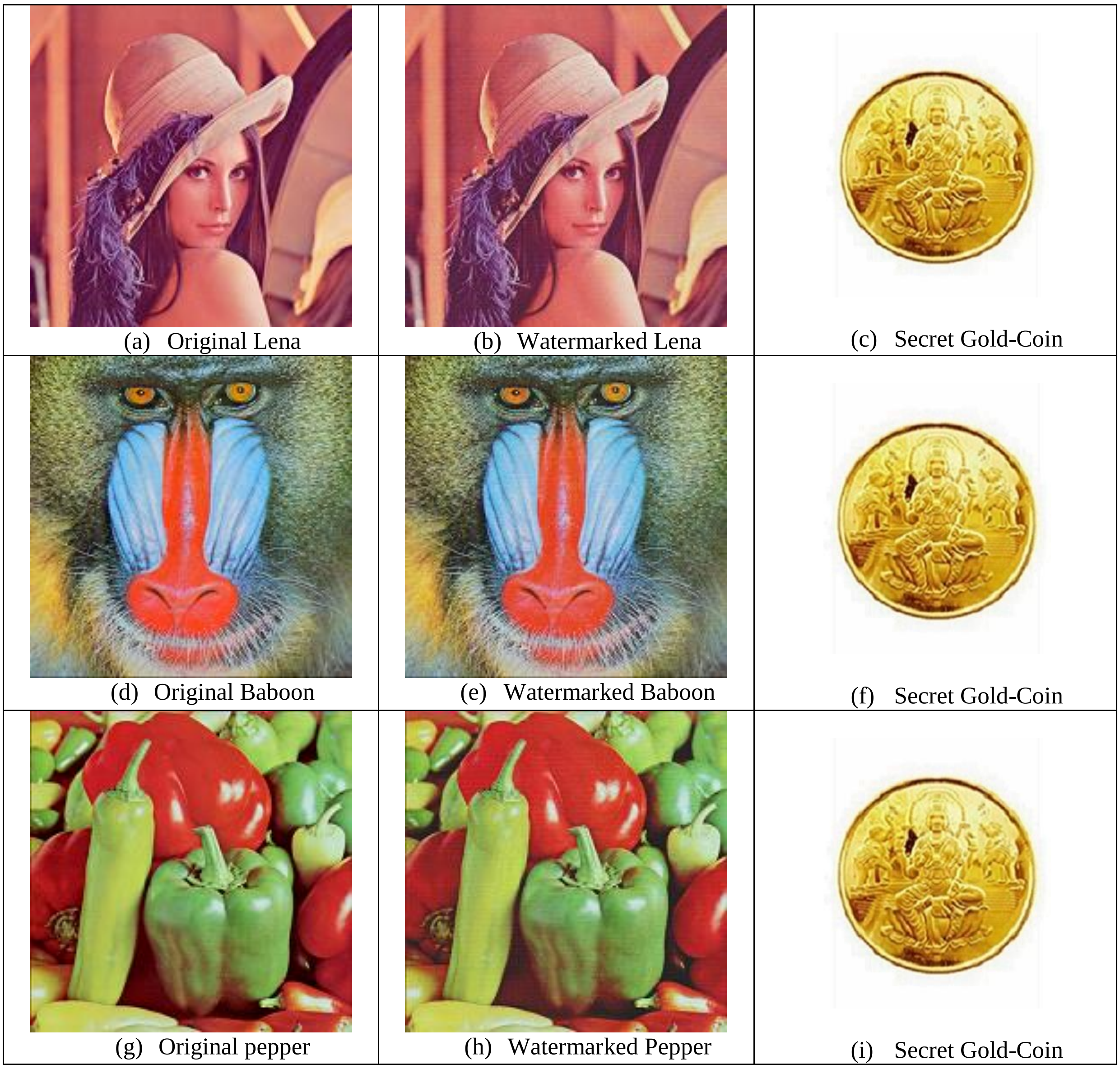
$$R'_1 = [238\ 77] \quad G'_1 = [95\ 197] \quad B'_1 = [20\ 102]$$

The modified pixel components are non-fractional since the least significant bit (i.e., LSB-0) of each transformed component is kept unaltered.

2.2.2.4 Results and Discussions

One dimensional Discrete Hartley Transform (1D-DHT) based watermarking (abbreviated as WDHT_1x2) is extensively analyzed and compared with respect to the obtained results. The scheme is analyzed with respect to the widely acceptable metrics such as the peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD), standard deviation error (SDE) and most importantly the payload. Benchmark (BMP) images [130, 131] and the secret watermark of varying sizes have been considered as shown in fig. 1.1. The WDHT_1x2 is compared against 2 x 2 block based watermarking using Separable Discrete

Hartley Transform (SDHT) (WDHT_2x2) and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes respectively. The WDHT_1x2 is proposed to offer variable payload by retaining perceptible quality in the watermarked images. Fig. 2.6 depicts the different states on embedding the watermark (i.e., the "Gold-Coin" image) into the original images viz. "Lena", "Baboon" and "Pepper" respectively.



(a) Original Lena

(b) Watermarked Lena

(c) Secret Gold-Coin

(d) Original Baboon

(e) Watermarked Baboon

(f) Secret Gold-Coin

(g) Original pepper

(h) Watermarked Pepper

(i) Secret Gold-Coin

Fig 2.6. Cover, watermarked and the authenticating watermark images in the proposed WDHT_1x2 technique

Simulation results of WDHT_1x2 is computed as well as analyzed against twenty benchmark images [130, 131] as shown in fig. 1.1. The average values of standard quality metrics with respect to varying payload are also summarized in table 2.2. The maximum value of peak signal to noise ratio (PSNR) for WDHT_1x2 is 51.07 dB for "San Diego" image at 0.5 bpB and that of minimum PSNR obtained is 25.19 dB at 3 bpB for the “Desert” image respectively. The PSNR value of “Desert” (along with most of the watermarked images) at 3 bpB falls below the acceptable level of PSNR value (i.e., < 30 dB) and hence, the opaque version of the watermarked image is obtained as soon as the payload exceeds 2.5 bpB [148]. Hence, the 1 x2 block based scheme (WDHT_1x2) ensured consistent results up to 2.5 bpB of payload. Regardless of the fact, the WDHT_1x2 is effective for the ability of providing variable payload (0.5 – 3 bpB). The minimum and maximum values of mean squared error (MSE) are computed at 0.5 and 3 bpB respectively. The MSE values are belonging to the range [0.50 (San Diego) - 196.66 (Desert)]. The image fidelity (IF) ranges between [0.999992 (Airplane) - 0.993754 (Desert)], structural similarity index (SSIM) ranges between [0.999871 (San Diego), 0.923728 (Splash)) and universal image quality index (UIQ) ranges between [0.999255 (San Diego) - 0.628142 (Splash)] respectively. It is apparent from table 2.2 that the values of IF, SSIM and UIQ lie between 0 and 1. These values are also closest to one which ensured high similarity between the host and the watermarked images.

Table 2.2. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to variable payload in WDHT_1x2 technique

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	50.968356	0.520289	0.999967	0.999503	0.989080
	1.0	44.315753	2.407160	0.999822	0.996745	0.968299
	1.5	42.537117	3.625506	0.999774	0.996334	0.938571
	2.0	37.289916	12.136367	0.999118	0.986724	0.882305
	2.5	35.332709	19.046248	0.998854	0.984757	0.815090
	3.0	30.142997	62.918830	0.995469	0.947837	0.721272
Baboon	0.5	51.032210	0.512695	0.999971	0.999851	0.997658
	1.0	44.357682	2.384031	0.999861	0.998932	0.991356
	1.5	42.720524	3.475584	0.999810	0.998727	0.985647
	2.0	37.355060	11.955679	0.999310	0.995094	0.969004
	2.5	35.720061	17.421047	0.999055	0.993986	0.948909
	3.0	30.239386	61.537757	0.996447	0.977247	0.906588
Pepper	0.5	48.386133	0.942902	0.999872	0.994562	0.984568
	1.0	41.629856	4.467810	0.999339	0.987105	0.961077
	1.5	40.381162	5.956129	0.999283	0.981620	0.935833
	2.0	34.392708	23.648824	0.996598	0.963228	0.877135
	2.5	33.313274	30.321655	0.996391	0.957022	0.824742
	3.0	27.083250	127.278568	0.981654	0.904356	0.722901

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Airplane	0.5	50.821815	0.538144	0.999984	0.999355	0.969061
	1.0	43.958937	2.613283	0.999928	0.996074	0.914369
	1.5	42.275839	3.850316	0.999891	0.995475	0.864764
	2.0	36.723077	13.828422	0.999621	0.984138	0.788756
	2.5	34.891348	21.083637	0.999405	0.981564	0.699060
	3.0	29.620510	70.962565	0.998053	0.943139	0.622412
Sailboat	0.5	50.898925	0.528673	0.999972	0.999564	0.989218
	1.0	44.072635	2.545756	0.999866	0.997083	0.962976
	1.5	42.481783	3.671994	0.999809	0.996780	0.949952
	2.0	36.929953	13.185148	0.999311	0.988160	0.904697
	2.5	35.428935	18.628884	0.999035	0.986738	0.866691
	3.0	28.947819	82.851126	0.995673	0.947036	0.785870
Earth	0.5	51.016963	0.514498	0.999976	0.999637	0.994575
	1.0	44.341154	2.393122	0.999889	0.997574	0.980594
	1.5	42.670544	3.515814	0.999823	0.997315	0.966918
	2.0	37.319000	12.055362	0.999430	0.989919	0.923371
	2.5	35.602244	17.900118	0.999092	0.988728	0.879937
	3.0	30.162737	62.633494	0.997032	0.961334	0.789260
San Diego	0.5	51.071997	0.508019	0.999983	0.999790	0.998316
	1.0	44.394806	2.363740	0.999927	0.998744	0.992946
	1.5	42.749797	3.452237	0.999884	0.998652	0.990210
	2.0	37.445656	11.708862	0.999634	0.994981	0.972938
	2.5	35.864106	16.852706	0.999432	0.994460	0.963497
	3.0	30.647925	56.012873	0.998246	0.979213	0.914195
Splash	0.5	50.681726	0.555786	0.999953	0.999244	0.955520
	1.0	43.919007	2.637421	0.999724	0.994877	0.888930
	1.5	41.521247	4.580950	0.999598	0.983882	0.810769
	2.0	36.469595	14.659557	0.998479	0.968511	0.713837
	2.5	34.149395	25.011561	0.997864	0.961258	0.624420
	3.0	29.274810	76.842114	0.992169	0.911605	0.532769
Oakland	0.5	49.694129	0.697697	0.999960	0.999125	0.995369
	1.0	43.872835	2.665611	0.999886	0.996394	0.979645
	1.5	41.646324	4.450901	0.999751	0.995626	0.976727
	2.0	36.809327	13.556502	0.999408	0.986392	0.928761
	2.5	34.624001	22.422303	0.998739	0.983572	0.917239
	3.0	29.653346	70.428053	0.996942	0.948638	0.819042

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	50.983241	0.518508	0.999983	0.999153	0.983734
	1.0	44.105738	2.526425	0.999930	0.994818	0.933596
	1.5	42.569179	3.598838	0.999879	0.994271	0.914131
	2.0	37.014419	12.931187	0.999635	0.980122	0.830084
	2.5	35.415863	18.685040	0.999371	0.977215	0.769184
	3.0	29.901185	66.521458	0.998129	0.924229	0.674356
Anhinga	0.5	48.873701	0.842772	0.999932	0.999760	0.929419
	1.0	42.007021	4.096172	0.999663	0.998551	0.905496
	1.5	40.329498	6.027407	0.999521	0.998209	0.833654
	2.0	35.454961	18.517578	0.998490	0.992266	0.797359
	2.5	33.761574	27.347831	0.997843	0.990530	0.762260
	3.0	29.012930	81.618251	0.993366	0.961755	0.702588
Athens	0.5	48.885059	0.840571	0.999933	0.999815	0.974626
	1.0	41.476182	4.628733	0.999623	0.998833	0.948711
	1.5	40.114283	6.333620	0.999498	0.998599	0.911380
	2.0	35.267831	19.332911	0.998439	0.992920	0.868069
	2.5	33.829023	26.926383	0.997863	0.991621	0.805397
	3.0	28.799418	85.731123	0.993066	0.959034	0.728978
Bardowl	0.5	48.212958	0.981260	0.999897	0.999280	0.996669
	1.0	41.079399	5.071545	0.999459	0.996209	0.985015
	1.5	39.973657	6.542060	0.999320	0.996240	0.980113
	2.0	33.542081	28.765516	0.996940	0.977775	0.937328
	2.5	32.716624	34.787075	0.996379	0.977814	0.918046
	3.0	25.842304	169.375249	0.981950	0.926170	0.843254
Barnfall	0.5	50.558089	0.571835	0.999880	0.999616	0.996691
	1.0	43.314525	3.031291	0.999283	0.997651	0.984759
	1.5	41.709875	4.386244	0.999139	0.997377	0.977340
	2.0	36.338066	15.110321	0.996534	0.988118	0.942340
	2.5	34.847977	21.29524	0.995871	0.986967	0.914824
	3.0	28.652603	88.678848	0.979431	0.935100	0.814199
Butrfly	0.5	49.557670	0.719968	0.999941	0.999787	0.992709
	1.0	42.293752	3.834467	0.999660	0.998449	0.979110
	1.5	40.820446	5.383140	0.999576	0.998329	0.968090
	2.0	35.927858	16.607126	0.998577	0.992908	0.935712
	2.5	34.400994	23.603747	0.998181	0.992123	0.897354
	3.0	29.008346	81.704449	0.992991	0.965279	0.814572

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	49.193921	0.782867	0.999873	0.999582	0.832578
	1.0	41.788158	4.307889	0.999232	0.996838	0.819441
	1.5	40.400107	5.930203	0.999068	0.996688	0.716497
	2.0	35.108020	20.057570	0.996527	0.984934	0.690786
	2.5	33.665126	27.961963	0.995692	0.983731	0.659271
	3.0	25.511152	182.795354	0.967326	0.902438	0.586585
Bodie	0.5	49.352944	0.754720	0.999857	0.999536	0.982208
	1.0	42.450208	3.698788	0.999244	0.997356	0.970942
	1.5	40.960603	5.212187	0.999054	0.997133	0.955883
	2.0	34.616443	22.461357	0.995435	0.978833	0.920251
	2.5	33.590372	28.447433	0.994708	0.978396	0.897309
	3.0	26.257597	153.929068	0.968221	0.898903	0.794482
Bluheron	0.5	49.534888	0.723754	0.999910	0.999656	0.992862
	1.0	42.793665	3.417541	0.999586	0.997758	0.975552
	1.5	41.012693	5.150044	0.999364	0.997464	0.960005
	2.0	36.594827	14.242872	0.998272	0.990711	0.916870
	2.5	34.722777	21.918087	0.997299	0.989550	0.863085
	3.0	29.422192	74.278148	0.990999	0.956767	0.760051
Colomtn	0.5	50.031172	0.645599	0.999950	0.999658	0.999658
	1.0	43.198432	3.113416	0.999763	0.997846	0.976301
	1.5	41.406643	4.703444	0.999637	0.997556	0.957916
	2.0	36.271521	15.343631	0.998831	0.990770	0.930232
	2.5	34.635475	22.363140	0.998274	0.989309	0.895806
	3.0	29.190476	78.348861	0.994031	0.950831	0.818775
Desert	0.5	45.674444	1.760499	0.999694	0.997437	0.994520
	1.0	39.298184	7.642965	0.998536	0.992678	0.985789
	1.5	37.977534	10.359217	0.998280	0.991239	0.977659
	2.0	32.495462	36.604468	0.993107	0.975148	0.952559
	2.5	31.197683	49.352825	0.991954	0.968253	0.925622
	3.0	25.194083	196.640085	0.962992	0.903908	0.848947
Average case	0.5	49.771520	0.723053	0.999924	0.999196	0.977452
	1.0	42.933400	3.492358	0.999611	0.996526	0.955245
	1.5	41.312940	5.010292	0.999498	0.995376	0.928603
	2.0	35.968290	17.335460	0.998085	0.985083	0.884120
	2.5	34.385480	24.568850	0.997565	0.982880	0.842387
	3.0	28.628250	96.554310	0.988709	0.940241	0.760055

The peak signal to noise ratio (PSNR) of the proposed watermarking technique (WDHT_1x2) is extensively analyzed as well as compared with the obtained PSNR values of Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129], respectively. The comparison has been made for five color images such as "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively. The PSNR values for DPTHDI [88] and DGTDHS [129] are obtained at 0.25 and 1 bpB respectively. Both schemes exploit the 2 x 2 block as the window size and treat the offered (fixed) payload values as low. On the contrary, the WDHT_1x2 computes the average PSNR for a spread from 0.5 to 3 bpB however, an acceptable visual imperceptibility (i.e., average PSNR ≥ 30 dB) is achieved for the payload range (0.5 – 2.5 bpB) [148]. The quality degradation of the watermarked images at 3 bpB is severe since the values of PSNR are less than 30 dB however, the quality improvement schemes may further be used to enhance the quality of the watermarked images by keeping the fabricated watermark intact. Compared to DPTHDI [88], the WDHT_1x2 offered equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for "Lena", 0.5, 1, 1.5, 2 and 2.5 bpB for "Baboon", 0.5, 1 and 1.5 bpB of payloads for "Pepper", 0.5, 1 and 1.5 bpB of payloads for "Airplane" and 0.5, 1, 1.5 and 2 bpB of payloads for "Sailboat" respectively. In contrast to DGTDHS [129], the PSNR obtained at 1 bpB in the WDHT_1x2 is comparatively less. However, the WDHT_1x2 could be an effective solution of embedding secret information since the proposed technique (WDHT_1x2) supports payload variation from 0.5 to 3 bpB.

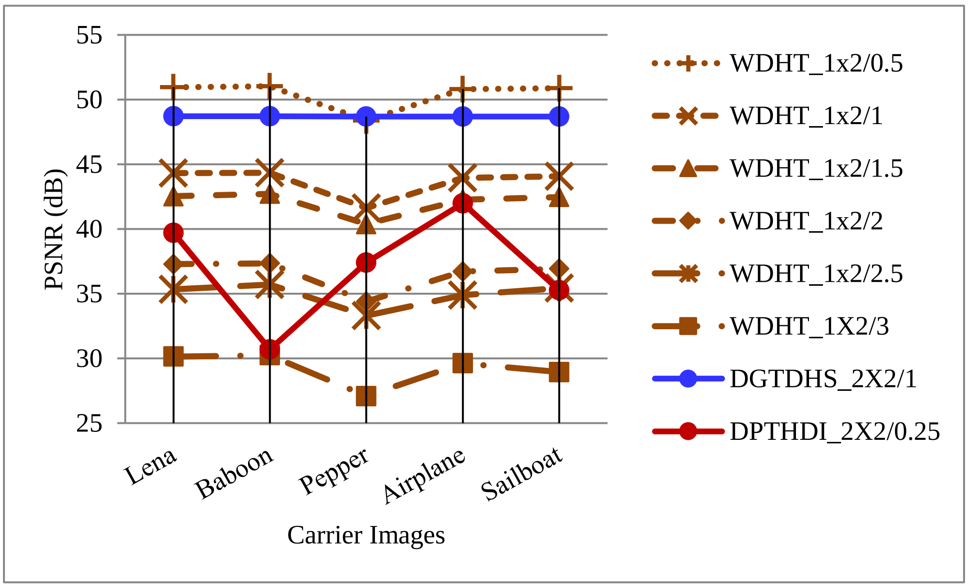


Fig. 2.7. Performance analysis of PSNR (dB) for variable payload based WDHT_1x2 and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 2.8 depicts the comparison of average peak signal to noise ratio (PSNR) among WDHT_1x2, WDHT_2x2, DPTHDI [88] and DGTDHS [129], respectively. The average PSNR of WDHT_1x2 and WDHT_2x2 are computed by averaging the PSNR values of twenty color images (fig. 1.1) for the payload variation [0.5 – 3 bpB]. In contrast to WDHT_2x2, the WDHT_1x2 obtains an improved average PSNR values while the payload is odd multiples of 0.5 i.e., 0.5, 1.5 and 2.5 bpB respectively. Unlikely, for even multiples of 0.5 i.e., the payload of 1, 2 and 3 bpB, the improvement in average PSNR values is minor. The average PSNR of 37.40 dB is computed for DPTHDI [88] by taking the average of PSNR values for “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images at 0.25 bpB of payload. Compared to DPTHDI [88], the average PSNR of WDHT_1x2 ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads. Again, an average PSNR (dB) of 48.70 dB is obtained for DGTDHS [129] where, the average PSNR is the result of taking the average of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” images at 1 bpB of payload. In contrast to DGTDHS [129], the WDHT_1x2 is lacking by 6 dB of average PSNR (dB) at 1 bpB however, the WDHT_1x2 is effective due to its variable payload. It is seen that the WDHT_1x2 offered perceptible quality (i.e., ≥ 30 dB) up to 2.5 bpB of payload however, the quality is severely degraded at 3 bpB. Since, the obtained average PSNR is 28.62 dB, a quality improvement scheme may be introduced into WDHT_1x2 (for raising the PSNR value as 30 dB or more) to obtain perceptible quality in the watermarked images [148].

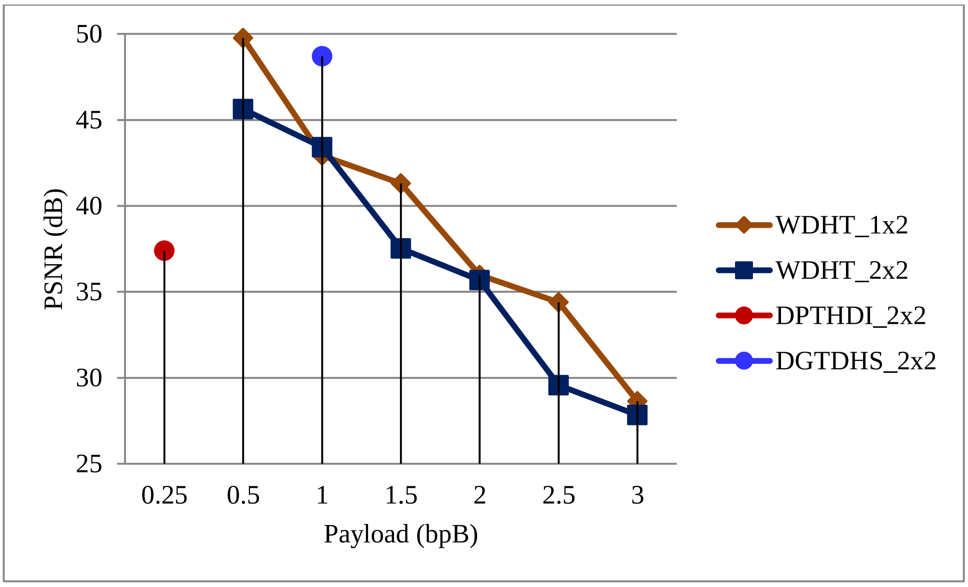


Fig. 2.8. Graphical representation of variation of average PSNR (dB) with respect to payload for WDHT_1x2, WDHT_2x2 and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

In fig. 2.9, a standard deviation (SD) analysis has been made for WDHT_1x2 scheme to measure the variation of standard deviation (SD) subsequent to embedding with respect to the payload variation 0.5 to 3 bpB. The standard deviation (SD) for each of the benchmark images such as “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” is computed by taking the average of standard deviation values corresponding to red, green and blue channels. It is seen from fig. 2.9 that the standard deviation (SD) is almost same up to the payload value of 2 bits per Byte (bpB). As the payload exceeds 2 bpB, the standard deviation (SD) for “Pepper” and “Sailboat” images decreases whereas, for “Lena”, “Baboon” and “Airplane” images, the standard deviation (SD) increases. The standard deviation (SD) at 0 bpB of payload designates the deviations of the original image. The dispersion made in standard deviation (SD) with respect to increasing payload is considerably minimal and for an observer, both U x V source and watermarked images obtained are seemingly alike.

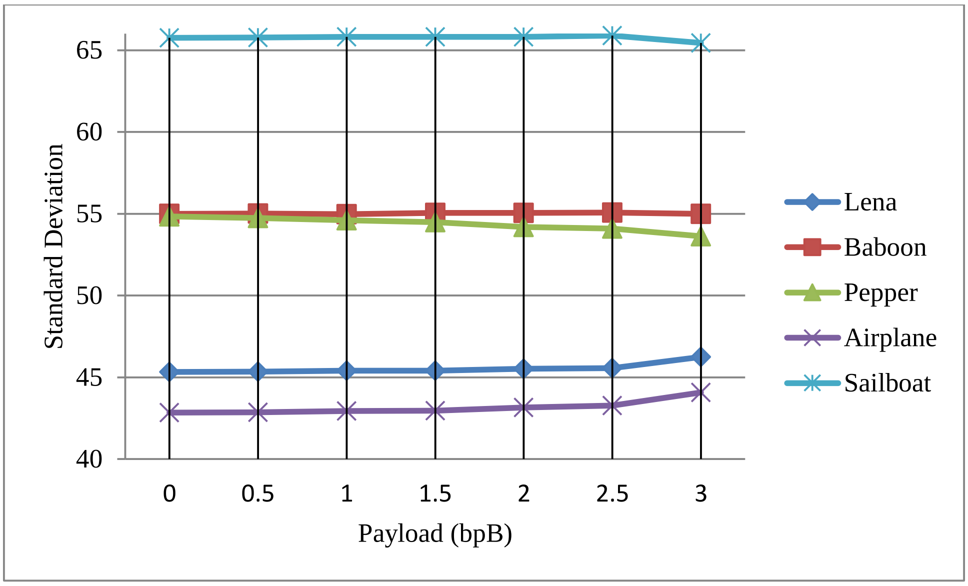


Fig. 2.9. Graphical representation of standard deviation (SD) for WDHT_1x2 with respect to 0, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

Standard deviation error (SDE) has been defined as the absolute difference of standard deviation (SD) values between the original and the watermarked images with respect to the payload variation of 0.5 to 3 bpB. In general, as the payload increases, the error increases too. On analysis it is seen from fig. 2.10 that the error is very low up to 2 bpB and further enhancement of payload, the error is substantially increases. The error is comparatively high for “Airplane” and “Pepper” images for the payload range [2.5 – 3 bpB] and as a consequence, the structured quality information of the watermarked image is deviated from the original image significantly.

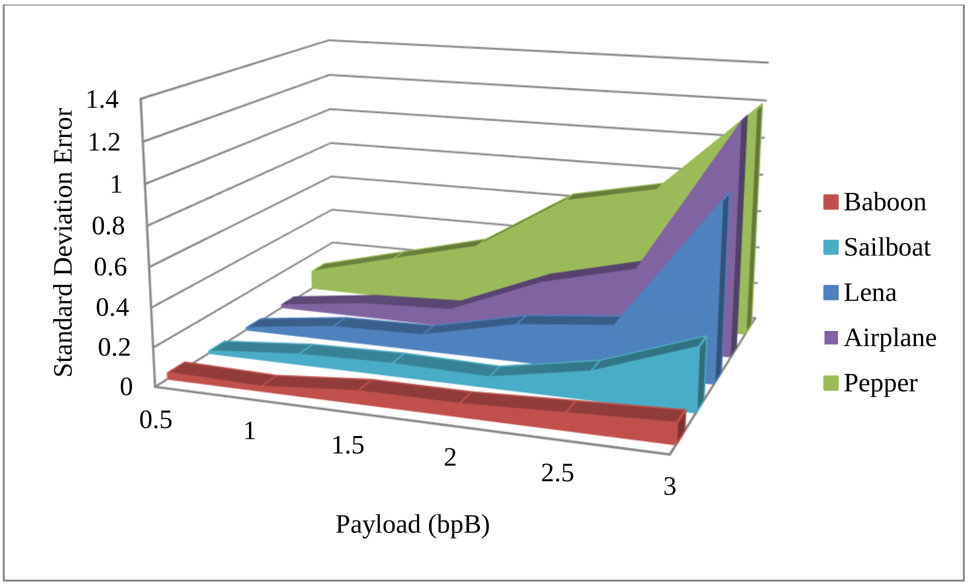


Fig. 2.10. Graphical representation of standard deviation error (SDE) for WDHT_1x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

Fig. 2.11 illustrates that the standard deviation error (SDE) for WDHT_2x2 and WDHT_1x2 are almost zero for "Lena", "Baboon", "Airplane" and "Sailboat" images up to 2 bpB. As the payload increases, the error increases too; however, the rate of increase of error is comparatively less for WDHT_1x2 over WDHT_2x2. The error may further be reduced by introducing the quality enhancement scheme of chapter 7 or GA based optimization of chapter 8. These quality improvement schemes are to be effective for the payload variation of 1.5 to 3 bpB.

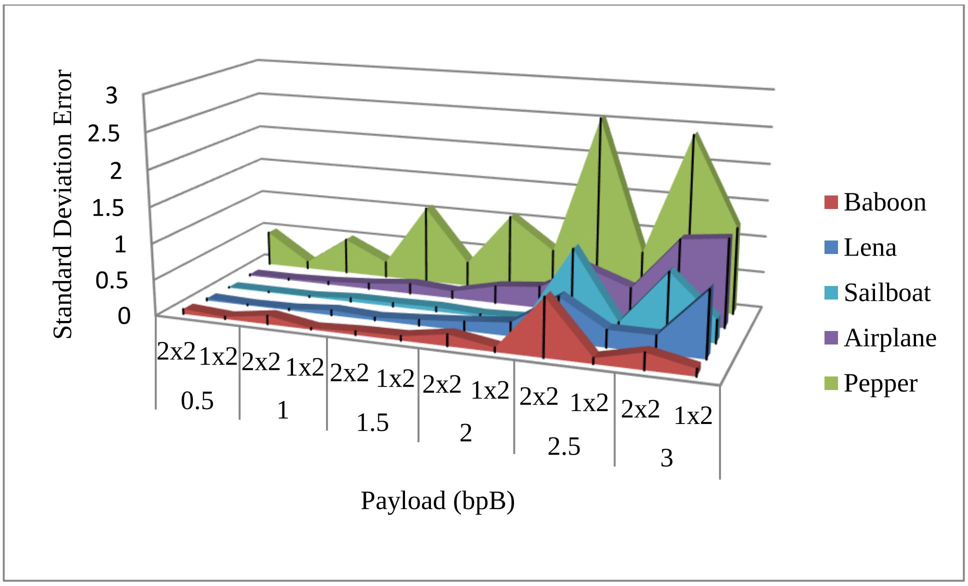


Fig. 2.11. Pictorial representation of variation of standard deviation error (SDE) between WDHT_1x2 and WDHT_2x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

2.3 Salient Features

It is observed that most of the existing schemes in transform domain are suffering due to the facts of high distortion, less payload, choosing gray-scale image as the cover and high computational overhead. In general, the watermark fabrication in transform domain offers improved security and higher robustness. Therefore, two novel watermarking schemes viz. WDHT_2x2 and WDHT_1x2 have been proposed in Discrete Hartley Transform (DHT) domain that offers variable payloads with respect to perceptible quality.

Some of the important features of the proposed techniques are discussed in the following section.

Watermarking based on Discrete Hartley Transform (DHT) is effective because DHT produces real transformed matrices with respect to real pixel matrices. Therefore, no fractional pixel components are generated during inverse transform and these schemes ensures less computational overhead. The pre-embedding adjustment ensures the non-occurrence of overflow and underflow situations. The higher transparency and variable payloads has been achieved by fabricating the watermark into the color images. In addition, it is seen that most of the existing schemes identified a region from each block for embedding; as a result the payload is comparatively less. Unlikely, proposed WDHT_2x2 and WDHT_1x2 schemes used variable payload (i.e., 0.5 to 3 bpB) with less degradation. The degradation in quality of the watermarked images can further be reduced by introducing the quality enhancement / genetic algorithm based optimizations as post-embedding operation. In contrast to Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes, both WDHT_2x2 and WDHT_1x2 schemes ensures variable payload with reduced quality degradation.

Chapter 3

Watermarking based on Legendre Transform (WLT)

3.1 Introduction

In this chapter, two novel fragile watermarking techniques have been outlined in Legendre Transform (LT) domain. Both schemes achieved variable payload with minimum degradation in fidelity; though, its primary concern is to authenticate color images. The carrier image is decomposed into sub-image blocks which are in turn transformed through Legendre Transform (LT). Watermark (along with a message digest) bits are fabricated into the transformed components in variable proportion to achieve variable payload and considerable quality distortion in the watermarked image. Applying Inverse Legendre Transform (ILT) on embedded blocks, the pixel components are generated in spatial domain. The recipient retrieves the watermark (as well as the message digest) through extraction operation and another message digest is re-computed to verify the authenticity. Simulation results ensured that the proposed techniques attained variable payload and acceptable quality distortion.

3.2 The Technique

Fragile watermarking based on Legendre Transform (LT) is the novel concept to verify the authenticity of color images. The carrier/cover image is a 24-bit color image which is decomposed into P x Q non-overlapping blocks (where, $1 \leq P \leq 2$ and $Q = 2$) in a sliding window manner. To address the overflow/underflow problem, adjustments of pixel components are done prior to embedding. Legendre Transform (LT) generates the P x Q sub-blocks of transformed components with reference to P x Q sub-image blocks of the carrier image. Secret bits corresponding to the message digest (MD) computed from the watermark, size and the content of the watermark are embedded into the transformed components starting from the least (Q – 1) bits position toward higher order bit position. It is observed that Legendre Transform (LT) is very much sensitive for small modification of transformed components during embedding. Therefore, to make sure the avoidance of overflow and underflow, an additional re-adjustment operation has been incorporated which adjusts the transformed components without affecting the fabricated bits. The inverse Legendre Transform (ILT) is used to transform each P x Q sub-block of embedded components into the spatial domain. The (Q – 1) unaltered bits at the least significant part of each transformed component ensure that the re-computed pixel components are non-fractional in nature. The process is repeated until the watermarked image is obtained. The recipient retrieves the secret bit-stream from the P x Q non-overlapping blocks (where, $1 \leq P \leq 2$ and $Q = 2$) of the 24 bit watermarked image in Legendre Transform (LT) domain. The message digest (MD), size and the content of the watermark are subsequently extracted from the transformed components. Optionally, an inverse Legendre Transform (ILT) is applied on the identical sub-image block to convert back the embedded components in spatial domain. An alphabet/color component (R/G/B) is constituted from each eight bits of the retrieved bit-stream which in succession restore the fabricated message digest (MD) along with the hidden watermark.

Proposed 2 x 2 and 1 x 2 block based schemes fabricates watermark in transform domain to authenticate color image. An intruder may alter the watermarked image in such an elegant way that the perceptibility of the watermarked image is not considerably degraded. The recipient re-computes the message digest (MD') from the extracted watermark and the same is compared against the extracted message digest (MD). If the extracted message digest (MD) perfectly matches with the re-computed message digest (MD'), then the authentication process is said to be successful, otherwise, it is unsuccessful. Therefore, any sort of visual or geometrical manipulations which may wipe out the originality of the watermarked image is easily detectable.

Section 3.2.1 of this chapter deals with 2 x 2 block based watermark fabrication that of section 3.2.2 describes 1 x 2 block based watermark fabrication.

3.2.1 2 x 2 Block based Watermark Fabrication

Legendre Transform (LT) [133, 134] produces integer output sequence for a given integer-valued sequence. Therefore, each 2 x 2 sub-block of pixel/transformed components are also considered as the linear sequence of four integer-valued numbers. The transform is applied on pixel components {p$_k$} to generate transformed components {t$_k$} as given in equation (3.1).

$$\begin{aligned} t_\gamma &= \sum_{k=0}^{\gamma} \binom{\gamma}{k} \binom{\gamma+k}{k} p_k \\ &= \sum_{k=0}^{\gamma} \binom{2k}{k} \binom{\gamma+k}{\gamma-k} p_k \end{aligned} \qquad (3.1)$$

where, $\binom{\gamma}{k}$ is a binomial co-efficient.

In general, Inverse Legendre Transform (ILT) [133, 134] does not always obtain an integer sequence as output; but, for a certain family of sequence $\{p_\gamma\}_{\gamma=1}^{\infty}$, it produces an integer-valued sequence. In this pretext, the inverse Legendre Transform (ILT) has been used to convert the transformed components back into spatial domain as given in equation (3.2) and equation (3.3).

$$\binom{2\gamma}{\gamma} p_\gamma = \sum_{i=0}^{\gamma} (-1)^{\gamma-k}\, d_{\gamma,k} t_k \tag{3.2}$$

where,

$$d_{\gamma,k} = \binom{2\gamma}{\gamma-k} - \binom{2\gamma}{\gamma-k-1}$$

$$= \frac{2k+1}{\gamma+k+1}\binom{2\gamma}{\gamma-k} \tag{3.3}$$

The formulation of Legendre Transform (LT) for image sub-block of size 2 x 2 has been expressed in equation (3.4).

$$t_k = \begin{cases} p_k : k = 0 \\ p_{k-1} + 2p_k : k = 1 \\ p_{k-2} + 6p_{k-1} + 6p_k : k = 2 \\ p_{k-3} + 12p_{k-2} + 30p_{k-1} + 20p_k \ : k = 3 \end{cases} \tag{3.4}$$

where, transformed component is denoted by t_k, pixel component is denoted by p_k and for all k, $0 \leq k \leq 3$.

Similarly, by applying the inverse Legendre Transform (ILT) of equation (3.2), the re-computed pixel components of each 2 x 2 masks are obtained as given in equation (3.5).

$$p_k = \begin{cases} t_k : k = 0 \\ \dfrac{t_k - t_{k-1}}{2} : k = 1 \\ \dfrac{t_k - 3t_{k-1} + 2t_{k-2}}{6} : k = 2 \\ \dfrac{t_k - 5t_{k-1} + 9t_{k-2} + 5t_{k-3}}{20} : k = 3 \end{cases} \tag{3.5}$$

where, pixel component is denoted by p_k, transformed component is denoted by t_k and for all k, $0 \leq k \leq 3$.

The proposed technique has been described in great detail at the following sections. The algorithm for insertion, the re-adjustment, the algorithm for extraction and an example are described in detail in sections 3.2.1.1, 3.2.1.2, 3.2.1.3 and 3.2.1.4 respectively. Results and discussions have been elaborated in section 3.2.1.5.

3.2.1.1 Insertion

The cover image is divided into 2 x 2 non-overlapping blocks in a row major order. The avoidance of overflow/underflow has been accomplished through an adjustment strategy prior to embedding. Legendre Transform (LT) is applied on each 2 x 2 sub-image block of red/green/blue channel to convert the pixel components into transformed components. Payload variation in the range (0.5 – 3 bpB) ensured the fabrication of the secret bits corresponding to the message digest, size and the content of the watermark into the transformed components. The secret bit fabrication is started from the first bit position of the least significant part (i.e., LSB-1) toward higher order bit position. Each transformed component except the fourth is capable of fabricating a maximum of four watermark bits. To address the occurrence of fractional pixel components, the least significant bit (i.e., LSB-0) of each transformed component is kept unaltered. It is observed that the components in the Legendre Transform (LT) domain are very much sensitive against small alteration during embedding. Therefore, to make sure the avoidance of overflow and underflow, an additional re-adjustment operation has been incorporated which may be used very often. Inverse Legendre Transform (ILT) is applied on 2 x 2 embedded blocks to obtain the pixel components. This process is repeated until and unless the entire bit-stream corresponding to the message digest (MD), watermark size and the content are embedded. The algorithm of insertion is given in algorithm 3.1.

Algorithm 3.1:

Input: Carrier/cover image (I) and authenticating watermark (W).

Output: Watermarked image (I′).

Method: The watermark (along with a message digest) is embedded into the carrier images in Legendre Transform (LT) domain where, the embedding is done in a block wise manner. Embedding bits in transform domain offers variable payload, less distortion and improved security. The detailed steps of embedding are as follows:

Step 1: Obtain a message digest (MD) from the secret watermark.

Step 2: The authenticating watermark size (in bits) is obtained by embedding the watermark into the three sub-matrices of the U x V color image as given in equation (3.6).

$$W_{size} = B \times \big(3 \times (U \times V)\big) - (MD + L) \qquad (3.6)$$

where, B, MD and L represents the embedding payload in terms of bits per Byte, the message digest obtained from the watermark and the header information corresponding to the size of the watermark respectively. The MD and L are consisting of 128 and 32 bits while the usual values of B are 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively.

Step 3: The cover image (I) is partitioned into 2 x 2 non-overlapping blocks in row major order. Each 2 x 2 block is consisting of four pixel components $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ *and* $p_{i+1,j+1}$ *of red/green/blue channel where, the values of i and j lies in the range* $0 \leq i \leq 1$*and* $0 \leq j \leq 1$.

Step 4: A pre-embedding adjustment is applied on pixel component (p) to avoid overflow and underflow based on the average payload of B bits per Byte as given in equation (3.7).

$$p = \begin{cases} \left(2^8 - 2^{2B - \left\lfloor \frac{2B}{3} \right\rfloor + 1}\right) : p \geq \left(2^8 - 2^{2B - \left\lfloor \frac{2B}{3} \right\rfloor + 1}\right) \\ 2^{2B - \left\lfloor \frac{2B}{3} \right\rfloor + 1} : p \leq 2^{2B - \left\lfloor \frac{2B}{3} \right\rfloor + 1} \end{cases} \quad (3.7)$$

Step 5: Apply Legendre Transform (LT) on 2 x 2 sub-matrices of pixel components corresponding to the red/green/blue channel to obtain the transformed components $t_{i,j}$, $t_{i,j+1}$, $t_{i+1,j}$ *and* $t_{i+1,j+1}$ *respectively.*

Step 6: λ_1 /λ_2 /λ_3 *bits from the secret bit-stream (corresponding to the message digest (MD), size (L) and the content (W) of the watermark) are subsequently fabricated on first/second/third transformed component for red/green/blue channel starting from the first bit position of the least significant part i.e., LSB-1 toward higher order bit position. Fourth transformed component is kept unaltered during embedding because a minor change in the component distorts the quality of the watermarked image significantly. The generalized form of* λ_1 /λ_2 /λ_3 *bits of secret information fabrication on first/second/third transformed component for the payload of B bits per Byte (bpB) is given in equation (3.8).*

$$(\lambda_1, \lambda_2, \lambda_3) = \begin{cases} (\lceil B \rceil, \lceil B \rceil, 2\lfloor B \rfloor): 0 < B \leq 1 \\ (\lceil B \rceil, \lfloor B + 1 \rfloor, \lfloor B + 1 \rfloor): 1 < B \leq 2 \\ (\lfloor B + 1 \rfloor, \lfloor B + 1 \rfloor, \lceil B + 1 \rceil): 2 < B \leq 3 \end{cases} \tag{3.8}$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all (λ_1, λ_2, λ_3), $0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 4$.

Step 7: Inverse Legendre Transform (ILT) is applied on each 2 x 2 sub-matrix of embedded components to convert it back into spatial domain. Since, LSB-0 is unaltered, first and second components are easily converted into spatial domain but the third component may generate fractional value during inverse transform. It is seen from equation (3.5) that for k = 3, the intermediate expression may not be divisible by 6. To make it divisible by 6, repeatedly add 2^{λ_3+1} with the third component until and unless it is divisible by 6.

Step 8: Apply re-adjustment operation over the embedded components in transform domain to address the overflow and underflow, if needed.

Step 9: Repeat steps 3 to 8 until and unless the embedding of the entire secret bit-stream (corresponding to the watermark size, content and the message digest MD) is done and the watermarked image (I') is produced.

Step 10: Stop.

3.2.1.2. Re-Adjustment

To address the problem of overflow and underflow, a pre-embedding adjustment of pixel components is done prior to embedding. However, it is already been discussed that Legendre Transform (LT) is very much sensitive against a very small changes on the transformed component. Therefore, an additional re-adjustment strategy has been incorporated which might be used very often. In general, the overflow and underflow may occur due to embedding followed by Inverse Legendre Transform (ILT) which may generate the following situations:

- The re-computed pixel component may be negative (-ve).
- The re-computed pixel component may be greater than the maximum value (i.e. 255).

In this phase, if the re-computed pixel component is negative (-ve), the operation applied for each 2 x 2 sub-matrix of transformed components is as here under:

$$t_{0,0} = t_{0,0} + 2^{\lambda_1+1} \tag{3.9}$$

where, $t_{0,0}$ is the first embedded component of the 2 x 2 sub-matrix in transform domain and λ_1 is used to specify the number of bits fabricated on $t_{0,0}$. The above process is repeated until and unless the all the re-computed pixel components becomes positive.

If the re-computed pixel component exceeds the maximum value of a byte (i.e., 255), then the operation applied for each 2 x 2 sub-matrix of transformed components is as here under:

$$t_{0,0} = t_{0,0} - 2^{\lambda_1+1} \tag{3.10}$$

where, $t_{0,0}$ is the first embedded component of the 2 x 2 sub-matrix in transform domain and λ_1 is used to specify the number of bits fabricated on $t_{0,0}$. The above process is repeated until and unless the all the re-computed pixel components becomes less than or equal to 255.

3.2.1.3. Extraction

At the recipient end, the watermarked image is decomposed into 2 x 2 non-overlapping blocks in a sliding window manner. Each 2 x 2 sub-image block of red/green/blue channel is converted into transform domain through Legendre Transform (LT). Varying numbers of fabricated secret bits are extracted from the transformed components as derived from the payload offered. Successive extractions of secret bits yields the re-construction of watermark size, content and the message digest (MD). Another message digest (MD′) is re-computed from the extracted watermark and the same is compared against the extracted message digest (MD). If both message digests are identical, then the authentication process is said to be successful; otherwise, it is to be unsuccessful. The extraction procedure is described as given in algorithm 3.2.

Algorithm 3.2:

Input: Watermarked image (I′).

Output: The authenticating watermark (W) and the 128 bits message digest (MD).

Method: The fabricated watermark (along with a message digest) is extracted from the watermarked image (I′) in Legendre Transform (LT) domain. Successive extracted bits reconstruct the watermark from which another message digest (MD) is obtained. Both message digests are compared between themselves to verify the authenticity. The exhaustive steps of extraction are as follows:

Step 1: The watermarked image (I') is decomposed into 2 x 2 non-overlapping blocks in row major order. Each 2 x 2 block is consisting of four pixel components $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$ corresponding to red/green/blue channel where, the values of i and j lies in the range $0 \le i \le 1$and $0 \le j \le 1$.

Step 2: Legendre Transform (LT) is applied on each 2 x 2 sub-matrix of pixel components corresponding to red/green/blue channel to obtain the transformed components $t_{i,j}$, $t_{i,j+1}$, $t_{i+1,j}$ and $t_{i+1,j+1}$ respectively.

Step 3: λ_1 /λ_2 /λ_3 bits of the fabricated secret bit-stream are successively extracted from the first/second/third transformed component of red/green/blue channel with the payload value of B bits per Byte (bpB) starting from the first bit position of the least significant part (LSB-1) toward higher order bit position. The generalized form of λ_1 /λ_2 /λ_3 secret bits extraction from first/second/third transformed component is given in equation (3.11).

$$(\lambda_1, \lambda_2, \lambda_3) = \begin{cases} (\lceil B \rceil, \lceil B \rceil, 2\lfloor B \rfloor) : 0 < B \le 1 \\ (\lceil B \rceil, \lfloor B+1 \rfloor, \lfloor B+1 \rfloor) : 1 < B \le 2 \\ (\lfloor B+1 \rfloor, \lfloor B+1 \rfloor, \lceil B+1 \rceil) : 2 < B \le 3 \end{cases} \tag{3.11}$$

where, $0 < B \le 3$, the difference between two successive payload values (ΔB) is 0.5 and for all (λ_1, λ_2, λ_3), $0 \le \lambda_1, \lambda_2, \lambda_3 \le 4$.

Step 4: For each 8 (eight) bits extraction, one alphabet/one primary (R/G/B) color component is constructed.

Step 5: Inverse Legendre Transform (ILT) is applied on each 2 x 2 sub-matrix of embedded components to re-obtain the pixel components in spatial domain.

Step 6: Repeat steps 1 to 5 to complete the extraction of the message digest (MD), size and content of the authenticating watermark.

Step 7: Obtain 128 bits message digest (MD') from the extracted watermark.

Step 8: Compare MD' with the extracted MD. If both are matches then the image is authorized, else unauthorized.

Step 9: Stop.

3.2.1.4. *Example*

Decomposing the carrier image into 2 x 2 non-overlapping blocks yields the 2 x 2 sub-matrices of red, green and blue channels such as R$_1$, G$_1$ and B$_1$ respectively. The 2 x 2 sub-matrices of pixel components are as follows:

$$R_1 = \begin{bmatrix} 245 & 69 \\ 21 & 112 \end{bmatrix} G_1 = \begin{bmatrix} 92 & 202 \\ 7 & 51 \end{bmatrix} B_1 = \begin{bmatrix} 25 & 119 \\ 220 & 245 \end{bmatrix}$$

A pre-embedding pixel adjustment has been incorporated to handle the overflow and underflow. It is done by modifying the upper and lower bounds of the pixel components in the specified range as given in equation (3.7). In this example, the payload is three (B = 3) and the modified upper bound (UB) and the lower bound (LB) are computed as follows:

$$UB = \left(2^8 - 2^{2\times3-\left\lfloor\frac{2\times3}{3}\right\rfloor+1}\right) = (2^8 - 2^5) = (256 - 32) = 224$$

$$LB = 2^{2\times3-\left\lfloor\frac{2\times3}{3}\right\rfloor+1} = 2^5 = 32$$

Thus, if any pixel component falls beyond this range then that pixel component is adjusted immediately and the 2 x 2 sub-matrices are obtained as follows:

$$R_1 = \begin{bmatrix} 224 & 69 \\ 32 & 112 \end{bmatrix} G_1 = \begin{bmatrix} 92 & 202 \\ 32 & 51 \end{bmatrix} B_1 = \begin{bmatrix} 32 & 119 \\ 220 & 224 \end{bmatrix}$$

Legendre Transform (LT) is applied on each 2 x 2 sub-matrix of pixel components to convert it from spatial domain into transform domain. The 2 x 2 sub-matrices of transformed components such as T(R1), T(G1) and T(B1) are obtained as given below:

$$T(R_1) = \begin{bmatrix} 224 & 362 \\ 830 & 4252 \end{bmatrix} T(G_1) = \begin{bmatrix} 92 & 496 \\ 1496 & 4496 \end{bmatrix} T(B_1) = \begin{bmatrix} 32 & 270 \\ 2066 & 12540 \end{bmatrix}$$

Let the secret bit-stream *“1010000101100000011001111011011000001”* is to be fabricated into the transformed components based on the embedding rule given in equation (3.8). To achieve the payload value of 3 bits per Byte, four bits are fabricated ($\lambda_1 = 4$, $\lambda_2 = 4$, $\lambda_3 = 4$) on first, second and third transformed components starting from LSB-1 toward higher order bit position. Hence, the 2 x 2 sub-matrices of embedded components are as here under:

$$T'(R_1) = \begin{bmatrix} 234 & 368 \\ 812 & 4252 \end{bmatrix} T'(G_1) = \begin{bmatrix} 64 & 486 \\ 1502 & 4496 \end{bmatrix} T'(B_1) = \begin{bmatrix} 44 & 262 \\ 2064 & 12540 \end{bmatrix}$$

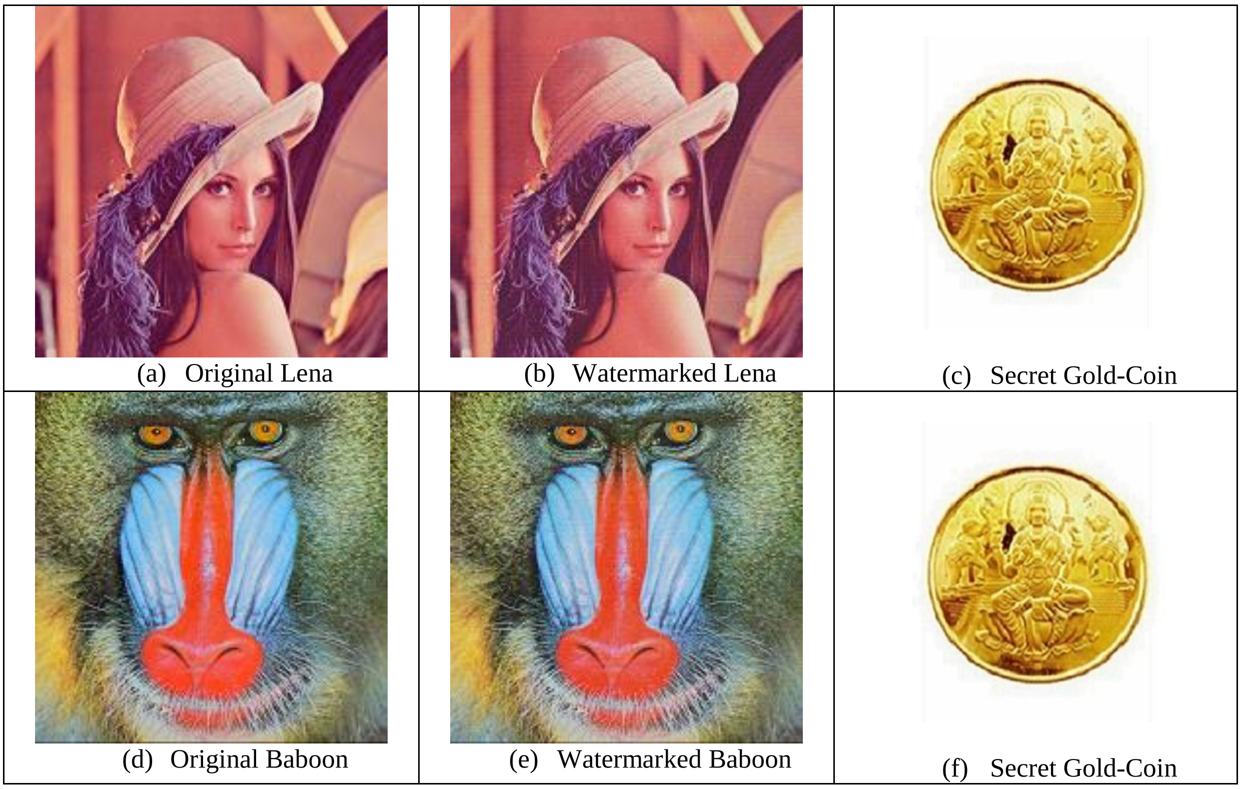
On application of one dimensional inverse Legendre Transform (ILT) over 2 x 2 sub-matrices of embedded components, the 2 x 2 sub-matrices of pixel components are obtained:

$$R'_1 = \begin{bmatrix} 234 & 67 \\ 40 & 116 \end{bmatrix} G'_1 = \begin{bmatrix} 64 & 211 \\ 34 & 52 \end{bmatrix} B'_1 = \begin{bmatrix} 44 & 109 \\ 233 & 217 \end{bmatrix}$$

It has been observed that the modified pixel components for 2 x 2 sub-matrices are non-fractional as the least significant bit (i.e., LSB-0) of first\second\third embedded component is kept unaltered whereas, the third component of the second sub-matrix is adjusted by adding 2^{4+1} i.e., 32 subsequently till it is divisible by 6.

3.2.1.5. Results and Discussions

The performance of 2 x 2 block based watermarking scheme in Legendre Transform (LT) domain (WLT_2x2) deals with the visual transparency at varying payload for the benchmark images [130, 131] as given in fig. 1.1. The peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD) and standard deviation error (SDE) are adopted to measure the visual image quality between the cover and watermarked images respectively. All cover images are 512 × 512 in dimension which fabricates the watermark of varying sizes. On embedding the watermark (i.e., the "Gold-Coin") into the carrier images (viz. "Lena", "Baboon" and "Pepper"), the watermarked images are obtained. The different states of modifications (before and after embedding the watermark) are shown in fig. 3.1. A comparative study has also been done among WLT_2x2, WDHT_2x2 and DPTHDI [88] and DGTDHS [129] respectively. In WLT_2x2, the quality metrics are computed at varying payload, analysis is done on those results and comparison is made to prove the effectiveness.



(a) Original Lena

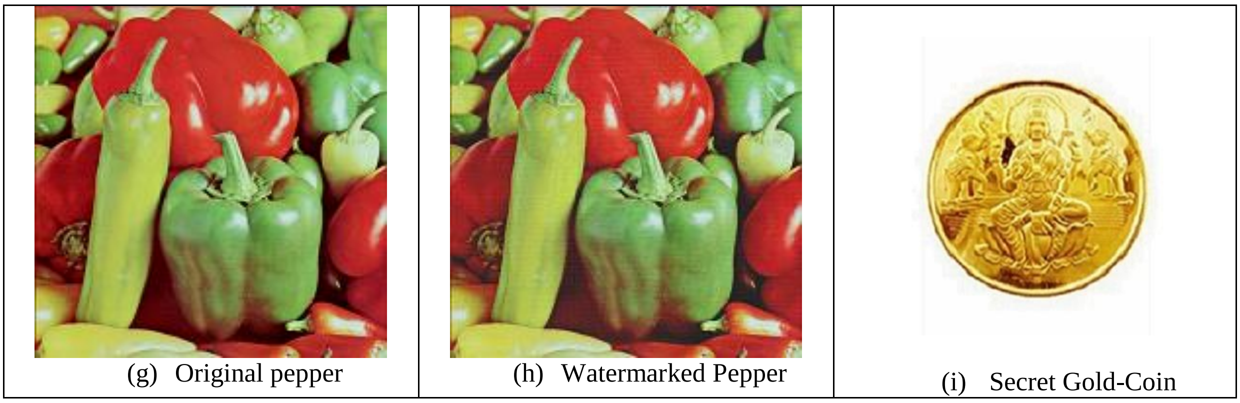
(b) Watermarked Lena

(c) Secret Gold-Coin

(d) Original Baboon

(e) Watermarked Baboon

(f) Secret Gold-Coin



(g) Original pepper

(h) Watermarked Pepper

(i) Secret Gold-Coin

Fig 3.1. Cover, watermarked and the authenticating watermark images in the proposed WLT_2x2 technique

Out of twenty benchmark images [130, 131] as given in fig. 1.1, the image quality is mostly affected for “Desert” image at 3 bpB and that of the quality is least affected for “Bluheron” image at 0.5 bpB of payload respectively. From a set of standard quality metrics such as peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ), the major two metrics namely peak signal to noise ratio (PSNR) and mean squared error (MSE) are widely used for analyzing the quality of the watermarked images. Both metrics maintain an inverse relationship with each other at the same time. Higher PSNR (or, lower MSE) values designate good image quality whereas, lower PSNR (or, higher MSE) values ensure poor image quality. Table 3.1 shows a MSE of 338.94 results in a PSNR of 22.82 dB, and a MSE of 0.77 results in a PSNR of 49.26 dB for two extreme payload values (0.5 and 3 bpB). The minimum PSNR obtained in WLT_2x2 does not provide perceptible image quality since the PSNR is less than 30 dB [148]. However, the WLT_2x2 is important for consideration as it offers variable payload in the range [0.5 – 3 bpB]. Later on, the quality improvement technique may also be applied over the watermarked images to enrich its quality without losing the embedded watermark. The minimum values of image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) are 0.94893 (Desert), 0.838956 (Bobcat) and 0.301821 (Splash) respectively whereas, the maximum values of IF, SSIM and UIQ are 0.999973 (Airplane), 0.999904 (San Diego) and 0.997381 (San Diego) respectively. These three quality metrics are lowest at 3 bpB and highest at 0.5 bpB, respectively. In general, the values of IF, SSIM and UIQ lie between 0 and 1; the values closer to one specify more similarity between the host and watermarked images. The experimental results are summarized against the average values for various metrics of twenty carrier images at variable payload.

Table 3.1. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WLT_2x2 technique

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	48.597498	0.898111	0.999943	0.999667	0.979761
	1.0	45.056542	2.029673	0.999871	0.999123	0.959146
	1.5	40.123397	6.320341	0.999600	0.997237	0.898161
	2.0	36.767718	13.687009	0.999143	0.992703	0.831967
	2.5	30.468036	58.381690	0.996354	0.971176	0.648356
	3.0	27.020561	129.129130	0.991696	0.944233	0.503675
Baboon	0.5	48.591863	0.899278	0.999952	0.999877	0.996092
	1.0	45.029024	2.042574	0.999891	0.999621	0.991029
	1.5	40.115164	6.332335	0.999664	0.998870	0.976473
	2.0	36.746634	13.753615	0.999273	0.996395	0.952637
	2.5	30.725022	55.027283	0.997103	0.985188	0.886801
	3.0	27.470962	116.408332	0.993834	0.976715	0.819914
Pepper	0.5	45.895704	1.673053	0.999845	0.990208	0.968212
	1.0	41.062445	5.091383	0.999516	0.982019	0.945524
	1.5	38.123879	10.015955	0.999095	0.979022	0.898301
	2.0	33.526633	28.868019	0.997298	0.961229	0.827298
	2.5	27.283054	121.555580	0.988398	0.922782	0.649104
	3.0	24.943521	208.318658	0.980747	0.895162	0.496055
Airplane	0.5	48.538964	0.910298	0.999973	0.999601	0.944582
	1.0	44.994920	2.058677	0.999941	0.998998	0.900462
	1.5	40.076365	6.389160	0.999817	0.996840	0.797924
	2.0	36.840460	13.459667	0.999614	0.992080	0.715679
	2.5	31.723598	43.724060	0.998746	0.981838	0.557968
	3.0	27.780963	108.388688	0.996894	0.954604	0.432999
Sailboat	0.5	48.314324	0.958623	0.999951	0.999706	0.983467
	1.0	44.559928	2.275555	0.999885	0.999178	0.967231
	1.5	39.882906	6.680203	0.999663	0.997609	0.923343
	2.0	36.302576	15.234306	0.999233	0.992906	0.872747
	2.5	29.729854	69.198211	0.996546	0.968981	0.715109
	3.0	26.518742	144.945964	0.992739	0.946995	0.596257
Earth	0.5	48.593479	0.898943	0.999946	0.999748	0.990599
	1.0	45.114612	2.002714	0.999880	0.999355	0.980311
	1.5	40.143354	6.291365	0.999626	0.997972	0.945617
	2.0	36.890433	13.305676	0.999213	0.994517	0.899650

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	30.714834	55.156520	0.996373	0.980146	0.752805
	3.0	26.939406	131.564805	0.991738	0.959339	0.613306
San Diego	0.5	48.604975	0.896567	0.999966	0.999904	0.997381
	1.0	45.118408	2.000965	0.999925	0.999755	0.994540
	1.5	40.164056	6.261446	0.999766	0.999225	0.985163
	2.0	36.914802	13.231225	0.999509	0.998065	0.973409
	2.5	31.872491	42.250433	0.998419	0.995161	0.929947
	3.0	27.696770	110.510440	0.995882	0.985969	0.858004
Splash	0.5	46.648097	1.406923	0.999862	0.989492	0.919325
	1.0	42.186192	3.930620	0.999599	0.984593	0.869336
	1.5	38.772883	8.625653	0.999179	0.981127	0.749618
	2.0	34.633025	22.375762	0.997792	0.969989	0.646495
	2.5	28.646320	88.807228	0.991063	0.940758	0.428735
	3.0	25.864859	168.497881	0.983800	0.901289	0.301821
Oakland	0.5	47.586542	1.133511	0.999933	0.999430	0.994396
	1.0	43.427854	2.953214	0.999818	0.998556	0.988077
	1.5	39.419605	7.432242	0.999573	0.997246	0.969765
	2.0	35.513242	18.270740	0.998886	0.992760	0.942024
	2.5	29.768230	68.589454	0.995622	0.980713	0.855958
	3.0	26.673254	139.879778	0.991598	0.961787	0.735221
Foster City	0.5	48.612614	0.894991	0.999968	0.999545	0.974108
	1.0	45.130751	1.995286	0.999929	0.998845	0.946926
	1.5	40.184898	6.231469	0.999778	0.996331	0.870710
	2.0	36.949230	13.126752	0.999530	0.990896	0.789979
	2.5	31.952090	41.483111	0.998516	0.978890	0.616129
	3.0	28.020619	102.569552	0.996293	0.943741	0.456696
Anhinga	0.5	47.728449	1.097072	0.999915	0.999712	0.872690
	1.0	43.010186	3.251335	0.999750	0.999319	0.853297
	1.5	38.813695	8.544975	0.999344	0.998569	0.814785
	2.0	35.889143	16.755831	0.998715	0.995456	0.809248
	2.5	29.910850	66.373594	0.994909	0.981120	0.699698
	3.0	26.907342	132.539723	0.989785	0.965315	0.608862
Athens	0.5	48.558146	0.906286	0.999927	0.999750	0.960121
	1.0	44.605298	2.251907	0.999819	0.999600	0.936098
	1.5	39.187490	7.840274	0.999373	0.998902	0.880374
	2.0	37.459800	11.670791	0.999070	0.996453	0.878110

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	31.941127	41.587955	0.996671	0.985869	0.729094
	3.0	27.430618	117.494771	0.990607	0.965121	0.587848
Bardowl	0.5	47.339213	1.199938	0.999878	0.999446	0.995658
	1.0	43.069315	3.207368	0.999671	0.998209	0.987093
	1.5	38.683068	8.805894	0.999113	0.997140	0.971578
	2.0	34.892953	21.075843	0.997831	0.984057	0.935055
	2.5	27.484521	116.045478	0.987989	0.934805	0.813875
	3.0	24.976757	206.730522	0.978860	0.920092	0.723527
Barnfall	0.5	48.949717	0.828149	0.999866	0.999781	0.994941
	1.0	45.292480	1.922349	0.999686	0.999412	0.989747
	1.5	39.811847	6.790404	0.998900	0.998240	0.967057
	2.0	36.513221	14.513034	0.997597	0.993641	0.943774
	2.5	30.629336	56.253133	0.990076	0.970177	0.819871
	3.0	27.031227	128.812376	0.978021	0.949453	0.684586
Butrfly	0.5	48.670968	0.883046	0.999937	0.999787	0.990622
	1.0	44.848470	2.1292826	0.999849	0.999634	0.980359
	1.5	39.316514	7.610776	0.999460	0.998787	0.948676
	2.0	37.243435	12.266956	0.999128	0.996665	0.933730
	2.5	31.684148	44.123045	0.996808	0.987781	0.813819
	3.0	27.489839	115.903457	0.991627	0.971979	0.674914
Bobcat	0.5	48.596957	0.898223	0.999873	0.999582	0.743136
	1.0	44.705551	2.200519	0.999690	0.999354	0.730157
	1.5	39.228867	7.765932	0.998909	0.997464	0.703488
	2.0	36.996485	12.984695	0.998166	0.991180	0.692386
	2.5	25.118073	200.111957	0.971736	0.868542	0.537552
	3.0	23.076604	320.198928	0.954726	0.838956	0.463534
Bodie	0.5	46.899420	1.327817	0.999766	0.999528	0.972632
	1.0	42.744241	3.456656	0.999384	0.998295	0.962971
	1.5	39.239816	7.746378	0.998648	0.997123	0.945727
	2.0	34.238252	24.505021	0.995637	0.979762	0.902805
	2.5	26.472683	146.491358	0.973269	0.891256	0.729494
	3.0	24.345058	239.097203	0.957276	0.869742	0.649734
Bluheron	0.5	49.264767	0.770200	0.999905	0.999835	0.991207
	1.0	45.743145	1.732869	0.999788	0.999481	0.982557
	1.5	39.993229	6.512645	0.999201	0.998122	0.944958
	2.0	36.816920	13.532821	0.998339	0.995182	0.898292

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	31.023947	51.367164	0.993699	0.981828	0.743238
	3.0	27.216936	123.420337	0.984879	0.958688	0.581916
Colomtn	0.5	48.019783	1.025892	0.999919	0.999758	0.979110
	1.0	43.929537	2.631034	0.999794	0.999389	0.971835
	1.5	39.629955	7.080839	0.999447	0.998297	0.948848
	2.0	36.197774	15.606404	0.998781	0.995283	0.931497
	2.5	30.252609	61.350682	0.995216	0.977803	0.822100
	3.0	26.955194	131.087379	0.989772	0.958342	0.713256
Desert	0.5	44.542082	2.284926	0.999641	0.997288	0.992252
	1.0	39.468780	7.348561	0.998836	0.992672	0.980884
	1.5	36.844878	13.445980	0.997946	0.991258	0.967647
	2.0	31.810824	42.854642	0.993388	0.969916	0.928163
	2.5	24.530458	229.104983	0.965391	0.894292	0.774425
	3.0	22.829464	338.948617	0.948930	0.875782	0.712674
Average Case	0.5	47.927680	1.089592	0.999898	0.998582	0.962015
	1.0	43.954880	2.825627	0.999726	0.997270	0.945879
	1.5	39.387790	7.636213	0.999305	0.995769	0.905411
	2.0	35.957180	17.553940	0.998307	0.988957	0.865247
	2.5	29.596560	82.849150	0.991145	0.958955	0.726204
	3.0	26.359430	160.722300	0.983985	0.937165	0.610740

This section describes the performance comparison of the proposed WLT_2x2 method over Varsaki et al.'s 2 x 2 block based schemes i.e., Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] in terms of peak signal to noise ratio (dB) and payload. Five benchmark images [130, 131] such as "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively are taken to fabricate the varying sizes of the secret "Gold-Coin" image as given in fig. 1.1. The primary constraint of DPTHDI [88] and DGTDHS [129] are their fixed payload values which are also considered as significantly low. In contrast to DPTHDI [88], the WLT_2x2 demonstrated equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for "Lena", 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for "Baboon", 0.5, 1 and 1.5 bpB of payloads for "Pepper", 0.5, 1 and 1.5 bpB of payloads for "Airplane" and 0.5, 1, 1.5 and 2 bpB of payloads for "Sailboat" respectively. Again, proposed WLT_2x2 offered comparatively less PSNR than DGTDHS [129] at 1 bpB; however, the WLT_2x2 provides variable payload that offers a spread from 0.5 to 3 bpB.

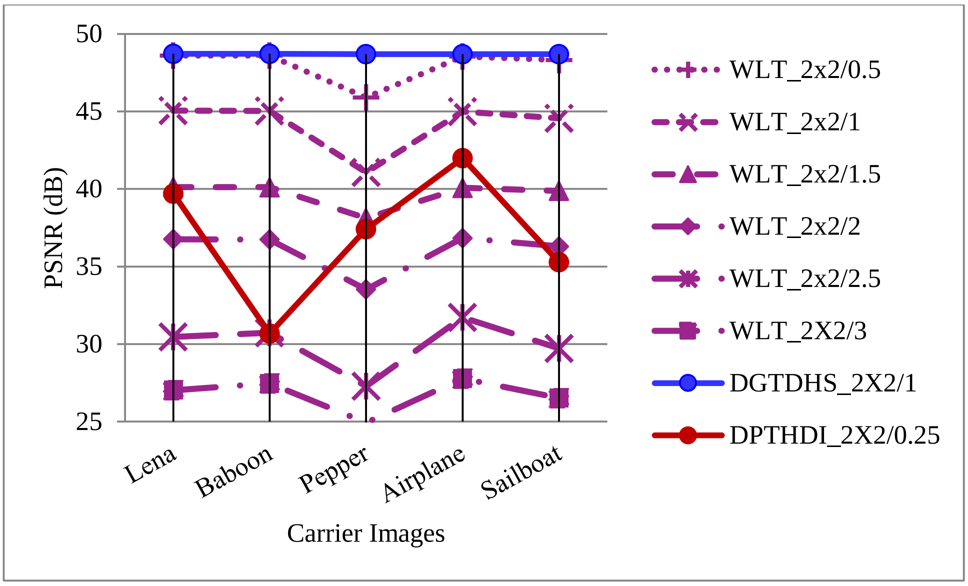


Fig. 3.2. Performance analysis of PSNR (dB) for variable payload based WLT_2x2 and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) techniques based on five color images

Table 3.1 reveals 47.92 dB and 26.35 dB as the highest and lowest average PSNR values as computed from twenty benchmark images (fig. 1.1). It is seen that the obtained average PSNR values are greater than or equal to 30 dB [148] for the payload range of 0.5 to 2 bpB. However, for the payload range 2.5 to 3 bpB, the average PSNR values become less than 30 dB which ensured quality distortion below the acceptable level. In further study, the average PSNR values (corresponding to the payload spreading from 2.5 to 3 bpB) might be improved to sustain it around 30 dB by introducing the quality improvement scheme. Fig. 3.3 illustrates the variation of average PSNR for WLT_2x2, WDHT_2x2, DPTHDI [88] and DGTDHS [129] respectively. In contrast to WDHT_2x2, the average PSNR of WLT_2x2 is improved with respect to 0.5, 1, 1.5, 2 and 2.5 bpB of payloads however, that of average PSNR is lacking over the WDHT_2x2 at 3 bpB. Nevertheless, the overall degradation of quality of WLT_2x2 has been reduced over WDHT_2x2. The average PSNR for DPTHDI [88] is 37.40 dB as computed from "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" images at 0.25 bpB of payload. In contrast to DPTHDI [88], the average PSNR of WLT_2x2 ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads. The average PSNR for DGTDHS [129] obtained is 48.70 dB by averaging the PSNR values of "Lighthouse", "Elaine", "Lenna", "Boat", "F16" images at 1 bpB of payload. Compared to DGTDHS [129], the WLT_2x2 is lacking in terms of average PSNR (dB) at 1 bpB however, the basic emphasis for this scheme is placed on the ability of offering variable payload for the spread of 0.5 to 3 bpB.

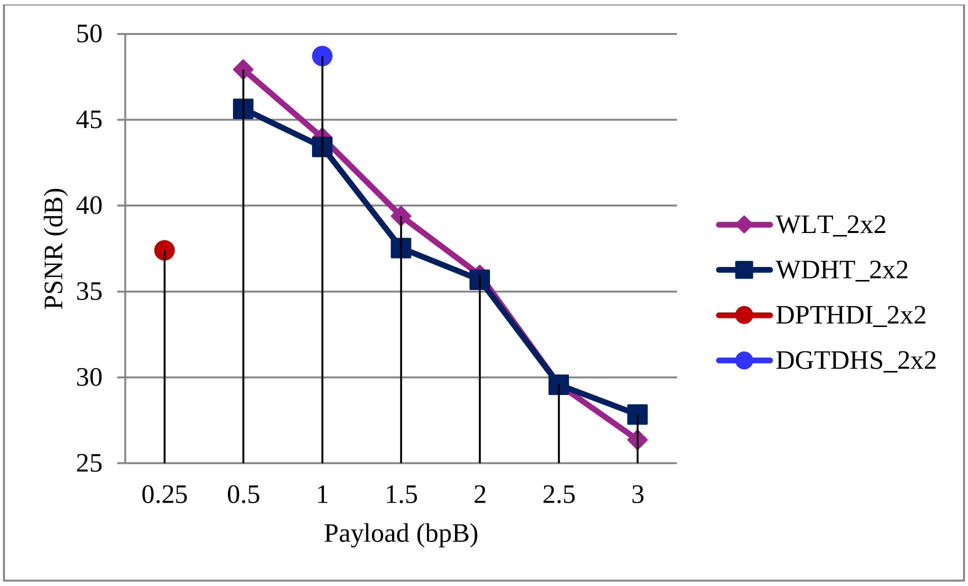


Fig. 3.3. Graphical representation of variation of average PSNR (dB) with respect to payload for WLT_2x2, WDHT_2x2 and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) techniques

In fig. 3.4, the standard deviation (SD) is computed by taking the average of standard deviation (SD) values of red, green and blue channels for "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively. The standard deviation (SD) at 0 bpB of payload represents the standard deviation (SD) of the original image. Standard deviation (SD) of the watermarked image is very close to the original one for the payload range [0.5 – 2 bpB]. Statistical comparison demonstrates that the changes made in the watermarked image with reference to the original image are comparatively high at 2.5 and 3 bpB of payload.

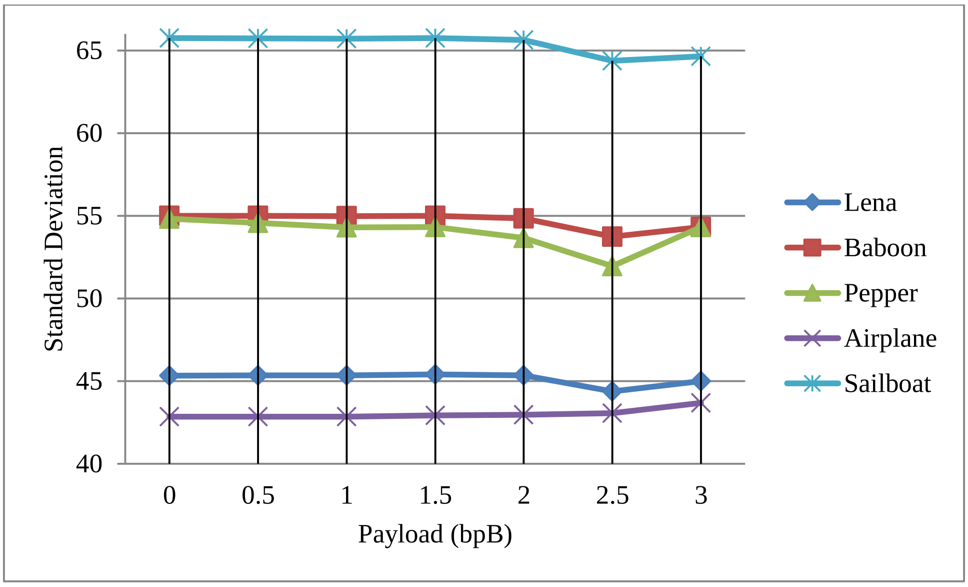


Fig. 3.4. Graphical representation of standard deviation (SD) for WLT_2x2 with respect to 0, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

In general, as the standard deviation is a measure of signal fluctuation from the mean, a better analysis has been made in terms of standard deviation error (SDE) where, the error is nothing but the absolute difference of standard deviation values between the original and the watermarked images with respect to increasing payload. The chart explains how the error is getting increased with the enhancement of payload (bpB). On analysis it is seen that the watermarked images are deviating from the original image with respect to increasing payload. It can also be seen that the error is minimum at (0.5 – 2 bpB) and afterward it is increasing. The error is comparatively high for "Pepper" and as a consequence, it creates significant deviations in the watermarked image as compared to the original one.

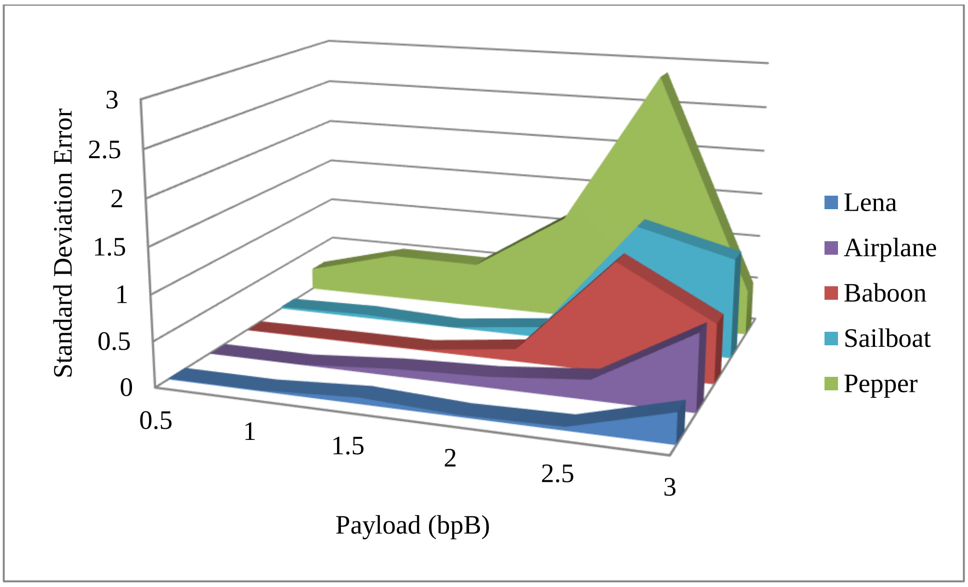


Fig. 3.5. Graphical representation of standard deviation error (SDE) for WLT_2x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

3.2.2. 1 x 2 Block based Watermark Fabrication

As discussed in section 3.2.1, the Legendre transform [133, 134] is an effective polynomial sequence which is applied on the pixel components {p_k} to generate transformed components {t_k} as given in equation (3.12).

$$t_\gamma = \sum_{k=0}^{\gamma} \binom{\gamma}{k} \binom{\gamma+k}{k} p_k$$

$$= \sum_{k=0}^{\gamma} \binom{2k}{k} \binom{\gamma+k}{\gamma-k} p_k \qquad (3.12)$$

where, $\binom{\gamma}{k}$ is a binomial co-efficient.

Similarly, the inverse Legendre Transform (ILT) is used to convert the transformed components into spatial domain as given in equation (3.13).

$$\binom{2\gamma}{\gamma} p_\gamma = \sum_{i=0}^{\gamma} (-1)^{\gamma-k}\, d_{\gamma,k} t_k \tag{3.13}$$

where,

$$d_{\gamma,k} = \binom{2\gamma}{\gamma-k} - \binom{2\gamma}{\gamma-k-1}$$

$$= \frac{2k+1}{\gamma+k+1}\binom{2\gamma}{\gamma-k} \tag{3.14}$$

In comparison with 2 x 2 block based formulation as discussed in section 3.2.1, this section deals with the formulation of Legendre Transform (LT) for 1 x 2 sub-blocks.

Legendre Transform (LT) converts each 1 x 2 sub-block (pair) of pixel components into the transformed components as given in equation (3.15).

$$t_k = \begin{cases} p_k : k = 0 \\ p_{k-1} + 2p_k : k = 1 \end{cases} \tag{3.15}$$

where, the transformed component and the pixel component are denoted by t_k and p_k respectively, and for all k, 0 ≤ k ≤ 1.

Similarly, by applying the inverse Legendre Transform (ILT) of equation (3.13), the re-computed pixel components of the 1 x 2 mask can be obtained using equation (3.16).

$$p_k = \begin{cases} t_k : k = 0 \\ \dfrac{t_k - t_{k-1}}{2} : k = 1 \end{cases} \tag{3.16}$$

where, the pixel component and the transformed component are denoted by p_k and t_k respectively, and for all k, 0 ≤ k ≤ 1.

The algorithm for insertion, the algorithm for extraction and an example are elaborately described in section 3.2.2.1, 3.2.2.2 and 3.2.2.3, respectively. Results, analysis and discussions have been elaborated in section 3.2.2.4.

3.2.2.2. Insertion

Split the cover image into 1 x 2 non-overlapping blocks in row major order. To address the overflow/underflow, pixel components of each block are adjusted prior to embedding. Apply

Legendre Transform (LT) on each 1 x 2 sub-image block / pair of pixel components to obtain the transformed components. Each transformed component is capable of fabricating a maximum of three watermark bits to achieve variable payload with perceptible quality. Secret bits corresponding to the message digest, size and the content of the watermark are subsequently fabricated into the transformed components starting from the 1st bit position of the least significant part (i.e., LSB-1) toward higher order bit position. The least significant bit (LSB-0) of each transformed component is kept unaltered for the generation of non-fractional pixel components subsequent to inverse transform. Inverse Legendre Transform (ILT) is applied on each 1 x 2 sub-block or pair of embedded components to re-compute the pixel components in spatial domain. The process is repeated until and unless the message digest, size and the content of the watermark are completely fabricated to produce the watermarked image in spatial domain.

Algorithm 3.3:

Input: Carrier/cover image (I) and authenticating watermark image (W).

Output: Watermarked image (I′).

Method: The watermark (along with a message digest) is embedded into the carrier images in Legendre Transform (LT) domain. Embedding bits in transform domain offers variable payload, less distortion and improved security. The detailed steps of embedding are as follows:

Step 1: Obtain a message digest (MD) from the authenticating watermark.

Step 2: The authenticating watermark size (in bits) is obtained by embedding the watermark bits into the three sub-matrices of the U x V color image as given in equation (3.17).

$$W_{size} = B \times \left(3 \times (U \times V)\right) - (MD + L) \qquad (3.17)$$

where, B, MD and L represents the embedding payload in terms of bits per Byte, the message digest obtained from the watermark and the header information corresponding to the size of the watermark respectively. The MD and L are consisting of 128 and 32 bits while the usual values of B are 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively.

Step 3: The cover image (I) is partitioned into 1 x 2 non-overlapping blocks in row major order. Each 1 x 2 block is consisting of *pair of pixel components p_i and p_{i+1} of red/green/blue channel where, the values of i lies in the range $0 \leq i \leq 1$.*

Step 4: A pre-embedding adjustment is applied on each pixel component p, to retain the value positive and less than, or equal to 255 i.e., to keep away from overflow and underflow. The modified upper and lower bounds corresponding to the payload value of B bits per byte of watermark is given in equation (3.18).

$$p = \begin{cases} \left(2^8 - 2^{\lceil B \rceil + 1}\right) : p \geq \left(2^8 - 2^{\lceil B \rceil + 1}\right) \\ 2^{\lceil B \rceil + 1} : p \leq 2^{\lceil B \rceil + 1} \end{cases} \quad (3.18)$$

Step 5: Legendre Transform (LT) is applied on 1 x 2 sub-matrices of pixel components of red/green/blue channel to compute the pair of transformed components t_i and t_{i+1} respectively.

Step 6: λ_1 /λ_2 bits from the secret bit-stream (corresponding the message digest (MD), size (L) and the content (W) of the watermark) are subsequently fabricated on first/second transformed component starting from the 1st bit position of the least significant part (i.e., LSB-1) toward higher order bit position. The generalized form of λ bits of secret information fabrication on each transformed component for the payload value of B bits per Byte (bpB) is given in equation (3.19).

$$\lambda = \begin{cases} \lfloor B \rfloor : \lambda = \lambda_1 \\ \lceil B \rceil : \lambda = \lambda_2 \end{cases} \quad (3.19)$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all λ, $0 \leq \lambda \leq 3$.

Step 7: Inverse Legendre Transform (ILT) is applied on each 1 x 2 sub-matrix or pair of embedded components to form the 1 x 2 sub-matrix or pair of pixel components in spatial domain.

Step 8: Repeat steps 3 to 7 to fabricate the message digest MD, size and content of the watermark which results the watermarked image (I') in spatial domain.

Step 9: Stop.

3.2.2.3. Extraction

At the receiving end the watermarked image is decomposed into 1 x 2 non-overlapping blocks and then each 1 x 2 sub-image block or pair of pixel components of red/green/blue channel is converted into transform domain based on Legendre Transform (LT). The reverse procedure is applied to extract the secret bits of the watermark size, content and message digest (MD) from the transformed components in variable proportion based on the extraction rule. Inverse Legendre Transform (ILT) is applied to convert back the pair of transformed components into the spatial domain. The process is repeated until and unless the fabricated message digest and the watermark are re-constructed. Message digest MD′ is obtained from the extracted watermark and the same is compared against the extracted message digest MD. If both message digests MD and MD′ are alike then the watermarked image is treated as authenticated. However, a single bit alteration in the watermarked image is treated as the lack of integrity which makes the watermarked image as unauthenticated. The extraction procedure is described in algorithm 3.4.

Algorithm 3.4:

Input: Watermarked image (I′).

Output: The authenticating watermark (W) and the 128 bits message digest (MD).

Method: The fabricated watermark (along with a message digest) is extracted from the watermarked image (I′) in Legendre Transform (LT) domain. Successive extracted bits reconstruct the watermark from which another message digest (MD) is obtained. Both message digests are compared against each other to verify the authenticity. The full steps of extraction are as follows:

Step 1: The watermarked image (I') is partitioned into 1 x 2 non-overlapping blocks in row major order. Each 1 x 2 block is consisting of pair of pixel components p_i and p_{i+1} of red/green/blue channel where, for all i, $0 \leq i \leq 1$.

Step 2: Legendre Transform (LT) is applied on 1 x 2 sub-matrices of pixel components of red/green/blue channel to compute the pair of transformed components t_i and t_{i+1}.

Step 3: λ_1 /λ_2 bits of the secret bit-stream are successively extracted from first/second embedded component starting from the 1st bit position of the least significant part (i.e., LSB-1) toward higher order bit position. To extract λ bits of secret information from *each pair of embedded components for the payload value of B bits per Byte (bpB), the extraction rule given in equation (3.20) is followed.*

$$\lambda = \begin{cases} \lfloor B \rfloor : \lambda = \lambda_1 \\ \lceil B \rceil : \lambda = \lambda_2 \end{cases} \qquad (3.20)$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all λ, $0 \leq \lambda \leq 3$.

Step 4: For each 8 (eight) bits extraction, construct one alphabet/one primary (R/G/B) color component.

Step 5: Inverse Legendre Transform (LT) is applied on 1 x 2 sub-matrix or pair of embedded components to re-compute the pixel components in spatial domain.

Step 6: Repeat steps 1 to 5 until and unless the fabricated message digest, size and the content of the watermark are recovered.

Step 7: Obtain 128 bits message digest MD' from the extracted watermark.

Step 8: Compare MD' with the extracted MD. If both are matches then the image is authorized, else unauthorized.

Step 9: Stop.

3.2.2.4. Example

Split the carrier image into 1 x 2 non-overlapping blocks of pixel components in a sliding window manner. The red, green and blue sub-matrices of pixel components having the identical block size have been considered for embedding secret bits. The 1 x 2 sub-matrices of adjusted pixel components i.e., R_1, G_1 and B_1 are as follows:

$$R_1 = [245\ 69] \quad G_1 = [92\ 202] \quad B_1 = [11\ 110]$$

The problem of overflow and underflow is resolved by a pre-embedding pixel adjustment which adjusts the upper and lower bounds of the pixel components in the specified range as given in equation (3.17). In this example, the payload value is three (i.e., B = 3) and the modified upper bound (UB) and the lower bound (LB) are obtained as follows:

$$UB = \left(2^8 - 2^{\lceil 3 \rceil + 1}\right) = (2^8 - 2^4) = (256 - 16) = 240$$

$$LB = 2^{\lceil 3 \rceil + 1} = 2^4 = 16$$

Pixel component outside this range is immediately adjusted bases on a pixel adjustment method to avoid the overflow and underflow. The 1 x 2 sub-matrices of adjusted pixel components are obtained as follows:

$$R_1 = [240\ 69] \quad G_1 = [92\ 202] \quad B_1 = [16\ 110]$$

Each 1 x 2 sub-matrix or pair of pixel components is converted into transform domain based on the Legendre Transform (LT). The 1 x 2 sub-matrices of transformed components corresponding to red, green and blue channels are obtained as here under:

$$T(R_1) = [240\ 378] \quad T(G_1) = [92\ 496] \quad T(B_1) = [16\ 236]$$

To fabricate the secret bit-stream *"101000010110000111"* into the transformed components, the embedding rule as given in equation (3.19) has been applied. In this example, three bits are fabricated ($\lambda = \lambda_1 = \lambda_2 = 3$) on each transformed component starting from LSB-1 toward the higher order bit position. The LSB-0 is kept unaffected to avoid the occurrence of fractional pixel components subsequent to embedding. Hence, the 1 x 2 sub-matrices of embedded components are:

$$T'(R_1) = [250\ 368] \quad T'(G_1) = [84\ 502] \quad T'(B_1) = [16\ 238]$$

Each pair of pixel components is re-computed from the pair of embedded components through the inverse Legendre Transform (ILT). The 1 x 2 sub-matrices consisting of pixel components are obtained as follows:

$$R'_1 = [250\ 59] \quad G'_1 = [84\ 209] \quad B'_1 = [16\ 111]$$

All re-computed pixel components become non-fractional, non-negative and less than or equal to 255.

3.2.2.5. Results and Discussions

The fabrication of varying sizes of the watermark (i.e., "Gold-Coin") into twenty benchmark color images [130, 131] of dimension 512 x 512 are accomplished in Legendre Transform (LT) domain based on a 1 x 2 block based watermarking technique (WLT_1x2). Quality and payload has been chosen as the principal parameters for assessing the performance of WLT_1x2. Standard quality metrics namely peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD) and standard deviation error (SDE) has been used to summarize the results. Comparison is made among WLT_1x2, WLT_2x2, WDHT_1x2 and

Varsaki et al.’s (Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] as well as Discrete Gould Transform based data hiding scheme (DGTDHS) [129]) schemes respectively. The extracted “Gold Coin” and the different states (before and after embedding the “Gold-Coin”) of modifications of carrier/cover images viz. “Lena”, “Baboon” and “Pepper” are shown in Fig 3.6.

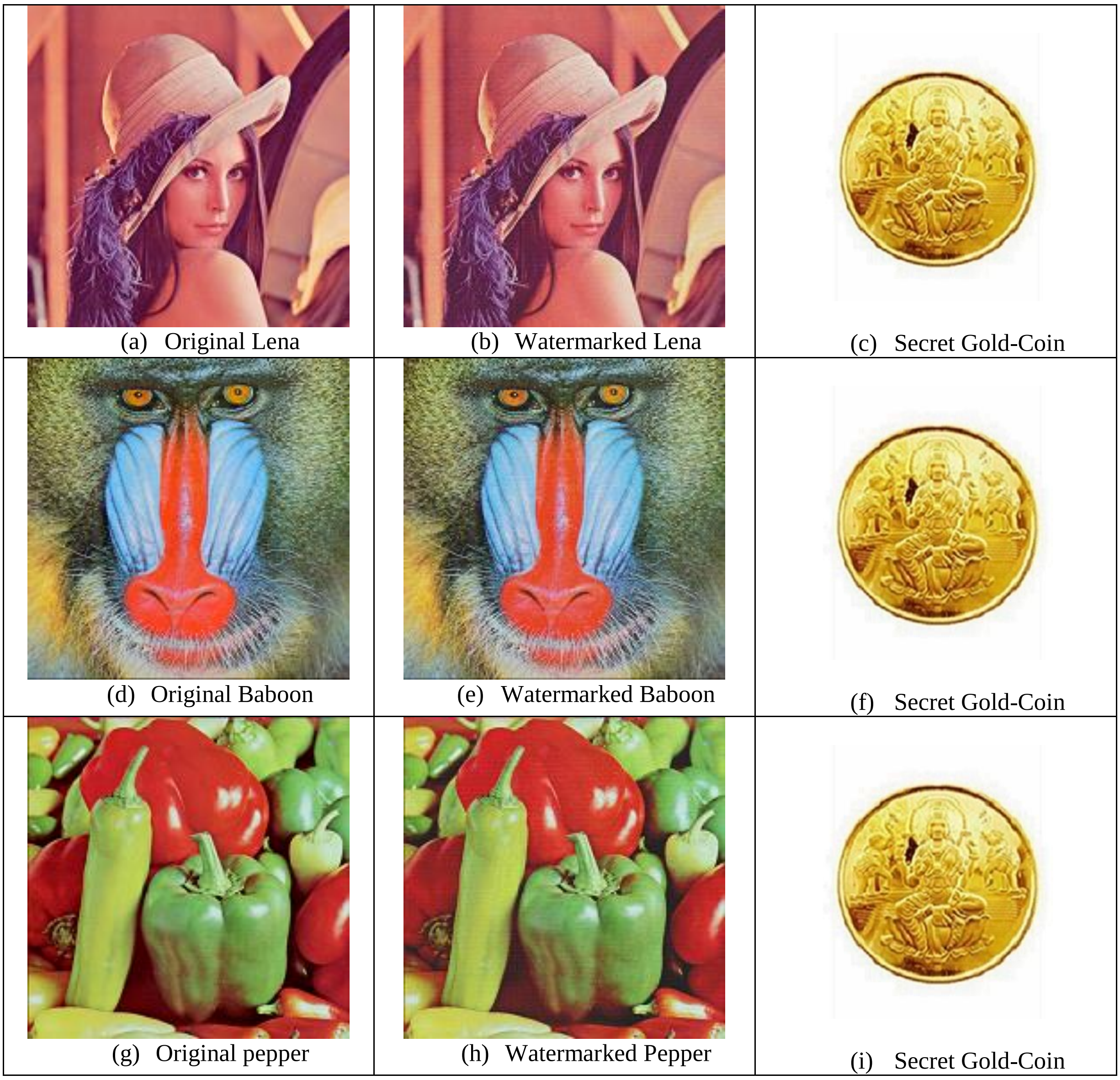

(a) Original Lena
(b) Watermarked Lena
(c) Secret Gold-Coin
(d) Original Baboon
(e) Watermarked Baboon
(f) Secret Gold-Coin
(g) Original pepper
(h) Watermarked Pepper
(i) Secret Gold-Coin

Fig 3.6. Cover, watermarked and authenticating watermark image in the proposed WLT_1x2 technique

The 1 x 2 block based watermarking (WLT_1x2) revealed the computed results of peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) for WLT_1x2 with respect to the payload range [0.5 – 3 bpB] as given in table 3.2. Average values are also computed from various metrics of twenty carrier images at variable payload to summarize the results.

The minimum PSNR obtained is 28.90 dB at 3 bpB of payload for the "Desert" while, the maximum PSNR obtained is 54.22 dB at 0.5 bpB of payload for the "Barnfall". Since, the minimum obtained PSNR is below 30 dB, the quality of "Desert" is considered to be severely degraded. However, the average PSNR of WLT_1x2 scheme is 31.68 dB (i.e., ≥ 30 dB) at 3 bpB and hence, the PSNR value of "Desert" is considered as an exceptional case and the watermarked images with high transparency are achieved for usual case [148]. The MSE analysis revealed the lowest MSE value of 0.24 for "Barnfall" at 0.5 bpB and that of the highest MSE value of 83.72 for "Desert" at 3 bpB respectively. Minimum and maximum values of IF, SSIM and UIQ (as obtained at 3 and 0.5 bpB) are belonging to the following ranges: [0.98738 (Desert) - 0.999992 (Airplane)], [0.947764 (Pepper) - 0.999898 (San Diego)] and [0.488121 (Splash) - 0.99933 (San Diego)]. Generally, the values of IF, SSIM and UIQ are lies into the range [0, 1]. Since, the obtained values of IF, SSIM and UIQ are closer to one, higher similarity between original and watermarked images is perceived with respect to increasing values of payload.

Table 3.2. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WLT_1x2 technique

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.126733	0.251424	0.999984	0.999645	0.994757
	1.0	46.673650	1.398670	0.999911	0.999211	0.970898
	1.5	44.255199	2.440958	0.999845	0.997642	0.955090
	2.0	39.333405	7.581232	0.999520	0.996031	0.884897
	2.5	37.164520	12.491893	0.999207	0.989866	0.844333
	3.0	32.452557	36.967887	0.997659	0.982703	0.699960
Baboon	0.5	54.063045	0.255138	0.999986	0.999870	0.998966
	1.0	46.674475	1.398404	0.999925	0.999697	0.994214
	1.5	44.205672	2.468954	0.999869	0.999043	0.989803
	2.0	39.316953	7.610006	0.999597	0.998431	0.972835
	2.5	37.159787	12.505514	0.999337	0.995696	0.958962
	3.0	32.487259	36.673670	0.998058	0.993099	0.913175
Pepper	0.5	48.030595	1.023342	0.999900	0.990116	0.977738
	1.0	44.546889	2.282398	0.999791	0.989079	0.961660
	1.5	40.240276	6.152515	0.999416	0.979037	0.941158
	2.0	37.303887	12.097387	0.998907	0.976256	0.886102
	2.5	33.260533	30.692129	0.997084	0.956312	0.835816
	3.0	30.323868	60.352263	0.994515	0.947764	0.697165

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	53.973755	0.260438	0.999992	0.999580	0.983352
	1.0	46.642680	1.408679	0.999959	0.999063	0.924539
	1.5	44.187715	2.479184	0.999929	0.997206	0.891684
	2.0	39.295268	7.648099	0.999781	0.995233	0.777706
	2.5	37.222470	12.326316	0.999647	0.988614	0.726228
	3.0	32.518097	36.414189	0.998957	0.980505	0.578756
Sailboat	0.5	53.335572	0.301663	0.999984	0.999693	0.995550
	1.0	46.465891	1.467206	0.999925	0.999326	0.976773
	1.5	43.763723	2.733430	0.999862	0.997909	0.964682
	2.0	39.119074	7.964763	0.999598	0.996561	0.913071
	2.5	36.538106	14.430110	0.999275	0.990402	0.879028
	3.0	32.095784	40.133027	0.997979	0.984732	0.771903
Earth	0.5	54.137597	0.250796	0.999985	0.999730	0.997664
	1.0	46.681633	1.396101	0.999917	0.999408	0.986370
	1.5	44.278366	2.427972	0.999855	0.998239	0.978507
	2.0	39.362953	7.529827	0.999553	0.997046	0.937696
	2.5	37.264283	12.208211	0.999268	0.992242	0.906100
	3.0	32.524425	36.361164	0.997814	0.986943	0.783107
San Diego	0.5	54.122957	0.251642	0.999990	0.999898	0.999330
	1.0	46.684078	1.395315	0.999948	0.999778	0.996197
	1.5	44.278823	2.427716	0.999909	0.999332	0.994065
	2.0	39.358912	7.536837	0.999719	0.998896	0.982969
	2.5	37.345053	11.983261	0.999553	0.997280	0.975788
	3.0	32.586833	35.842397	0.998664	0.995199	0.938069
Splash	0.5	49.266754	0.769847	0.999917	0.989394	0.963196
	1.0	45.155668	1.983871	0.999809	0.988233	0.895883
	1.5	41.318746	4.799607	0.999512	0.981487	0.855803
	2.0	37.952895	10.418155	0.999004	0.978390	0.726278
	2.5	34.578656	22.657639	0.997708	0.964811	0.664045
	3.0	31.144117	49.965309	0.995291	0.953640	0.488121
Oakland	0.5	51.286666	0.483519	0.999968	0.999417	0.997710
	1.0	45.888116	1.675979	0.999903	0.999101	0.992149
	1.5	42.583242	3.587203	0.999778	0.997449	0.986788
	2.0	38.637934	8.897888	0.999489	0.996455	0.965074
	2.5	35.567022	18.045883	0.998876	0.990783	0.947340
	3.0	31.714663	43.814112	0.997457	0.986169	0.8747541

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	54.153834	0.249860	0.999991	0.999519	0.993394
	1.0	46.686176	1.394641	0.999950	0.998943	0.963030
	1.5	44.276420	2.429060	0.999913	0.996807	0.943610
	2.0	39.389892	7.483264	0.999733	0.994761	0.852896
	2.5	37.396761	11.841431	0.999577	0.987160	0.801263
	3.0	32.615526	35.606369	0.998733	0.977295	0.642843
Anhinga	0.5	50.833323	0.536720	0.999958	0.999809	0.882627
	1.0	45.879051	1.679481	0.999870	0.999594	0.869365
	1.5	42.437131	3.709943	0.999715	0.998657	0.883403
	2.0	37.710038	11.017333	0.999154	0.997813	0.812991
	2.5	35.487719	18.378431	0.998591	0.993645	0.802309
	3.0	31.403904	47.064118	0.996393	0.986573	0.715827
Athens	0.5	51.884716	0.421316	0.999966	0.999864	0.971366
	1.0	46.182203	1.566246	0.999874	0.999663	0.956086
	1.5	43.344067	3.010742	0.999759	0.999003	0.963165
	2.0	37.579178	11.354354	0.999091	0.998278	0.877342
	2.5	36.063297	16.097211	0.998714	0.993945	0.859437
	3.0	31.873118	42.244335	0.996626	0.983354	0.737404
Bardowl	0.5	50.122462	0.632170	0.999934	0.999465	0.997727
	1.0	45.519226	1.824558	0.999815	0.999160	0.994449
	1.5	42.039237	4.065898	0.999584	0.996874	0.988759
	2.0	37.492234	11.583955	0.998834	0.995516	0.968705
	2.5	34.145141	25.036073	0.997426	0.980587	0.933755
	3.0	30.660413	55.852041	0.994323	0.974086	0.877491
Barnfall	0.5	54.225905	0.245747	0.999961	0.999762	0.998852
	1.0	46.822993	1.351390	0.999782	0.999447	0.992625
	1.5	44.144272	2.504108	0.999592	0.998245	0.989757
	2.0	39.051513	8.089636	0.998696	0.997231	0.963632
	2.5	36.843526	13.450168	0.997745	0.990659	0.947512
	3.0	32.349955	37.851648	0.993818	0.984484	0.863827
Butrfly	0.5	52.613555	0.356225	0.999974	0.999852	0.995605
	1.0	46.341627	1.509793	0.999892	0.999602	0.988149
	1.5	43.619462	2.825752	0.999799	0.998800	0.986605
	2.0	37.924652	10.486127	0.999255	0.997900	0.945157
	2.5	36.309110	15.211402	0.998923	0.994634	0.930100
	3.0	31.880293	42.174605	0.997037	0.987838	0.824046

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	52.199342	0.391873	0.999944	0.999695	0.749617
	1.0	46.239258	1.545804	0.999782	0.999105	0.740411
	1.5	43.420696	2.958085	0.999584	0.997132	0.745851
	2.0	37.692885	11.060934	0.998440	0.994582	0.700370
	2.5	35.679807	17.583268	0.997515	0.985634	0.690683
	3.0	30.799445	54.092344	0.992346	0.972174	0.608680
Bodie	0.5	50.407332	0.592034	0.999893	0.999533	0.981783
	1.0	45.415760	1.868548	0.999675	0.999275	0.970274
	1.5	42.061886	4.044750	0.999286	0.997336	0.962983
	2.0	38.486375	9.213886	0.998393	0.996011	0.943656
	2.5	34.077052	25.431686	0.995404	0.974503	0.905474
	3.0	30.814801	53.901415	0.990481	0.967301	0.842001
Bluheron	0.5	54.188895	0.247851	0.999969	0.999756	0.996671
	1.0	47.132107	1.258547	0.999846	0.999437	0.987203
	1.5	44.352599	2.386824	0.999708	0.998100	0.977976
	2.0	39.831876	6.759159	0.999172	0.997098	0.946550
	2.5	37.651839	11.165969	0.998631	0.992711	0.923303
	3.0	32.258962	38.653077	0.995269	0.986794	0.780911
Colomtn	0.5	52.498062	0.365825	0.999971	0.999762	0.985223
	1.0	46.275919	1.532810	0.999880	0.999486	0.976688
	1.5	43.369253	2.993333	0.999765	0.998476	0.973281
	2.0	38.833227	8.506631	0.999336	0.997375	0.946811
	2.5	36.457339	14.700984	0.998852	0.993505	0.930043
	3.0	32.392913	37.479084	0.997068	0.988608	0.857504
Desert	0.5	45.918758	1.664196	0.999731	0.997251	0.994053
	1.0	43.383755	2.983353	0.999535	0.996765	0.990937
	1.5	38.734373	8.702480	0.998623	0.990758	0.981069
	2.0	36.104940	15.943597	0.997557	0.989134	0.965933
	2.5	31.412550	46.970514	0.992729	0.964885	0.926515
	3.0	28.902098	83.727957	0.987380	0.956721	0.871975
Average Case	0.5	52.069490	0.477581	0.999950	0.998581	0.972759
	1.0	46.064560	1.646090	0.999849	0.998169	0.956395
	1.5	43.045560	3.457426	0.999665	0.995877	0.947702
	2.0	38.488900	9.439154	0.999141	0.994450	0.898534
	2.5	35.881230	18.160400	0.998203	0.985894	0.869402
	3.0	31.689950	45.258550	0.995793	0.978799	0.768376

Proposed WLT_1x2 is compared against Varsaki et al.'s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] as well as Discrete Gould Transform based data hiding scheme (DGTDHS) [129] in terms of PSNR (dB) and payload (bpB) as depicted in fig. 3.7. Five color images namely, "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively have been considered to compute the results. The PSNR variation for DPTHDI [88] and DGTDHS [129] is observed at 0.25 and 1 bpB respectively however, the payload values are fixed as well as consisting low fabrication density. On the contrary, the WLT_1x2 is focused on variable payload for the range [0.5 – 3 bpB]. In contrast to DPTHDI [88], proposed WLT_1x2 ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads for "Lena", 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Baboon", 0.5, 1, 1.5 and 2 bpB of payloads for "Pepper", 0.5, 1 and 1.5 bpB of payloads for "Airplane" and 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for "Sailboat" respectively. In comparison with DGTDHS [129], proposed WLT_1x2 gives slightly less PSNR at 1 bpB of payload; interestingly, the proposed scheme offers variable payload that offers a spread from 0.5 to 3 bpB. The obtained PSNR values in WDHT_1x2 for all five images are above the threshold limit (i.e., ≥ 30 dB) and hence, the obtained watermarked images are perceived as the high-quality watermarked images [148].

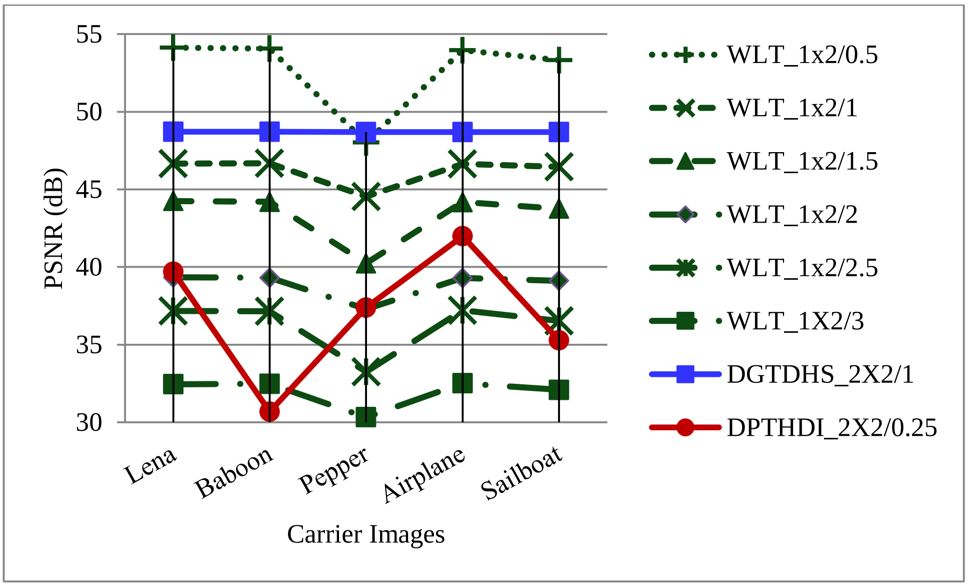


Fig. 3.7. Performance analysis of PSNR (dB) for variable payload based WLT_1x2 and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) techniques based on five color images

It is seen from table 3.2 that the maximum and minimum values of average PSNR are 52.06 dB and 31.68 dB respectively. The computation of average PSNR for WLT_1x2 is made on twenty benchmark images as given in fig. 1.1. In average case, high transparency is achieved for the payload variation of 0.5 to 3 bpB since, the average PSNR values are falling above 30 dB [148]. Fig. 3.8 illustrates the average PSNR analysis among WLT_1x2, WLT_2x2, WDHT_1x2, DPTHDI [88] and DGTDHS [129] respectively. In contrast to WLT_2x2 and WDHT_1x2, the WLT_1x2 proves the betterment in terms of average PSNR for payload range [0.5 – 3 bpB]. In DPTHDI [88], the average PSNR of 37.40 dB is computed from "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" images at 0.25 bpB of payload. On the contrary, the average PSNR for DGTDHS [129] is 48.70 dB as computed from "Lighthouse", "Elaine", "Lenna", "Boat", "F16" images at 1 bpB of payload. Compared to DPTHDI [88], the average PSNR of WLT_1x2 ensured better PSNR for payload range [0.5 – 2 bpB] and that of average PSNR value for WLT_1x2 is slightly lacking than DGTDHS [129] at 1 bpB however, the WLT_1x2 offered variable payload for a range from 0.5 to 3 bpB.

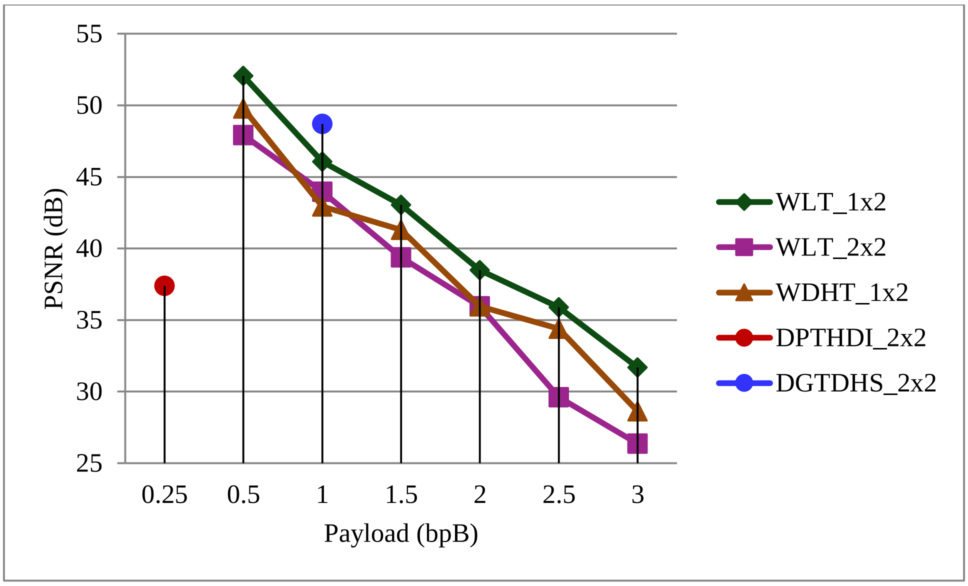


Fig. 3.8. Graphical representation of variation of average PSNR (dB) with respect to payload for WLT_1x2, WLT_2x2, WDHT_1x2 and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

In fig. 3.9, the standard deviation (SD) analysis has been made for "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively by computing the standard deviation (SD) as the average values analogous to red, green and blue channels. The graph ensured that the standard deviation (SD) is almost same up to the payload value of 2 bits per Byte (bpB). On the contrary, at 2.5 and 3 bpB of payload, the standard deviation (SD) decreases for "Pepper" and "Sailboat" at the same time as increases for "Lena", "Baboon" and "Airplane". It is to be noted that the standard deviation (SD) at 0 bpB of payload designates the standard deviation (SD) of the original image.

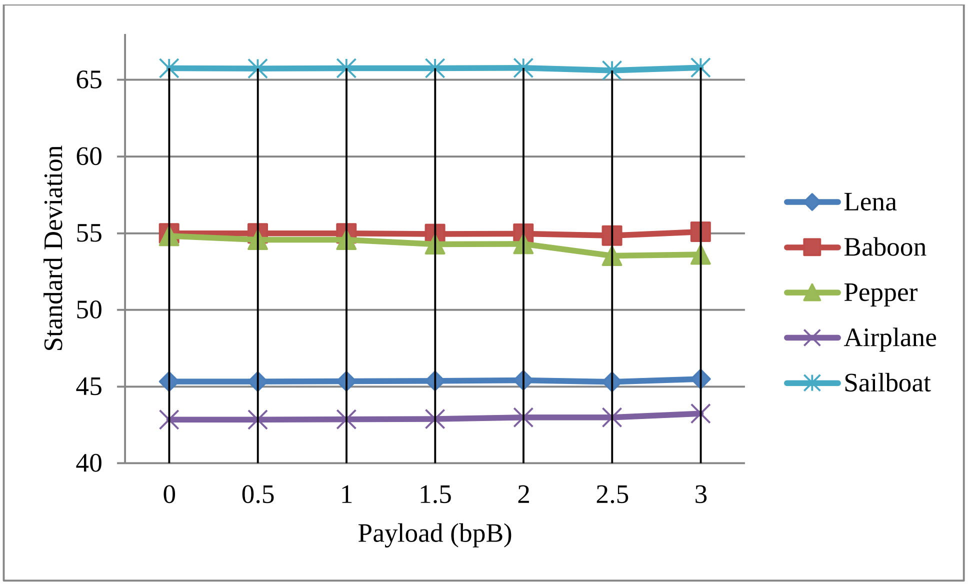


Fig. 3.9. Graphical representation of standard deviation (SD) for WLT_1x2 with respect to 0, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

An extensive analysis of structural quality distortion of the watermarked images is measured in terms of standard deviation error (SDE). The analysis is carried out for WLT_1x2 by comparing the watermarked images against the original image for the payload variation of 0.5 to 3 bpB. The standard deviation error (SDE) is very low up to 2 bpB and afterward it is increasing which results significant deviations in the watermarked images from the original image. It is to be noted that the standard deviation error (SDE) is comparatively high for “Airplane” and “Pepper” images at higher payload values, and thus a significant dispersion is observed in the watermarked images.

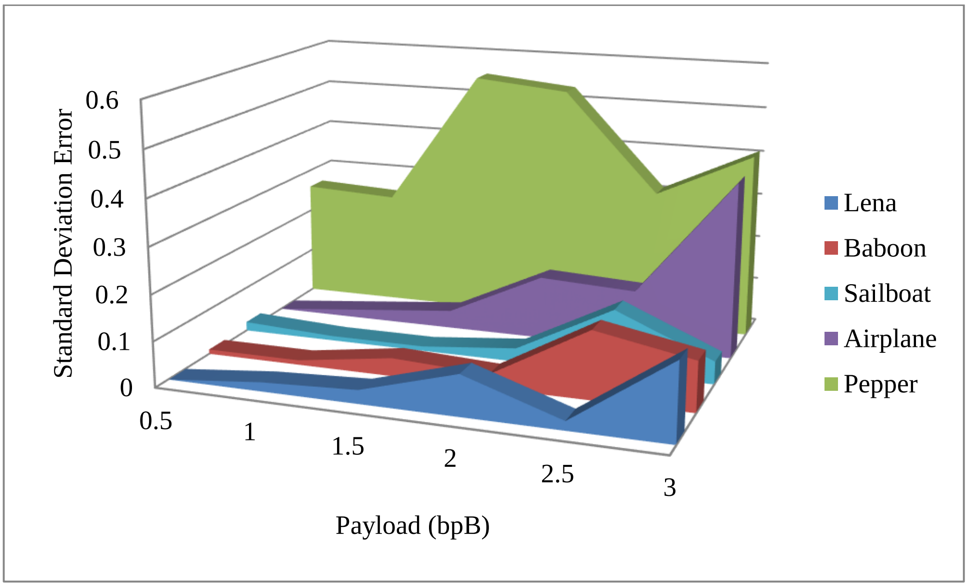


Fig. 3.10. Graphical representation of standard deviation error (SDE) for WLT_1x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

In fig. 3.11, the standard deviation error (SDE) is analyzed for WLT_2x2 and WLT_1x2 techniques. The error is almost 0 for “Lena”, “Baboon”, “Airplane” and “Sailboat” for the payload range (0.5 – 2 bpB). As the payload increases from 2 bpB, the error also increases however, the rate of increase has been minimized tremendously by choosing the block size as 1 x 2 instead of 2 x 2. The error rate is apparently high for “Pepper” which could be improved by incorporating the quality enhancement of chapter 7 or GA based optimization of chapter 8.

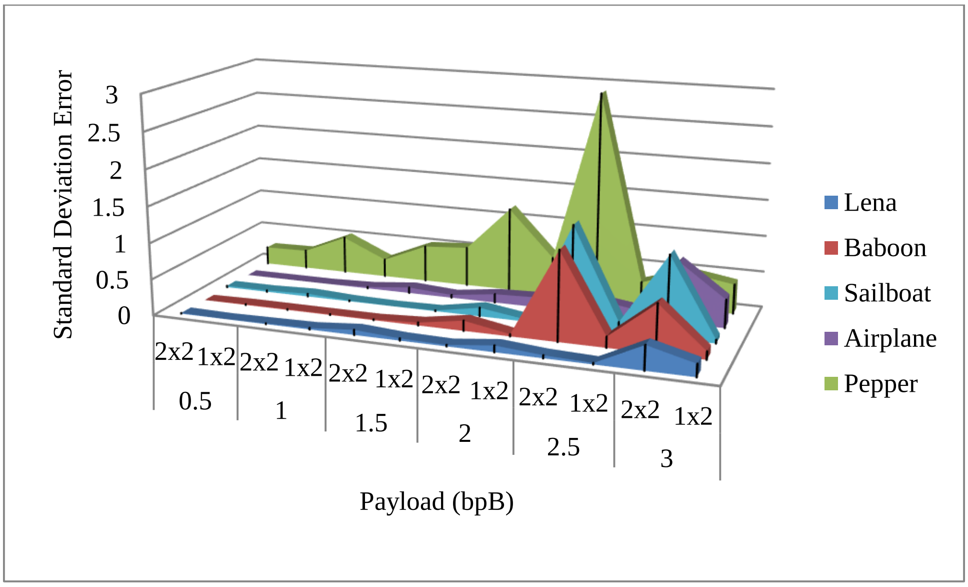


Fig. 3.11. Pictorial representation of variation of standard deviation error (SDE) between WLT_1x2 and WLT_2x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

3.3. Salient Features

To address the shortcomings of existing techniques, two novel watermarking schemes viz. WLT_2x2 and WLT_1x2 have been proposed in Legendre Transform (LT) domain that offers variable payloads with acceptable visual imperceptibility. The basic shortcomings includes: high distortion, less payload, lacking in usage of color image as the cover and the high computational overhead.

Few important features of the proposed techniques have been summarized in this section as follows.

Both schemes perform a pure integer calculation which is a significant advantage of Legendre Transform (LT). The least significant bit of each transformed component is kept unaltered to avoid the formation of fractional pixel components during inverse transform. Thus, the computational overhead is significantly reduced. The overflow/underflow situation has been tactfully avoided through the pre-embedding pixel adjustment. A re-adjustment of transformed component has also been incorporated as the transformed components in Legendre Transform (LT) domain are highly sensitive against a small change in the watermarked image. The higher transparency and variable payloads has also been achieved by embedding the watermark into the color images. Most of the existing schemes offers fixed payload which is relatively low as compared to proposed WLT_2x2 and WLT_1x2 schemes. Both schemes offers variable payload (i.e., 0.5 to 3 bpB) with reference to perceptible quality of the watermarked image. The distortion can further be reduced by introducing the quality enhancement as proposed in chapter 7 and genetic algorithm (GA) based optimization of chapter 8 as post-embedding operation. In contrast to Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes, both WLT_2x2 and WLT_1x2 schemes ensured variable payload with reduced quality degradation.

Chapter 4

Watermarking based on Binomial Transform (WBT)

4.1 Introduction

This chapter deals with the fragile watermarking schemes based on Binomial Transform (BT) to verify the authenticity of color images. Binomial Transform (BT) converts the non-overlapping blocks of the carrier image into transform domain. Secret bits (along with a message digest) are concealed into the transformed components based on a predefined embedding rule to achieve variable payload by allowing a considerable quality distortion. Embedded blocks are re-transformed into the spatial domain through inverse Binomial Transform (IBT) which in succession generates the watermarked image. The recipient retrieves the secret bits through the reverse operation and the re-computed message digest obtained from the secret bits is compared against the extracted message digest for authentication. Simulation results demonstrate that the proposed scheme offers variable payload and considerable quality distortion.

4.2 The Technique

Two novel approaches of fragile watermarking in Binomial Transform (BT) domain have been proposed to authenticate color images. The carrier image is divided into P x Q non-overlapping blocks (where, $1 \leq P \leq 2$ and $Q = 2$) of red, green and blue channels. The pixel adjustment prior to embedding ensured the non-occurrence of overflow and underflow. The P x Q sub-image blocks are subsequently converted into transform domain based on Binomial Transform (BT). The secret bits corresponding to the message digest (MD) obtained from the watermark, size and the content of the watermark are embedded into the transformed components starting from the least significant bit (i.e., LSB-0) position toward higher order bit position. The components of Binomial Transform (BT) are highly sensitive against a small modification subsequent to embedding. Therefore, a rarely used re-adjustment operation has also been incorporated to ensure the avoidance of overflow and underflow. Due to this re-adjustment the fabricated watermark bits are not affected. Inverse Binomial Transform (IBT) is applied on the identical sub-image block to obtain the pixel components in spatial domain. The repetitive execution of the above steps yields the construction of the watermarked image. At the recipient end, the watermarked image is decomposed into P x Q non-overlapping blocks (where, $1 \leq P \leq 2$ and $Q = 2$) corresponding to red, green and blue channels. Consequently, the secret bits are extracted from the transformed components based on a pre-defined extraction rule. The inverse Binomial Transform (IBT) is applied on the identical sub-image block to convert back the embedded components into spatial domain. Each 8 (eight) bits extraction of the secret bit-stream, construct one alphabet/one primary (R/G/B) color component. The process is repeated until and unless the message digest (MD), size and the content of the watermark are extracted.

Any kind of intentional or unintentional perturbation (such as filtering, blurring, lossy compression, channel noise addition etc.) on the watermarked image may destroy the fabricated watermark which results the authentication process as failed. If the alteration is minor then it is really hard to distinguish the attacked image from the watermarked image. However, the frequency distribution is transformed in such a way that the decoder fails to extract the hidden data. In this technique, the recipient operate the authentication process by comparing the extracted message digest MD with the re-computed message digest MD', where, MD' is obtained from the extracted watermark. If the bit-streams corresponding to the extracted message digest MD and the re-computed message digest MD' are identical, then the authentication process is considered to be successful, otherwise, it is said to be unsuccessful. Hence, the proposed techniques can extensively be used for tamper detection.

Section 4.2.1 of this chapter deals with 2 x 2 block based watermark fabrication that of section 4.2.2 describes 1 x 2 block based watermark fabrication.

4.2.1 2 x 2 Block based Watermark Fabrication

Binomial Transform (BT) [135, 136] produces integer output sequence with respect to an integer-valued input sequence. According to this concept, each 2 x 2 sub-block of pixel/transformed components is treated as the linear sequence of four integer-valued numbers. On applying Binomial Transform (BT) over the pixel components $\{p_\gamma\}$, the transformed components $\{t_\gamma\}$ are obtained as given in equation (4.1).

$$t_\gamma = \sum_{k=0}^{\gamma} (-1)^k \binom{\gamma}{k} p_k \tag{4.1}$$

Similarly, the inverse Binomial Transform (IBT) is used to convert back the transformed components into the spatial domain as given in equation (4.2).

$$p_\gamma = \sum_{k=0}^{\gamma} (-1)^k \binom{\gamma}{k} t_k \tag{4.2}$$

The formulation of Binomial Transform (BT) for image sub-block of size 2 x 2 has been expressed in equation (4.3).

$$t_k = \begin{cases} p_k : k = 0 \\ -p_k + p_{k-1} : k = 1 \\ p_k - 2p_{k-1} + p_{k-2} : k = 2 \\ -p_k + 3p_{k-1} - 3p_{k-2} + p_{k-3} : k = 3 \end{cases} \tag{4.3}$$

where, k is a non-negative number lies in the range $0 \leq k \leq 3$, the transformed and pixel components are denoted as t_k and p_k respectively.

By applying the inverse Binomial Transform (IBT) of equation (4.2), the pixel components corresponding to each 2 x 2 masks are obtained using equation (4.4).

$$p_k = \begin{cases} t_k : k = 0 \\ -t_k + t_{k-1} : k = 1 \\ t_k - 2t_{k-1} + t_{k-2} : k = 2 \\ -t_k + 3t_{k-1} - 3t_{k-2} + t_{k-3} : k = 3 \end{cases} \tag{4.4}$$

where, k is a non-negative number lies in the range $0 \leq k \leq 3$, the pixel and transformed components are denoted as p_k and t_k respectively.

It can be observed from equation (4.3) and (4.4) that the Binomial Transform (BT) is a self-inverse symmetric transformation.

The proposed technique has been elaborately discussed in the following sections. The algorithm for insertion, the re-adjustment, the algorithm for extraction and an example are described in detail in section 4.2.1.1, 4.2.1.2, 4.2.1.3 and 4.2.1.4 respectively. Results and discussions have been described in great detail at section 4.2.1.5.

4.2.1.1 Insertion

The carrier image is decomposed into 2 x 2 non-overlapping blocks in a row major order. The pre-embedding pixel adjustment is made to avoid overflow and underflow. Binomial Transform (BT) is applied on each 2 x 2 sub-image block of red/green/blue channel to obtain the transformed components. Embedding is done with the payload variation of 0.5 to 3 bpB by fabricating the secret bits corresponding to the message digest, size and content of the watermark into the transformed components. The secret bit fabrication is started from the least significant bit (i.e., LSB-0) position toward higher order bit position. Binomial Transform (BT) is highly sensitive against a minor change in the values of transformed components during embedding. To avoid overflow and underflow, an additional re-adjustment operation is applied in transform domain. Inverse Binomial Transform (IBT) is applied on 2 x 2 embedded blocks to re-compute the pixel components. Successive steps ensured the generation of the watermarked image in spatial domain.

Algorithm 4.1:

Input: Carrier/cover image (I) and authenticating watermark image (W).

Output: Watermarked image (I′).

Method: The watermark (along with a message digest) is embedded into the carrier images in Binomial Transform (BT) domain. Embedding bits in transform domain offers variable payload, less distortion and improved security. The detailed steps of embedding are as follows:

Step 1: Obtain a message digest (MD) from the secret watermark.

Step 2: The authenticating watermark size (in bits) is obtained by embedding the watermark into the three sub-matrices of the U x V color image as given in equation (4.5).

$$W_{size} = B \times \big(3 \times (U \times V)\big) - (MD + L) \qquad (4.5)$$

where B, MD and L represents the embedding payload in terms of bits per Byte, the message digest obtained from the watermark and the header information corresponding to the size of the watermark respectively. The MD and L are consisting of 128 and 32 bits while the usual values of B are 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively.

Step 3: The cover image (I) is decomposed into 2 x 2 non-overlapping blocks in row major order. Each block is consisting of four pixel components $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ *and* $p_{i+1,j+1}$ *of red/green/blue channel where, for all i and j,* $0 \leq i, j \leq 1$.

Step 4: Pixel components (p) are adjusted to avoid overflow and underflow for the payload value of B bits per Byte as given in equation (4.6).

$$p = \begin{cases} \left(2^8 - 2^{\lfloor B \rfloor + 2}\right) : p \geq \left(2^8 - 2^{\lfloor B \rfloor + 2}\right) \\ 2^{\lfloor B \rfloor + 2} : p \leq 2^{\lfloor B \rfloor + 2} \end{cases} \qquad (4.6)$$

Step 5: Apply Binomial Transform (BT) on 2 x 2 sub-matrices of pixel components to obtain the transformed components $t_{i,j}$, $t_{i,j+1}$, $t_{i+1,j}$ *and* $t_{i+1,j+1}$ *respectively.*

Step 6: $\lambda_1/\lambda_2/\lambda_3/\lambda_4$ bits from the secret bit-stream (corresponding to the message digest (MD), size (L) and the content (W) of the watermark) are subsequently retrieved and then fabricated on first/second/third/fourth transformed component starting from the least significant bit (LSB-0) position toward higher order bit position. The generalized form of λ bits of secret information fabrication on each transformed component for the payload value of B bits per Byte (bpB) is derived from the expression given in equation (4.7).

$$\lambda = \begin{cases} \lceil B \rceil : \lambda = \lambda_1 \\ \lceil B \rceil - 1 : \lambda = \lambda_2 \\ \lfloor B \rfloor : \lambda = \lambda_3 \\ \lfloor B \rfloor + 1 : \lambda = \lambda_4 \end{cases} \tag{4.7}$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all ($\lambda_1, \lambda_2, \lambda_3, \lambda_4$), $0 \leq \lambda_1, \lambda_2, \lambda_3, \lambda_4 \leq 4$.

Step 7: Inverse Binomial Transform (IBT) is applied on each 2 x 2 sub-matrix of embedded components to get back the pixel components.

Step 8: Apply re-adjustment operation on the embedded components to avoid overflow and underflow, if necessary.

Step 9: Repeat steps 3 to 8 for the embedding of watermark size, content and the message digest MD respectively. The block embedding operation in succession produces the watermarked image (I') in spatial domain.

Step 10: Stop.

4.2.1.2 Re-Adjustment

Binomial Transform (BT) is highly sensitive against a small change on the value of transformed component. Therefore, the additional re-adjustment operation (as discussed in section 3.2.1.2) has been applied to avoid the overflow/underflow. In general, the overflow and underflow may occur due to embedding followed by inverse Binomial Transform (IBT) which results the following situations:

- The re-computed pixel component may be negative (-ve).
- The re-computed pixel component may be greater than the maximum value (i.e., 255).

The concept of re-adjust phase is to handle the above two serious problems by adjusting the first transformed component of the 2 x 2 mask.

In this phase, if the re-computed pixel component is negative (-ve), the operation applied for each 2 x 2 sub-matrix of transformed components is as follows:

$$t_{0,0} = t_{0,0} + 2^{\lambda_1+1} \tag{4.8}$$

where, $t_{0,0}$ is the first component of the 2 x 2 sub-matrix and λ_1 is used to specify the number of bits fabricated on $t_{0,0}$. The above process is repeated until and unless all the re-computed pixel components become positive.

If the re-computed pixel component exceeds the maximum value of a byte (i.e., greater than 255), then the operation applied for each 2 x 2 sub-matrix is as follows:

$$t_{0,0} = t_{0,0} - 2^{\lambda_1+1} \tag{4.9}$$

where, $t_{0,0}$ is the first component of the 2 x 2 sub-matrix and λ_1 is used to specify the number of bits fabricated on $t_{0,0}$. The process is repeated till the re-computed pixel components become less than or equal to 255.

4.2.1.3 Extraction

The recipient decomposed the watermarked image into 2 x 2 non-overlapping blocks in a sliding window manner. Binomial Transform (BT) is applied on each 2 x 2 sub-image block of red/green/blue channel to obtain the components in transform domain. The reverse methodology is followed to extract the fabricated bits corresponding to the watermark size, content and the message digest (MD) from the transformed components. The process is repeated till the watermark (along with the message digest) is extracted. Message digest MD′ is re-computed from the extracted watermark which in turn is compared with the extracted message digest MD for authentication. The extraction procedure is described in algorithm 4.2.

Algorithm 4.2:

Input: Watermarked image (I′).

Output: The authenticating watermark (W) and the 128 bits message digest (MD).

Method: The watermark (along with a message digest MD) is extracted from the watermarked image (I′) in Binomial Transform (BT) domain. Successive extracted bits reconstruct the watermark from which another message digest (MD′) is obtained. Both MD and MD′ are compared between each other to verify the authenticity. Detailed steps of extraction are as follows:

Step 1: The watermarked image (I') is partitioned into 2 x 2 non-overlapping blocks in row major order. Each 2 x 2 block is consists of four pixel components $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$ of red/green/blue channel where, the values of i and j lies in the range $0 \leq i \leq 1$and $0 \leq j \leq 1$.

Step 2: Binomial Transform (BT) is applied on 2 x 2 sub-matrix of pixel components to obtain the transformed components $t_{i,j}$, $t_{i,j+1}$, $t_{i+1,j}$ and $t_{i+1,j+1}$ respectively.

Step 3: λ_1 /λ_2 /λ_3 /λ_4 bits of the secret bit-stream are successively extracted from first/second/third/fourth transformed component of red/green/blue channel for the payload value of B bits per Byte (bpB) starting from the least significant bit (LSB-0) position toward higher order bit position. The generalized form of λ bits of secret information extraction from each transformed component is expressed in equation (4.10).

$$\lambda = \begin{cases} \lceil B \rceil : \lambda = \lambda_1 \\ \lceil B \rceil - 1 : \lambda = \lambda_2 \\ \lfloor B \rfloor : \lambda = \lambda_3 \\ \lfloor B \rfloor + 1 : \lambda = \lambda_4 \end{cases} \qquad (4.10)$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all ($\lambda_1, \lambda_2, \lambda_3, \lambda_4$), $0 \leq \lambda_1, \lambda_2, \lambda_3, \lambda_4 \leq 4$.

Step 4: An alphabet / primary (R/G/B) component is reconstructed from each 8 (eight) bits of watermark extraction.

Step 5: Inverse Binomial Transform (IBT) is applied on each 2 x 2 sub-matrix of embedded components to convert it back into spatial domain.

Step 6: Repeat steps 1 to 5 to complete the extraction of the message digest (MD), size and the content of the authenticating watermark.

Step 7: Obtain 128 bits message digest MD' from the extracted watermark.

Step 8: Compare MD' with the extracted MD. If both are matches then the image is authorized, else unauthorized.

Step 9: Stop.

4.2.1.4 Example

The carrier image is decomposed into 2 x 2 non-overlapping blocks in row major order. Three 2 x 2 sub-matrices namely R$_1$, G$_1$ and B$_1$ corresponding to red, green and blue channels have been considered for embedding. The 2 x 2 sub-matrices of pixel components are as given below:

$$R_1 = \begin{bmatrix} 230 & 72 \\ 17 & 155 \end{bmatrix} G_1 = \begin{bmatrix} 62 & 215 \\ 56 & 22 \end{bmatrix} B_1 = \begin{bmatrix} 111 & 172 \\ 251 & 7 \end{bmatrix}$$

The pre-embedding pixel adjustment ensures the non-occurrence of overflow and underflow. The adjustment resets the upper and lower bounds of the pixel components in the specified range as given in equation (4.5). In this example, the offered payload value is three (B = 3) and therefore, the modified upper bound (UB) and the lower bound (LB) values are obtained as follows:

$$UB = \left(2^8 - 2^{\lfloor 3 \rfloor + 2}\right) = (2^8 - 2^5) = 224$$
$$LB = 2^{\lfloor 3 \rfloor + 2} = 2^5 = 32$$

Therefore, if any pixel component goes beyond this range then that pixel component is adjusted immediately. The 2 x 2 sub-matrices of pixel components followed by adjustment are as follows:

$$R_1 = \begin{bmatrix} 224 & 72 \\ 32 & 155 \end{bmatrix} G_1 = \begin{bmatrix} 62 & 215 \\ 56 & 32 \end{bmatrix} B_1 = \begin{bmatrix} 111 & 172 \\ 224 & 32 \end{bmatrix}$$

Binomial Transform (BT) is applied on 2 x 2 sub-matrices of pixel components to obtain the components in transform domain. The 2 x 2 sub-matrices of transformed components such as T(R$_1$), T(G$_1$) and T(B$_1$) are computed as here under:

$$T(R_1) = \begin{bmatrix} 224 & 152 \\ 112 & -51 \end{bmatrix} T(G_1) = \begin{bmatrix} 62 & -153 \\ -312 & -447 \end{bmatrix} T(B_1) = \begin{bmatrix} 111 & -61 \\ -9 & 235 \end{bmatrix}$$

Let the secret bit-stream *“010111010110010011001111011011000000”* is to be fabricated into the transformed components of red/green/blue sub-matrices. To achieve the payload value of 3 bits per Byte, three/two/three/four bits are fabricated (λ_1 = 3, λ_2 = 2, λ_3 = 3, λ_4 = 4) on first/second/third/fourth component starting from LSB-0 toward higher order bit position. The 2 x 2 sub-matrices of embedded components in transform domain are obtained as follows:

$$T'(R_1) = \begin{bmatrix} 226 & 155 \\ 117 & -54 \end{bmatrix} T'(G_1) = \begin{bmatrix} 58 & -154 \\ -313 & -447 \end{bmatrix} T'(B_1) = \begin{bmatrix} 110 & -62 \\ -9 & 224 \end{bmatrix}$$

Application of inverse Binomial Transform (IBT) on 2 x 2 sub-matrices of embedded components yields the 2 x 2 sub-matrices of pixel components as follows:

$$R'_1 = \begin{bmatrix} 226 & 71 \\ 33 & 166 \end{bmatrix} \; G'_1 = \begin{bmatrix} 58 & 212 \\ 53 & 28 \end{bmatrix} \; B'_1 = \begin{bmatrix} 110 & 172 \\ 225 & 45 \end{bmatrix}$$

It has been observed that the modified pixel components are non-fractional, non-negative and less than or equal to 255. Thus, no additional re-adjustment is necessary in this example.

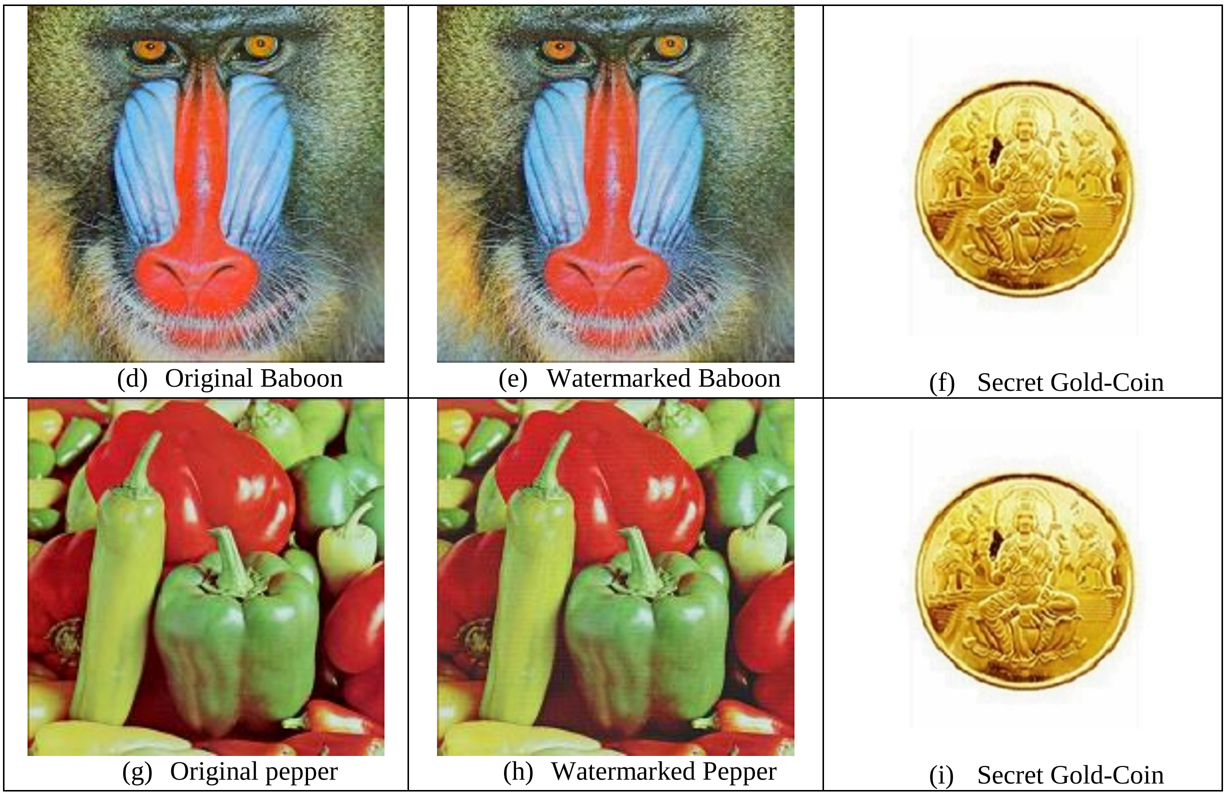
4.2.1.5 Results and Discussions

Results and discussions of the proposed 2 x 2 block based watermark fabrication using Binomial Transform (BT) (WBT_2x2) is elaborated in this section. The quality of the watermarked images is measured by means of peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD) and standard deviation error (SDE) with respect to increasing payload (i.e., from 0.5 to 3 bpB). Benchmark (BMP) images [130, 131] as given in fig. 1.1 are taken to compute results where, all cover images are 512 × 512 in dimension however, the watermark is of variable sizes. Comparative analysis is made among the proposed 2 x 2 block based watermarking using Binomial Transform (BT) i.e., WBT_2x2, 2 x 2 block based watermarking using Legendre Transform (WLT_2x2) of section 3.2.1, 2 x 2 block based watermarking using Separable Discrete Hartley Transform (WDHT_2x2) of section 2.2.1 and Varsaki et al.'s Discrete Pascal Transform based data hiding technique (DPTHDI) [88] as well as Discrete Gould Transform based data hiding scheme (DGTDHS) [129] schemes respectively. Simulation results ensured that the WBT_2x2 is improvised in terms of quality over the above mentioned schemes. Fig. 4.1 illustrates three benchmark images namely “Lena”, “Baboon” and “Pepper” which allows the fabrication of the secret watermark (i.e., “Gold-Coin” image) into their original copies to obtain the corresponding watermarked images.

(a) Original Lena

(b) Watermarked Lena

(c) Secret Gold-Coin



(d) Original Baboon

(e) Watermarked Baboon

(f) Secret Gold-Coin

(g) Original pepper

(h) Watermarked Pepper

(i) Secret Gold-Coin

Fig 4.1. Cover, watermarked and the authenticating watermark image in the proposed WBT_2x2 technique

Performance of WBT_2x2 scheme is evaluated for twenty benchmark images (fig. 1.1) in terms of the peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) with respect to 0.5, 1.0, 1.5, 2.0, 2.5 and 3 bits per Byte (bpB) of payloads. In general, the quality of watermarked images sharply decreases as payload increases and that of quality increases while payload values decreases. However, the quality (which is widely measured in terms of PSNR) is not uniform in nature as each watermarked image has different pixel distribution which may stipulate larger or smaller change in the pixel values. In proposed WBT_2x2, the maximum PSNR obtained is 50.22 dB at 0.5 bpB of payload for the "Foster City" image while the minimum PSNR obtained is 24.13 dB at 3 bpB of payload for "Desert" image which falls below the acceptable level of quality (i.e., < 30 dB). Since, the WBT_2x2 scheme offered variable payload of 0.5 to 3 bpB, the quality of watermarked image is sacrificed at 2.5 and 3 bpB respectively. The lowest MSE is 0.61 for "Foster City" at payload of 0.5 bpB and the highest MSE is 251.04 for "Desert" at 3 bpB of payload. It is to be noted that the IF, SSIM and UIQ are maximum at 0.5 bpB and minimum at 3 bpB whereas the default range is [0, 1]. The minimum values of IF, SSIM and UIQ are 0.962091 (Desert), 0.777996 (Splash) and 0.402609 (Splash) and that of maximum values are 0.999981 (Airplane), 0.999143 (San

Diego) and 0.998400 (San Diego) respectively. Experimental results are summarized based on the average values for different metrics of twenty color images at variable payload.

Table 4.1. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WBT_2x2 technique

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	50.170722	0.625184	0.999960	0.997008	0.987649
	1.0	45.118747	2.000808	0.999873	0.993480	0.960032
	1.5	40.418927	5.904561	0.999626	0.976591	0.911432
	2.0	37.216494	12.343289	0.999225	0.962475	0.842190
	2.5	33.430747	29.512476	0.998126	0.904155	0.747471
	3.0	29.655912	70.386450	0.995452	0.849978	0.613376
Baboon	0.5	50.123143	0.632071	0.999966	0.998939	0.997615
	1.0	45.030284	2.041982	0.999892	0.997617	0.990923
	1.5	40.299848	6.068697	0.999678	0.990959	0.979034
	2.0	37.111273	12.645996	0.999332	0.985008	0.954806
	2.5	33.334384	30.174624	0.998401	0.962395	0.927795
	3.0	29.870780	66.988821	0.996459	0.937869	0.870926
Pepper	0.5	46.481090	1.462080	0.999862	0.987135	0.973145
	1.0	40.904093	5.280451	0.999499	0.975842	0.944967
	1.5	38.290845	9.638196	0.999126	0.958368	0.908773
	2.0	33.791417	27.160555	0.997481	0.931489	0.834235
	2.5	31.540436	45.607541	0.995895	0.875314	0.752145
	3.0	26.584003	142.784151	0.986523	0.797012	0.606739
Airplane	0.5	50.072236	0.639523	0.999981	0.996426	0.963779
	1.0	45.064240	2.026078	0.999942	0.992383	0.903582
	1.5	40.449521	5.863112	0.999832	0.973023	0.819725
	2.0	37.381659	11.882678	0.999659	0.957475	0.730899
	2.5	33.527566	28.861820	0.999174	0.891525	0.621979
	3.0	30.594748	56.702929	0.998377	0.843049	0.529024
Sailboat	0.5	49.734037	0.691315	0.999965	0.997561	0.989584
	1.0	44.535756	2.288256	0.999884	0.994459	0.967239
	1.5	40.120757	6.324185	0.999681	0.980610	0.932043
	2.0	36.700315	13.901089	0.999301	0.967742	0.879520
	2.5	33.116329	31.728347	0.998401	0.920695	0.812116
	3.0	28.918834	83.405930	0.995823	0.866499	0.694976
Earth	0.5	50.211011	0.619411	0.999963	0.997722	0.994337

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	1.0	45.141506	1.990351	0.999881	0.995050	0.980562
	1.5	40.419312	5.904037	0.999649	0.982365	0.953228
	2.0	37.293281	12.126968	0.999277	0.971207	0.905575
	2.5	33.392249	29.775248	0.998220	0.924139	0.832098
	3.0	29.551507	72.099068	0.995372	0.882359	0.716144
San Diego	0.5	50.199054	0.621119	0.999976	0.999143	0.998400
	1.0	45.170963	1.976896	0.999926	0.998155	0.994701
	1.5	40.379111	5.958942	0.999778	0.993137	0.987085
	2.0	37.374979	11.900971	0.999556	0.988981	0.974717
	2.5	33.430964	29.511000	0.998901	0.969416	0.951479
	3.0	30.566345	57.074988	0.997871	0.951371	0.912400
Splash	0.5	47.457569	1.167678	0.999882	0.985263	0.941465
	1.0	42.090334	4.018342	0.999587	0.976021	0.872433
	1.5	39.041302	8.108679	0.999219	0.950364	0.773914
	2.0	35.032559	20.409127	0.997947	0.926462	0.668459
	2.5	32.370873	37.669773	0.996400	0.846163	0.543910
	3.0	28.105581	100.582458	0.989727	0.777996	0.402609
Oakland	0.5	48.711909	0.874760	0.999946	0.997839	0.996125
	1.0	43.408005	2.966742	0.999816	0.995162	0.988299
	1.5	39.602149	7.126321	0.999587	0.984584	0.973778
	2.0	35.939597	16.562300	0.998992	0.974057	0.946178
	2.5	32.669059	35.170164	0.997969	0.935108	0.900605
	3.0	28.862488	84.495087	0.994778	0.894706	0.823447
Foster City	0.5	50.223314	0.617659	0.999978	0.995890	0.984041
	1.0	45.051875	2.031855	0.999927	0.990980	0.948486
	1.5	40.458959	5.850385	0.999791	0.968346	0.886898
	2.0	37.292126	12.130193	0.999567	0.948631	0.798883
	2.5	33.495382	29.076497	0.998962	0.869838	0.687486
	3.0	30.457331	58.525775	0.997924	0.803105	0.565828
Anhinga	0.5	48.518520	0.914594	0.999929	0.998785	0.880234
	1.0	43.679812	2.786757	0.999786	0.997237	0.872376
	1.5	39.670912	7.014377	0.999461	0.991024	0.850791
	2.0	36.405731	14.876720	0.998859	0.985776	0.831270
	2.5	32.864935	33.619159	0.997421	0.959081	0.775645
	3.0	29.302451	76.354596	0.994134	0.927131	0.708812
Athens	0.5	48.616577	0.894175	0.999928	0.998795	0.968605

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	1.0	45.935568	1.657766	0.999867	0.997506	0.960615
	1.5	40.004009	6.496499	0.999481	0.991019	0.933399
	2.0	38.279313	9.663824	0.999228	0.986392	0.909774
	2.5	32.893488	33.398848	0.997334	0.953360	0.823433
	3.0	30.027346	64.616832	0.994830	0.921000	0.739184
Bardowl	0.5	47.654216	1.115985	0.999886	0.998745	0.996966
	1.0	43.326743	3.022776	0.999689	0.996563	0.989377
	1.5	39.079715	8.037274	0.999189	0.991711	0.981374
	2.0	35.269403	19.325912	0.998010	0.977802	0.941884
	2.5	31.735635	43.603047	0.995595	0.958484	0.914238
	3.0	26.694159	139.208057	0.985655	0.900331	0.809472
Barnfall	0.5	49.699070	0.696904	0.999893	0.998321	0.997215
	1.0	45.119722	2.000359	0.999679	0.996230	0.991761
	1.5	41.204910	4.927076	0.999215	0.988917	0.981908
	2.0	37.693296	11.059886	0.998175	0.979736	0.959416
	2.5	33.741662	27.473506	0.995537	0.942646	0.910624
	3.0	29.768063	68.592093	0.988026	0.889526	0.812251
Butrfly	0.5	49.015265	0.815743	0.999942	0.998879	0.994139
	1.0	45.030771	2.041753	0.999854	0.997032	0.984814
	1.5	40.121117	6.323661	0.999553	0.991160	0.971612
	2.0	37.450141	11.696777	0.999172	0.983670	0.939897
	2.5	33.135576	31.588040	0.997775	0.958807	0.893069
	3.0	29.761611	68.694061	0.995048	0.924211	0.797556
Bobcat	0.5	49.066413	0.806193	0.999886	0.998370	0.748072
	1.0	45.687098	1.755377	0.999752	0.996261	0.742328
	1.5	40.168160	6.255532	0.999122	0.986605	0.726236
	2.0	37.697992	11.047934	0.998442	0.977612	0.704836
	2.5	33.100930	31.841045	0.995545	0.942118	0.665275
	3.0	24.558832	2.276130	0.967817	0.807958	0.537441
Bodie	0.5	48.151928	0.995147	0.999826	0.998603	0.976576
	1.0	43.250448	3.076348	0.999457	0.996467	0.967516
	1.5	39.740820	6.902370	0.998812	0.990070	0.955687
	2.0	34.889922	21.090562	0.996232	0.970337	0.917372
	2.5	32.397619	37.438498	0.993437	0.946504	0.884105
	3.0	26.259486	153.862103	0.971950	0.835771	0.735750
Bluheron	0.5	49.847640	0.673467	0.999917	0.998128	0.993673

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	1.0	46.053261	1.613445	0.999802	0.996377	0.985625
	1.5	41.454936	4.651432	0.999431	0.987311	0.964817
	2.0	38.885506	8.404844	0.998969	0.981136	0.938061
	2.5	34.711920	21.972947	0.997314	0.950131	0.871103
	3.0	31.066716	50.863786	0.993759	0.914513	0.776174
Colomtn	0.5	49.506922	0.728430	0.999943	0.998396	0.983158
	1.0	44.270656	2.432286	0.999809	0.996516	0.977354
	1.5	40.316882	6.044942	0.999527	0.988014	0.962691
	2.0	36.812822	13.545595	0.998941	0.981069	0.945311
	2.5	33.437579	29.466084	0.997698	0.949278	0.897753
	3.0	29.406951	74.539279	0.994191	0.906242	0.821658
Desert	0.5	44.838152	2.134347	0.999661	0.996140	0.992907
	1.0	39.477933	7.333089	0.998845	0.990441	0.982180
	1.5	37.403745	11.822405	0.998171	0.985402	0.976314
	2.0	32.209715	39.093884	0.993973	0.962814	0.937033
	2.5	30.124353	63.189514	0.990424	0.942658	0.914165
	3.0	24.133351	251.041319	0.962091	0.853428	0.792643
Average Case	0.5	48.914940	0.886289	0.999915	0.996804	0.967884
	1.0	44.167390	2.717086	0.999738	0.993489	0.950259
	1.5	39.932300	6.761134	0.999396	0.982479	0.921537
	2.0	36.536380	15.543460	0.998517	0.969994	0.878016
	2.5	32.922580	34.033910	0.996926	0.930091	0.816325
	3.0	28.707320	87.154700	0.989790	0.874203	0.713321

Fig. 4.2 depicts an interesting study (WBT_2x2) for the variation of PSNR values in the watermarked images on embedding the varying amount of secret information for the payload range [0.5 – 3 bpB] in Binomial Transform (BT) domain. Color images such as “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” are taken to compute PSNR values for payload range [0.5 – 3 bpB]. The PSNR values of above mentioned five images are also computed for DPTHDI [88] and DGTDHS [129] respectively. However, the basic disadvantage of DPTHDI [88] and DGTDHS [129], having the window size of 2 x 2, is their fixed and relatively low payload values of 0.25 and 1 bpB respectively. Unlikely, the WBT_2x2 is designed and implemented based on the principle of variable payload feature that offers a spread from 0.5 to 3 bpB. An acceptable visual imperceptibility (i.e., PSNR ≥ 30 dB) is perceived for the payload range [0.5 – 3 bpB] however, the imperceptibility falls down

(i.e., the acceptable level of PSNR values becomes less than 30 dB) as soon as the payload reaches to the maximum limit of 3 bpB [148]. In contrast to DPTHDI [88], the WBT_2x2 ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2 and 2.5 bpB for “Baboon”, 0.5, 1 and 1.5 bpB of payloads for “Pepper”, 0.5 and 1 bpB of payloads for “Airplane” and 0.5, 1, 1.5 and 2 bpB of payloads for “Sailboat” respectively. Compared to DGTDHS [129], the WBT_2x2 ensured low PSNR value at 1 bpB of payload. In spite of the fact, the WBT_2x2 is to be important due to the incorporation of variable payload [0.5 – 3 bpB] feature.

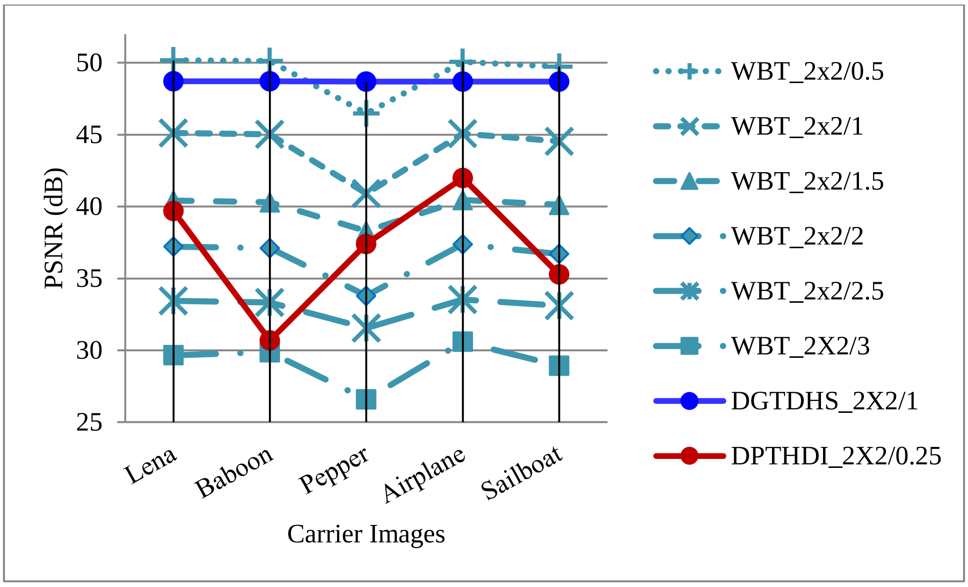


Fig. 4.2. Performance analysis of PSNR (dB) for variable payload based WBT_2x2 and fixed payload based Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

The average peak signal to noise ratio (PSNR) versus payload graph is plotted in fig. 4.3 to analyze the performance of WBT_2x2, WLT_2x2 and WDHT_2x2 schemes in comparison with Varsaki et al.’s DPTHDI [88] and DGTDHS [129] respectively. In contrast to WLT_2x2, slight improvement in average PSNR is observed for WBT_2x2 for the payload range 0.5 to 2 bpB however, exceeding this payload limit leads to massive improvement in PSNR values. In contrast to WDHT_2x2, the WBT_2x2 obtains better PSNR for payload range [0.5 – 3 bpB]. The average PSNR for DPTHDI [88] is 37.40 dB by considering the average of PSNR values for “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images at 0.25 bpB of payload. The average PSNR of WBT_2x2 ensures better transparency over DPTHDI [88] at 0.5, 1 and 1.5 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB by considering the average of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat”, “F16” images at 1 bpB of payload. The advantage of WBT_2x2 is its ability of offering variable payload however, the average PSNR at 1 bpB slightly lacks over the DGTDHS [129]. The WBT_2x2 revealed that the quality of the watermarked images are belonging to the perceptible level for the payload range 0.5 to 2.5 bpB however, the severe quality distortion (i.e., average PSNR value of less than 30 dB) is observed at 3 bpB. To address this issue, a quality improvement process such as an optimization scheme or quality enhancement scheme may be included to enrich the image quality above the acceptable level (PSNR ≥ 30 dB).

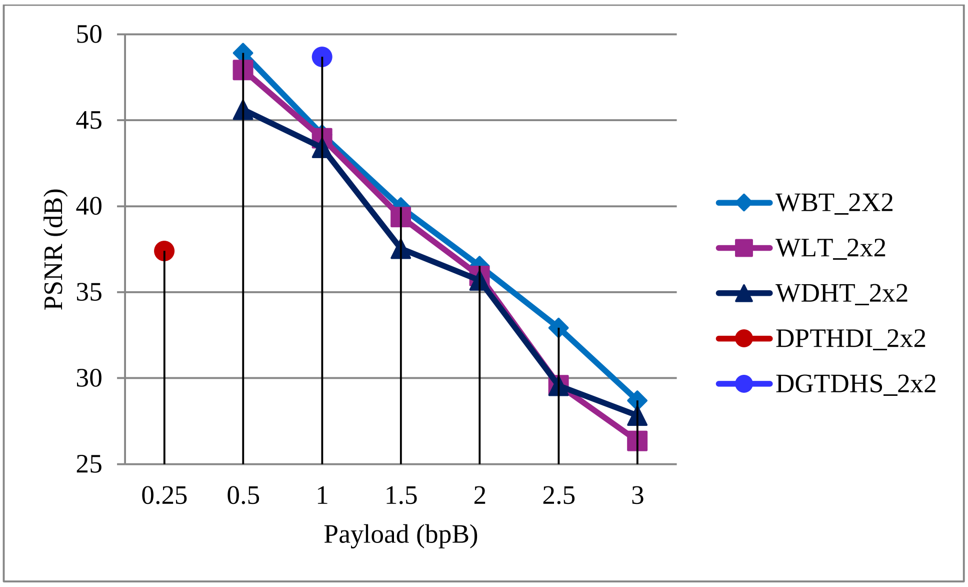


Fig. 4.3. Graphical representation of variation of average PSNR (dB) with respect to payload for WBT_2x2, WLT_2x2, WDHT_2x2 and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

Fig. 4.4 depicts the standard deviation (SD) analysis of five color images [130, 131] viz. "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively. The standard deviation (SD) of a color image is computed based on the average of standard deviation (SD) values for red, green and blue channels. It is seen from fig. 4.4 that the standard deviation (SD) is almost same up to the payload value of 2 bits per Byte (bpB). However, as the payload increases from 2 bpB, the standard deviation (SD) decreases in usual case such as for "Lena", "Baboon", "Pepper" and "Sailboat" respectively whereas, the standard deviation (SD) for "Airplane" increases. As the payload of 0 bpB designates the original image prior to embedding, the standard deviation (SD) at that payload is nothing but the standard deviation (SD) of the original image. Therefore, the standard deviation (SD) values are computed for the watermarked images at payload range [0.5 – 3 bpB] and are compared against the reference image to measure the gradual enhancement of dispersion.

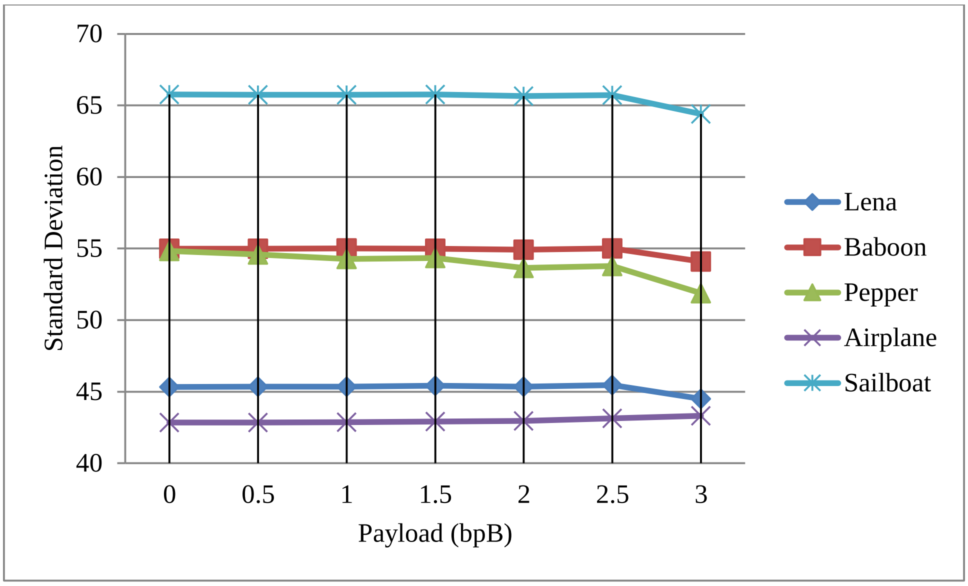


Fig. 4.4. Graphical representation of standard deviation (SD) for WBT_2x2 with respect to 0, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

The standard deviation (SD) analysis is made to measure the rate of signal fluctuation from the mean; unlikely, the standard deviation error (SDE) is the absolute difference of standard deviation (SD) between the original and the watermarked images with respect to increasing payload. Five color images (fig. 1.1) have been considered for carrying out the analysis. Fig. 4.5 illustrates that the standard deviation error (SDE) increases as the payload (bpB) increases however, from 0.5 to 2 bpB of payload, the error is almost negligible. The standard deviation error (SDE) is comparatively high for “Pepper” and as a consequence, it creates noteworthy dispersions at higher payload values (i.e. > 2 bpB).

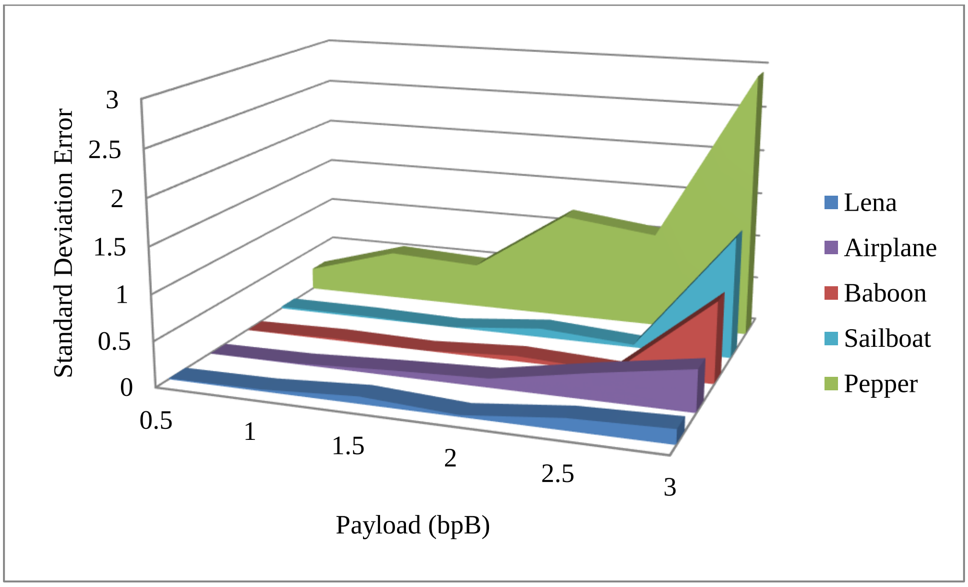


Fig. 4.5. Graphical representation of standard deviation error (SDE) for WBT_2x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

4.2.2 1 x 2 Block based Watermark Fabrication

Binomial Transform (BT) [135, 136] accepts a sequence of integer-valued numbers as input and produces a sequence of integer-valued numbers as output. The transform is applied on pixel components $\{p_\gamma\}$ to generate transformed components $\{t_\gamma\}$ as given in equation (4.11).

$$t_\gamma = \sum_{k=0}^{\gamma} (-1)^k \binom{\gamma}{k} p_k \tag{4.11}$$

Similarly, the inverse Binomial Transform (IBT) is used to convert transformed components back into spatial domain as given in equation (4.12).

$$p_\gamma = \sum_{k=0}^{\gamma} (-1)^k \binom{\gamma}{k} t_k \tag{4.12}$$

Compared to 2 x 2 block based formulation of section 4.2.1, the 1 x 2 block based formulation of Binomial Transform (BT) has been made to convert each pair of pixel components into transform domain as expressed in equation (4.13).

$$t_k = \begin{cases} p_k : k = 0 \\ -p_k + \ p_{k-1} : k = 1 \end{cases} \tag{4.13}$$

where, transformed component is denoted by t_k, pixel component is denoted by p_k and for all k, $0 \leq k \leq 1$.

Similarly, on applying the inverse Binomial Transform (IBT) of equation (4.12), the re-computed pixel components of a 1 x 2 masks can be obtained using equation (4.14).

$$p_k = \begin{cases} t_k : k = 0 \\ -t_k + \ t_{k-1} : k = 1 \end{cases} \tag{4.14}$$

where, pixel component is denoted by p_k, transformed component is denoted by t_k and for all k, $0 \leq k \leq 1$.

Sections 4.2.2.1 and 4.2.2.2 described the algorithm for insertion and the algorithm for extraction respectively. The example is given in section 4.2.2.3 and the results and discussions have been elaborated in section 4.2.2.4.

4.2.2.1 Insertion

Split the carrier image into 1 x 2 non-overlapping blocks consisting of pair of pixel components. The pixel components are adjusted prior to embedding to avoid the occurrence of overflow and underflow. Binomial Transform (BT) converts each 1 x 2 sub-image block or pair of pixel components of red/green/blue channel into transform domain. Variable payload with considerable degradation in fidelity is achieved by fabricating the message digest, size and content of the watermark into the transformed components starting from the least significant bit position (i.e., LSB-0) toward higher order bit position. Each transformed component is capable of fabricating an average of up to three watermark bits. Inverse Binomial Transform (IBT) is applied on 1 x 2 block (pair) of embedded components to obtain the pixel components. Successive steps yield the watermarked image in spatial domain.

Algorithm 4.3:

Input: Carrier/cover image (I) and authenticating watermark image (W).

Output: Watermarked image (I′).

Method: Binomial Transform (BT) is applied on each 1 x 2 sub-block of the carrier image to convert the carrier image into transform domain. Watermark (along with a message digest) is fabricated into transformed components in varying proportions to achieve variable payload, less distortion and improved security. The detailed steps of embedding are as follows:

Step 1: Obtain a message digest (MD) from the secret watermark.

Step 2: The authenticating watermark size (in bits) is obtained by embedding the watermark into the three sub-matrices of the U x V color image as given in equation (4.15).

$$W_{size} = B \times \left(3 \times (U \times V)\right) - (MD + L) \qquad (4.15)$$

where, B, MD and L represents the embedding payload in terms of bits per Byte, the message digest obtained from the watermark and the header information corresponding to the size of the watermark respectively. The MD and L are consisting of 128 and 32 bits while the usual values of B are 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively.

Step 3: The cover image (I) is partitioned into 1 x 2 non-overlapping blocks in row major order. Each 1 x 2 block is consisting of a pair of pixel components p_i and p_{i+1} of red/green/blue channel where, the values of i lies in the range $0 \leq i \leq 1$.

Step 4: A pre-embedding adjustment is applied on each pixel component p, to retain the value positive and less than or equal *to 255 i.e., to keep away from overflow and underflow. The pixel component is adjusted by setting up the upper and lower bounds based on the payload value of B bits per byte as given in equation (4.16).*

$$p = \begin{cases} \left(2^8 - 2^{\lceil B\rceil+1}\right) : p \geq \left(2^8 - 2^{\lceil B\rceil+1}\right) \\ 2^{\lceil B\rceil+1} : p \leq 2^{\lceil B\rceil+1} \end{cases} \qquad (4.16)$$

Step 5: Apply Binomial Transform (BT) on 2 x 2 sub-matrices pixel components corresponding to red/green/blue channel to obtain the transformed components t_i and t_{i+1} respectively.

Step 6: λ_1 /λ_2 bits from the secret bit-stream S (i.e., the combinations of the message digest (MD), size (L) and the content (W) of the watermark) are subsequently fabricated on first/second transformed component (t_c) starting from the least significant bit position (i.e., LSB-0) toward higher order bit position. The generalized form of λ bits of secret information fabrication on each transformed component (t_c) for the payload value of B bits per Byte (bpB) can be expressed as given in equation (4.17).

$$t'_c = t_c + S(\lambda)$$

$$\lambda = \begin{cases} \lfloor B\rfloor : \lambda = \lambda_1 \\ \lceil B\rceil : \lambda = \lambda_2 \end{cases} \qquad (4.17)$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all λ, $0 \leq \lambda \leq 3$. The '+' operator indicates the fabrication of λ bits of secret information (S) into the transformed component (t_c) whereas t'_c denote the embedded component.

Step 7: On application of Inverse Binomial Transform (IBT) over 1 x 2 sub-matrices of embedded components, the pixel components with identical block sizes are obtained in spatial domain.

Step 8: Repeat steps 3 to 7 till the message digest MD, size and the content of the watermark is embedded to obtain the watermarked image (I').

Step 9: Stop.

4.2.2.1 Extraction

The recipient decomposed the watermarked image into 1 x 2 non-overlapping blocks in row major order. By applying Binomial Transform (BT) on each 1 x 2 sub-image block or pair of pixel components, the transformed components are obtained with the identical block size. The reverse procedure is successively applied to extract the fabricated bit stream from which the message digest (MD), size and the content of the watermark are reconstructed. A new message digest (MD′) is re-computed from the extracted watermark which in turn is compared against the extracted message digest (MD). If both MD and MD′ are identical then the authentication procedure is considered to be successful. However, a single bit modification in the watermarked image is treated as the lack of authenticity. The extraction procedure is described in algorithm 4.4.

Algorithm 4.4:

Input: Watermarked image (I′).

Output: The authenticating watermark (W) and the 128 bits message digest (MD).

Method: The fabricated watermark (along with a message digest) is extracted from the watermarked image (I′) in Binomial Transform (BT) domain. Successive extracted bits reconstruct the watermark from which another message digest (MD′) is obtained. Both message digests are compared against each other to verify the authenticity. The detailed steps of extraction are as follows:

Step 1: The watermarked image (I′) is partitioned into 1 x 2 non-overlapping blocks in row major order. Each 1 x 2 block is consisting of pair of pixel components p_i and p_{i+1} of red/green/blue channel where, for all i, $0 \leq i \leq 1$.

Step 2: Apply Binomial Transform (BT) on 1 x 2 sub-matrices of pixel components corresponding to RGB color channels to obtain the transformed components t_i and t_{i+1} respectively.

Step 3: λ_1 /λ_2 bits of the fabricated secret bit-stream are successively extracted from first/second embedded component starting from the least significant bit position (i.e., LSB-0) toward higher order bit position. In general, the reverse procedure is applied to extract λ bits for the payload value of B bits per Byte (bpB) from each pair of embedded components as per the extraction rule given in equation (4.18).

$$\lambda = \begin{cases} \lfloor B \rfloor : \lambda = \lambda_1 \\ \lceil B \rceil : \ \lambda = \lambda_2 \end{cases} \tag{4.18}$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5 and for all λ, $0 \leq \lambda \leq 3$.

Step 4: For each 8 (eight) bits extraction, construct one alphabet/one primary (R/G/B) color component.

Step 5: Inverse Binomial Transform (IBT) is applied on 1 x 2 sub-matrix or pair of embedded components to restore back the pixel components.

Step 6: Repeat steps 1 to 5 until and unless the fabricated message digest, size and the content of the watermark is recovered.

Step 7: Obtain 128 bits message digest MD' from the extracted watermark.

Step 8: Compare MD' with the extracted MD. If both are matches then the image is authorized, else unauthorized.

Step 9: Stop.

4.2.2.2 Example

Decomposition of the carrier image into non-overlapping blocks yields three 1 x 2 sub-matrices of pixel components corresponding to red (R_1), green (G_1) and blue (B_1) channels. The 1 x 2 sub-matrices are represented as follows:

$$R_1 = [249\ 78] \quad G_1 = [119\ 217] \quad B_1 = [15\ 130]$$

In this example, overflow and underflow situations are avoided based on a pre-embedding pixel adjustment strategy in which upper bound (UB) and lower bound (LB) of the pixel components are redefined for payload value of 3 bpB based on equation (4.16).

$$UB = \left(2^8 - 2^{\lceil 3 \rceil + 1}\right) = (2^8 - 2^4) = (256 - 16) = 240$$
$$LB = 2^{\lceil 3 \rceil + 1} = 2^4 = 16$$

Therefore, any pixel component beyond this range is instantaneously adjusted. The 1 x 2 adjusted sub-matrices of pixel components are obtained as given below:

$$R_1 = [240\ 78] \quad G_1 = [119\ 217] \quad B_1 = [16\ 130]$$

Binomial Transform (BT) converts each 1 x 2 sub-matrices of pixel components into the transformed components i.e., $T(R_1)$, $T(G_1)$ and $T(B_1)$ as follows:

$$T(R_1) = [240\ 162]\quad T(G_1) = [119\ -98]\quad T(B_1) = [16\ -114]$$

Insertion of the secret bit stream *"111101110010110000"* into the transformed components is done by replacing the designated bit(s) of the components. In this example, three bits are fabricated ($\lambda = \lambda_1 = \lambda_2 = 3$) on each component starting from LSB-0 toward the higher order bit positions. Hence, the 1 x 2 sub-matrices of embedded components are obtained as follows:

$$T'(R_1) = [248\ 165]\quad T'(G_1) = [115\ -98]\quad T'(B_1) = [19\ -112]$$

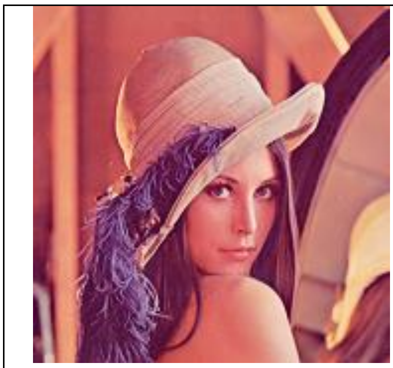
Applying inverse Binomial Transform (IBT) on each pair of embedded components yields the generation of pair of pixel components in spatial domain as given below:

$$R'_1 = [248\ 83]\quad G'_1 = [115\ 213]\quad B'_1 = [19\ 131]$$

The pixel components obtained after inverse transform phase are non-negative, non-fractional and less than or equal to 255.

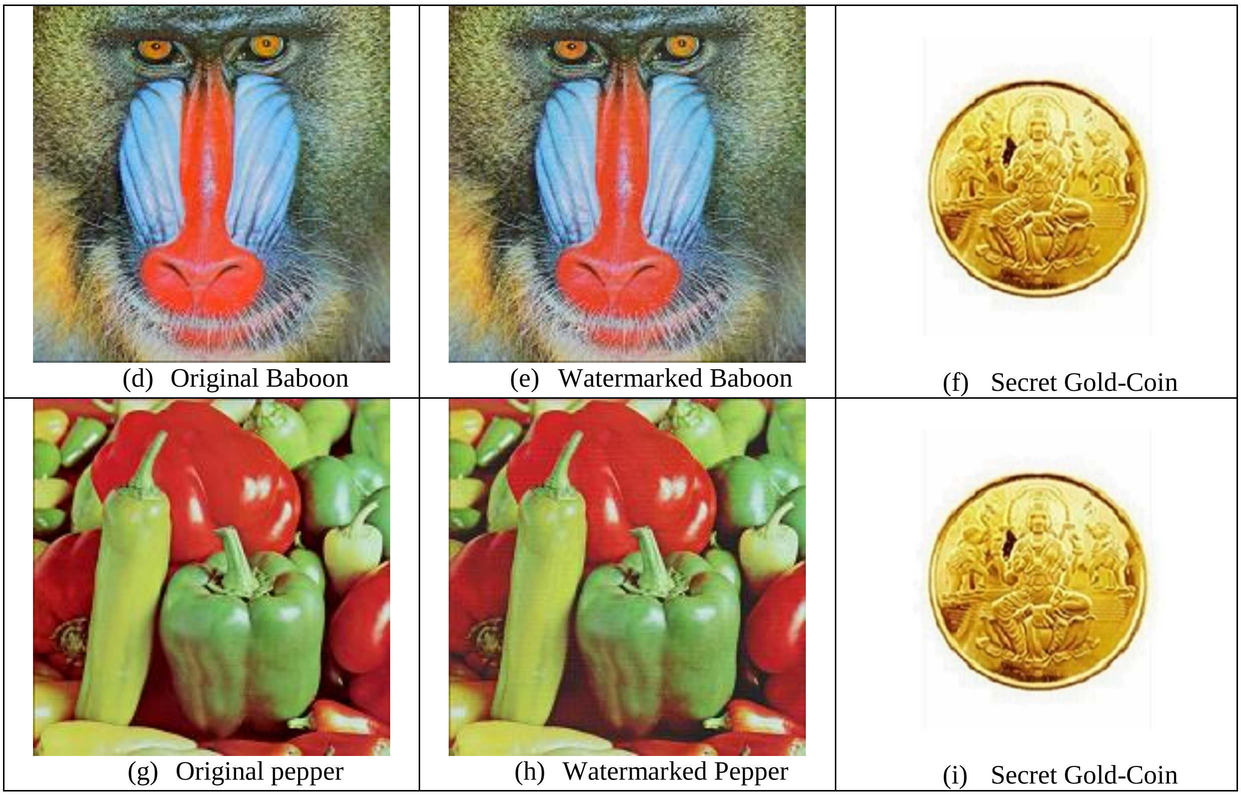
4.2.2.3 Results and Discussions

The 1 x 2 block based watermarking (WBT_1x2) is analyzed in terms of peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD) and standard deviation error (SDE) respectively. In this experiment, twenty cover images and the varying sizes of the watermark are used as given in fig. 1.1. Comparison is made among WBT_1x2, WLT_1x2, WBT_2x2, WDHT_1x2 and Varsaki et al.'s Discrete Pascal Transform based data hiding technique (DPTHDI) [88] as well as Discrete Gould Transform based data hiding scheme (DGTDHS) [129] schemes respectively. Different states of modifications of carrier/cover images viz. "Lena", "Baboon" and "Pepper" (along with the secret image i.e., "Gold-Coin") are shown in fig 4.6.


(a) Original Lena

(b) Watermarked Lena

(c) Secret Gold-Coin



(d) Original Baboon

(e) Watermarked Baboon

(f) Secret Gold-Coin

(g) Original pepper

(h) Watermarked Pepper

(i) Secret Gold-Coin

Fig 4.6. Cover, watermarked and authenticating watermark image using the proposed WBT_1x2 technique

The performance of WBT_1x2 technique has been investigated for twenty color images of dimension 512 x 512 that fabricates the varying sizes of the secret watermark (Gold-Coin) as depicted in fig. 1.1. Table 4.2 reveals that the lowest peak signal to noise ratio (PSNR) is 30.30 dB at 3 bpB of payload for the "Desert" image whereas, the highest PSNR of 54.15 dB is obtained at 0.5 bpB of payload for the "Foster City" image. Since, the lowest PSNR of WBT_1x2 retained above 30 dB, the watermarked images obtained in this experiment are considered to be well perceptible [148]. The lowest mean squared error (MSE) is 0.24 for "Foster City" at 0.5 bpB and that of the highest value is 60.56 for "Desert" at 3 bpB of payload. The minimum values of image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) are 0.990742 (Desert), 0.916937 (Splash) and 0.614041 (Splash) respectively while the maximum values of the respective parameters are 0.999992 (Airplane), 0.999871 (San Diego) and 0.999255 (San Diego). The values of IF, SSIM and UIQ are maximum at 0.5 bpB and minimum at 3 bpB. The usual values of IF, SSIM and UIQ are lies in between 0 and 1 however, the values closest to one designate higher similarity between the watermarked and the original images. The average values of above mentioned quality metrics are also computed against the payload variation (0.5 – 3 bpB) to summarize the experimental results.

Table 4.2. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WBT_1x2 technique

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.145421	0.250344	0.999984	0.999567	0.994298
	1.0	49.422982	0.742646	0.999953	0.997964	0.984533
	1.5	45.209355	1.959498	0.999876	0.996060	0.961489
	2.0	41.931087	4.168421	0.999736	0.990094	0.930268
	2.5	38.001243	10.302817	0.999366	0.980491	0.863166
	3.0	35.045561	20.348117	0.998710	0.958427	0.804445
Baboon	0.5	54.142422	0.250517	0.999986	0.999850	0.998909
	1.0	49.322226	0.760077	0.999959	0.999290	0.996941
	1.5	45.152829	1.985168	0.999895	0.998551	0.991192
	2.0	41.842384	4.254435	0.999774	0.996395	0.984180
	2.5	38.038402	10.215040	0.999462	0.992436	0.961345
	3.0	35.080401	20.185530	0.998930	0.984531	0.946324
Pepper	0.5	53.876715	0.266323	0.999977	0.999142	0.988760
	1.0	46.422549	1.481922	0.999862	0.988717	0.971649
	1.5	41.628498	4.469207	0.999578	0.980432	0.948885
	2.0	39.363372	7.529101	0.999308	0.972701	0.925003
	2.5	34.818586	21.439848	0.997985	0.954377	0.862340
	3.0	32.388097	37.520669	0.996534	0.928655	0.806525
Airplane	0.5	54.138654	0.250734	0.999992	0.999486	0.982639
	1.0	49.423420	0.742571	0.999978	0.997591	0.955813
	1.5	45.219193	1.955064	0.999944	0.995332	0.906119
	2.0	41.943551	4.156475	0.999880	0.988289	0.848951
	2.5	37.921425	10.493921	0.999699	0.975112	0.748263
	3.0	35.027072	20.434927	0.999415	0.949709	0.679919
Sailboat	0.5	54.110007	0.252394	0.999987	0.999685	0.995188
	1.0	49.095326	0.800843	0.999959	0.998367	0.987344
	1.5	44.801712	2.152331	0.999891	0.996862	0.968884
	2.0	41.622117	4.475779	0.999774	0.991955	0.946722
	2.5	37.681921	11.088893	0.999442	0.984148	0.898676
	3.0	34.641352	22.332899	0.998876	0.965642	0.855513
Earth	0.5	54.155713	0.249752	0.999985	0.999672	0.997370
	1.0	49.456888	0.736871	0.999956	0.998448	0.992869
	1.5	45.314007	1.912844	0.999886	0.997017	0.981303
	2.0	41.960197	4.140574	0.999753	0.992524	0.964393

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	38.339737	9.530300	0.999431	0.985410	0.922044
	3.0	35.142888	19.897178	0.998808	0.968165	0.877525
San Diego	0.5	54.137201	0.250818	0.999990	0.999871	0.999255
	1.0	49.422528	0.742724	0.999972	0.999407	0.997982
	1.5	45.308735	1.915167	0.999928	0.998861	0.994815
	2.0	41.931605	4.167924	0.999844	0.997157	0.990232
	2.5	38.415310	9.365895	0.999651	0.994593	0.978897
	3.0	35.205602	19.611920	0.999269	0.987901	0.966078
Splash	0.5	54.000320	0.258850	0.999977	0.998787	0.965256
	1.0	47.299330	1.211008	0.999879	0.987145	0.932784
	1.5	42.665134	3.520196	0.999646	0.980663	0.876070
	2.0	40.237964	6.155790	0.999398	0.970784	0.809579
	2.5	35.743814	17.326024	0.998311	0.951071	0.698233
	3.0	33.336353	30.160949	0.997081	0.916937	0.614041
Oakland	0.5	54.064019	0.255081	0.999986	0.999721	0.998422
	1.0	48.289669	0.964080	0.999942	0.998432	0.995446
	1.5	43.812021	2.703200	0.999835	0.996825	0.988897
	2.0	40.938579	5.238686	0.999691	0.993053	0.980118
	2.5	36.919088	13.218175	0.999194	0.986579	0.956272
	3.0	34.083711	25.392717	0.998491	0.971824	0.929705
Foster City	0.5	54.156509	0.249706	0.999991	0.999422	0.992571
	1.0	49.501043	0.729417	0.999974	0.997207	0.980486
	1.5	45.338529	1.902074	0.999932	0.994671	0.950269
	2.0	42.0130316	4.090507	0.999853	0.986608	0.911753
	2.5	38.337413	9.535401	0.999661	0.974182	0.825845
	3.0	35.271881	19.314889	0.999312	0.944386	0.752901
Anhinga	0.5	52.081071	0.402692	0.999969	0.999698	0.882230
	1.0	48.553767	0.907201	0.999930	0.999048	0.878763
	1.5	42.530476	3.631053	0.999721	0.997392	0.852973
	2.0	40.529487	5.756142	0.999558	0.995556	0.846254
	2.5	35.666891	17.635640	0.998649	0.987366	0.795268
	3.0	33.353204	30.044147	0.997695	0.977765	0.782614
Athens	0.5	51.881060	0.421671	0.999966	0.999551	0.970407
	1.0	49.219801	0.778216	0.999937	0.998957	0.966760
	1.5	43.862507	2.671957	0.999786	0.996632	0.933554
	2.0	40.681061	5.558712	0.999554	0.994818	0.924583

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	36.570150	14.324033	0.998856	0.983514	0.844652
	3.0	32.903013	33.325681	0.997337	0.974610	0.828587
Bardowl	0.5	52.530426	0.363109	0.999963	0.999768	0.998743
	1.0	47.790922	1.081404	0.999890	0.998969	0.996695
	1.5	42.733976	3.464836	0.999647	0.997110	0.987255
	2.0	39.820619	6.776702	0.999316	0.994812	0.982496
	2.5	35.152941	19.851174	0.997974	0.982620	0.936995
	3.0	32.050540	40.553307	0.995892	0.971961	0.922987
Barnfall	0.5	53.104854	0.318122	0.999950	0.999696	0.998398
	1.0	49.326944	0.759251	0.999880	0.998758	0.996527
	1.5	44.747922	2.179154	0.999648	0.997324	0.988320
	2.0	42.045685	4.059866	0.999353	0.994132	0.981635
	2.5	37.318068	12.057950	0.998010	0.984384	0.944240
	3.0	34.743097	21.815771	0.996401	0.970458	0.924125
Butrfly	0.5	52.514547	0.364439	0.999974	0.999703	0.995016
	1.0	49.101758	0.799658	0.999943	0.999161	0.992913
	1.5	44.132874	2.510688	0.999822	0.997665	0.978576
	2.0	40.768547	5.447855	0.999613	0.995496	0.970585
	2.5	36.828016	13.498289	0.999045	0.988001	0.921336
	3.0	33.241831	30.824584	0.997821	0.979680	0.903887
Bobcat	0.5	52.161319	0.395319	0.999944	0.99954	0.749370
	1.0	49.393287	0.747741	0.999894	0.998862	0.747211
	1.5	44.116661	2.520079	0.999646	0.997069	0.731240
	2.0	40.917033	5.264741	0.999258	0.993617	0.725343
	2.5	36.765949	13.692583	0.998073	0.985136	0.685597
	3.0	33.260272	30.693973	0.995677	0.967503	0.672923
Bodie	0.5	53.142233	0.315396	0.999946	0.999759	0.979354
	1.0	47.720609	1.099054	0.999807	0.998815	0.975527
	1.5	42.866619	3.360612	0.999407	0.996798	0.962574
	2.0	40.675674	5.565611	0.999036	0.994437	0.956664
	2.5	35.175258	19.749427	0.996503	0.977525	0.911896
	3.0	32.937328	33.063397	0.994132	0.960210	0.891834
Bluheron	0.5	52.656006	0.352760	0.999956	0.999649	0.995693
	1.0	49.623876	0.709075	0.999913	0.998578	0.992728
	1.5	44.002853	2.586991	0.999683	0.996602	0.976992
	2.0	41.548144	4.552666	0.999444	0.993030	0.966331

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	36.527689	14.464767	0.998228	0.982955	0.906858
	3.0	35.259492	19.370066	0.997627	0.974007	0.884805
Colomtn	0.5	53.079946	0.319952	0.999974	0.999730	0.984703
	1.0	49.026124	0.813706	0.999936	0.998820	0.982228
	1.5	43.594896	2.841781	0.999777	0.997454	0.968763
	2.0	41.431376	4.676734	0.999634	0.994503	0.962075
	2.5	36.591907	14.252452	0.998886	0.986794	0.921596
	3.0	34.430010	23.446570	0.998168	0.975677	0.907048
Desert	0.5	52.888865	0.334344	0.999952	0.999715	0.998172
	1.0	44.970361	2.070351	0.999675	0.996624	0.992990
	1.5	39.716066	6.941825	0.998922	0.991728	0.980486
	2.0	37.969517	10.378356	0.998393	0.988490	0.976216
	2.5	32.460623	36.899293	0.994367	0.967714	0.930770
	3.0	30.308732	60.562965	0.990742	0.955618	0.918753
Average Case	0.5	53.348370	0.306116	0.999972	0.999601	0.973238
	1.0	48.619170	0.933891	0.999912	0.997458	0.965909
	1.5	43.837740	2.859186	0.999724	0.995052	0.946433
	2.0	41.008550	5.320754	0.999509	0.990923	0.929169
	2.5	36.663720	14.947100	0.998540	0.980220	0.875714
	3.0	33.885520	27.945010	0.997346	0.964183	0.843527

Comparisons among WBT_1x2, DPTHDI [88] and DGTDHS [129] are made in terms of peak signal to noise ratio (dB) as given in fig. 4.7. The comparison is made for five color images viz. “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” respectively. The basic drawback of both 2 x 2 block based approaches (i.e., DPTHDI [88] and DGTDHS [129]) is their fixed (as well as low) payload of 0.25 and 1 bpB respectively. In contrast, the WBT_1x2 is focused on variable payload that offers a spread from 0.5 to 3 bpB by keeping perceptible quality in the watermarked images (PSNR ≥ 30 dB) [148]. Compared to DPTHDI [88], proposed WBT_1x2 ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB for “Baboon”, 0.5, 1, 1.5 and 2 bpB of payloads for “Pepper”, 0.5, 1, 1.5 and 2 bpB of payloads for “Airplane” and 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for “Sailboat” respectively. In comparison with the DGTDHS [129], proposed WBT_1x2 ensured higher PSNR (dB) at 0.5 and 1 bpB of payloads for “Lena”, 0.5 and 1 bpB of payloads for “Baboon”, 0.5 bpB of payload for “Pepper”, 0.5 and 1 bpB of payloads for “Airplane” and 0.5 and 1 bpB of payloads for “Sailboat” respectively.

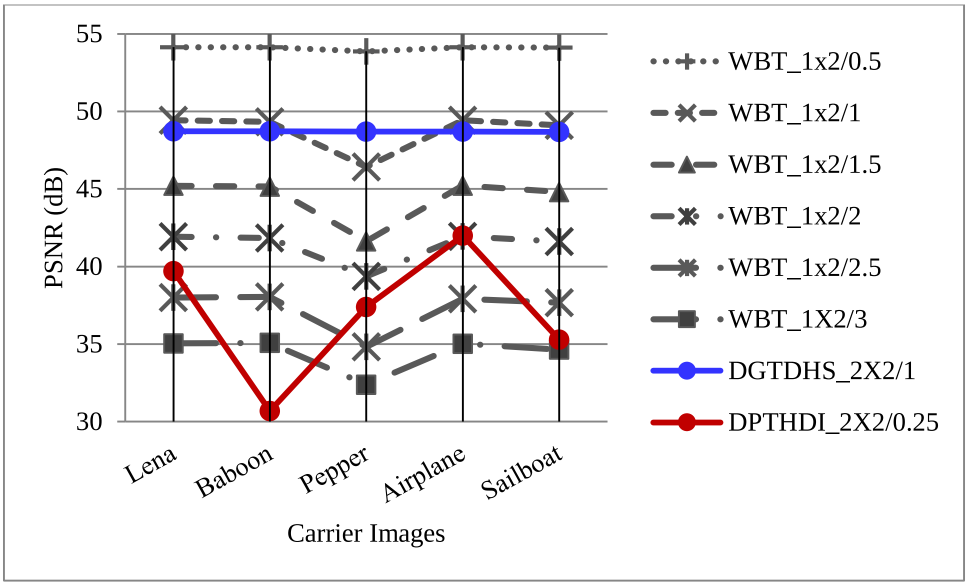


Fig. 4.7. Performance analysis of PSNR (dB) for variable payload based WBT_1x2 and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes computed using five color images

Table 4.2 revealed the maximum value of average PSNR of 53.34 dB at 0.5 bpB and that of the minimum value obtained is 33.38 dB at 3 bpB respectively. To compute the average PSNR values, twenty benchmark images (fig. 1.1) have been considered. Fig. 4.8 depicts the variation of average PSNR values with respect to proposed variable payload based schemes (WBT_1x2, WBT_2x2, WLT_1x2 and WDHT_1x2) and Varsaki et al.'s fixed payload based schemes (DPTHDI [88] and DGTDHS [129]) respectively. It is observed that the WBT_1x2 obtains a high PSNR over the WBT_2x2, WLT_1x2 and WDHT_1x2 with respect to the payload range [0.5 – 3 bpB]. Therefore, the WBT_1x2 offered high fidelity over the WBT_2x2, WLT_1x2 and WDHT_1x2 respectively. The average PSNR for DPTHDI [88] is 37.40 dB which is obtained by taking the average of PSNR values for "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" images at 0.25 bpB of payload. Compared to DPTHDI [88], the WBT_1x2 ensured equal or higher PSNR at 0.5, 1, 1.5 and 2 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB which is obtained by taking the average of PSNR values for "Lighthouse", "Elaine", "Lenna", "Boat", "F16" images at 1 bpB of payload. Compared to DGTDHS [129], the average PSNR values of WBT_1x2 is almost identical at 1 bpB of payload however, the WBT_1x2 provides payload variation in the range of 0.5 to 3 bpB. It is to be noted that the obtained average PSNR values of WBT_1x2 is greater than or equal to 30 dB with respect to varying payload (0.5 – 3 bpB) and hence, the obtained watermarked images are satisfying the criteria of perceptible fidelity as well as excellent transparency [148].

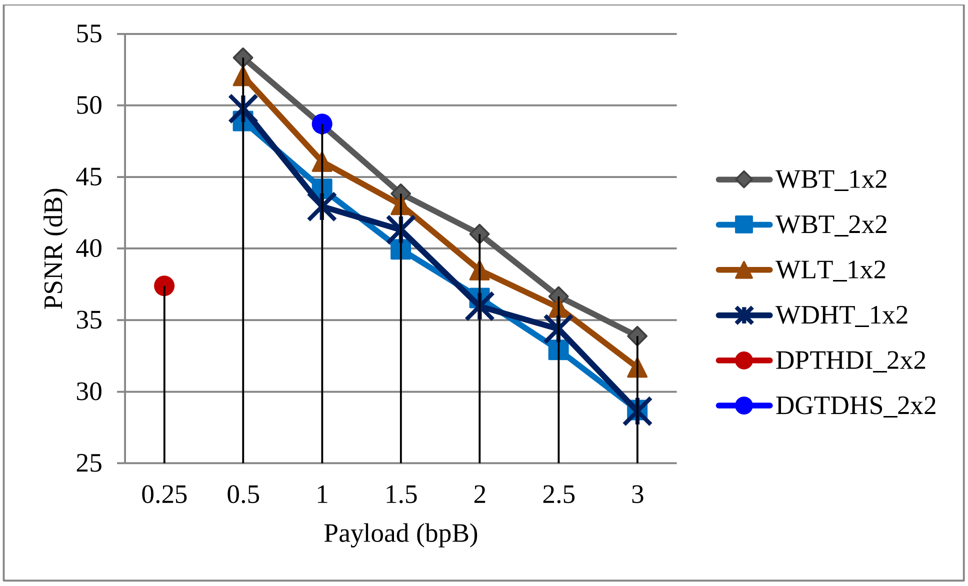


Fig. 4.8. Graphical representation of variation of average PSNR (dB) with respect to payload for WBT_1x2 and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) techniques

In fig. 4.9, the standard deviation (SD) is computed by taking the average of standard deviation (SD) values of red, green and blue channels for “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” respectively. Standard deviation (SD) at 0 bpB of payload represents the standard deviation (SD) of the original image. Statistical comparison demonstrates that the changes made in the watermarked image with reference to the original image (except the “Pepper”) is very less for the payload range [0.5 – 3 bpB] and hence, it is really very much tough for an observer to differentiate between the source and the watermarked images.

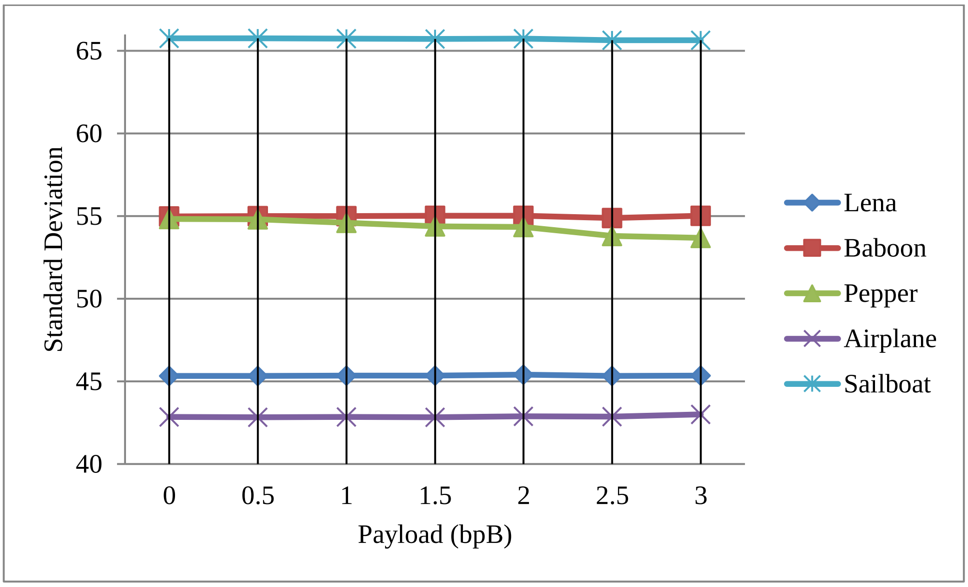


Fig. 4.9. Graphical representation of standard deviation (SD) for WBT_1x2 with respect to 0, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

In fig. 4.10, the standard deviation error (SDE) analysis is carried out for five color images such as “Lena”, “Baboon”, “Airplane” and “Sailboat”, respectively. The standard deviation error (SDE) for these images has been computed separately at varying payload (i.e., 0 bpB - 3 bpB) by taking the average of standard deviation error (SDE) values corresponding to red, green and blue channels. It is observed that the standard deviation error (SDE) increases as the payload increases and the watermarked images are gradually deviating from the original image with respect to the increasing payload. It can also be seen that the error is almost zero for “Lena”, “Baboon”, “Airplane” and “Sailboat” up to 3 bpB of payload; however, the error is comparatively high for “Pepper” image. Hence, the selection of “Pepper” image as the cover is not an effective choice as the standard deviation error (SDE) is elevated at higher payload values.

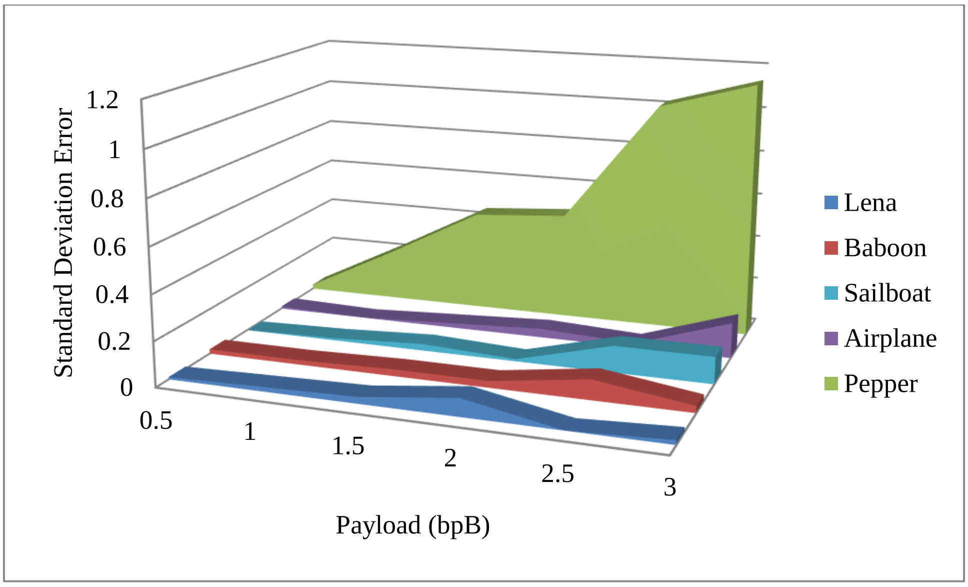


Fig. 4.10. Graphical representation of standard deviation error (SDE) for WBT_1x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

In fig. 4.11, the performance of 1 x 2 block based watermarking in Binomial Transform domain (WBT_1x2) is compared against the 2 x 2 block based watermarking in Binomial Transform domain (WBT_2x2) in terms of standard deviation error (SDE) analysis. It is clearly seen from the figure that the error is almost 0 for “Lena”, “Baboon”, “Airplane” and “Sailboat” up to the payload value of 2.5 bpB. When the payload increases from 2.5 to 3 bpB, the error also increases; however, the rate of increase has been minimized tremendously for the 1 x 2 block based watermarking than the 2 x 2 block based watermarking in Binomial Transform (BT) domain. It ensures the effectiveness of choosing 1 x 2 as the window size instead of 2 x 2 for fabricating watermark into the carrier image.

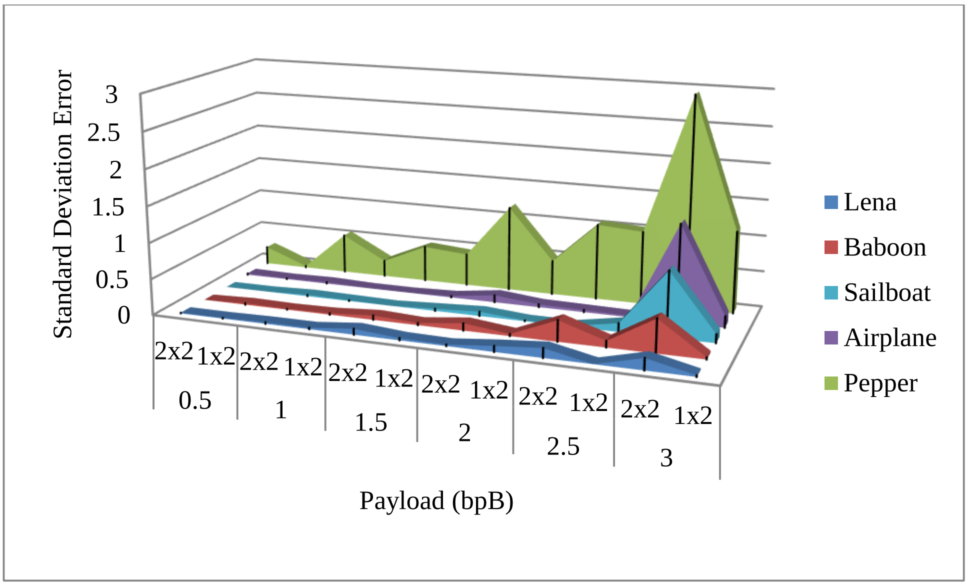


Fig. 4.11. Pictorial representation of variation of standard deviation error (SDE) between WBT_2x2 and WBT_1x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

4.3 Salient Features

In contrast to 2 x 2 and 1 x 2 block based watermarking in Discrete Hartley Transform (DHT) domain or Legendre Transform (LT) domain, the Binomial Transform (BT) based watermarking techniques i.e., both WBT_2x2 and WBT_1x2 schemes starts fabrication in the transformed component from LSB-0 toward higher order bit position. As a consequence, lower order bit positions are properly utilized for embedding and the higher order bits are least altered to achieve better PSNR values at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads.

In addition, the drawbacks of existing techniques [88, 129] including the high distortion, less payload, lacking in usage of color image as the cover and the high computational overhead etc. can be resolved by introducing the WBT_2x2 and WBT_1x2 schemes. Both schemes offered low robustness against attacks; however, a variable payload of up to 3 bpB with perceptible visual imperceptibility is achieved.

Some of the important features of the proposed techniques are mentioned as follows.

Binomial Transform (BT) is highly acceptable due to its high computational complexity. To address the problem of overflow/underflow, a pixel adjustment is done prior to embedding. However, as Binomial Transform (BT) is highly sensitive against small modification of a transformed component, a re-adjustment operation has also been incorporated. It is seen that most of the existing schemes offers fixed (as well as low) payload; however, proposed WBT_2x2 and WBT_1x2 schemes are based on variable payload (i.e., 0.5 to 3 bpB) by keeping perceptible quality of the watermarked image. In contrast to Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes, both WBT_2x2 and WBT_1x2 schemes ensure variable payload with reduced quality degradation. The degradation of quality can further be minimized by introducing the quality enhancement of chapter 7 / genetic algorithm based optimization of chapter 8.

Chapter 5

Watermarking based on Stirling Transform (WST)

5.1 Introduction

The objective of Stirling Transform (ST) based fragile watermarking schemes is to verify the authenticity of color images by fabricating varying number of secret bits within perceptible quality of degradation. The cover image is decomposed into non-overlapping blocks which in turn are transformed through Stirling Transform (ST). Watermark and the message digest (obtained from the watermark) are embedded into the transformed components. The pixel components are re-computed from each block of embedded components by applying inverse Stirling Transform (IST). The decoder extracts the hidden watermark and the message digest. Message digest is also re-computed from the extracted watermark which in turn is compared with the extracted message digest to verify the authenticity. Simulation results demonstrate the effectiveness of the proposed schemes over the existing watermarking schemes.

5.2 The Technique

Consider a 24 bit color image as the cover and decompose the image into P x Q non-overlapping blocks ($1 \leq P \leq 2$ and $Q = 2$) in a row major order. Stirling Transform (ST) converts each P x Q non-overlapping block of pixel components into transformed components. To avoid the occurrence of overflow and underflow, pixel components are adjusted prior to embedding. The fabrication of secret bits corresponding to the message digest (MD), size and the content of the watermark has been made at the least significant part of transformed components in varying proportions. The pixel components are restored by applying the inverse Stirling Transform (IST) on identical blocks of embedded components. To handle unavoidable overflow/underflow, an additional re-adjustment operation may be applied in transform domain which does not affect the fabricated bits. The above process is repeated to obtain the final watermarked image. The decoder receives the watermarked image, splits the image into P x Q non-overlapping blocks (where, $1 \leq P \leq 2$ and $Q = 2$) and then converts each block into transform domain based on Stirling Transform (ST). The secret bits corresponding to the message digest (MD), size and the content of the watermark are extracted from the transformed components. Inverse Stirling Transform (IST) is applied on each P x Q block of the embedded components to restore the pixel components. The process is repeated until and unless the watermark is reconstructed.

A minor alteration in the watermarked image may violate the principle of authentication. As a result, attacks such as filtering, blurring, lossy compression etc. can destroy the fabricated data of the watermarked image and the image is considered as tampered. To address this problem, a message digest (MD') is obtained from the extracted watermark at the recipient end. The integrity of the watermarked image has been verified by comparing the extracted message digest (MD) against the re-computed message digest (MD'). If both message digests are identical then the watermarked image is considered to be authenticated.

Section 5.2.1 of this chapter deals with 2 x 2 block based watermark fabrication that of section 5.2.2 describe 1 x 2 block based watermark fabrication.

5.2.1 2 x 2 Block based Watermark Fabrication

Stirling Transform (ST) [137-141] generates integer output sequence for a given integer-valued sequence as input. Stirling Transform (ST) is applied to convert the pixel components $\{p_k\}$ into the transformed components $\{t_k\}$ using the formula as given in equation (5.1).

$$t_\gamma = \sum_{k=1}^{\gamma} \left\{ \begin{matrix} \gamma \\ k \end{matrix} \right\} p_k \tag{5.1}$$

where, k = 1, 2, 3,... and $\left\{ \begin{matrix} \gamma \\ k \end{matrix} \right\}$, also denoted as S(γ, k) is the Stirling number of the second kind which specify the partitions of a set of size γ into *k* parts as given in equation (5.2).

$$S(\gamma, k) = \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^\gamma \tag{5.2}$$

By applying the inverse Stirling Transform (IST), the pixel components $\{p_k\}$ are re-generated from the transformed components $\{t_k\}$ as given in equation (5.3).

$$p_\gamma = \sum_{k=1}^{\gamma} s(\gamma, k) t_k \tag{5.3}$$

where, *s*(*γ*, *k*) is a Stirling number of the first kind which is computed from the following recurrence relation.

$$s(\gamma, k) = s(\gamma - 1, k - 1) + (\gamma - 1) s(\gamma - 1, k)$$

where, the initial conditions s(0, 0) = 1 and $s(\gamma, 0)$ = $s(0, \gamma)$= 0 for all $\gamma, \gamma \geq 1$.

The formulation has been made for 2 x 2 sub-image blocks of the cover image. Stirling Transform (ST) converts the 2 x 2 sub-blocks of pixel components into the 2 x 2 sub-blocks of transformed components as expressed in equation (5.4).

$$t_k = \begin{cases} p_k : k = 1 \\ p_{k-1} + p_k : k = 2 \\ p_{k-2} + 3p_{k-1} + p_k : k = 3 \\ p_{k-3} + 7p_{k-2} + 6p_{k-1} + p_k : k = 4 \end{cases} \tag{5.4}$$

where, transformed components are denoted as t_k, pixel components are denoted as p_k and for all k, $1 \leq k \leq 4$.

Similarly, the inverse Stirling Transform (IST) of equation (5.3) generates the 2 x 2 sub-matrices of pixel components from the 2 x 2 sub-matrices of transformed components as given in equation (5.5).

$$p_k = \begin{cases} t_k : k = 1 \\ -t_{k-1} + t_k : k = 2 \\ 2t_{k-2} - 3t_{k-1} + t_k : k = 3 \\ -6t_{k-3} + 11t_{k-2} - 6t_{k-1} + t_k : k = 4 \end{cases} \tag{5.5}$$

where, pixel and transformed components are denoted as p_k and t_k respectively and k ranges from [1, 4] i.e., $1 \leq k \leq 4$.

The proposed technique has been described in detail at the following sections. The algorithm for insertion, the re-adjustment, the algorithm for extraction and an example are described in detail in section 5.2.1.1, 5.2.1.2, 5.2.1.3 and 5.2.1.4 respectively. Results and discussions have been elaborated in section 5.2.1.5.

5.2.1.1 Insertion

Decompose the carrier image into 2 x 2 non-overlapping blocks in a row major order. To avoid overflow and underflow, adjustment of pixel components have been carried out at the pre-embedding stage. Each 2 x 2 sub-block of pixel components corresponding to the red, green and blue channels are transformed through Stirling Transform (ST). The message digest obtained from the watermark, size and content of the watermark generate a secret bit-stream from which varying number of bits is fabricated into the transformed components. First, third and fourth transformed components of each 2 x 2 sub-block (in row major order) may be modified to fabricate the secret bits; however, the second transformed component is modified to not to embed secret bits rather, to avoid severe distortion of watermarked image's quality. The pixel components are re-computed by applying the inverse Stirling Transform (IST) on identical blocks of embedded components. To handle the overflow and underflow situations, an additional re-adjustment operation is applied in transform domain which does not affect the fabricated bits. The above process is repeated till the fabrication of secret bits is over and the watermarked image is obtained in spatial domain.

Algorithm 5.1:

Input: Carrier/cover image (I) and authenticating watermark image (W).

Output: Watermarked image (I′).

Method: Stirling Transform (ST) is used to embed the watermark (along with a message digest) into the carrier images by converting the image from spatial domain into transform domain. Embedding bits in transform domain offers variable payload, considerable quality distortion and improved security.

Step 1: Obtain a message digest (MD) from the authenticating watermark.

Step 2: The authenticating watermark size (in bits) is obtained by embedding the watermark into the three sub-matrices of the U x V color image as given in equation (5.6).

$$W_{size} = B \times \big(3 \times (U \times V)\big) - (MD + L) \tag{5.6}$$

where, B, MD and L represents the embedding payload in terms of bits per Byte, the message digest obtained from the watermark and the header information corresponding to the size of the watermark respectively. The MD and L are consisting of 128 and 32 bits while the usual values of B are 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively.

Step 3: The cover image (I) is partitioned into 2 x 2 non-overlapping blocks in row major order. Each 2 x 2 block is consisting of four pixel components $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$ of red/green/blue channel where, the values of i and j lies in the range $0 \leq i \leq$ 1and $0 \leq j \leq 1$.

Step 4: A pre-embedding adjustment is applied on each pixel component p, to avoid overflow and underflow based on the payload value of B bits per Byte as given in equation (5.7).

$$p = \begin{cases} \left(2^8 - 2^{2B - \left\lfloor \frac{2B}{5} \right\rfloor}\right) : p \geq \left(2^8 - 2^{2B - \left\lfloor \frac{2B}{5} \right\rfloor}\right) \\ 2^{2B - \left\lfloor \frac{2B}{5} \right\rfloor} : p \leq 2^{2B - \left\lfloor \frac{2B}{5} \right\rfloor} \end{cases} \tag{5.7}$$

Step 5: Apply Stirling Transform (ST) on 2 x 2 sub-matrices of pixel components corresponding to the red, green and blue channels to obtain 2 x 2 sub-matrices of transformed components $t_{i,j}$, $t_{i,j+1}$, $t_{i+1,j}$ and $t_{i+1,j+1}$ respectively.

Step 6: λ_1 /λ_2 /λ_3 bits from the secret bit-stream (corresponding to the message digest (MD), size (L) and the content (W) of the watermark) are subsequently fabricated on first/third/fourth transformed component of red/green/blue channel starting from the least significant bit position (i.e., LSB-0) toward higher order bit position. It has been observed that a small modification on the first transformed component affects other components significantly. Thus, second transformed component is altered by adding the difference of the pre-embedding and post-embedding value of the first component. The generalized form of λ_1 /λ_2 /λ_3 bits of secret information fabrication on first/third/fourth transformed components for the payload value of B bits per Byte (bpB) is given in equation (5.8).

$$(\lambda_1, \lambda_2, \lambda_3) = \begin{cases} \lfloor B \rfloor + 1, \lceil B \rceil, 2B - 1 : 0 < B \leq 1.5 \\ \lceil B \rceil + 1, \lfloor B \rfloor + 1, 2B - 2 : 1.5 < B \leq 3 \end{cases} \tag{5.8}$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all (λ_1, λ_2, λ_3), $0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 4$.

Step 7: Inverse Stirling Transform (IST) is applied over each 2 x 2 sub-matrix of pixel components corresponding to red/green/blue channel to generate the transformed components $t_{i,j}$, $t_{i,j+1}$, $t_{i+1,j}$ and $t_{i+1,j+1}$ respectively.

Step 8: Apply re-adjustment operation on the embedded components in transform domain to avoid overflow and underflow situations, if necessary.

Step 9: Repeat steps 3 to 8 to fabricate the watermark size, content and the message digest MD respectively. On execution of successive block embedding operation, the watermarked image (I') is obtained.

Step 10: Stop.

5.2.1.2 Re-Adjustment

Stirling Transform (ST) is very much sensitive about a small change. Therefore, an additional re-adjustment has been used to avoid overflow and underflow issues. The overflow and underflow may occur during inverse transform phase which may generate the following situations:

- The fourth pixel component may be negative (-ve).
- The fourth pixel component may be greater than the maximum value (i.e., 255).

This re-adjustment operation is slightly different from sections 3.2.1.2 and 4.2.1.2 since the above two problems are handled by adjusting the fourth transformed component of the 2 x 2 mask (instead of the first transformed component).

In this phase, if the converted value is negative (-ve), the operation applied for each 2 x 2 sub-matrix is as follows:

$$t_{1,1} = t_{1,1} + 2^{\lambda_3} \tag{5.9}$$

Here, $t_{1,1}$ is the fourth transformed component of a 2 x 2 sub-matrix and λ_3 bits are fabricated on fourth transformed component. The process is repeated until and unless the fourth pixel component in spatial domain becomes positive.

If the converted value (pixel component) exceeds the maximum value of a byte (i.e., 255), then the operation applied for each 2 x 2 sub-matrix is as follows:

$$t_{1,1} = t_{1,1} - 2^{\lambda_3} \tag{5.10}$$

Here, $t_{1,1}$ is the fourth transformed component of a 2 x 2 sub-matrix and λ_3 bits are fabricated on fourth transformed component. The above process is repeated until and unless the fourth pixel component becomes less than or equal to 255.

5.2.1.3 Extraction

Consider the watermarked image as a set of 2 x 2 non-overlapping blocks of pixel components in row major order. Each 2 x 2 sub-block of red/green/blue channel is converted into transform domain based on Stirling Transform (ST). Based on the successive extraction of the secret bits from first, third and fourth transformed components, the message digest (MD), size and content of the watermark is reconstructed. The extracted watermark re-constitutes a message digest (MD′). Compare both extracted message digest (MD) and the re-computed message digest (MD′). If they are alike then the watermarked image is considered to be authentic. The extraction procedure is described in algorithm 5.2.

Algorithm 5.2:

Input: Watermarked image (I′).

Output: The authenticating watermark (W) and the 128 bits message digest (MD).

Method: Stirling Transform (ST) is used to extract the watermark (along with a message digest) from the watermarked image (I′) by converting the image from spatial domain into transform domain. Successive extracted bits forms the watermark bit-stream and re-generate a message digest to verify the authenticity. The extraction procedure is described as follows:

Step 1: The watermarked image (I′) is partitioned into 2 x 2 non-overlapping blocks in row major order. Each 2 x 2 block is consisting of four pixel components $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$ of red/green/blue channel where, for all i and j, $0 \leq i, j \leq 1$.

Step 2: Apply forward Stirling Transform (ST) on 2 x 2 sub-matrices of pixel components corresponding to the red, green and blue channels where, each sub-matrix is consisting of four transformed components $t_{i,j}$, $t_{i,j+1}$, $t_{i+1,j}$ and $t_{i+1,j+1}$ respectively.

Step 3: λ_1 /λ_2 /λ_3 bits of the secret watermark (W) are extracted from first/third/fourth transformed component (f_c) starting from the least significant bit (i.e., LSB-0) position toward higher order bit position. To achieve the payload value of B bits per Byte (bpB), the generalized form of λ_1 /λ_2 /λ_3 bits watermark extraction from first/third/fourth transformed components is given in equation (5.11).

$$(\lambda_1, \lambda_2, \lambda_3) = \begin{cases} \lfloor B \rfloor + 1, \lceil B \rceil, 2B - 1 : 0 < B \leq 1.5 \\ \lceil B \rceil + 1, \lfloor B \rfloor + 1, 2B - 2 : 1.5 < B \leq 3 \end{cases} \qquad (5.11)$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all (λ_1, λ_2, λ_3), $0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 4$.

Step 4: For each 8 (eight) bits extraction, construct one alphabet/one primary (R/G/B) component.

Step 5: Inverse Stirling Transform (IST) is applied on 2 x 2 sub-matrices of embedded components to re-compute the pixel components in spatial domain.

Step 6: Repeat steps 1 to 5 to complete the extraction of the message digest (MD), size and content of the authenticating watermark.

Step 7: Obtain 128 bits message digest MD' from the extracted watermark.

Step 8: Compare MD' with the extracted MD. If both are matches then the image is authorized, else unauthorized.

Step 9: Stop.

5.2.1.4 Example

The carrier image can be considered as the set of 2 x 2 non-overlapping blocks or pair of pixel components. Each block corresponding to red, green and blue channels are consisting of pair of pixel components. The 2 x 2 sub-matrices of pixel components are as follows:

$$R_1 = \begin{bmatrix} 212 & 198 \\ 1 & 45 \end{bmatrix} \; G_1 = \begin{bmatrix} 97 & 16 \\ 230 & 14 \end{bmatrix} \; B_1 = \begin{bmatrix} 156 & 166 \\ 118 & 65 \end{bmatrix}$$

Overflow and underflow situations are avoided through the pre-embedding adjustments of pixel components as given in equation (5.7). As a consequence, the upper bound (UB) and the lower bound (LB) at 3 bpB of payload (i.e., B = 3) has been computed as follows:

$$UB = \left(2^8 - 2^{2\times3-\left\lfloor\frac{2\times3}{5}\right\rfloor}\right) = (2^8 - 2^5) = (256 - 32) = 224$$

$$LB = 2^{2\times3-\left\lfloor\frac{2\times3}{5}\right\rfloor} = 2^5 = 32$$

The 2 x 2 sub-matrices of pixel components followed by the adjustment prior to embedding are obtained as follows:

$$R_1 = \begin{bmatrix} 212 & 198 \\ 32 & 45 \end{bmatrix} \; G_1 = \begin{bmatrix} 97 & 32 \\ 224 & 32 \end{bmatrix} \; B_1 = \begin{bmatrix} 156 & 166 \\ 118 & 65 \end{bmatrix}$$

The 2 x 2 sub-matrices of pixel components corresponding to red/green/blue channel are converted into the transform domain based on Stirling Transform (ST). The 2 x 2 sub-matrices of transformed components such as T(R$_1$), T(G$_1$) and T(B$_1$) are obtained as given below:

$$T(R_1) = \begin{bmatrix} 212 & 410 \\ 838 & 1835 \end{bmatrix} \; T(G_1) = \begin{bmatrix} 97 & 129 \\ 417 & 1697 \end{bmatrix} \; T(B_1) = \begin{bmatrix} 156 & 322 \\ 772 & 2091 \end{bmatrix}$$

Suppose, the secret bits-stream *“010000010110010011111001011011010101”* is to be embedded into the first/third/fourth transformed components in 4: 4: 4 embedding ratio (i.e., $\lambda_1 = 4$, $\lambda_2 = 4$, $\lambda_3 = 4$) at the payload value of 3 bpB. The embedding starts from LSB-0 toward higher order bits position. It is seen that a small modification on the first transformed component affects other components significantly. Thus, the second transformed component is modified by adding the difference of the pre-embedding and post-embedding value of the first component to avoid severe distortion in the resultant watermarked image. Hence, the 2 x 2 sub-matrices of embedded components are as given below:

$$T'(R_1) = \begin{bmatrix} 210 & 408 \\ 840 & 1830 \end{bmatrix} \; T'(G_1) = \begin{bmatrix} 98 & 130 \\ 431 & 1705 \end{bmatrix} \; T'(B_1) = \begin{bmatrix} 150 & 316 \\ 779 & 2090 \end{bmatrix}$$

Applying inverse Stirling Transform (ST) on each 2 x 2 sub-matrices of embedded components yields the pixel components having the identical block size as given below:

$$R'_1 = \begin{bmatrix} 210 & 198 \\ 36 & 18 \end{bmatrix} \; G'_1 = \begin{bmatrix} 98 & 32 \\ 237 & -39 \end{bmatrix} \; B'_1 = \begin{bmatrix} 150 & 166 \\ 131 & -8 \end{bmatrix}$$

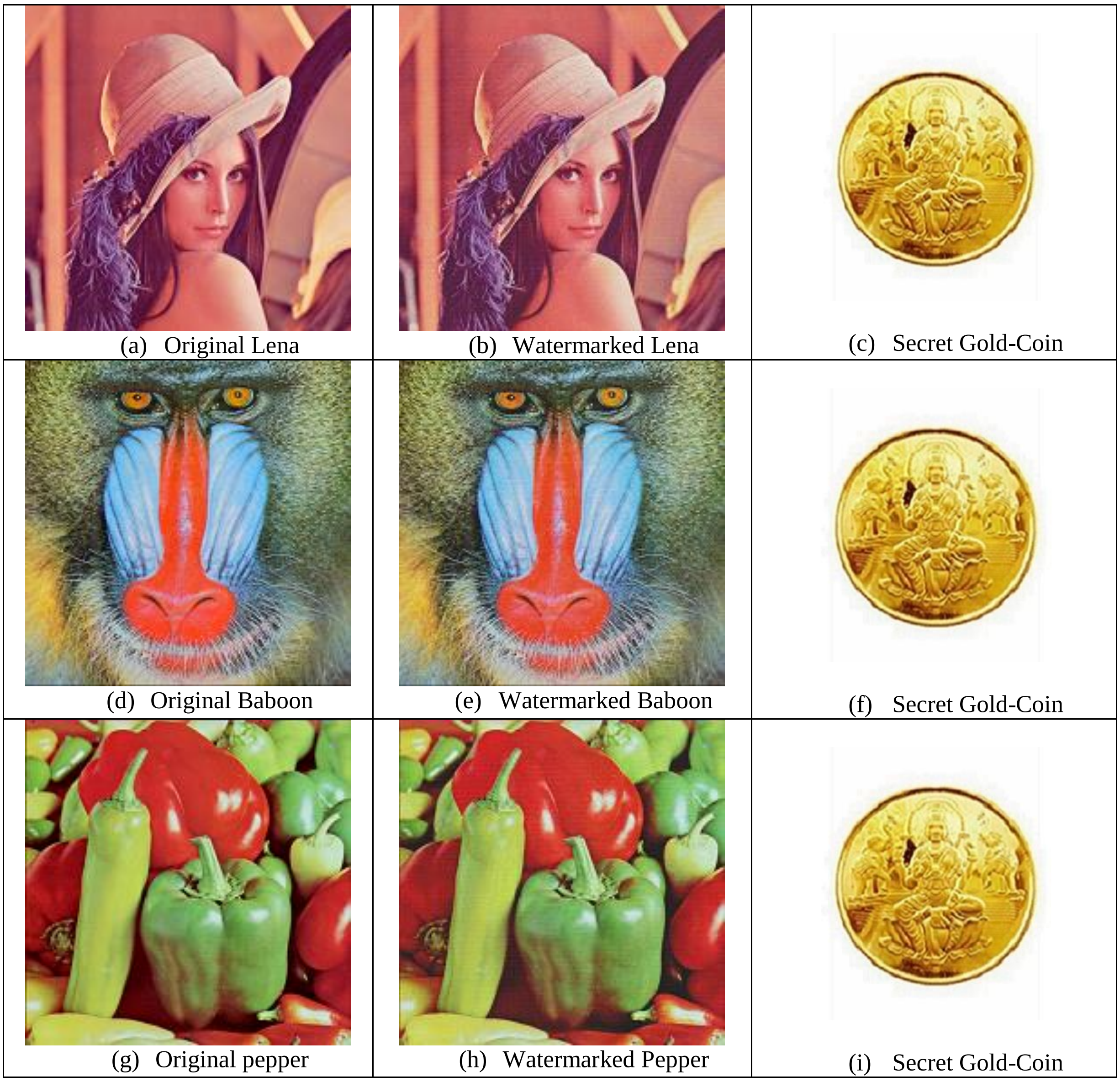
The re-adjustment operation has been applied on the fourth transformed component to keep the pixel component positive.

$$R''_1 = \begin{bmatrix} 210 & 198 \\ 36 & 18 \end{bmatrix} \; G''_1 = \begin{bmatrix} 98 & 32 \\ 237 & 7 \end{bmatrix} \; B''_1 = \begin{bmatrix} 150 & 166 \\ 131 & 8 \end{bmatrix}$$

It has been observed that the modified pixel components for each 2 x 2 sub-matrices are modified slightly to fabricate the secret data.

5.2.1.5 Results and Discussions

To evaluate the performance of WST_2x2 scheme, twenty color images [130, 131] of dimension 512 x 512 and varying sizes of the secret watermark as given in fig. 1.1 has been taken to compute the experimental results. In this experiment, peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD) and standard deviation error (SDE) have been used to measure the quality of the benchmark images prior to embedding with respect to increasing payload. The 2 x 2 block based watermarking using Stirling Transform (WST_2x2), 2 x 2 block based watermarking using Binomial Transform (WBT_2x2) of section 4.2.1, 2 x 2 block based watermarking using Legendre Transform (WLT_2x2) of section 3.2.1, 2 x 2 block based watermarking using Separable Discrete Hartley Transform (WDHT_2x2) of section 2.2.1 and Varsaki et al.'s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] as well as Discrete Gould Transform based data hiding scheme (DGTDHS) [129] are compared among themselves. Experimental results demonstrated the improvement of WST_2x2 scheme over the above schemes. The original images “Lena”, “Baboon” and “Pepper” along with the secret watermark i.e., the “Gold-Coin” are shown in fig. 5.1.



(a) Original Lena
(b) Watermarked Lena
(c) Secret Gold-Coin
(d) Original Baboon
(e) Watermarked Baboon
(f) Secret Gold-Coin
(g) Original pepper
(h) Watermarked Pepper
(i) Secret Gold-Coin

Fig 5.1. Cover, watermarked and the authenticating watermark images in the proposed WST_2x2 technique

The experiment is carried out for twenty benchmark [130, 131] images and the results are computed for standard quality metrics as given in table 5.1. The WST_2x2 offered variable payload for a spread from 0.5 to 3 bpB. The metrics peak signal to noise ratio (PSNR) and mean squared error (MSE) are conflicting to each other and thus, increasing the values for one ensured decreasing the values of the other. Hence, the PSNR range of (54.55 – 24.27 dB) ensured the MSE range of (0.22 – 242.72). The extreme values of PSNR and MSE are computed at 0.5 and 3 bpB of payloads for “Bluheron” and “Desert” respectively. Since, the PSNR of “Desert” image is below the acceptable level of quality (i.e., < 30 dB) at 3 bpB of payload, the quality distortion of the watermarked image is severe [148]. However, the major advantages of WST_2x2 are its ability of producing perceptible watermarked images up to 2. 5 bpB; however, the variable payload property which offered a spread from 0.5 to 3 bpB is also an interesting study. The minimum values of image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) are evaluated as 0.963544 (Desert), 0.849215 (Bobcat) and 0.356289 (Splash) respectively that of the maximum values are obtained as 0.999992 (Airplane), 0.99998 (Athens) and 0.999331 (San Diego) respectively. The IF, SSIM and UIQ values closely analyzed for twenty images and as the values get closer to one, higher similarity is observed between the original and watermarked images respectively. To summarize the experimental results, the average values are computed for the above mentioned metrics at varying payload.

Table 5.1. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WST_2x2 technique

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.164743	0.249233	0.999984	0.999684	0.994767
	1.0	46.964473	1.308076	0.999917	0.998864	0.972223
	1.5	42.593858	3.578446	0.999774	0.995779	0.932038
	2.0	38.165341	9.920789	0.999379	0.988355	0.856693
	2.5	33.964756	26.097850	0.998357	0.972760	0.746727
	3.0	29.049186	80.939730	0.994919	0.924426	0.572819
Baboon	0.5	54.143899	0.250432	0.999986	0.999885	0.999015
	1.0	46.946248	1.313577	0.999930	0.999560	0.994371
	1.5	42.559506	3.606863	0.999809	0.998306	0.985194
	2.0	38.106903	10.055183	0.999469	0.994908	0.960632
	2.5	33.961653	26.116504	0.998619	0.989147	0.929329
	3.0	29.137025	79.319110	0.995816	0.967461	0.854546
Pepper	0.5	53.611422	0.283098	0.999975	0.998541	0.987707
	1.0	44.764449	2.170878	0.999801	0.988995	0.962575
	1.5	39.907341	6.642724	0.999388	0.978647	0.926362
	2.0	34.384418	23.694005	0.997750	0.958208	0.851756
	2.5	31.774676	43.212828	0.996079	0.941231	0.749628
	3.0	26.618959	141.639520	0.9867461	0.876316	0.573539
Airplane	0.5	54.124779	0.251537	0.999992	0.999625	0.983309
	1.0	46.882472	1.333009	0.999961	0.998684	0.926516
	1.5	42.548979	3.615617	0.999896	0.995114	0.851058
	2.0	38.309582	9.596705	0.999725	0.986747	0.742766

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	34.050856	25.585547	0.999267	0.969962	0.620699
	3.0	29.788467	68.270590	0.998045	0.923726	0.487933
Sailboat	0.5	54.096070	0.253205	0.999987	0.999741	0.995591
	1.0	46.707774	1.387723	0.999930	0.999038	0.977414
	1.5	42.283069	3.843911	0.999806	0.996422	0.946679
	2.0	37.631313	11.218868	0.999436	0.989544	0.890667
	2.5	33.628477	28.198926	0.998580	0.976671	0.810917
	3.0	28.572379	90.332159	0.995475	0.930588	0.660508
Earth	0.5	54.167048	0.2491010	0.999985	0.999763	0.997650
	1.0	46.962071	1.308799	0.999922	0.999156	0.986980
	1.5	42.631991	3.547163	0.999789	0.996866	0.965279
	2.0	38.367791	9.468936	0.999433	0.991219	0.915308
	2.5	34.036033	25.673021	0.998461	0.979336	0.826371
	3.0	29.230157	77.636259	0.995097	0.943626	0.683087
San Diego	0.5	54.167580	0.249070	0.999990	0.999911	0.999331
	1.0	46.978876	1.303745	0.999951	0.999681	0.996348
	1.5	42.627037	3.551212	0.999867	0.998809	0.990547
	2.0	38.439007	9.314931	0.999653	0.996718	0.977967
	2.5	34.106937	25.257282	0.999059	0.992108	0.951559
	3.0	29.898786	66.558216	0.997517	0.979407	0.896941
Splash	0.5	53.781897	0.272201	0.999975	0.997960	0.965398
	1.0	46.235137	1.547271	0.999847	0.987386	0.921610
	1.5	40.721861	5.506735	0.999463	0.979328	0.814776
	2.0	35.677226	17.593719	0.998225	0.964102	0.687027
	2.5	32.557576	36.084664	0.996573	0.941453	0.541256
	3.0	27.774357	108.553682	0.989304	0.875316	0.356289
Oakland	0.5	53.978528	0.260152	0.999985	0.999776	0.998472
	1.0	46.182496	1.566140	0.999909	0.998917	0.992473
	1.5	41.585054	4.514138	0.999734	0.996458	0.980723
	2.0	36.689134	13.936923	0.999135	0.990360	0.953465
	2.5	33.186176	31.222145	0.998174	0.980517	0.903029
	3.0	28.543164	90.941875	0.994493	0.948998	0.799023
Foster City	0.5	54.174437	0.248677	0.999991	0.999581	0.993350
	1.0	46.980956	1.303120	0.999953	0.998495	0.964620
	1.5	42.634034	3.545495	0.999873	0.994310	0.910005
	2.0	38.472545	9.243273	0.999671	0.984607	0.816163

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	34.129745	25.124983	0.999105	0.962974	0.690431
	3.0	29.937945	65.960788	0.997654	0.908219	0.531239
Anhinga	0.5	53.822131	0.269691	0.999979	0.999952	0.883977
	1.0	45.466053	1.847034	0.999858	0.999617	0.861105
	1.5	41.773208	4.322743	0.999668	0.998276	0.848309
	2.0	36.091629	15.992538	0.998775	0.993956	0.790044
	2.5	32.793328	34.178066	0.997383	0.984927	0.736867
	3.0	28.804003	85.640664	0.993427	0.955287	0.667116
Athens	0.5	54.102421	0.252835	0.999979	0.999980	0.972700
	1.0	45.677388	1.759306	0.999859	0.999704	0.945779
	1.5	43.243296	3.081418	0.999753	0.998674	0.930861
	2.0	37.563564	11.395252	0.999089	0.995375	0.842753
	2.5	33.599866	28.385314	0.997736	0.984714	0.766283
	3.0	30.468380	58.377073	0.995330	0.955375	0.682383
Bardowl	0.5	53.810332	0.270425	0.999972	0.999913	0.999162
	1.0	45.259598	1.936959	0.999804	0.999242	0.993332
	1.5	41.611237	4.487005	0.999544	0.997401	0.983589
	2.0	35.424332	18.648637	0.998081	0.984958	0.936136
	2.5	33.599866	28.385314	0.997736	0.984714	0.766283
	3.0	30.468380	58.377073	0.995330	0.955375	0.682383
Barnfall	0.5	54.282906	0.242543	0.999962	0.999855	0.998867
	1.0	47.363563	1.193229	0.999811	0.999469	0.994089
	1.5	42.916972	3.321873	0.999464	0.997528	0.983822
	2.0	37.983780	10.344329	0.998270	0.991936	0.952441
	2.5	33.905664	26.455374	0.995667	0.981609	0.896169
	3.0	28.986759	82.111576	0.985831	0.937558	0.753519
Butrfly	0.5	54.125218	0.251511	0.999982	0.999940	0.996048
	1.0	45.944795	1.654248	0.999882	0.999566	0.985322
	1.5	43.046839	3.224010	0.999771	0.998427	0.974032
	2.0	37.713142	11.009462	0.999219	0.995301	0.926050
	2.5	33.586060	28.475691	0.998002	0.987196	0.856406
	3.0	30.145495	62.882645	0.995469	0.963432	0.766293
Bobcat	0.5	54.068113	0.254840	0.999964	0.999913	0.750419
	1.0	45.767689	1.723103	0.999756	0.999234	0.735491
	1.5	43.117686	3.171843	0.999553	0.997526	0.726083
	2.0	37.271079	12.189122	0.998281	0.991056	0.681794

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	32.719871	34.761079	0.995117	0.977446	0.630835
	3.0	25.003439	205.464336	0.970931	0.849215	0.518817
Bodie	0.5	54.100674	0.252937	0.999957	0.999898	0.980116
	1.0	45.929134	1.660224	0.999713	0.999244	0.972946
	1.5	41.381889	4.730330	0.999178	0.997081	0.958985
	2.0	35.277093	19.291723	0.996524	0.980106	0.914135
	2.5	32.342398	37.917572	0.993324	0.970140	0.871558
	3.0	26.063160	160.977166	0.970976	0.874486	0.704601
Bluheron	0.5	54.552339	0.227953	0.999972	0.999860	0.996625
	1.0	47.770808	1.086424	0.999867	0.999390	0.988570
	1.5	43.529062	2.885187	0.999647	0.997563	0.972040
	2.0	38.811429	8.549434	0.998952	0.991956	0.929569
	2.5	34.143825	25.043663	0.996933	0.980743	0.840402
	3.0	29.816627	67.829343	0.991689	0.945686	0.687261
Colomtn	0.5	54.155071	0.249788	0.999980	0.999863	0.985545
	1.0	46.448683	1.473031	0.999884	0.999398	0.976752
	1.5	42.116545	3.994163	0.999687	0.997678	0.964590
	2.0	37.085190	12.722174	0.999007	0.992846	0.929304
	2.5	33.776644	27.253100	0.997869	0.983486	0.880542
	3.0	28.869144	84.365706	0.993421	0.948345	0.776281
Desert	0.5	53.083555	0.319686	0.999953	0.999696	0.998276
	1.0	43.639913	2.812477	0.999562	0.996969	0.991177
	1.5	38.803547	8.564965	0.998669	0.991942	0.979303
	2.0	32.412219	37.312847	0.994228	0.971549	0.931746
	2.5	30.333553	60.217811	0.990919	0.962702	0.897271
	3.0	24.279596	242.728468	0.963544	0.880076	0.762683
Average Case	0.5	54.025660	0.257946	0.999978	0.999667	0.973816
	1.0	46.243630	1.584419	0.999856	0.998030	0.956985
	1.5	42.081650	4.186792	0.999617	0.995107	0.931214
	2.0	36.993840	14.074940	0.998615	0.986690	0.874321
	2.5	33.309900	31.182340	0.997148	0.975192	0.795628
	3.0	28.572770	98.945300	0.990051	0.927146	0.670863

In order to validate the WST_2x2 scheme, an extensive analysis of average PSNR (dB) is made with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. Computed results are also compared against DPTHDI [88] and DGTDHS [129] to ensure the enhancement in quality of the watermarked images (fig. 5.2). Five color images viz. “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” have been considered out of twenty benchmark images (as these are also referenced for DPTHDI [88] DGTDHS [129]) for carrying out the analysis. The PSNR values of DPTHDI [88] and DGTDHS [129] are computed at 0.25 and 1 bpB of fixed payload, respectively. The payload of both 2 x 2 block based schemes is considered as very low. In contrast to DPTHDI [88], proposed WST_2x2 scheme ensures equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2 and 2.5 bpB for “Baboon”, 0.5, 1 and 1.5 bpB of payloads for “Pepper”, 0.5, 1 and 1.5 bpB of payloads for “Airplane” and 0.5, 1, 1.5 and 2 bpB of payloads for “Sailboat” respectively. In contrast to DGTDHS [129], the average PSNR of WST_2x2 scheme is comparable however, the WST_2x2 scheme supports varying payload that offers a spread from 0.5 to 3 bpB.

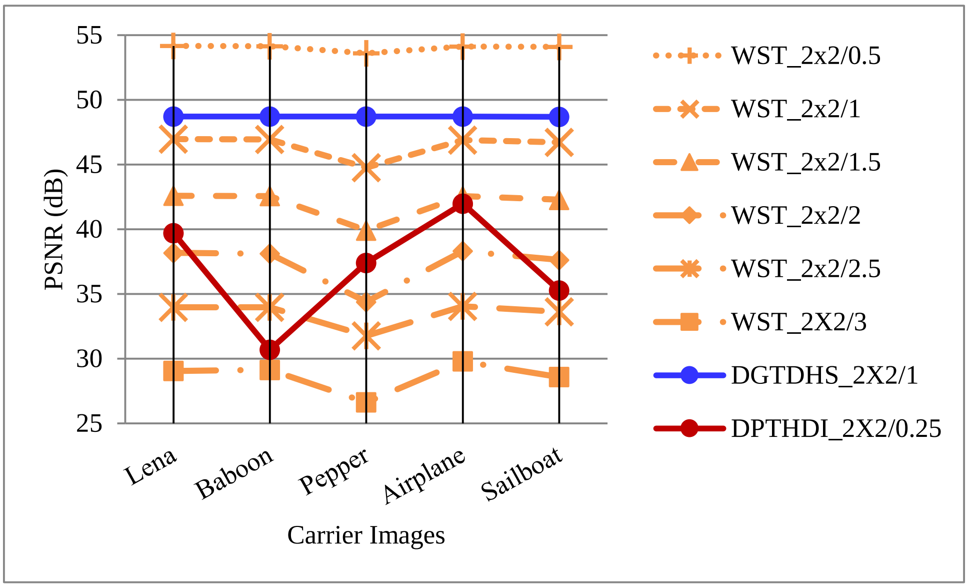


Fig. 5.2. Performance analysis of PSNR (dB) for variable payload based WST_2x2 and fixed payload based Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Table 5.1 revealed that the average peak signal to noise ratio (PSNR) of twenty color images is 54.02 dB at 0.5 bpB and 28.47 dB at 3 bpB, respectively. Fig. 5.3 illustrates the comparative results of WST_2x2, WBT_2x2, WLT_2x2, WDHT_2x2, DPTHDI [88] and DGTDHS [129] in terms of average PSNR and payload with diverse cover images. In contrast to WLT_2x2 and WDHT_2x2, the WST_2x2 obtains tremendous improvement in average PSNR for payload values lies between 0.5 and 3 bpB. The WST_2x2 is also achieved an improved average PSNR over the WBT_2x2 for the payload range [0.5 – 2 bpB] however, the PSNR is almost identical at 2.5 and 3 bpB of payloads. Therefore, WST_2x2 is considered as superior than the WBT_2x2, WLT_2x2 and WDHT_2x2 respectively. An average PSNR of 37.40 dB is obtained at 0.25 bpB for DPTHDI [88] where, “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images has been specified as the cover images. It is seen that the average PSNR for WST_2x2 scheme is superior over DPTHDI [88] at 0.5, 1 and 1.5 bpB of payloads. Again, the average PSNR of 48.70 dB is obtained at 1 bpB of payload for DGTDHS [129] where, the cover images are “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” respectively. Compared to DGTDHS [129], the WST_2x2 provides variable payload however, the average PSNR is lagging at 1 bpB. Fig. 5.3 also demonstrated that the average PSNR for WST_2x2 is less than 30 dB for the payload value 3 bpB. However, the quality could be improved by utilizing a post-embedding quality enhancement or GA based optimization which is to be discussed on chapter 7 and chapter 8.

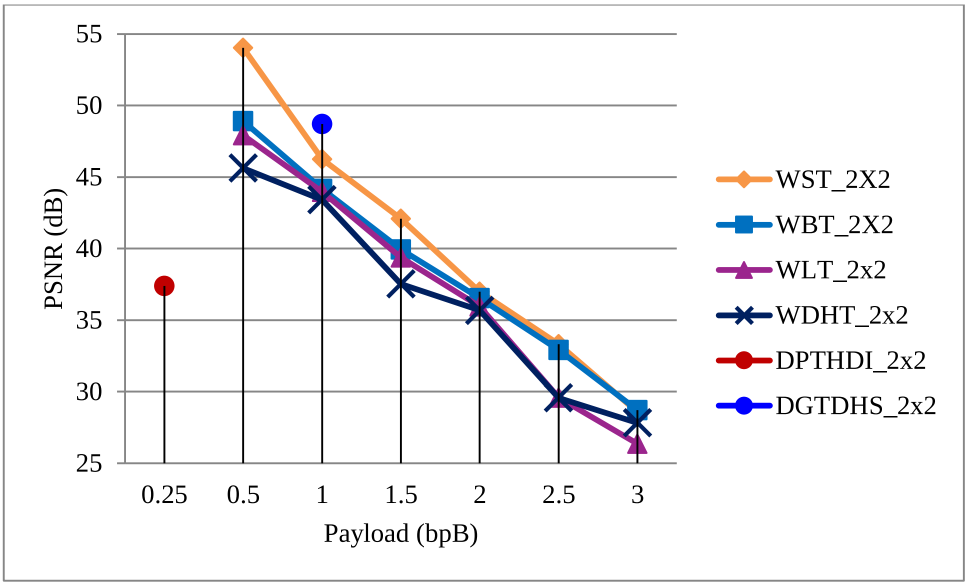


Fig. 5.3. Graphical representation of variation of average PSNR (dB) with respect to payload for WST_2x2, WBT_2X2, WLT_2X2, WDHT_2X2 and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

Fig. 5.4 shows the analysis of standard deviation (SD) values for “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat”, respectively. As far as the payload of 0 bpB is concerned, the standard deviation (SD) at that payload is nothing but the standard deviation (SD) of the original image. It is seen from fig. 5.4 that the Standard deviation (SD) for “Lena”, “Baboon”, “Airplane” and “Sailboat” remains constant up to the payload value of 3 bits per Byte (bpB) however, slight dispersion is observed for “Pepper” for the payload range (2.5 – 3 bpB). Standard deviation (SD) values ensured minimal difference between original and watermarked images for the payload range 0.5 – 3 bpB and therefore, an observer cannot easily differentiate between the source and the watermarked images.

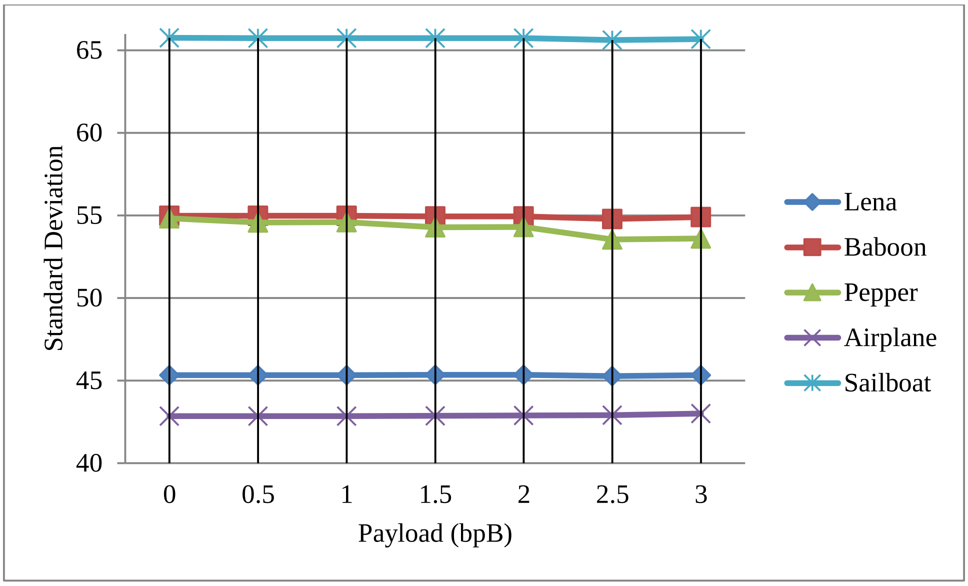


Fig. 5.4. Graphical representation of standard deviation (SD) for WST_2x2 with respect to 0, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

Fig. 5.5 illustrates a graphical representation of standard deviation error (SD) analysis where, the standard deviation (SD) for five benchmark images [130, 131] are computed and analyzed to validate the WST_2x2 scheme. The standard deviation error (SDE) is computed as the absolute difference of standard deviation (SD) values between the original and the watermarked images. The dispersion of error (SDE) is almost zero for “Lena”, “Baboon”, “Airplane” and “Sailboat” images for payload range [0.5 – 3 bpB]. Unlikely, the error is comparatively high for “Pepper” as soon as the payload exceeds 2 bpB of payload and hence, the dispersion is clearly observed.

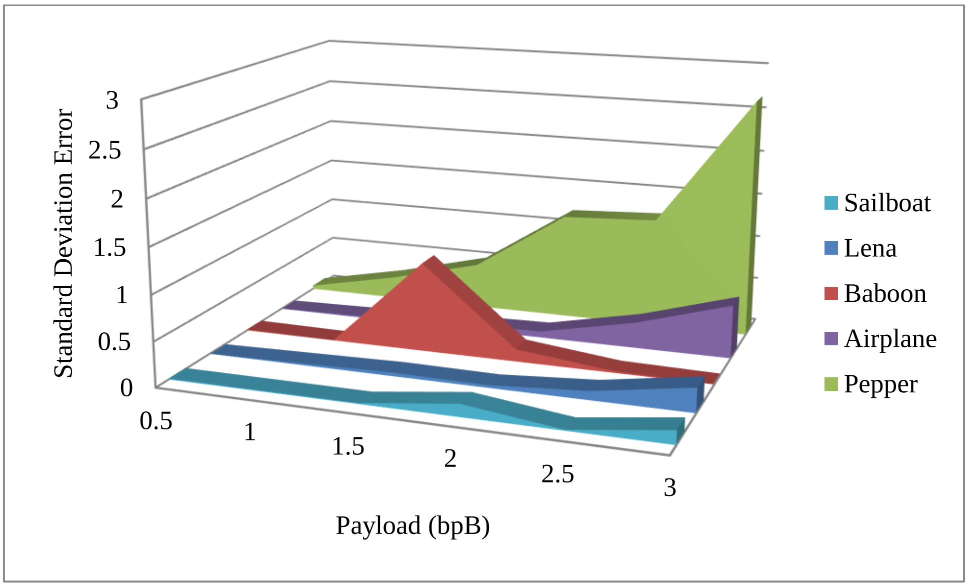


Fig. 5.5. Graphical representation of standard deviation error (SDE) for WST_2x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

5.2.2 1 x 2 Block based Watermark Fabrication

As discussed in section 5.2.1, the pixel components {p_k} can be converted into the transformed components {t_k} based on Stirling Transform (ST) [137-141] as given in equation (5.12).

$$t_\gamma = \sum_{k=1}^{\gamma} \left\{ {\gamma \atop k} \right\} p_k \tag{5.12}$$

where, k = 1, 2, … and $\left\{ {\gamma \atop k} \right\}$, also denoted as S(γ, k) is the Stirling number of the second kind which specify the partitions of a set of size γ into *k* parts as given in equation (5.13).

$$S(n,k) = \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n \tag{5.13}$$

The inverse Stirling Transform (IST) re-computes the pixel components {p_k} from the transformed components {t_k} as given in equation (5.14).

$$p_n = \sum_{k=1}^{\gamma} s(\gamma,k) t_k \tag{5.14}$$

where, *s*(*γ*, *k*) is a Stirling number of the first kind which is computed from the following recurrence relation.

$$s(\gamma,k) = s(\gamma-1,k-1) + (\gamma-1)s(\gamma-1,k)$$

where, the initial conditions s(0, 0) = 1 and $s(\gamma,0)$ = $s(0,\gamma)$= 0 for all γ, $\gamma \geq 1$.

In WST_1x2, the size of each image block is 1 x 2 instead of 2 x 2 as discussed in section 5.2.1. Hence, p = [p_0 p_1] be the 1 x 2 sub-matrix of pixel components and t = [t_0 t_1] represents the 1 x 2 sub-matrix of transformed components as given in equation (5.15).

$$t = pS \xrightarrow{yields} [t_0 \quad t_1] = [p_0 \quad p_1] \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \xrightarrow{yields} \begin{cases} t_0 = p_0 \\ t_1 = p_0 + p_1 \end{cases} \tag{5.15}$$

where, t and p are 1 x 2 matrices and S is the 2 x 2 forward Stirling Transform matrix.

On applying inverse Stirling Transform (IST) over the 1 x 2 sub-matrix of transformed components (t_0, t_1), the pixel components (p_0, p_1) are obtained as given in equation (5.16).

$$p = tS^{-1} \xrightarrow{yields} [p_0 \quad p_1] = [t_0 \quad t_1] \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \xrightarrow{yields} \begin{cases} p_0 = t_0 \\ p_1 = t_1 - t_0 \end{cases} \tag{5.16}$$

where, t and p are 1 x 2 matrices and S^{-1} is the 2 x 2 inverse Stirling Transform matrix.

The proposed technique has been elaborately described at the following sections. The algorithm for insertion, the algorithm for extraction and an example are described in detail in section 5.2.2.1, 5.2.2.2 and 5.2.2.3 respectively. Results and discussions have been elaborated in section 5.2.2.4.

5.2.2.1 Insertion

The carrier image can be considered as consisting of 1 x 2 non-overlapping blocks where, each 1 x 2 sub-image block represents the pair of pixel components. The pixel components are adjusted to stay away from overflow and underflow. Each pair of pixel components are successively converted into transform domain based on Stirling Transform (ST). Arnold's cat map of section 1.3 has been used to scramble the watermark based on a secret key which results enhanced security of the secret watermark. Both transformed components of each 1 x 2 sub-matrix can fabricate varying number of secret bits corresponding to the message digest, size and content of the scrambled watermark. A maximum of three bits from the least significant part may be modified to embed the secret information. The re-computation of pair of pixel components is accomplished based on the inverse Stirling Transform (IST). Repeat the above process in a block-wise manner to obtain the watermarked image in spatial domain.

Algorithm 5.3:

Input: Carrier/cover image (I), 64 bit secret key and the secret watermark (W).

Output: Watermarked image (I′).

Method: Stirling Transform (ST) is used to fabricate the watermark (along with a message digest) into the carrier images by converting the image from spatial domain into transform domain. Embedding bits in transform domain offers variable payload, less distortion and improved security. The comprehensive steps of embedding are as follows:

Step 1. 128 bits message digest (MD) is generated from the authenticating message/image.

Step 2. The authenticating watermark size (in bits) is obtained by embedding the watermark into the three sub-matrices of the U x V color image as given in equation (5.20).

$$W_{size} = B \times \left(3 \times (U \times V)\right) - (MD + L) \qquad (5.20)$$

where, B, MD and L represents the average embedding payload in terms of bits per Byte, the message digest obtained *from the watermark and the header information corresponding to the size of the watermark respectively. The MD and L are consisting of 128 and 32 bits while the usual values of B are 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively.*

Step 3. Watermark is scrambled through a 64 bits/8 characters secret key. A pseudo-random number (r) is obtained by summing up the ASCII values of all characters (S) and then modulo it with the period (T), where the period (T) is 192 as mentioned in section 1.3. ASCII value of each character (A_i) is checked cyclically with respect to the pseudo-random number (r) where, i fall in the range 1 to 8. If the ASCII value of a character (A_i) is greater than the pseudo-random number (r) then obtain their difference and subtract it from the summation (S); otherwise, add the difference accordingly. Mathematical expressions are given in equation (5.21) and (5.22).

$$r = S\ mod\ T$$

where, $$S = \sum_{i=1}^{8} A_i \text{ and } T = 192 \qquad (5.21)$$

$$S = \begin{cases} S - (r - A_i) : A > r \\ S + (r - A_i) : A \leq r \end{cases} \qquad (5.22)$$

Each pair of pixel components of the secret watermark is scrambled by transforming it through r iterations as mentioned in equation (1.3) of section 1.3.

Step 4. The cover image (I) is partitioned into (Red\Green\Blue) pairs of pixel components (p_i, p_{i+1}) in a sliding window manner.

Step 5. A pre-embedding adjustment is applied on each pixel component p, to avoid the overflow/underflow for the payload of B bits per byte of watermark as given in equation (5.23).

$$p = \begin{cases} \left(2^8 - 2^{\lceil B \rceil + 1}\right) : p \geq \left(2^8 - 2^{\lceil B \rceil + 1}\right) \\ 2^{\lceil B \rceil + 1} : p \leq 2^{\lceil B \rceil + 1} \end{cases} \qquad (5.23)$$

Step 6. Stirling Transform (ST) is applied over each non-overlapping pair of pixel components (p_i, p_{i+1}) to generate transformed components (t_i, t_{i+1}) in a sliding window manner. The *transformed components are represented in matrix form as given in equation (5.24).*

$$[t_i \quad t_{i+1}] = [p_i \quad p_{i+1}]\begin{bmatrix} s_{i,j} & s_{i+1,j} \\ s_{i,j+1} & s_{i+1,j+1} \end{bmatrix} \quad (5.24)$$

where, $s_{m,n}$ = 1|1|0|1 for 0 ≤ m, n ≤ 1.

Step 7. λ_1 /λ_2 bits from the secret bit-stream S (corresponding to the message digest (MD), size (L) and the content (W) of the watermark) are subsequently fabricated on first/second transformed component (t_c) starting from the least significant bit position (i.e., LSB-0) toward higher order bit position. To achieve the payload of B bits per Byte (bpB), the generalized form of λ bits of secret information fabrication on each transformed component (t_c) is given in equation (5.25).

$$t'_c = t_c + W(\lambda)$$

$$\lambda = \begin{cases} \lfloor B \rfloor : \lambda = \lambda_1 \\ \lceil B \rceil : \lambda = \lambda_2 \end{cases} \quad (5.25)$$

where, 0 < B ≤ 3, the difference between two successive payload values (ΔB) is 0.5, and for all λ, 0 ≤ λ ≤ 3. The '+' operator indicates the fabrication of λ bits of watermark (W) into the transformed component (t_c) whereas, t'_c denote the embedded component.

Step 8. Inverse Stirling Transform (IST) is applied on each pair of embedded components (t'_i, t'_{i+1}) to re-compute the pixel components (p'_i, p'_{i+1}) as given in equation (5.26).

$$[p'_i \quad p'_{i+1}] = [t'_i \quad t'_{i+1}]\begin{bmatrix} s'_{i,j} & s'_{i+1,j} \\ s'_{i,j+1} & s'_{i+1,j+1} \end{bmatrix} \quad (5.26)$$

where $s'_{m,n}$ = 1|0|-1|1 for 0 ≤ m, n ≤ 1.

Step 9. Repeat steps 3 to 8 to embed the complete authenticating message/image size, content and for message digest MD in a channel wise manner. The block embedding operation in succession generates the watermarked image (I').

Step 10. Stop.

5.2.2.2 Extraction

The watermarked image *(I')* can be treated as the set of 1 x 2 non-overlapping blocks of pixel components in row major order. The transformation of each block/pair of pixel components into the pair of transformed components is carried out by applying Stirling Transform (ST). The reverse procedure is applied to extract the fabricated bits of the scrambled watermark size, content and message digest (MD) from the transformed components in variable proportion as similar to the embedding procedure. Successive extraction operations ensure the retrieval of message digest and the watermark in scrambled form. Arnold's cat map of section 1.3 has been used to unscramble the secret watermark (along with the message digest) based on a secret key. Therefore, the extraction process is carried out in transform domain; the message digest (MD), size and the content of the watermark are also reconstructed. Obtain another message digest (MD*′)* from the extracted watermark and the same is compared with the extracted message digest (MD). Identical contents of MD and MD′ yields the successful authentication however, a single bit manipulation may hamper the authentication system and makes the process as unauthenticated. The extraction procedure is described in algorithm 5.4.

Algorithm 5.4:

Input: Watermarked image (I′) and the 64 bit secret key.

Output: Watermark image (W) and the message digest (MD).

Method: Stirling Transform (ST) is applied on each pair of pixel components to extract the watermark (along with a message digest) from the watermarked image by converting the image from spatial domain into transform domain. Successive extracted bits forms the message digest, size and content of the scrambled watermark which are unscrambled by utilizing Arnold's cat map along with a secret key. Extracted message digest is compared with the re-computed message digest for authentication. The detailed steps of extraction are as follows:

Step 1. The watermarked image (I') is partitioned into (Red\Green\Blue) pairs of pixel components (p_i, p_{i+1}) in a sliding window manner.

Step 2. Stirling Transform (ST) is applied on each pair of pixel components (p_i, p_{i+1}) to generate transformed components (t_i, t_{i+1}) which are represented in matrix form as given in equation (5.27).

$$[t_i \quad t_{i+1}] = [p_i \quad p_{i+1}] \begin{bmatrix} s_{i,j} & s_{i+1,j} \\ s_{i,j+1} & s_{i+1,j+1} \end{bmatrix} \qquad (5.27)$$

where, $s_{m,n}$ = 1|1|0|1 for $0 \le m, n \le 1$.

Step 3. λ_1 / λ_2 bits of the secret bit-stream (S) are extracted from first/second transformed component starting from the least significant bit position (LSB-0) toward higher order bit position. For payload value of B bits per Byte (bpB), the generalized form of λ bits watermark extraction from each transformed component is given in equation (5.28).

$$\lambda = \begin{cases} \lfloor B \rfloor : \lambda = \lambda_1 \\ \lceil B \rceil : \lambda = \lambda_2 \end{cases} \tag{5.28}$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all N, $0 \leq N \leq 3$.

Step 4. The secret key generates the pseudo-random number (r) by adding the ASCII values of all characters (S) followed by modulo it with the period (T) where, T = 192. ASCII value of each character (A_i) is checked cyclically respecting the pseudo-random number (r) where, for all i, $1 \leq i \leq 8$. If the ASCII value of a character (A_i) is greater than the pseudo-random number (r) then obtain their difference and subtract it from the summation (S); otherwise, add the difference accordingly. Equations (5.29) and (5.30) give following expressions.

$$r = S \; mod \; T$$

where, $$S = \sum_{i=1}^{8} A_i \text{ and } T = 192 \tag{5.29}$$

$$S = \begin{cases} S - (r - A_i) : A > r \\ S + (r - A_i) : A \leq r \end{cases} \tag{5.30}$$

Each pair of pixel components of the scrambled watermark is unscrambled by transforming it through (192 – r) iterations as mentioned in equation (1.3) of section 1.3.

Step 5. For each eight bits extraction of the secret bit-stream (S), construct one R/G/B component/ alphabet.

Step 6. The inverse Stirling Transform (IST) is applied on each pair of embedded components (t'_i, t'_{i+1}) to generate the pixel components (p'_i, p'_{i+1}) in spatial domain as given in equation (5.31).

$$[p'_i \quad p'_{i+1}] = [t'_i \quad t'_{i+1}] \begin{bmatrix} s'_{i,j} & s'_{i+1,j} \\ s'_{i,j+1} & s'_{i+1,j+1} \end{bmatrix} \tag{5.31}$$

where $s'_{m,n}$ = 1|0|-1|1 for $0 \leq m, n \leq 1$.

Step 7. Repeat step 1 to step 5 until and unless the fabricated message digest, size and the content of the watermark is recovered.

Step 8. Obtain 128 bits message digest MD' from the extracted watermark.

Step 9. Compare message digest MD' with the extracted message digest MD. If MD and MD' are alike, the image is authorized, else unauthorized.

Step 10. Stop.

5.2.2.3 Example

Consider the carrier image as the set of 1 x 2 non-overlapping blocks consisting of pixel components. Each block of pixel components corresponding to the red, green and blue channels are denoted as R_1, G_1 and B_1 respectively. The 1 x 2 sub-matrices are as follows:

$$R_1 = [164\ 3]\quad G_1 = [253\ 57]\quad B_1 = [71\ 31]$$

Pixel components are adjusted prior to embedding as given in equation (5.23) and the modified upper bound (UB) and lower bound (LB) are computed to avoid the overflow and underflow. The example deals with the payload value of three (i.e., B = 3) to obtain the upper bound (UB) and the lower bound (LB) as follows:

$$UB = \left(2^8 - 2^{\lceil 3 \rceil + 1}\right) = (2^8 - 2^4) = (256 - 16) = 240$$
$$LB = 2^{\lceil 3 \rceil + 1} = 2^4 = 16$$

The pixel components exceeding this range are adjusted immediately. The 1 x 2 adjusted sub-matrices are obtained as given below:

$$R_1 = [164\ 16]\quad G_1 = [240\ 57]\quad B_1 = [71\ 31]$$

Stirling Transform (ST) converts each 1 x 2 sub-matrix or pair of pixel components into transformed components. The 1 x 2 sub-matrices of transformed components are $T(R_1)$, $T(G_1)$ and $T(B_1)$ as given below:

$$T(R_1) = [164\ 180]\quad T(G_1) = [240\ 297]\quad T(B_1) = [71\ 102]$$

Let the secret bit-stream "*011101110011100010*" is to be fabricated into the transformed components based on the embedding rule as given in equation (5.25). In this example, three bits are fabricated ($\lambda = \lambda_1 = \lambda_2 = 3$) on each transformed component starting from LSB-0 toward the higher order bit positions. Hence, the 1 x 2 sub-matrices of embedded components becomes:

$$T'(R_1) = [166\ 181]\quad T'(G_1) = [243\ 302]\quad T'(B_1) = [65\ 98]$$

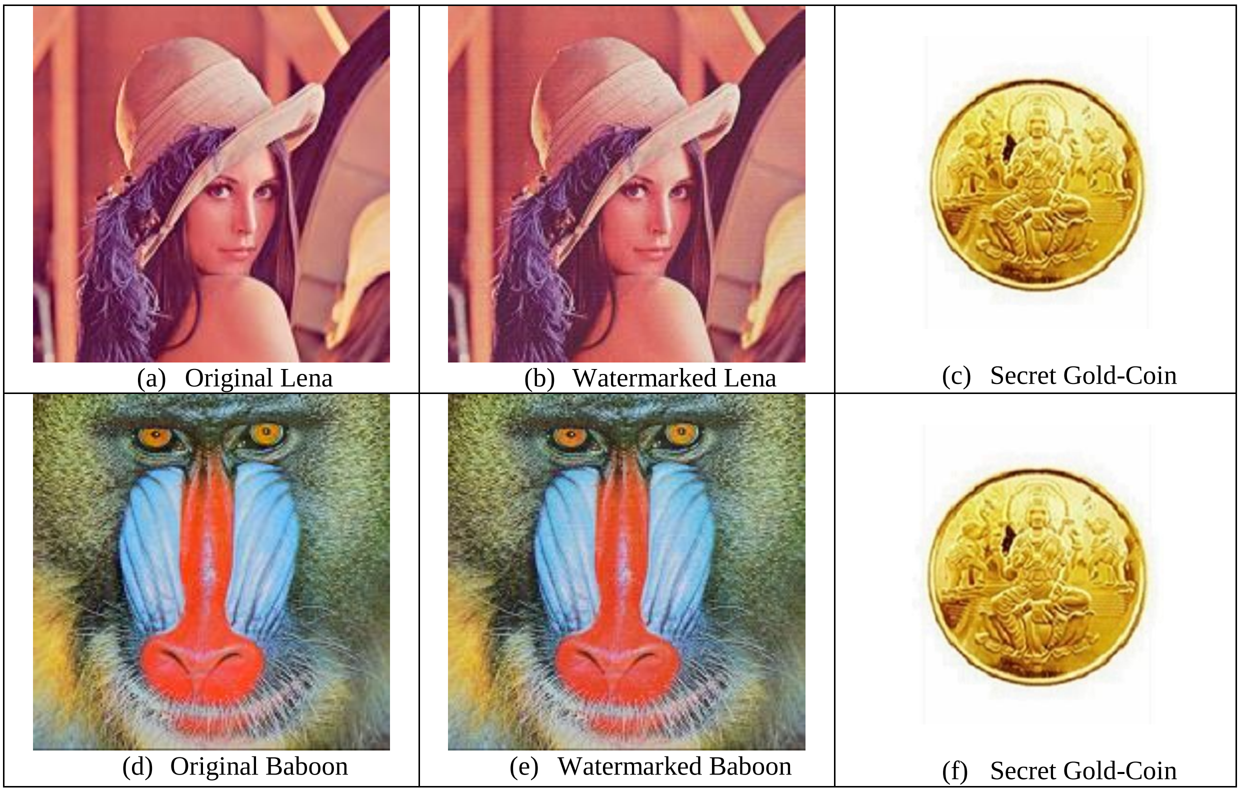
Application of inverse Stirling Transform (IST) on each pair of embedded components yields the re-generated pairs of pixel components in spatial domain as given below:

$$R'_1 = [166\ 15] \quad G'_1 = [243\ 59] \quad B'_1 = [65\ 33]$$

The re-computed pixel components are non-fractional, non-negative and less than or equal to 255.

5.2.2.4 Results and Discussions

The performance metrics such as the peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD) and standard deviation error (SDE) has been considered to compute the results of the proposed 1 x 2 block based watermark fabrication (WST_1x2) scheme with respect to increasing payload. Twenty cover images of dimension 512 x 512 are taken to fabricate the varying sizes of the secret watermark as given in fig. 1.1. Comparison is made among the WST_1x2, WST_2x2, WBT_1x2, WLT_1x2, WDHT_1x2 and Varsaki et al.'s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] schemes respectively. Extracted "Gold Coin" and the different states of modifications of carrier images viz. "Lena", "Baboon" and "Pepper" (before and after embedding the watermark) are shown in fig. 5.6.



(a) Original Lena

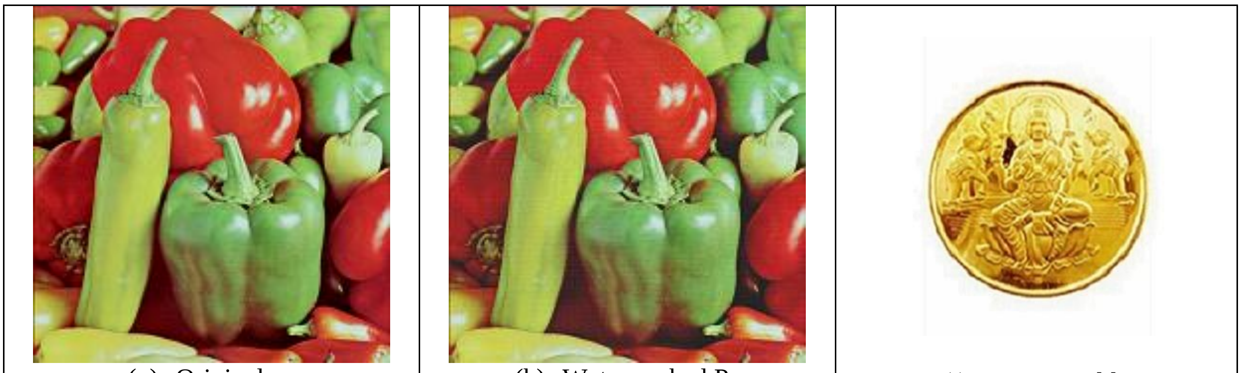
(b) Watermarked Lena

(c) Secret Gold-Coin

(d) Original Baboon

(e) Watermarked Baboon

(f) Secret Gold-Coin


(g) Original pepper

(h) Watermarked Pepper

(i) Secret Gold-Coin

Fig 5.6. Cover, watermarked and authenticating watermark image in the proposed WST_1x2 technique

Table 5.2 reveals the perceptual quality distortion of the proposed WST_1x2, which is quantified in terms of the peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) respectively. These quality metrics are computed for twenty color images (as given in fig. 1.1) with respect to 0.5, 1.0, 1.5, 2.0, 2.5 and 3 bits per Byte (bpB) of payloads. The quality of watermarked images sharply decreases as payload increases. It can also be noticed that the WST_1x2 doesn't give a uniform PSNR for all watermarked images since each watermarked image has different pixel distribution; therefore, the watermark demands larger or smaller changes in the pixel values. The WST_1x2 scheme discloses the minimum PSNR of 31.22 dB at 3 bpB of payload for the "Desert" image and that of the maximum PSNR obtained is 54.15 dB at 0.5 bpB of payload for the "Foster City" image. The minimum PSNR value of greater than 30 dB is the indication perceptible image quality [148]. The inverse relationship between PSNR and MSE ensures that the increase in PSNR causes decrease in MSE and vice-versa. The minimum and maximum values of MSE are achieved as 0.24 for "Foster City" at payload of 0.5 bpB and 49.01 for "Desert" at 3 bpB of payload respectively. Minimum values of IF, SSIM and UIQ are 0.992463 (Desert), 0.959043 (Pepper) and 0.612765 (Splash) respectively whereas, the maximum values of IF, SSIM and UIQ are 0.999992 (Airplane), 0.999919 (Athens) and 0.999333 (San Diego) respectively. Minimum values corresponding to the metrics IF, SSIM and UIQ are computed at 3 bpB of payload however, the maximum values of IF and UIQ are computed at 0.5 bpB while the SSIM is computed at 1 bpB respectively. In addition, the values are close to one which indicates high similarity between the original and watermarked images with respect to increasing payload. The average values are computed for various metrics of twenty different carrier images at variable payload to summarize the experimental results.

Table 5.2. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WST_1x2 technique

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.126228	0.251453	0.999984	0.999646	0.994776
	1.0	50.038663	0.644486	0.999959	0.999683	0.985716
	1.5	45.839374	1.694895	0.999893	0.998066	0.966949
	2.0	42.777034	3.430653	0.999782	0.998256	0.937498
	2.5	39.009105	8.169017	0.999483	0.991818	0.884246
	3.0	36.011306	16.291075	0.998969	0.991217	0.803432
Baboon	0.5	54.041626	0.256399	0.999986	0.999870	0.998967
	1.0	50.027607	0.646129	0.999965	0.999871	0.997203
	1.5	45.778715	1.718734	0.999909	0.999227	0.992504
	2.0	42.726648	3.470687	0.999816	0.999274	0.985570
	2.5	38.945177	8.290154	0.999561	0.996347	0.968267
	3.0	36.002119	16.325570	0.999136	0.996199	0.947010
Pepper	0.5	48.034773	1.022357	0.999900	0.990122	0.977761
	1.0	46.665162	1.401406	0.999868	0.990282	0.972085
	1.5	41.028002	5.131922	0.999506	0.980052	0.949937
	2.0	39.662453	7.028053	0.999347	0.979978	0.928282
	2.5	34.276428	24.290561	0.997661	0.959526	0.869532
	3.0	32.943205	33.018690	0.996918	0.959043	0.801502
Airplane	0.5	53.971148	0.260594	0.999992	0.999579	0.983422
	1.0	49.968752	0.654945	0.999981	0.999628	0.958382
	1.5	45.808036	1.707169	0.999951	0.997736	0.916333
	2.0	42.721697	3.474646	0.999900	0.997954	0.859607
	2.5	39.250148	7.727971	0.999778	0.990816	0.783594
	3.0	36.107674	15.933564	0.999543	0.990030	0.680961
Sailboat	0.5	53.339840	0.301367	0.999984	0.999694	0.995547
	1.0	49.683212	0.699453	0.999964	0.999722	0.988372
	1.5	45.247688	1.942278	0.999902	0.998313	0.973706
	2.0	42.398556	3.743041	0.999811	0.998444	0.951132
	2.5	38.265023	9.695674	0.999514	0.992025	0.910316
	3.0	35.545869	18.133991	0.999088	0.991559	0.852439
Earth	0.5	54.155713	0.249752	0.999985	0.999732	0.997668
	1.0	50.043001	0.643843	0.999961	0.999767	0.993535
	1.5	45.884052	1.677548	0.999900	0.998573	0.984585
	2.0	42.788046	3.421966	0.999796	0.998706	0.969008

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
	2.5	39.195086	7.826574	0.999531	0.993778	0.934218
	3.0	36.105048	15.943199	0.999046	0.993294	0.874835
San Diego	0.5	54.121904	0.251703	0.999990	0.999899	0.999333
	1.0	50.037489	0.644660	0.999976	0.999910	0.998168
	1.5	45.870015	1.682979	0.999937	0.999457	0.995690
	2.0	42.783038	3.425914	0.999872	0.999518	0.991459
	2.5	39.258961	7.712305	0.999712	0.997798	0.983078
	3.0	36.168298	15.712689	0.999414	0.997598	0.966718
Splash	0.5	49.270565	0.769172	0.999917	0.989397	0.963107
	1.0	47.535985	1.146784	0.999885	0.989492	0.935045
	1.5	42.272921	3.852904	0.999599	0.982457	0.885904
	2.0	40.580076	5.689482	0.999435	0.982565	0.819594
	2.5	35.830677	16.982930	0.998231	0.968249	0.730973
	3.0	34.106459	25.260060	0.997500	0.967503	0.612765
Oakland	0.5	51.278461	0.484433	0.999968	0.999417	0.997711
	1.0	48.69399	0.878377	0.999947	0.999427	0.995700
	1.5	43.736225	2.750792	0.999825	0.997787	0.989885
	2.0	41.608390	4.489948	0.999732	0.997920	0.981900
	2.5	36.973995	13.052113	0.999162	0.992137	0.961992
	3.0	34.863847	21.217567	0.998723	0.991817	0.930522
Foster City	0.5	54.156509	0.249706	0.999991	0.999518	0.993383
	1.0	50.058037	0.641618	0.999977	0.999580	0.982171
	1.5	45.880965	1.678741	0.999940	0.997399	0.958548
	2.0	42.808839	3.405621	0.999879	0.997700	0.920433
	2.5	39.297331	7.644466	0.999727	0.989504	0.850956
	3.0	36.215523	15.542753	0.999446	0.988555	0.751462
Anhinga	0.5	50.822348	0.538078	0.999958	0.999807	0.882629
	1.0	49.533637	0.723963	0.999944	0.999856	0.881614
	1.5	43.669832	2.793168	0.999786	0.998910	0.873729
	2.0	41.719398	4.376636	0.999664	0.999092	0.854138
	2.5	37.109516	12.651112	0.999031	0.995248	0.841166
	3.0	34.534856	22.887307	0.998244	0.994607	0.773523
Athens	0.5	51.881060	0.421671	0.999966	0.999864	0.971365
	1.0	51.1353861	0.500658	0.999960	0.999919	0.970370
	1.5	45.214676	1.957098	0.999843	0.999235	0.962239
	2.0	42.756419	3.446976	0.999724	0.999430	0.937980

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	38.421398	9.352775	0.999253	0.996791	0.924943
	3.0	35.030227	20.420087	0.998368	0.996110	0.818906
Bardowl	0.5	50.094251	0.636290	0.999934	0.999464	0.997727
	1.0	49.020251	0.814807	0.999916	0.999472	0.997147
	1.5	43.322751	3.025555	0.999688	0.997372	0.990238
	2.0	41.458075	4.648071	0.999526	0.997380	0.984834
	2.5	35.677007	17.594608	0.998175	0.983329	0.949627
	3.0	33.580345	28.513187	0.997089	0.983225	0.921430
Barnfall	0.5	53.104854	0.318122	0.999950	0.999784	0.998572
	1.0	50.239177	0.615407	0.999899	0.999812	0.996867
	1.5	45.981757	1.640229	0.999733	0.998680	0.993409
	2.0	42.888873	3.343435	0.999462	0.998814	0.985736
	2.5	39.002730	8.181017	0.998610	0.993452	0.971552
	3.0	35.830111	16.985144	0.997188	0.993011	0.928967
Butrfly	0.5	52.514547	0.364439	0.999974	0.999853	0.995615
	1.0	50.882130	0.530722	0.999962	0.999869	0.993905
	1.5	45.546494	1.813138	0.999871	0.999054	0.988846
	2.0	42.676235	3.511210	0.999750	0.999160	0.977159
	2.5	38.733965	8.703296	0.999383	0.996292	0.965616
	3.0	35.154385	19.844573	0.998594	0.995988	0.898541
Bobcat	0.5	52.161319	0.395319	0.999944	0.999696	0.749616
	1.0	51.007421	0.515630	0.999927	0.999725	0.748645
	1.5	45.291796	1.922651	0.999729	0.998063	0.742750
	2.0	42.696930	3.494518	0.999507	0.998185	0.729118
	2.5	38.017639	10.263994	0.998548	0.990576	0.713496
	3.0	34.836581	21.351195	0.996988	0.990071	0.666705
Bodie	0.5	50.196920	0.621424	0.999888	0.999554	0.978414
	1.0	48.106353	1.005645	0.999822	0.999581	0.975564
	1.5	43.250645	3.076208	0.999451	0.997818	0.967319
	2.0	41.322708	4.795230	0.999164	0.997849	0.959780
	2.5	35.262399	19.357103	0.996477	0.978216	0.921333
	3.0	33.734047	27.521720	0.995091	0.977917	0.893563
Bluheron	0.5	52.656006	0.352760	0.999956	0.999772	0.996039
	1.0	50.119309	0.632629	0.999922	0.999779	0.992999
	1.5	46.284380	1.529827	0.999812	0.998504	0.987934
	2.0	43.018555	3.245075	0.999603	0.998606	0.971361

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	40.433785	5.884394	0.999279	0.994634	0.959578
	3.0	36.228471	15.496485	0.998101	0.994274	0.885957
Colomtn	0.5	52.128962	0.398276	0.999968	0.999769	0.984849
	1.0	49.555255	0.720368	0.999943	0.999805	0.982616
	1.5	44.851718	2.127690	0.999833	0.998717	0.978928
	2.0	42.284818	3.842363	0.999699	0.998868	0.968216
	2.5	38.033871	10.225704	0.999201	0.994232	0.954791
	3.0	35.213179	19.577734	0.998470	0.993821	0.906710
Desert	0.5	45.961851	1.647764	0.999735	0.997282	0.993767
	1.0	45.142962	1.989683	0.999685	0.997345	0.993226
	1.5	39.369014	7.519325	0.998803	0.991453	0.982390
	2.0	38.479549	9.228378	0.998555	0.991442	0.978998
	2.5	32.230406	38.908072	0.993917	0.968752	0.939261
	3.0	31.227859	49.011100	0.992463	0.968320	0.916325
Average Case	0.5	51.900940	0.489554	0.999949	0.998586	0.972513
	1.0	49.374690	0.802561	0.999923	0.998626	0.966967
	1.5	44.506450	2.562143	0.999746	0.996344	0.954091
	2.0	42.007820	4.275595	0.999601	0.996457	0.934590
	2.5	37.661230	12.625690	0.998712	0.988176	0.900927
	3.0	34.971970	21.749380	0.997919	0.987708	0.841614

The comparison of average PSNR values among the WST_1x2, DPTHDI [88] and DGTDHS [129] showed that the WST_1x2 yields less quality degradation and varying payload that offers a spread from 0.5 to 3 bpB as evident from fig. 5.7. Unlike to WST_1x2, the payload values of DPTHDI [88] and DGTDHS [129] are 0.25 and 1 bpB respectively which are considered as fixed and significantly low. In contrast to DPTHDI [88], the WST_1x2 ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB for “Baboon”, 0.5, 1, 1.5 and 2 bpB of payloads for “Pepper”, 0.5, 1, 1.5 and 2 bpB of payloads for “Airplane” and 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for “Sailboat” respectively. Compared to the DGTDHS [129], the WST_1x2 ensured higher PSNR (dB) at 0.5 and 1 bpB of payloads for “Lena”, 0.5 and 1 bpB of payloads for “Baboon”, 0.5 bpB of payload for “Pepper”, 0.5 and 1 bpB of payloads for “Airplane” and 0.5 and 1 bpB of payloads for “Sailboat” respectively. Since, the PSNR values for above mentioned five images never falls below 30 dB, the quality of the watermarked images are considered to be well perceptible [148].

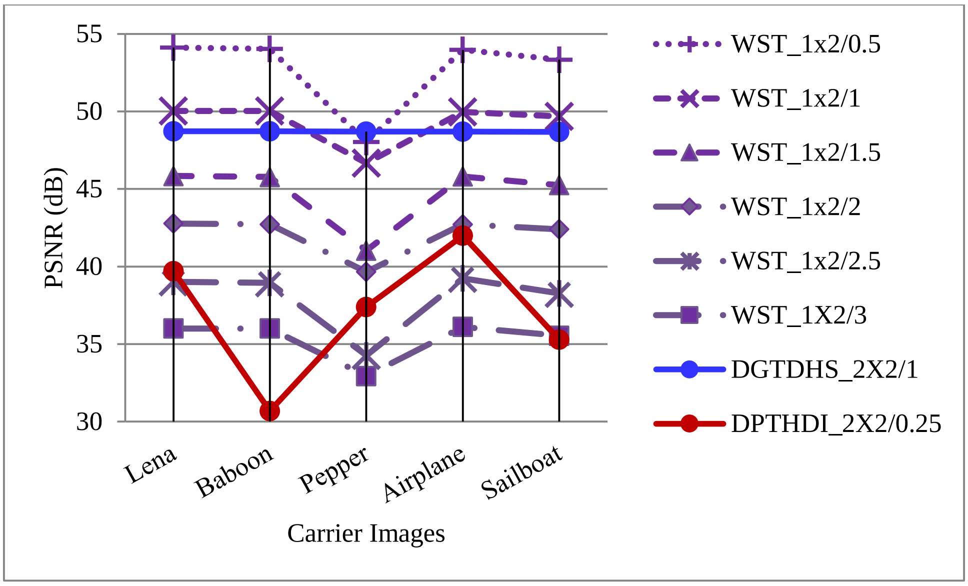


Fig. 5.7. Performance analysis of PSNR (dB) for variable payload based WST_1x2 and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Table 5.2 shows the lowest and highest average PSNR values of WST_1x2 as 34.97 and 51.90 dB respectively. Twenty color images of dimension 512 x 512 as given in fig. 1.1 are taken to compute the average PSNR values with respect to increasing payload. Fig. 5.8 depicts the comparative analysis of visual clarity in terms of average PSNR with respect to payload values for WST_1x2, WST_2x2, WBT_1x2, WLT_1x2, WDHT_1x2, DPTHDI [88] and DGTDHS [129] respectively. In contrast to WST_2x2 and WBT_1x2, the WST_1x2 obtains tremendous improvement in average PSNR values for the payload range [1 – 3 bpB]. The WST_1x2 also provides the higher PSNR values for payload range [0.5 – 3 bpB] in comparison with the WLT_1x2 and WDHT_1x2 schemes respectively. The average PSNR for DPTHDI [88] is obtained as 37.40 dB by considering the average of PSNR values for “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” at 0.25 bpB of payload. In contrast to DPTHDI [88], the WST_1x2 ensured equal or higher average PSNR values at 0.5, 1, 1.5, 2 and 2.5 bpB of payloads. The average PSNR value for DGTDHS [129] is 48.70 dB by considering the average of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” images at 1 bpB of payload. Compared to DGTDHS [129], the WST_1x2 ensured equal or higher average PSNR values at 0.5 and 1 bpB of payloads. In addition, the offered payload also draws the attention since it is varying for the dispersion of 0.5 to 3 bpB. The WST_1x2 scheme produces high-quality watermarked images in output as the average PSNR values obtained with respect to the payload range (0.5 – 3 bpB) becomes greater than or equal to 30 dB [148].

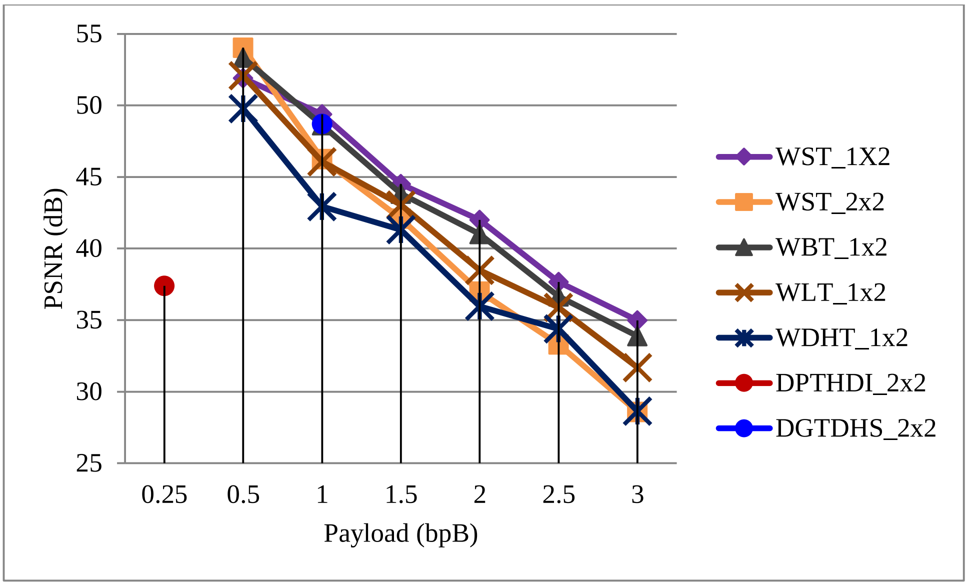


Fig. 5.8. Graphical representation of variation of average PSNR (dB) with respect to payload for WST_1X2, WST_2X2, WBT_1X2, WLT_1X2, WDHT_1x2 and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

In fig. 5.9, the standard deviation (SD) for "Lena", "Baboon", "Airplane" and "Sailboat" have been computed by averaging the standard deviation (SD) values corresponding to red, green and blue channels for payload range (0 – 3 bpB). Standard deviation (SD) at 0 bpB is nothing but the standard deviation (SD) of the original image. The deviation for the watermarked images in comparison with the original images at varying payload for "Lena", "Baboon", "Airplane" and "Sailboat" are remained constant up to 3 bpB. On the contrary, the deviation is slightly decreasing for "Pepper" while the payload value increases from 1 bpB.

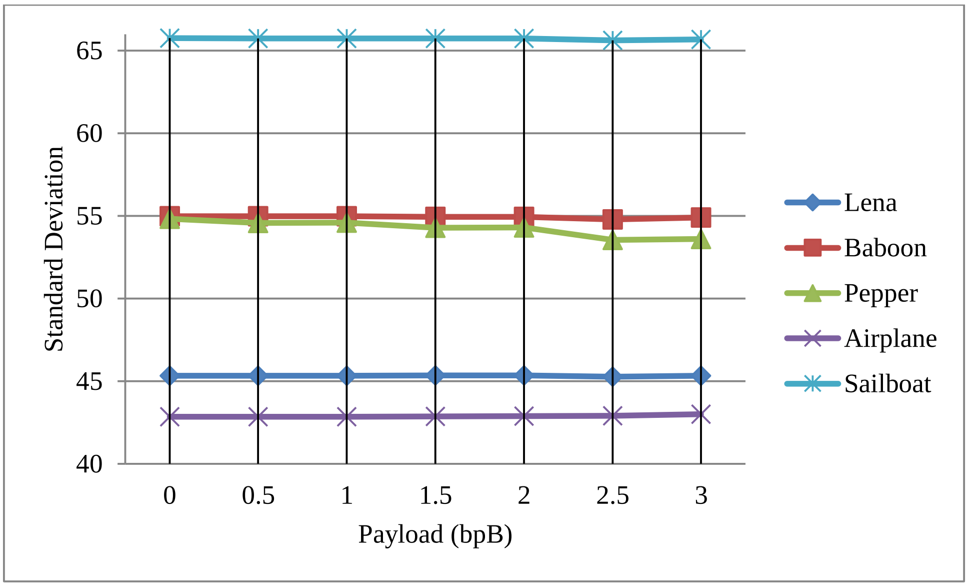


Fig. 5.9. Graphical representation of standard deviation (SD) for WST_1x2 with respect to 0, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

In fig. 5.10, the standard deviation error (SDE) analysis has been made for five different images such as “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” respectively. The standard deviation error (SDE) is obtained by evaluating the absolute difference of standard deviation (SD) values between the watermarked and original images at varying payload. The graph also illustrates the variation of standard deviation error (SDE) values with respect to increasing payload (bpB). The standard deviation error (SDE) is almost zero for the payload variation from the spread of 0.5 to 3 bpB for “Lena”, “Baboon”, “Airplane” and “Sailboat” respectively. The maximum standard deviation error (SDE) is 1.2 which is observed for “Pepper” image at 3 bpB of payload. However, the error is treated as very low and hence, the changes made in the watermarked images in comparison to the original images are not perceivable.

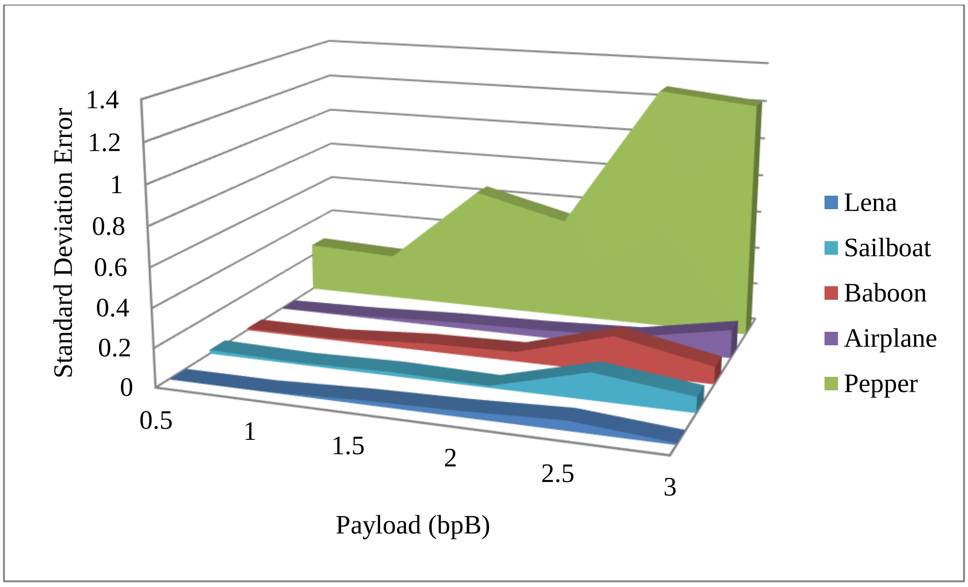


Fig. 5.10. Graphical representation of standard deviation error (SDE) for WST_1x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

Fig. 5.11 illustrates the comparative analysis of standard deviation error (SDE) between WST_2x2 and WST_1x2 respectively. Five color images viz. “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” are taken to analyze the performance. The standard deviation error (SDE) values are almost 0 for “Lena”, “Baboon”, “Airplane” and “Sailboat” images for the payload spread of 0.5 to 2.5 bpB however, the standard deviation error (SDE) is maximum at 3 bpB of payload for “Pepper” image. The rate of increase of error has been minimized considerably for the 1 x 2 block over the 2 x 2 block based watermarking for the payload values 0.5 to 3 bpB. Therefore, the structural quality distortion made to the watermarked images of WST_1x2 is much less than the WST_2x2 at respective payload values.

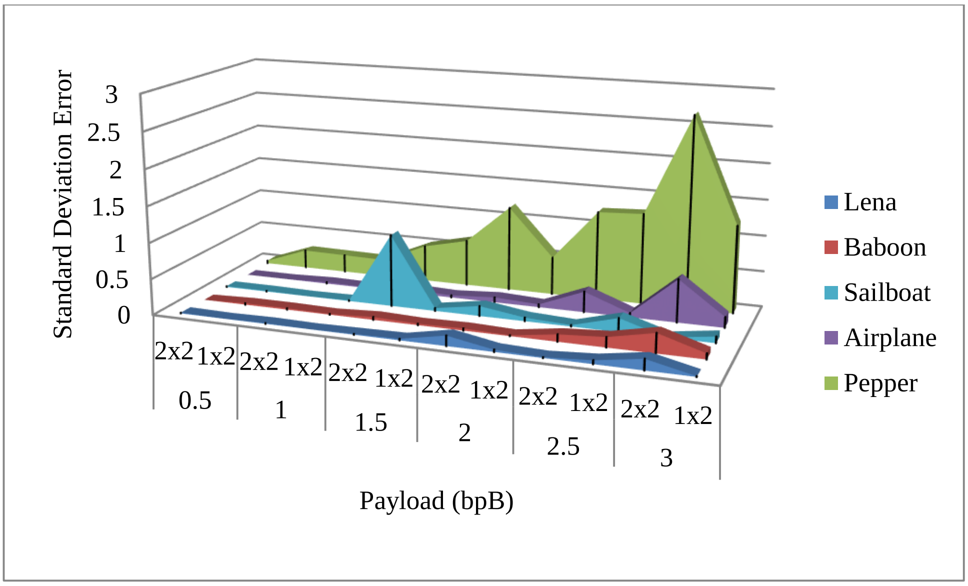


Fig. 5.11. Pictorial representation of variation of standard deviation error (SDE) between WST_2x2 and WST_1x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

5.3 Salient Features

The WST_2x2 and WST_1x2 are two novel fragile watermarking schemes in Stirling Transform (BT) domain which shows improvement of performances compared to the existing schemes [88, 129] because the existing techniques suffer from high distortion, low as well as fixed payload, lacking in usage of color image as the cover and the high computational overhead.

Some of the relevant features of the proposed techniques have been pointed out in the following section.

Security of the proposed watermarking technique (WST_1x2) has been enhanced by introducing an additional level of security by scrambling the watermark through Arnold's cat map. Arnold's cat map is a chaos based scheme which is utilized for transforming the values of each pair of pixel components of a given watermark. The process of scrambling is applied in spatial domain prior the fabrication. The scrambled bits of the watermark are embedded into the transformed components in Stirling Transform (ST) domain to improve the security.

Both of the proposed schemes perform a pure integer calculation which is a significant advantage of Stirling Transform (ST) in terms of computational complexity. To address the problem of overflow/underflow, a pixel adjustment is done prior to embedding. However, as Stirling Transform (ST) is highly sensitive against small modification of a transformed component for 2 x 2 blocks, a re-adjustment operation has also been incorporated to avoid the rarely occurred overflow/underflow by adjusting the transformed components. It is seen that most of the existing schemes offers fixed payload which is relatively low; however, proposed WST_2x2 and WST_1x2 schemes are based on variable payload (i.e., 0.5 to 3 bpB) by keeping perceptible quality of the watermarked image. On application of quality enhancement scheme of chapter 7, the quality of the watermarked images has been improved significantly. As the watermarking is considered as an optimization problem, it is solved by the Genetic Algorithm (GA) based optimization of chapter 8 following to embedding operation which ensures rapid enhancement in fidelity. In contrast to Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes, both WST_2x2 and WST_1x2 are capable of embedding variable payload with considerable quality degradation.

Chapter 6

Watermarking based on Group of Linear Transformations for Dihedral Group of Order 4 (WGD4)

6.1 Introduction

A group of linear transformations projected on a discrete signal is termed as G-lets in the context of group theory. G-lets are constructed for the dihedral group of order n (i.e., D_n) that can deal with two types of transformations: reflection and rotation. The authentication of color image is carried out based on the fragile watermarking schemes based on group of linear transformations for dihedral group of order 4 (G-lets D4). The dihedral group of order 4 i.e., D_4 is considered as the symmetry group of the square in which the vertices are on the unit circle formed at angles 0, π /2, π, and 3π /2. The carrier image has been decomposed into non-overlapping blocks and then each block is converted into the transform domain by applying forward transform in G-lets D4 domain. The message digest obtained from the watermark, size of the watermark and the content of the watermark are embedded into the transformed components in varying proportion. The pixel components are re-computed from each block of embedded components by applying inverse transform in G-lets D4 domain. The recipient extracts the fabricated message digest, size and content of the watermark. Another message digest is computed from the extracted watermark which in turn is compared against the extracted message digest to verify the authenticity. Performance of the proposed schemes demonstrates the superiority over existing schemes.

6.2 The Technique

In this chapter authenticity of color images has been verified through the fragile watermarking techniques based on group of linear transformations for dihedral group of order 4 (G-lets D4). The carrier image is partitioned into m x n non-overlapping blocks (where, $1 \leq m \leq 2$ and $n = 2$) corresponding to red, green and blue channels. Each m x n sub-image block of the carrier image is converted into transform domain based on the forward transform in G-lets D4 domain. The secret bits corresponding to the message digest (MD), size and the content of the watermark are subsequently fabricated into the transformed components at the least significant part. An inverse transform in G-lets D4 domain is applied on the identical sub-image block to generate the pixel components in spatial domain. The pixel components become non-fractional even after embedding and no overflow or underflow conditions are arises at any stage. Successive block embedding operations produces the watermarked image. At the recipient end, the watermarked image is partitioned into m x n non-overlapping blocks (where $1 \leq m \leq 2$ and $n = 2$) corresponding to red, green and blue channels. Consequently, the fabricated secret bits consisting of message digest (MD), size and content of the watermark are extracted from the transformed components. An inverse transform in G-lets

D4 domain is applied on the identical sub-image block to convert back the embedded components in spatial domain. Each 8 (eight) bits extraction of the watermark, construct one alphabet/one primary (R/G/B) color component. The process is repeated till the watermark is reconstructed.

The transmission of watermarked image through the public medium is highly vulnerable due to the intentional/unintentional attacks such as filtering, blurring and lossy compression etc. The attacks alters the content of the watermarked image in such a way that visually the image is not considerably degraded, however, the frequency distribution is transformed in such a manner that the recipient fails to extract the hidden data. But, in the proposed techniques, the recipient operate the authentication process by matching the extracted message digest MD with the re-computed message digest MD', where MD' can be obtained from the extracted watermark image. If the extracted message digest MD matches with the re-computed message digest MD', then the authentication process is said to be successful, otherwise, it is said to be unsuccessful.

Section 6.2.1 of this chapter deals with 2 x 2 block based watermark fabrication that of section 6.2.2 describes 1 x 2 block based watermark fabrication.

6.2.1 2 x 2 Block based Watermark Fabrication

A group of linear transformations in group theory is termed as G-lets [142]. The G-lets are constructed based on the dihedral group D_n where, an ‘n’ sided polygon has exactly n rotational symmetries and n reflection symmetries. A sparse diagonal representation matrix can also be constructed from each rotation/reflection transformation. The polygon can be considered as having the vertices on the unit circle with labeling as 0, 1, …, n – 1 starting at (1, 0) and proceeding anti-clockwise at an angles of 360/n degrees (i.e., 2π/n radians) or its multiples. The rotations R_0, R_1,…., R_{n-1} and reflections S_0, S_1,…, S_{n-1} are obtained for the line that passes through the origin and makes an angle of 2πk/n with the horizontal axis. Since the elements of D_n are considered as a set of linear transformations for a given plane, the elements of D_n i.e., the rotation and reflection matrices can be represented in 2 x 2 matrix form as given in equation (6.1).

$$R_k = \begin{pmatrix} \cos\left(\frac{2\pi k}{n}\right) & -\sin\left(\frac{2\pi k}{n}\right) \\ \sin\left(\frac{2\pi k}{n}\right) & \cos\left(\frac{2\pi k}{n}\right) \end{pmatrix} \tag{6.1}$$

$$S_k = \begin{pmatrix} \cos\left(\frac{2\pi k}{n}\right) & \sin\left(\frac{2\pi k}{n}\right) \\ \sin\left(\frac{2\pi k}{n}\right) & -\cos\left(\frac{2\pi k}{n}\right) \end{pmatrix} \tag{6.2}$$

The dihedral group of order 4 i.e., D4 is considered as the symmetry group of the square in which the vertices are on the unit circle formed at angles 0, π /2, π, and 3π /2. The matrix representation is given by,

$$R_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad R_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad R_3 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$S_0 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad S_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad S_3 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

If G_r represents the vector of r^{th} G-let in G-lets D4 domain and P represents the vector of pixel components then the forward transform in G-lets D4 domain yields the vector of transformed components (denoted as T) as given in equations (6.3) and (6.4):

$$T = G_r \times P \tag{6.3}$$

where,

$$G_r = \begin{cases} R_{r-1} : 1 \leq r \leq 4 \\ S_{r-5} : 5 \leq r \leq 8 \end{cases} \tag{6.4}$$

where, r is the positive integer, for all r, 1 ≤ r ≤ 8.

Hence, the forward transform of G-lets D4 domain is applied on each 2 x 2 sub-matrix of pixel components (i.e., $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$) to obtain the equivalent 2 x 2 sub-matrix of transformed components (i.e., $t_{i,j}$, $t_{i,j+1}$, $t_{i+1,j}$ *and* $t_{i+1,j+1}$). Forward transform of G-lets D4 domain also yields the equivalent linear equations as given in equation (6.5).

$$\begin{bmatrix} t_{i,j} & t_{i+1,j} \\ t_{i,j+1} & t_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} g_{i,j} & g_{i+1,j} \\ g_{i,j+1} & g_{i+1,j+1} \end{bmatrix} \begin{bmatrix} p_{i,j} & p_{i+1,j} \\ p_{i,j+1} & p_{i+1,j+1} \end{bmatrix}$$

and,

$$\begin{aligned} t_{i,j} &= g_{i,j}p_{i,j} + g_{i,j+1}p_{i+1,j} \\ t_{i,j+1} &= g_{i,j}p_{i,j+1} + g_{i,j+1}p_{i+1,j+1} \\ t_{i+1,j} &= g_{i+1,j}p_{i,j} + g_{i+1,j+1}p_{i+1,j} \\ t_{i+1,j+1} &= g_{i+1,j}p_{i,j+1} + g_{i+1,j+1}p_{i+1,j+1} \end{aligned} \tag{6.5}$$

where, $g_{m,n}$ = 0, 1 or -1 for the r^{th} G-let (i.e., G_r) based on the value of r, for all r, 1 ≤ r ≤ 8 and 0 ≤ m, n ≤ 1.

Similarly, the inverse transform in G-lets D4 domain is applied on each 2 x 2 sub-matrix of transformed components (i.e., $t_{i,j}$, $t_{i,j+1}$, $t_{i+1,j}$ *and* $t_{i+1,j+1}$) to re-generate the equivalent 2 x 2 sub-matrix of non-negative pixel components (i.e., $|p_{i,j}|$, $|p_{i,j+1}|$, $|p_{i+1,j}|$ *and* $|p_{i+1,j+1}|$). The equivalent linear equations of inverse transform in G-lets D4 domain are also given in equation (6.6).

$$\begin{bmatrix} |p_{i,j}| & |p_{i+1,j}| \\ |p_{i,j+1}| & |p_{i+1,j+1}| \end{bmatrix} = \begin{bmatrix} g_{i,j} & g_{i+1,j} \\ g_{i,j+1} & g_{i+1,j+1} \end{bmatrix} \begin{bmatrix} t_{i,j} & t_{i+1,j} \\ t_{i,j+1} & t_{i+1,j+1} \end{bmatrix}$$

and,

$$\begin{aligned} |p_{i,j}| &= g_{i,j}t_{i,j} + g_{i,j+1}t_{i+1,j} \\ |p_{i,j+1}| &= g_{i,j}t_{i,j+1} + g_{i,j+1}t_{i+1,j+1} \\ |p_{i+1,j}| &= g_{i+1,j}t_{i,j} + g_{i+1,j+1}t_{i+1,j} \\ |p_{i+1,j+1}| &= g_{i+1,j}t_{i,j+1} + g_{i+1,j+1}t_{i+1,j+1} \end{aligned} \qquad (6.6)$$

where, $g_{m,n}$ = 0, 1 or -1 of the rth G-let (i.e., G_r) based on the identical value of r, $1 \leq r \leq 8$ and $0 \leq m, n \leq 1$.

The proposed technique consists of the algorithm for insertion, the algorithm for extraction and the example which are described in section 6.2.1.1, 6.2.1.2 and 6.2.1.3, respectively. Results and discussions have been elaborated in section 6.2.1.4.

6.2.1.1 Insertion

Consider the 24-bit cover image which consists of 2 x 2 non-overlapping blocks. A pseudo-random number (r) is obtained from the hash function and the 64-bit secret key. The parameter r is subject to a circular shift (bit-wise rotation) and thus the pseudo-random number (r) is modified for the succeeding blocks. Based on the pseudo-random number (r), the rth G-let is selected and then the forward transform in G-lets D4 domain of equation (6.6) is applied on each 2 x 2 sub-matrix of pixel components corresponding to the red, green and blue channels to obtain the transformed components of identical block size. Secret bits corresponding to the watermark size, content and the message digest are fabricated on transformed components in varying proportion. The sign of each transformed component constitutes the embedding order i.e., for positive component, secret bits embedding are starting from least significant bit position toward higher order bit position however, the order of embedding is reversed for negative transformed components. Inverse transform in G-lets D4 domain is applied on each 2 x 2 embedded blocks to re-compute the pixel components. Repetitive actions of the embedding process yield the watermarked image in spatial domain.

Algorithm 6.1:

Input: Carrier/cover image (I) and authenticating watermark image (W).

Output: Watermarked image (I′).

Method: Forward transform in G-Lets D4 domain is used to fabricate the watermark (along with a message digest) into the carrier images by converting the image from spatial domain into transform domain. Embedding of watermark bits into transform domain offers variable payload, less distortion and improved security. The detailed steps of embedding are described as follows:

Step 1: Obtain the message digest (MD) from the secret watermark.

Step 2: The authenticating watermark size (in bits) is obtained by embedding the watermark into the three sub-matrices of the U x V color image as given in equation (6.7).

$$W_{size} = B \times \big(3 \times (U \times V)\big) - (MD + L) \qquad (6.7)$$

where, B, MD and L represents the embedding payload in terms of bits per Byte, the message digest obtained from the watermark and the header information corresponding to the size of the watermark respectively. The MD and L are consisting of 128 and 32 bits while the usual values of B are 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively.

Step 3: The cover image (I) is partitioned into 2 x 2 non-overlapping blocks in row major order. Each 2 x 2 block is consisting of four pixel components $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$ of red/green/blue channel where, for all i and j, $0 \leq i, j \leq 1$.

Step 4: A 64-bit bit-sequence is obtained from the secret key which is consisting of eight characters. The pseudo-random number (r) is computed from the hash function H(k) where, k is treated as the decimal equivalent of the 64-bit bit-sequence of the secret key. The parameter r is subject to a circular shift (bit-wise rotation) of r bit positions in clock-wise direction. The hash based pseudo random number (r) is obtained as given in equation (6.8).

$$r = H(k) \xrightarrow{yields} (k\%8) + 1 \qquad (6.8)$$

Step 5: Based on the pseudo-random number (r), the r^{th} G-let is selected and then the forward transform in G-lets D4 domain of equation

(6.6) is applied on each 2 x 2 sub-matrix of pixel components corresponding to the red, green and blue channels to obtain the transformed components of identical block size.

Step 6: $\lambda_1/\lambda_2/\lambda_3/\lambda_4$ bits from the secret bit-stream (corresponding to the message digest (MD), size (L) and the content (W) of the watermark) are fabricated on first/second/third/fourth transformed component for red/green/blue channel. To achieve the payload value of B bits per Byte (bpB), the generalized form of λ bits of secret information fabrication on each transformed component is given in equation (6.9).

$$\lambda = \begin{cases} \lfloor B \rfloor : \lambda = \lambda_1 | \lambda_2 \\ \lceil B \rceil : \lambda = \lambda_3 | \lambda_4 \end{cases} \tag{6.9}$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all (λ_1, λ_2, λ_3, λ_4), $0 \leq \lambda_1$, λ_2, λ_3, $\lambda_4 \leq 3$. If the transformed component is positive then the secret bits are fabricated in usual order i.e., starting from the least significant bit (LSB-0) position toward higher order bit position; otherwise, the fabrication is done in reverse order.

Step 7: The inverse transform of G-lets D4 domain of equation (6.7) is applied on each 2 x 2 sub-matrix of embedded components based on the identical pseudo-random number (r) to re-compute the non-negative pixel components in spatial domain.

Step 8: Repeat steps 3 to 6 for embedding the size, content and the message digest MD respectively. Successive steps yield the watermarked image (I') in spatial domain.

Step 9: Stop.

6.2.1.2 Extraction

At the receiving end, the watermarked image is partitioned into 2 x 2 non-overlapping blocks in a sliding window manner. A pseudo-random number (r) is obtained from the hash function and the 64-bit secret key. The parameter r is subject to a circular shift (bit-wise rotation) and thus the pseudo-random number (r) is modified for the succeeding blocks. Based on the pseudo-random number (r), the rth G-let is selected and then the forward transform in G-lets

D4 domain of equation (6.6) is applied on each 2 x 2 sub-matrix of pixel components (corresponding to the red, green and blue channels) to obtain the transformed components of identical block size. Secret bits corresponding to the watermark size, content and the message digest (MD) are extracted from the transformed components in varying proportion. The sign of each transformed component decides the extraction order i.e., for positive component, secret bits extraction are started from the least significant bit position toward higher order bit position however, the order of extraction is reverse for negative transformed components. Inverse transform in G-lets D4 domain converts each 2 x 2 sub-block of transformed components into the spatial domain. The process is repeated until and unless the watermark size, content and the message digest (MD) respectively are extracted. Message digest MD′ is re-computed from the extracted watermark which in turn is compared with the extracted message digest MD for authentication. The extraction procedure is described in algorithm 6.2.

Algorithm 6.2:

Input: Watermarked image (I′).

Output: The authenticating watermark (W) and the 128 bits message digest (MD).

Method: Forward transform in G-lets D4 domain is used to extract the watermark (along with a message digest) from the watermarked image (I′). Successive extracted bits construct the watermark from which another message digest MD′ is re-computed. The message digest MD′ is compared against the extracted message digest MD for authentication. The steps of extraction are mentioned below:

Step 1: The watermarked image (I') is partitioned into 2 x 2 non-overlapping blocks in row major order. Each 2 x 2 block is consisting of four pixel components $p_{i,j}$, $p_{i,j+1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$ of red/green/blue channel where, for all i and j, $0 \leq i, j \leq 1$.

Step 2: A 64-bit bit-sequence is obtained from the secret key which is consisting of eight characters. The pseudo-random number (r) is computed from the hash function H(k) where, k is treated as the decimal equivalent of the 64-bit bit-sequence of the secret key. The parameter r is subject to a circular shift (bit-wise rotation) of r bit positions in clock-wise direction. The hash based pseudo random number (r) can be obtained as given in equation (6.10).

$$r = H(k) \xrightarrow{yields} (k\%8) + 1 \qquad \textit{(6.10)}$$

Step 3: Based on the pseudo-random number (r), the rth G-let is selected and the forward transform in G-lets D4 domain of equation (6.5) is applied on each 2 x 2 sub-matrix of pixel components corresponding to the red, green and blue channels to obtain the transformed components of identical block size.

Step 4: λ_1 /λ_2 /λ_3 /λ_4 bits of the fabricated secret bit-stream are successively extracted from the first/second/third/fourth transformed component of red/green/blue channel for the payload of B bits per Byte (bpB). The generalized form of λ bits of secret data extraction from each transformed component is given in equation (6.11).

$$\lambda = \begin{cases} \lfloor B \rfloor : \lambda = \lambda_1 | \lambda_2 \\ \lceil B \rceil : \lambda = \lambda_3 | \lambda_4 \end{cases} \tag{6.11}$$

where, $0 < B \leq 3$, the difference between two successive payload values (ΔB) is 0.5, and for all (λ_1, λ_2, λ_3, λ_4), $0 \leq \lambda_1, \lambda_2, \lambda_3, \lambda_4 \leq 3$. If the transformed component is positive then the secret bits are extracted in usual order i.e., starting from the least significant bit (LSB-0) toward higher order bit position; otherwise, the extraction is done in reverse order.

Step 5: For each 8 (eight) bits extraction, construct one alphabet/one primary (R/G/B) component.

Step 6: The inverse transformation of G-lets D4 domain as given in equation (6.7) is applied on each 2 x 2 sub-matrix of embedded components to convert it back into spatial domain.

Step 7: Repeat steps 1 to 7 to complete the extraction of the message digest (MD), size and content of the authenticating watermark from the watermarked image.

Step 8: Obtain 128 bits message digest MD' from the extracted watermark.

Step 9: Compare message digest MD' with the extracted message digest MD. If both are alike then the image is considered as authorized, else unauthorized.

Step 10: Stop.

6.2.1.3 Example

The carrier image is partitioned into 2 x 2 non-overlapping blocks in row major order. Three 2 x 2 sub-matrices namely R$_1$, G$_1$ and B$_1$ corresponding to red, green and blue channels have been considered for watermark fabrication. The 2 x 2 sub-matrices are as given below:

$$R_1 = \begin{bmatrix} 230 & 72 \\ 17 & 155 \end{bmatrix} G_1 = \begin{bmatrix} 62 & 215 \\ 56 & 22 \end{bmatrix} B_1 = \begin{bmatrix} 111 & 172 \\ 251 & 7 \end{bmatrix}$$

Consider the secret key (K) = “skghosal” from which the pseudo random number r = (7809650151167322995 % 8) +1 = 4 is obtained. Since, r = 4, G$_4$ i.e., rotation matrix R$_3$ is multiplied with the 2 x 2 sub-matrix of pixel components by applying forward transform in G-lets D4 domain. Thus the 2 x 2 sub-matrices of transformed components for R/G/B channel are expressed as given below:

$$T(R_1) = \begin{bmatrix} 17 & 155 \\ -230 & -72 \end{bmatrix} T(G_1) = \begin{bmatrix} 56 & 22 \\ -62 & -215 \end{bmatrix} T(B_1) = \begin{bmatrix} 251 & 7 \\ -111 & -172 \end{bmatrix}$$

Suppose, the watermark bit stream *“01100110011001001100111101101100011O*” is to be fabricated into the transformed components for the payload value of 3 bits per Byte. In this example, three bits are fabricated (λ_1 = 3, λ_2 = 3, λ_3 = 3, λ_4 = 3) on each transformed component. The order of embedding may vary for each component based on its sign. Hence, the 2 x 2 sub-matrices of embedded components are obtained as follows:

$$T'(R_1) = \begin{bmatrix} 22 & 156 \\ -228 & -78 \end{bmatrix} T'(G_1) = \begin{bmatrix} 58 & 22 \\ -57 & -215 \end{bmatrix} T'(B_1) = \begin{bmatrix} 254 & 6 \\ -104 & -174 \end{bmatrix}$$

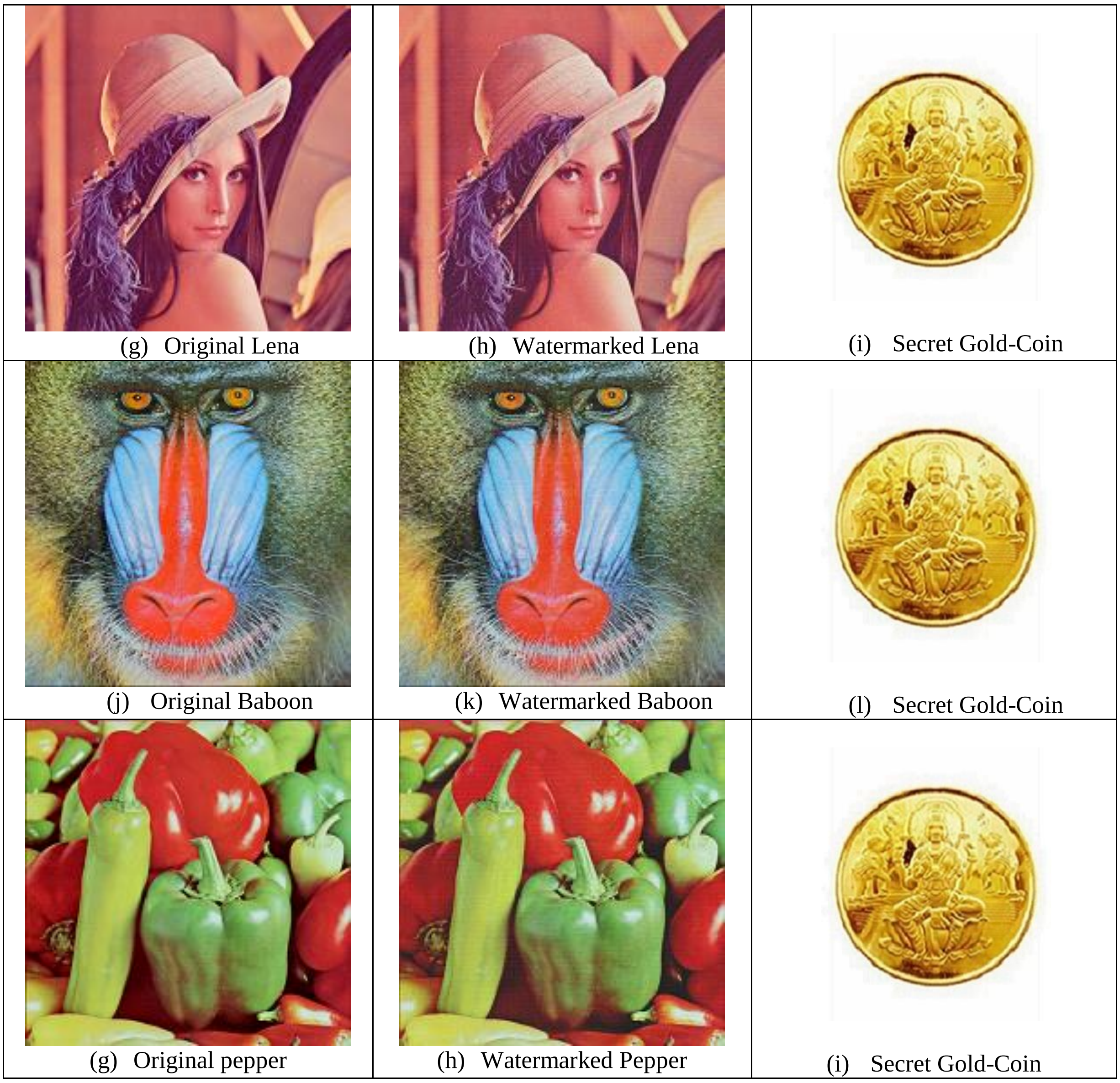
Inverse transform in G-lets D4 domain is applied on 2 x2 sub-matrices of embedded components corresponding to red, green and blue channels. The 2 x 2 sub-matrices of pixel components in spatial domain are as here under:

$$R'_1 = \begin{bmatrix} 228 & 78 \\ 22 & 156 \end{bmatrix} G'_1 = \begin{bmatrix} 57 & 215 \\ 58 & 22 \end{bmatrix} B'_1 = \begin{bmatrix} 104 & 174 \\ 254 & 6 \end{bmatrix}$$

Modified pixel components are non-fractional, non-negative and less than or equal to 255. Hence, there is no possibility of overflow as well as underflow.

6.2.1.4 Results and Discussions

Results and discussions of the 2 x 2 block based watermark fabrication in G-lets D4 domain (WGD4_2x2) has been computed in this section. Performance of WGD4_2x2 scheme is evaluated based on the standard quality metrics such as the peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD) and standard deviation error (SDE) respectively. Twenty benchmark (BMP) images [130, 131] of dimension 512 x 512 and the varying sizes of the secret watermark of fig. 1.1 are taken to compute results. Proposed WGD4_2x2 is compared against the Stirling Transform based watermarking (WST_2x2), Binomial Transform based watermarking (WBT_2x2), Legendre Transform based watermarking (WLT_2x2), Separable Discrete Hartley Transform based watermarking (WDHT_2x2) and Varsaki et al.'s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] respectively. The extracted "Gold Coin", the pre-embedding and post-embedding states of "Lena", "Baboon" and "Pepper" images are depicted in Fig 6.1.



(g) Original Lena

(h) Watermarked Lena

(i) Secret Gold-Coin

(j) Original Baboon

(k) Watermarked Baboon

(l) Secret Gold-Coin

(g) Original pepper

(h) Watermarked Pepper

(i) Secret Gold-Coin

Fig 6.1. Cover, watermarked and authenticating watermark image in the proposed WGD4_2x2 technique

Table 6.1 shows that the performance of WGD4_2x2 in terms of peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) with respect to 0.5, 1.0, 1.5, 2.0, 2.5 and 3 bits per Byte (bpB) of payloads. The PSNR variation for each watermarked image is non-uniform in nature as the pixel distribution is different. Proposed WGD4_2x2 offered the minimum PSNR of 33.27 dB at 3 bpB of payload for the "Athens" image which is considered as the perceptible quality [148]. Similarly, the maximum PSNR of 54.19 dB is achieved at 0.5 bpB of payload for the "Lena" image. The minimum MSE is 0.24 for "Lena" at 0.5 bpB of payload and the maximum MSE is 30.61 for "Athens" at 3 bpB of payload. The IF, SSIM and UIQ values are maximum at 3 bpB and that of values are minimum at 0.5 bpB; the ranges of values are [0.999992 (Airplane), 0.99597 (Bobcat)], [0.999911 (San Diego), 0.953761 (Splash)] and [0.999334 (San Diego), 0.674813 (Splash)] respectively. It is seen that IF, SSIM and UIQ are close to one and specifies high similarity between the watermarked and original images. Average values of these metrics are also computed to summarize the results.

Table 6.1. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WGD4_2x2 technique

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Lena	0.5	54.196500	0.247417	0.999984	0.999686	0.994808
	1.0	51.180434	0.495492	0.999968	0.999177	0.990166
	1.5	45.874016	1.681429	0.999893	0.997242	0.968557
	2.0	43.586399	2.847347	0.999820	0.995217	0.952793
	2.5	39.195771	7.825340	0.999506	0.990560	0.899367
	3.0	36.987859	13.010513	0.999178	0.977376	0.852527
Baboon	0.5	54.156000	0.249735	0.999986	0.999886	0.999022
	1.0	51.150645	0.498902	0.999973	0.999666	0.998044
	1.5	45.866279	1.684427	0.999910	0.998893	0.993536
	2.0	43.578024	2.852842	0.999849	0.998101	0.989794
	2.5	39.182386	7.849494	0.999584	0.996401	0.977350
	3.0	36.973256	13.054334	0.999309	0.991508	0.964454
Pepper	0.5	54.034890	0.256797	0.999978	0.999440	0.990092
	1.0	51.056199	0.509871	0.999957	0.998449	0.985853
	1.5	45.710007	1.746142	0.999853	0.995253	0.968625
	2.0	43.429146	2.952335	0.999753	0.991771	0.955604
	2.5	38.974094	8.235137	0.999306	0.983147	0.910624
	3.0	36.721173	13.834484	0.998831	0.966136	0.869308

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Airplane	0.5	54.120917	0.251761	0.999992	0.999626	0.983152
	1.0	51.115621	0.502942	0.999985	0.999045	0.969501
	1.5	45.821825	1.701758	0.999951	0.996784	0.917705
	2.0	43.541479	2.876950	0.999917	0.994126	0.884326
	2.5	39.166327	7.878574	0.999774	0.988852	0.797686
	3.0	36.936750	13.164527	0.999623	0.972097	0.729255
Sailboat	0.5	54.114845	0.252113	0.999987	0.999740	0.995608
	1.0	51.125217	0.501832	0.999974	0.999321	0.992255
	1.5	45.820563	1.702252	0.999914	0.997739	0.974804
	2.0	43.530657	2.884128	0.999854	0.996071	0.962674
	2.5	39.171397	7.869382	0.999602	0.992346	0.923803
	3.0	36.958311	13.099334	0.999338	0.981759	0.888247
Earth	0.5	54.194826	0.247512	0.999985	0.999764	0.997682
	1.0	51.165703	0.497175	0.999970	0.999380	0.995612
	1.5	45.878919	1.679532	0.999900	0.997978	0.985270
	2.0	43.600432	2.838161	0.999831	0.996586	0.977454
	2.5	39.204837	7.809020	0.999536	0.992909	0.945958
	3.0	37.010580	12.942621	0.999232	0.983046	0.915674
San Diego	0.5	54.188650	0.247865	0.999990	0.999911	0.999334
	1.0	51.168158	0.496894	0.999981	0.999770	0.998738
	1.5	45.870933	1.682623	0.999937	0.999229	0.995846
	2.0	43.589835	2.845095	0.999893	0.998659	0.993702
	2.5	39.193157	7.830051	0.999708	0.997264	0.985349
	3.0	36.991276	13.000281	0.999516	0.993299	0.976714
Splash	0.5	54.058783	0.255388	0.999977	0.999176	0.966074
	1.0	51.077490	0.507377	0.999955	0.997618	0.949486
	1.5	45.761562	1.725536	0.999848	0.992738	0.890165
	2.0	43.495327	2.907686	0.999745	0.987939	0.851843
	2.5	39.035420	8.119668	0.999280	0.977124	0.753681
	3.0	36.828726	13.496081	0.998805	0.953761	0.674813
Oakland	0.5	54.129326	0.251274	0.999986	0.999804	0.998596
	1.0	51.128024	0.501508	0.999973	0.999479	0.997538
	1.5	45.827287	1.699619	0.999908	0.998209	0.992294
	2.0	43.545439	2.874328	0.999844	0.996891	0.988177
	2.5	39.147892	7.912087	0.999574	0.994028	0.971982
	3.0	36.926973	13.194197	0.999290	0.985574	0.954429

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	54.186913	0.247964	0.999991	0.999578	0.993367
	1.0	51.160064	0.497821	0.999982	0.998923	0.988152
	1.5	45.885079	1.677151	0.999940	0.996309	0.960189
	2.0	43.600782	2.837932	0.999898	0.993372	0.939759
	2.5	39.249766	7.728650	0.999724	0.987114	0.868517
	3.0	37.040848	12.852733	0.999541	0.967732	0.804575
Anhinga	0.5	52.448813	0.369997	0.999971	0.999895	0.882825
	1.0	49.741529	0.690124	0.999947	0.999654	0.885574
	1.5	43.671849	2.791871	0.999785	0.998739	0.866645
	2.0	41.304457	4.815424	0.999630	0.997548	0.871204
	2.5	36.548115	14.396895	0.998895	0.995547	0.830565
	3.0	34.226687	24.570367	0.998114	0.985924	0.827083
Athens	0.5	51.855675	0.424143	0.999966	0.999907	0.971388
	1.0	49.354203	0.754501	0.999939	0.999683	0.974436
	1.5	43.159558	3.141409	0.999749	0.998836	0.952678
	2.0	40.704024	5.529398	0.99956	0.997752	0.958569
	2.5	35.722293	17.412095	0.998608	0.996008	0.903621
	3.0	33.271913	30.611810	0.997554	0.986211	0.902312
Bardowl	0.5	52.469129	0.368270	0.999963	0.999885	0.999054
	1.0	49.895188	0.666133	0.999933	0.999614	0.998920
	1.5	43.874298	2.664713	0.999734	0.998701	0.994909
	2.0	41.426090	4.682430	0.999533	0.997460	0.993406
	2.5	36.519543	14.491923	0.998558	0.994528	0.979349
	3.0	34.067087	25.490105	0.997463	0.987761	0.971972
Barnfall	0.5	53.432516	0.295004	0.999955	0.999812	0.998833
	1.0	50.605045	0.565686	0.999913	0.999444	0.998217
	1.5	45.742370	1.733178	0.999730	0.998452	0.993474
	2.0	43.589555	2.845278	0.999559	0.997177	0.991764
	2.5	38.809760	8.552720	0.998647	0.994487	0.973846
	3.0	36.485555	14.605781	0.997681	0.985441	0.964612
Butrfly	0.5	52.335775	0.379754	0.999972	0.999887	0.995565
	1.0	49.745587	0.689479	0.999950	0.999630	0.995474
	1.5	43.652411	2.804395	0.999801	0.998714	0.986614
	2.0	41.206259	4.925546	0.999651	0.997653	0.985142
	2.5	36.279247	15.316363	0.998917	0.995645	0.957801
	3.0	33.856677	26.755470	0.998108	0.988886	0.953222

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	52.050556	0.405531	0.999942	0.999737	0.749630
	1.0	49.505535	0.728663	0.999897	0.999076	0.750896
	1.5	43.364621	2.996527	0.999578	0.996760	0.737987
	2.0	40.917814	5.263793	0.999259	0.993751	0.738413
	2.5	35.986241	16.38536	0.997695	0.987138	0.710499
	3.0	33.562039	28.633627	0.995970	0.970313	0.704485
Bodie	0.5	54.002689	0.258708	0.999957	0.999882	0.980279
	1.0	51.081650	0.506891	0.999916	0.999619	0.979328
	1.5	45.675361	1.760128	0.999708	0.998846	0.973565
	2.0	43.377612	2.987576	0.999506	0.997784	0.971706
	2.5	39.262679	7.705705	0.998698	0.995518	0.956686
	3.0	37.079435	12.739044	0.997834	0.988510	0.949351
Bluheron	0.5	53.637012	0.281435	0.999965	0.999752	0.996472
	1.0	50.769326	0.544687	0.999933	0.999141	0.995349
	1.5	45.994941	1.635257	0.999800	0.997741	0.986629
	2.0	43.825167	2.695030	0.999671	0.995812	0.982368
	2.5	39.274936	7.683989	0.999058	0.995296	0.956451
	3.0	36.895575	13.289932	0.998371	0.986803	0.949305
Colomtn	0.5	53.872445	0.266585	0.999979	0.999826	0.985477
	1.0	50.917212	0.526452	0.999958	0.999502	0.985171
	1.5	45.151517	1.985768	0.999844	0.998344	0.977006
	2.0	42.788451	3.421647	0.999732	0.996997	0.975418
	2.5	38.245540	9.739267	0.999239	0.994296	0.952039
	3.0	35.928869	16.603260	0.998704	0.985666	0.943658
Desert	0.5	53.587786	0.284643	0.999959	0.999841	0.998602
	1.0	50.670424	0.557234	0.999920	0.999565	0.998393
	1.5	45.247898	1.942184	0.999719	0.998616	0.995443
	2.0	42.904125	3.331714	0.999516	0.997139	0.994032
	2.5	37.719466	10.993443	0.998417	0.993755	0.981883
	3.0	35.222601	19.535307	0.997177	0.985746	0.977339
Average Case	0.5	53.553700	0.292095	0.999974	0.999752	0.973793
	1.0	50.680660	0.561983	0.999951	0.999288	0.971355
	1.5	45.192560	2.021795	0.999825	0.997706	0.955597
	2.0	42.877050	3.460732	0.999701	0.995890	0.947907
	2.5	38.294440	10.086760	0.999116	0.992098	0.911853
	3.0	35.998610	17.224190	0.998482	0.981177	0.888667

Analysis of peak signal to noise ratio (dB) for WGD4_2x2 is made at 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively. The WGD4_2x2 is compared against the Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) in terms of PSNR (dB) as given in fig. 6.2. Comparison has been made over "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively. The PSNR of the watermarked images of both DPTHDI [88] and DGTDHS [129] are computed with respect to 0.25 bpB and 1 bpB of fixed payloads. The offered payload is significantly low as far as the proposed 2 x 2 block based scheme is concerned. However, the WGD4_2x2 is focused on variable payload that offers a spread from 0.5 to 3 bpB of payload with acceptable visual imperceptibility. In contrast to DPTHDI [88], the WGD4_2x2 provides equal or higher PSNR (dB) at 0.5,1, 1.5, 2 and 2.5 bpB of payloads for "Lena", 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Baboon", 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for "Pepper", 0.5, 1, 1.5 and 2 bpB of payloads for "Airplane" and 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for "Sailboat" respectively. Compared to DGTDHS [129], the WGD4_2x2 provides higher PSNR (dB) at 0.5 and 1 bpB of payloads for "Lena", 0.5 and 1 bpB of payloads for "Baboon", 0.5 and 1 bpB of payloads for "Pepper", 0.5 and 1 bpB of payloads for "Airplane" and 0.5 and 1 bpB of payloads for "Sailboat" respectively. All five images offered acceptable PSNR values of more than 30 dB [148].

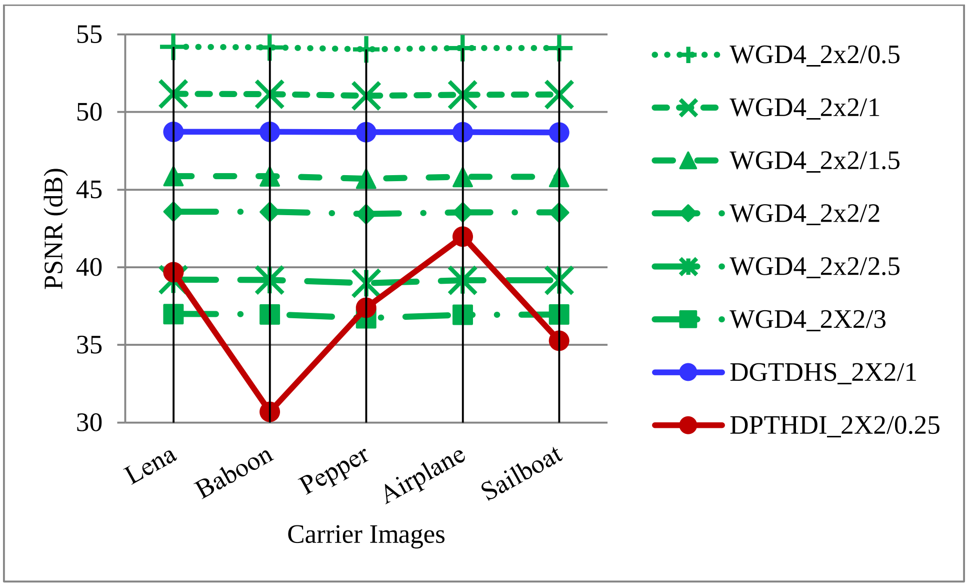


Fig. 6.2. Performance analysis of PSNR (dB) for variable payload based WGD4_2x2 and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five benchmark images [130, 131]

Fig. 6.3 illustrates an average PSNR analysis of WGD4_2x2 against WST_2x2, WBT_2x2, WLT_2x2, WDHT_2x2, DPTHDI [88] and DGTDHS [129] respectively. In WGD4_2x2, the highest average PSNR of 53.55 dB is obtained at 0.5 bpB and that of lowest value obtained is 35.99 dB at 3 bpB respectively. Compared to WST_2x2, WBT_2x2, WLT_2x2 and WDHT_2x2, massive improvement of average PSNR values (in actual sense, quality) is observed in WGD4_2x2 for payload range [0.5 – 3 bpB]. The average PSNR obtained for DPTHDI [88] is 37.40 dB as computed from “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” at 0.25 bpB of payload. In contrast to DPTHDI [88], the WGD4_2x2 ensured equal or higher average PSNR for the payload range [0.5 – 2.5 bpB]. The average PSNR for DGTDHS [129] is 48.70 dB as computed from the “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” at 1 bpB of payload. Compared to DGTDHS [129], the WGD4_2x2 ensured equal or higher average PSNR at 0.5 and 1 bpB respectively. The WGD4_2x2 provides average PSNR of 35.99 dB at 3 bpB which is higher than the acceptable PSNR value of 30 dB.

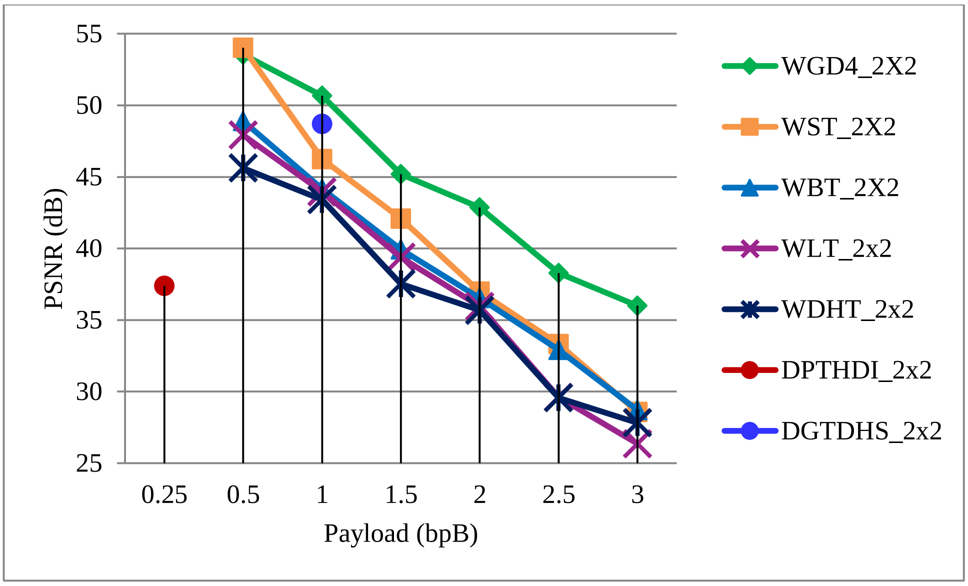


Fig. 6.3. Graphical representation of variation of average PSNR (dB) with respect to payload for WGD4_2x2,WST_2x2, WBT_2x2, WLT_2x2, WDHT_2x2 and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

In fig. 6.4, the standard deviation (SD) analysis has been carried out for five color images such as “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” respectively. The computation of standard deviation (SD) is done by taking the average of standard deviation values corresponding to red, green and blue channels. Fig. 6.4 also illustrates that the variation of standard deviation (SD) up to 3 bpB is almost identical to the standard deviation (SD) of the original image prior to embedding (i.e., at 0 bpB of payload). However, the changes made in the standard deviation (SD) is minimum on embedding i.e., the WGD4_2x2 scheme embeds data in such a way that the difficulty occurs to detect the difference between U x V source and watermarked images.

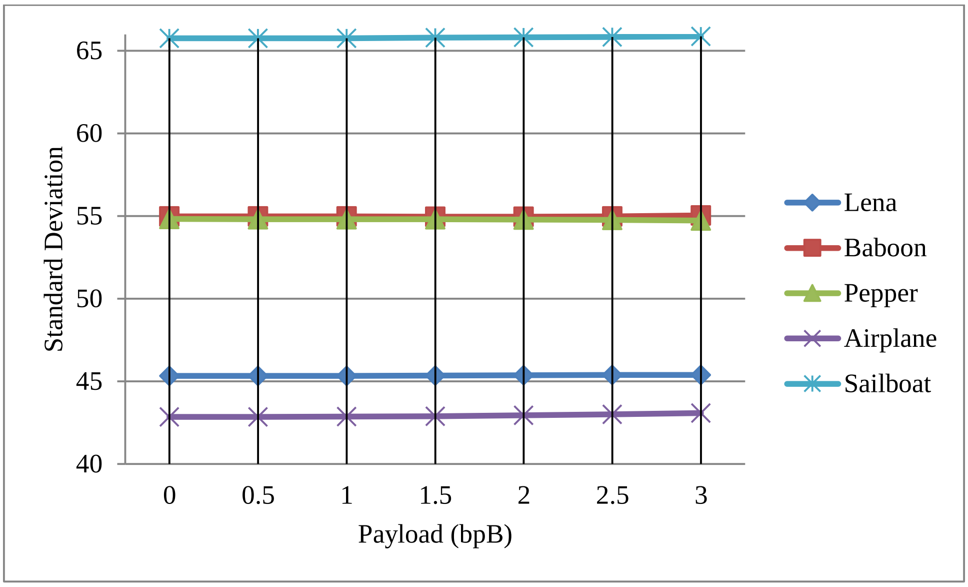


Fig. 6.4. Graphical representation of standard deviation (SD) for WGD4_2x2 with respect to 0, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

It is seen from fig 6.4 that all the lines representing the standard deviation (SD) with respect to increasing payload (i.e., 0 - 3 bpB) are forming a straight line. To analyze the dispersion more clearly, a standard deviation error (SDE) analysis has been made. Fig. 6.5 illustrates the dispersion of error in the watermarked images referencing to the original image as 0.2 in worst case however, errors are almost equal to zero in most of the cases.

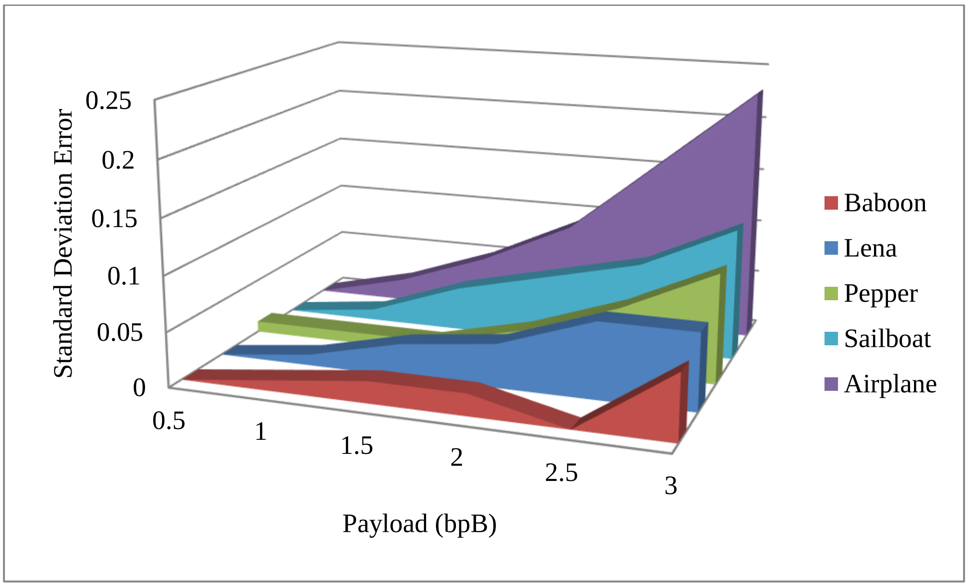


Fig. 6.5. Graphical representation of standard deviation error (SDE) for WGD4_2x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

6.2.2 1 x 2 Block based Watermark Fabrication

As discussed in section 6.2.1, the G-lets are constructed based on the dihedral group D_n where, the elements of D_n are treated as a set of linear transformations for a given plane, and the elements of D_n (i.e., rotation and reflection matrices) are represented in 2 x 2 forms as follows:

$$R_k = \begin{pmatrix} \cos\left(\frac{2\pi k}{n}\right) & -\sin\left(\frac{2\pi k}{n}\right) \\ \sin\left(\frac{2\pi k}{n}\right) & \cos\left(\frac{2\pi k}{n}\right) \end{pmatrix} \qquad (6.12)$$

$$S_k = \begin{pmatrix} \cos\left(\frac{2\pi k}{n}\right) & \sin\left(\frac{2\pi k}{n}\right) \\ \sin\left(\frac{2\pi k}{n}\right) & -\cos\left(\frac{2\pi k}{n}\right) \end{pmatrix} \qquad (6.13)$$

The dihedral group of order 4 (D_4) is the symmetry group of the square. The vertices are on the unit circle formed at angles 0, π /2, π, and 3π /2. The matrix representation is given by,

$$R_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad R_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad R_3 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$S_0 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad S_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad S_3 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

If G_r represents the vector of r^{th} G-let in G-lets D4 domain and P represents the vector of pixel components then the forward transform in G-lets D4 domain yields the vector of transformed components (denoted as T) as follows:

$$T = P \times G_r \qquad (6.14)$$

where,

$$G_r = \begin{cases} R_{r-1} : 1 \le r \le 4 \\ S_{r-5} : 5 \le r \le 8 \end{cases} \qquad (6.15)$$

where, r is the positive integer, for all r, $1 \le r \le 8$.

Unlike to 2 x 2 block based formulation of section 6.2.1, the forward transform of G-lets D4 domain is applied on each pair / 1 x 2 sub-matrix of pixel components (i.e., p_i and p_{i+1}) to obtain the equivalent 1 x 2 sub-matrix of transformed components (i.e., t_i and t_{i+1}). Forward transform of G-lets D4 domain also yields the equivalent linear equations as given in equation (6.16).

$$[t_i \quad t_{i+1}] = [p_i \quad p_{i+1}] \begin{bmatrix} g_{i,j} & g_{i+1,j} \\ g_{i,j+1} & g_{i+1,j+1} \end{bmatrix}$$

and,

$$\begin{aligned} t_{i,j} &= p_{i,}g_{i,j} + p_{i+1}g_{i,j+1} \\ t_{i+1} &= p_i g_{i+1,j} + p_{i+1}g_{i+1,j+1} \end{aligned} \qquad (6.5)$$

where, $g_{m,n}$ = 0, 1 or -1 of the r^{th} G-let (i.e., G_r) based on the value of r, for all r, $1 \leq r \leq 8$ and $0 \leq m, n \leq 1$.

Similarly, the inverse transform of G-lets D4 domain is applied on each pair / 1 x 2 sub-matrix of transformed components (i.e., t_i and t_{i+1}) to re-generate the equivalent 1 x 2 sub-matrix of non-negative pixel components (i.e., p_i and p_{i+1}). The equivalent linear equations are expressed as given in equation (6.17).

$$[|p_i| \quad |p_{i+1}|] = [t_i \quad t_{i+1}] \begin{bmatrix} g_{i,j} & g_{i+1,j} \\ g_{i,j+1} & g_{i+1,j+1} \end{bmatrix}$$

and,

$$\begin{aligned} |p_{i,j}| &= t_{i,}g_{i,j} + t_{i+1}g_{i,j+1} \\ |p_{i+1}| &= t_i g_{i+1,j} + t_{i+1}g_{i+1,j+1} \end{aligned} \qquad (6.17)$$

where, $g_{m,n}$ = 0, 1 or -1 of the r^{th} G-let (i.e., G_r) based on the identical value of r, $1 \leq r \leq 8$ and $0 \leq m, n \leq 1$.

The proposed technique has been elaborately described in the following sections. The algorithm for insertion, the algorithm for extraction and an example are described in detail in section 6.2.2.1, 6.2.2.2 and 6.2.2.3 respectively. Results and discussions have been elaborated in section 6.2.2.4.

6.2.2.1 Insertion

This section partitions the cover image into 1 x 2 sub-matrix or pair of pixel components in row major order. It also computes the pseudo-random number (r) corresponding to each sub-matrix based on the hash function and the 64-bit secret key as discussed in section 6.2.1.1. Based on the value of the pseudo-random number (r), the r^{th} G-let is selected. Each pair of pixel components are converted into transform domain by applying forward transform in G-lets D4 domain. The bit-stream corresponding to each pair of transformed components is decomposed into two segments: Vector Co-ordinate Area (VCA) and Vector Modification Area (VMA). The value of P_{VMA} is determined to generate 'n' number of vector co-ordinates. A reference value (f) is computed to obtain the summation value (S_V). Insertion of secret bits (corresponding to the message digest, size and the content of the watermark) is achieved by replacing the VMA portion with the summation value; however, the sign of each transformed pair decides the order of embedding. Consequently, the signs are reassigned on the transformed components. On applying inverse transform in G-lets D4 domain, the pairis applied on each embedded components to re-compute the pair of pixel components with non-negative values. Successive embedding operations generate the watermarked image.

Algorithm 6.3:

Input: *C*over image (I), an authenticating watermark (W) and a secret key (K) of 64 bits.

Output: Watermarked image (I').

Method: The carrier image (I) is converted into transform domain based on group of linear transformations for dihedral group of order 4 (G-lets D4). Watermark bits are fabricated into each pair of transformed components based on the proposed 1 x 2 block based watermarking scheme in G-lets D4 domain. The detailed steps of embedding are described as follows:

Step 1. O*btain 128 bits message digest MD from the authenticating watermark image.*

Step 2. The authenticating watermark size (in bits) is obtained by embedding the watermark into the three sub-matrices of the U x V color image as given in equation (6.18).

$$W_{size} = B \times \left(3 \times (U \times V)\right) - (MD + L) \qquad (6.18)$$

where, B, MD and L represents the embedding payload in terms of bits per Byte, the message digest obtained from the watermark and the header information corresponding to the size of the watermark respectively. The MD and L are consisting of 128 and 32 bits while the usual values of B are 0.5, 1, 1.5, 2, 2.5 and 3 bpB respectively.

Step 3. The cover image (I) is partitioned into non-overlapping (Red\Green\Blue) pair of pixel components (p_i, p_{i+1}) in a sliding window manner.

Step 4. A 64-bit bit-sequence is obtained from the secret key which is consisting of eight characters. The pseudo-random number (r) is computed from the hash function H(k) where, k is treated as the decimal equivalent of the 64-bit bit-sequence of the secret key. The parameter r is subject to a circular shift (bit-wise rotation) of r bit *positions in clock-wise direction. The hash based pseudo random number (r) can be obtained as given in equation (6.19).*

$$r = H(k) \xrightarrow{yields} (k\%8) + 1 \tag{6.19}$$

Step 5. Based on the pseudo-random number (r), the forward transform in G-lets D4 domain of equation (6.16) is applied on each 1 x 2 sub-matrix of pixel components corresponding to the red, green and blue channels to obtain the transformed components of identical block size.

Step 6. λ bits from the secret bit-stream S (corresponding to the message digest (MD), size (L) and the content (W) of the watermark) are fabricated on each pair of transformed components (t_i, t_{i+1}) based on the following embedding rules as discussed below:

a) Check the sign s_i and s_{i+1} for the pair of transformed components (t_i, t_{i+1}).

$$d = \begin{cases} (b_0b_1b_2 \ldots b_{\lambda-1})_2 : s_i = s_{i+1} \\ (b_{\lambda-1}b_{\lambda-2}b_{\lambda-3} \ldots b_0)_2 : s_i \neq s_{i+1} \end{cases} \tag{6.20}$$

where, for all λ, 1 ≤ λ ≤ 6 while the secret data (d) is obtained by retrieving λ number of bits from the secret bit-stream (S) corresponding to the message digest MD, size (L) and content of the watermark (W) as given in equation (6.20).

b) Consider the transformed components as positive integers during embedding i.e., (t_i, t_{i+1}) = ($|t_i|$, $|t_{i+1}|$).

c) Assign P_{VMA} = λ for VMA and n is obtained from P_{VMA} for generating the vector co-ordinates of VCA as per given rule:

$$n = \begin{cases} P_{VMA} - 1 : \lambda \; is \; even \\ P_{VMA} : \lambda \; is \; odd \end{cases} \tag{6.21}$$

d) Allocate the VCA evenly according to n, and generate n vectors denoted as (g_1, g_2, g_3, ... g_n).

e) Allocate the VMA, consisting of λ bits into two halves i.e., λ_1 and λ_2 where, $\lambda_1 = \left\lfloor \frac{\lambda}{2} \right\rfloor$, $\lambda_2 = \lambda - \lambda_1$.

f) Calculate the reference value f by using equation (6.22) with respect to vectors (g_1, g_2, g_3, ... g_n) to embed λ secret bits.

$$f(g_1, g_2, \dots, g_n) = \sum_{i=0}^{n} g_i(2^i - 1) mod\ 2^\lambda \quad (6.22)$$

g) *Calculate the summation value,* $S_V = (d + f) \% 2^\lambda$.

h) *Obtain the pair of embedded components* (t'_i, t'_{i+1}) *by replacing the VMA with* S_V *as given below.*

$$t'_i = \left(t_i \ \& \left(2^8 - 2^{\lambda_1}\right)\right) + \left\lfloor \frac{S_V}{2^{\left\lceil \frac{\lambda}{2} \right\rceil}} \right\rfloor$$

$$t'_{i+1} = \left(t_{i+1} \ \& \left(2^8 - 2^{\lambda_2}\right)\right) + (S_V \ \%2^{\left\lceil \frac{\lambda}{2} \right\rceil})$$

Here, '&' is the bitwise-AND operator and all calculations in this context are purely integer based.

i) *The sign* s_i *and* s_{i+1} *of the pair of transformed components* (t_i, t_{i+1}) *are re-assigned to the pair of embedded components* (t'_i, t'_{i+1}).

Step 7. The inverse transform of G-lets D4 domain of equation (6.19) is applied on each 1 x 2 sub-matrix of embedded components based on the identical pseudo-random number (r) to re-compute the non-negative pixel components in spatial domain.

Step 8. Repeat step 3 to step 7 to embed the size, content and the message digest MD respectively. The block embedding operation in a row yields the watermarked image (I').

Step 9. Stop.

6.2.2.2 Extraction

The watermarked image *(I')* is received in spatial domain. During extraction, the watermarked image has been taken as the input and partitioned into pair of pixel components in the sliding window manner. The identical pseudo-random number (r) is computed for each block based on the hash function and the 64-bit secret key as discussed in section 6.2.1.2. Based on the pseudo-random number (r), the rth G-let is selected. Each pair of pixel components are converted into transform domain by applying forward transform in G-lets D4 domain. The bit-stream corresponding to each pair of transformed components is decomposed into two segments: Vector Co-ordinate Area (VCA) and Vector Modification

Area (VMA). The value of P_{VMA} is determined to generate ‘n’ number of vector co-ordinates. A reference value (f) is computed to obtain the fabricated secret bits which are retrieved based on the sign of each transformed pair. Successive extraction operations ensure the retrieval of the message digest MD, size and the content of the watermark. Obtain message digest (MD′) from the extracted watermark which in turn is compared with the extracted message digest (MD) for authentication. Even a single bit alteration leads to the failure of the authentication process.

Algorithm 6.4:

Input: Watermarked image (I′) and the 64 bit secret key.

Output: The watermark image (W) and the message digest (MD).

Method: The watermarked image (I′) is converted from the spatial domain into transform domain based on group of linear transformations for dihedral group of order 4 (G-lets D4). 128 bits message digest (MD), size and content of the watermark are extracted from each pair of transformed components in G-lets D4 domain. Successive extraction operations re-generate the watermark image (W) in spatial domain. Obtain message digest (MD′) from the extracted watermark which in turn is compared with the extracted message digest (MD) for authentication. The steps of extraction are listed below:

Step 1. The watermarked image (I′) partitioned into (Red\Green\Blue) pairs of pixel components (p_i, p_{i+1}) in row major order.

Step 2. A 64-bit bit-sequence is obtained from the secret key which is consisting of eight characters. The pseudo-random number (r) is computed from the hash function H(k) where, k is treated as the decimal equivalent of the 64-bit bit-sequence of the secret key. The parameter r is subject to a circular shift (bit-wise rotation) of r bit positions in clock-wise direction. As a consequence, the pseudo random number (r) is modified iteratively. The hash based pseudo random number (r) can be obtained as given in equation (6.23).

$$r = H(k) \xrightarrow{yields} (k\%8) + 1 \qquad (6.23)$$

Step 3. Based on the pseudo-random number (r), the forward transform in G-lets D4 domain of equation (6.16) is applied on each 1 x 2 sub-matrix of pixel components corresponding to the red, green and blue channels to obtain the transformed components of identical block size.

Step 4. *λ bits of the fabricated secret information where, $1 \leq \lambda \leq 6$ are extracted from each pair of transformed components (t_i, t_{i+1}) based on the following extraction rules as discussed below:*

a) Consider the transformed components as positive integers during extraction i.e., (t_i, t_{i+1}) = ($|t_i|$, $|t_{i+1}|$).

b) Assign $P_{VMA} = \lambda$ for VMA and n is obtained from P_{VMA} for generating the vector co-ordinates of VCA as per given rule:

$$n = \begin{cases} P_{VMA} - 1 : \lambda \, is \, even \\ P_{VMA} : \lambda \, is \, odd \end{cases} \qquad (6.24)$$

(c) Allocate the VCA evenly according to n, and generate n vectors denoted as (g_1, g_2, g_3, ... , g_n).

(d) λ_1 and λ_2 bits are extracted from each pair of transformed components (t_i, t_{i+1}) i.e., from VMA where, $\lambda_1 = \lfloor \lambda / 2 \rfloor$, $\lambda_2 = \lambda - \lambda_1$. Intermediate data (d) is a decimal number obtained by concatenating ('+') two extracted bit sequences $(b'_0b'_1...b'_{\lambda1-1})_2$ and $(b''_0b''_1...b''_{\lambda2-1})_2$ which are obtained from t_i and t_{i+1} as given in equation (6.25).

$$d = (b'_0 b'_1 \ldots b'_{\lambda_1 - 1})_2 + (b''_0 b''_1 \ldots b''_{\lambda_2 - 1})_2 \qquad (6.25)$$

(e) Calculate reference value f by using equation (6.26) with respect to vectors (g_1, g_2, g_3, ... g_n) to extract λ secret bits.

$$f(g_1, g_2, \ldots, g_n) = \sum_{i=0}^{n} g_i(2^i - 1) mod \; 2^{\lambda} \qquad (6.26)$$

(f) Secret data (d') consisting of λ bits of information are recovered from each pair of transformed components (t_i, t_{i+1}) where, $d' = (2^{\lambda} - f + d) \% 2^{\lambda}$.

(g) The signs s_i and s_{i+1} ensured the order of retrieval of the secret data (d'). The signs are re-assigned to the pair of transformed components (t_i, t_{i+1}).

Step 5. *For each eight bits extraction of the watermark bit stream (W), construct one R/G/B color component or a textual alphabet.*

Step 6. *The inverse transform of G-lets D4 domain of equation (6.19) is applied on each pair / 1 x 2 sub-matrix of embedded components based* *on the identical pseudo-random number (r) to re-compute the non-negative pixel components in spatial domain.*

Step 7. Repeat steps 1 to 6 for red/green/blue channel to complete the extraction as per the size of the authenticating watermark.

Step 8. Obtain 128 bits message digest MD' from the extracted authenticating watermark (W).

Step 9. Compare MD' with extracted MD. If both are same then the image is authorized, else unauthorized.

Step 10. Stop.

6.2.2.3 Example

Consider the pair of pixel components (174, 105) for a given cover image that can fabricate the secret data 14. The embedding and extraction processes for an average payload of 2 bpB have been described in the following section.

Embedding: - The following steps are carried out to embed the secret data:

1) Let, the pair of pixel components (p_0, p_1) = (174, 105), secret data (d) = 14 (equivalent to the four bit binary stream 1110), number of bits to be embedded (λ) = 4 and secret key (K) = “skghosal”.

2) Pseudo-random number, r = (7809650151167322995 % 8) +1 = 4.

3) Since, r = 4, G_4 will be computed i.e., rotation matrix R_3 is multiplied with the pair of pixel components based on the forward transform in G-lets D4 domain. Therefore, the pair of transformed components is obtained as T_4 = (-105, 174).

4) Since, sign s_0 is –ve and s_1 is +ve, the bits-stream of the secret data (d) is retrieved in reverse order i.e., d = $(0111)_2$ = 7. Consequently, the transformed components are considered as positive integers i.e., (t_0, t_1) = (|-105|, |174|) = (105,174).

5) Here, P_{VMA} = 4, n = (4 – 1) = 3, (t_0, t_1) = (105, 174) = $(01101001, 10101110)_2$.

6) Three vector are generated as, (g_1, g_2, g_3) = $(0110, 1010, 1011)_2$ = (6, 10, 11).

7) Compute, f (g_1, g_2, g_3) = (6 x 1 + 10 x 3 + 11 x 7) % 2^4 = 113 % 16 = 1 = $(0001)_2$.

8) Calculate the summation value, S_V = (7 + 1) % 2^4 = 8 = $(1000)_2$.

9) Since, VMA = 01 || 10, λ_1 = $\lfloor 4 / 2 \rfloor$ = 2, λ_2 = 4 – 2 = 2, replace the VMA by S_V as follows:

$t'_i = (105 \ \& \ (2^8 – 2^2)) + \lfloor 8/ 2^{\lceil 4/2 \rceil} \rfloor = (105 \ \& \ (256 – 4) + \lfloor 8/4 \rfloor$

$= (105 \ \& \ 252) + 2$

$= 106.$

$t'_{i+1} = (174 \ \& \ (2^8 – 2^2)) + 8 \ \% \ 2^{\lceil 4/2 \rceil} = (174 \ \& \ (256 – 4)) + 8 \ \% \ 4$

$= (174 \ \& \ 252) + 0$

$= 172.$

10) Pair of embedded components $(t'_0, t'_1) = (106, 172) = (01101010, 10101100)_2$.

11) As sign s_0 is –ve and s_1 is +ve, the signs are re-assigned and hence, the embedded components becomes, $(t'_0, t'_1) = (-106, 172)$.

12) Inverse transform in G-lets D4 domain (for r = 4) re-generates the pair of pixel components with non-negative values, $(p'_0, p'_1) = (|172|, |-(-106)|)$
$= (172, 106)$.

Extraction: - The following steps are followed to recover the secret data:

1) Pair of pixel components $(p_0, p_1) = (172, 106)$, no. of bits to be extracted $(\lambda) = 4$ and secret key (K) = “skghosal”.

2) Obtain the pseudo random number r = (7809650151167322995 % 8) +1 = 4.

3) Since, r = 4, G_4 will be the domain of G-lets fro extraction i.e., rotation matrix R_3 is multiplied with the pair of pixel components based on the forward transform in G-lets D4 domain. Therefore, the pair of transformed components is obtained as $T_4 = (-106, 172)$.

4) Since, sign s_0 is –ve and s_1 is +ve, $(t_0, t_1) = (|-106|, |172|) = (106,172)$.

5) Here, $P_{VMA} = 4$, n = (4 – 1) = 3, $(t_0, t_1) = (106, 172) = (01101010, 10101100)_2$ and $d = (1000)_2 = 8$.

6) Generate three vectors $(g_1, g_2, g_3) = (0110, 1010, 1011)_2 = (6, 10, 11)$.

7) Compute, $f = (6 \text{ x } 1 + 10 \text{ x } 3 + 11 \text{ x } 7) \ \% \ 2^4 = 1$.

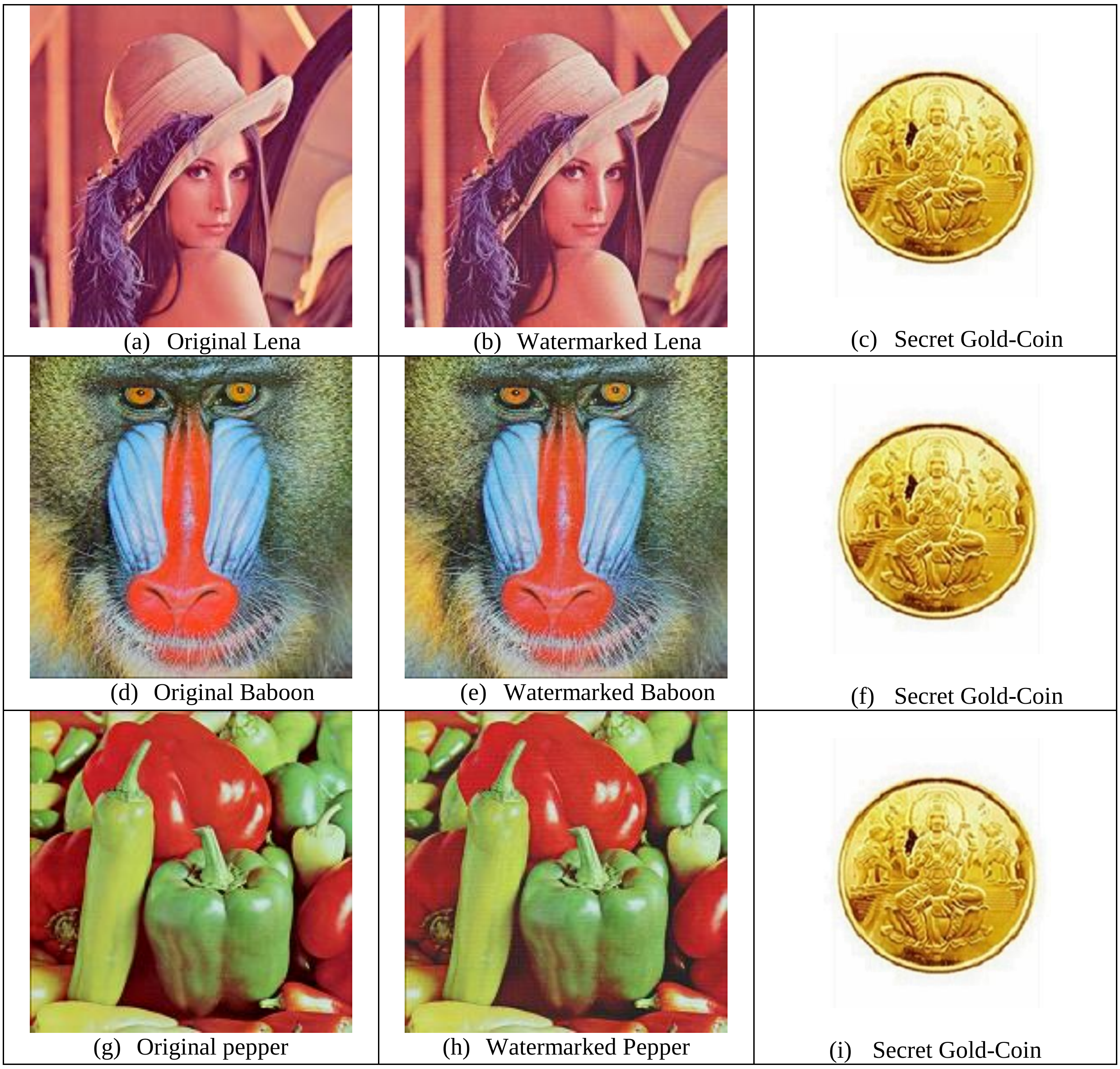
8) The recovered data, $d' = (2^4 – 1 + 8) \ \% \ 2^4 = (16 – 1 + 8) \ \% \ 16 = 7 = (0111)_2$.

9) Since, sign s_0 is –ve and s_1 is +ve, so re-assigning the sign to the embedded components yields, $(t'_0, t'_1) = (-106, 172)$ and secret data is retrieved in reverse order, $d' = (1110)_2 = 14$.

10) Inverse transform in G-lets D4 domain (for r = 4) re-generates the pair of pixel components with non-negative values, $(|p'_0|, |p'_1|) = (|172|, |- (-106)|)$
$= (172, 106)$.

6.2.2.4 Results and Discussions

Results and discussions of WGD4_1x2 has been computed for twenty benchmark (BMP) images [130, 131] as given in fig. 1.1. The quality of the watermarked images are quantified as well as analyzed through peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM), universal image quality index (UIQ), standard deviation (SD) and standard deviation error (SDE) respectively at varying payload. Comparison is made among the WGD4_1x2, WGD4_2x2, WST_1x2, WBT_1x2, WLT_1x2, WDHT_1x2 and Varsaki et al.'s DPTHDI [88] as well as DGTDHS [129] respectively. The extracted "Gold Coin" and the different states of alteration for "Lena", "Baboon" and "Pepper" images are shown in fig. 6.6.


(a) Original Lena (b) Watermarked Lena (c) Secret Gold-Coin

(d) Original Baboon (e) Watermarked Baboon (f) Secret Gold-Coin

(g) Original pepper (h) Watermarked Pepper (i) Secret Gold-Coin

Fig 6.6. Cover, watermarked and authenticating watermark image in the proposed WGD4_1x2 technique

The results corresponding to the standard quality metrics are computed for WGD4_1x2 for payload range [0.5 – 3 bpB]. Table 6.2 revealed the minimum value of peak signal to noise ratio (PSNR) as 35.80 dB at 3 bpB of payload for the “Athens” image and that of the maximum value is 54.80 dB at 0.5 bpB of payload for the “Barnfall” image. The PSNR of “Athens” image ensured that the watermarked images obtained are highly perceptible since, the minimum value of PSNR is much greater than 30 dB [148]. The lowest mean squared error (MSE) is 0.21 for “Barnfall” at payload of 0.5 bpB and that of the highest value is 17.08 for “Athens” at 3 bpB of payload. Other quality metrics such as the image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) are 0.997765 (Bobcat), 0.962135 (Splash) and 0.674038 (Splash) respectively at 3 bpB and that of are 0.999992 (Airplane), 0.999881 (San Diego) and 0.999232 (San Diego) respectively at 0.5 bpB. The usual values of IF, SSIM and UIQ ranges from [0, 1], the closer the IF, SSIM and UIQ to one, the more similar is the watermarked image to the original image. Experimental results are summarized against the average values for these metrics at variable payload.

Table 6.2. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WGD4_1x2 technique

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.122957	0.251642	0.999984	0.999580	0.994002
	1.0	51.131130	0.501149	0.999968	0.999168	0.988280
	1.5	46.337986	1.511060	0.999904	0.997502	0.967810
	2.0	44.130790	2.511894	0.999840	0.995884	0.949337
	2.5	40.027658	6.461219	0.999591	0.989022	0.893646
	3.0	37.933751	10.464179	0.999340	0.982169	0.847512
Baboon	0.5	54.136782	0.250843	0.999986	0.999854	0.998885
	1.0	51.145136	0.499535	0.999973	0.999710	0.997788
	1.5	46.354334	1.505382	0.999920	0.999129	0.993683
	2.0	44.124373	2.515608	0.999866	0.998530	0.989673
	2.5	40.063733	6.407772	0.999660	0.996235	0.976525
	3.0	37.976329	10.362092	0.999451	0.993891	0.964771
Pepper	0.5	53.996503	0.259077	0.999978	0.999374	0.989625
	1.0	51.068759	0.508398	0.999957	0.998630	0.984935
	1.5	46.186196	1.564806	0.999867	0.995734	0.968033
	2.0	43.932869	2.629016	0.999778	0.992245	0.953083
	2.5	39.776900	6.845265	0.999422	0.982093	0.906500
	3.0	37.671417	11.115745	0.999060	0.972496	0.867577

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	54.091841	0.253452	0.999992	0.999493	0.980928
	1.0	51.113820	0.503150	0.999985	0.998986	0.964849
	1.5	46.410737	1.485958	0.999957	0.997183	0.918575
	2.0	44.200140	2.472101	0.999929	0.995092	0.881420
	2.5	40.007817	6.490805	0.999814	0.988398	0.787803
	3.0	37.977256	10.359879	0.999703	0.980043	0.724689
Sailboat	0.5	54.093148	0.253376	0.999987	0.999661	0.994812
	1.0	51.111340	0.503438	0.999974	0.999330	0.990131
	1.5	46.369968	1.499973	0.999924	0.998086	0.974810
	2.0	44.124878	2.515316	0.999872	0.996726	0.959938
	2.5	40.022493	6.468908	0.999673	0.991824	0.918359
	3.0	37.977700	10.358820	0.999477	0.986669	0.885807
Earth	0.5	54.139491	0.250686	0.999985	0.999682	0.997291
	1.0	51.144086	0.499656	0.999970	0.999366	0.994636
	1.5	46.326587	1.515031	0.999910	0.998089	0.984247
	2.0	44.117527	2.519577	0.999850	0.996823	0.974504
	2.5	39.924596	6.616383	0.999607	0.990952	0.938855
	3.0	37.929899	10.473464	0.999376	0.986402	0.910354
San Diego	0.5	54.125503	0.251495	0.999990	0.999881	0.999232
	1.0	51.130127	0.501265	0.999981	0.999761	0.998490
	1.5	46.344507	1.508792	0.999943	0.999284	0.995672
	2.0	44.129625	2.512568	0.999906	0.998809	0.993064
	2.5	40.017896	6.475758	0.999758	0.996888	0.983862
	3.0	37.936742	10.456975	0.999610	0.995059	0.975656
Splash	0.5	54.049560	0.255931	0.999977	0.999030	0.964135
	1.0	51.085170	0.506481	0.999955	0.997791	0.945965
	1.5	46.278074	1.532049	0.999863	0.993528	0.892648
	2.0	43.993130	2.592789	0.999769	0.988566	0.847727
	2.5	39.942726	6.588821	0.999417	0.975649	0.749511
	3.0	37.789328	10.818013	0.999043	0.962135	0.674038
Oakland	0.5	54.106179	0.252616	0.999986	0.999746	0.998429
	1.0	51.123478	0.502033	0.999973	0.999483	0.997114
	1.5	46.301053	1.523965	0.999918	0.998439	0.992029
	2.0	44.098118	2.530862	0.999864	0.997412	0.987274
	2.5	40.011010	6.486035	0.999649	0.993408	0.970474
	3.0	37.932461	10.467288	0.999434	0.989498	0.954411

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	54.159628	0.249526	0.999991	0.999424	0.992236
	1.0	51.146496	0.499379	0.999982	0.998854	0.984754
	1.5	46.449943	1.472604	0.999947	0.996713	0.959131
	2.0	44.185606	2.480388	0.999911	0.994412	0.933501
	2.5	40.153781	6.276278	0.999775	0.986232	0.863273
	3.0	38.031348	10.231646	0.999635	0.977104	0.804730
Anhinga	0.5	52.331008	0.380171	0.999970	0.999836	0.882762
	1.0	49.769127	0.685752	0.999947	0.999623	0.885525
	1.5	45.289094	1.923848	0.999852	0.998988	0.871648
	2.0	43.035249	3.232625	0.999752	0.998282	0.866474
	2.5	38.115277	10.035814	0.999234	0.994610	0.833830
	3.0	36.246472	15.432385	0.998818	0.991767	0.823809
Athens	0.5	51.861877	0.423538	0.999966	0.999864	0.971347
	1.0	49.368925	0.751948	0.999940	0.999667	0.974375
	1.5	45.110883	2.004435	0.999840	0.998827	0.958119
	2.0	42.659928	3.524419	0.999718	0.997916	0.951207
	2.5	37.979281	10.355050	0.999172	0.995133	0.911736
	3.0	35.803467	17.089665	0.998634	0.991708	0.900060
Bardowl	0.5	52.484000	0.367012	0.999963	0.999858	0.999006
	1.0	50.012786	0.648338	0.999935	0.999610	0.998831
	1.5	45.481886	1.840312	0.999816	0.999146	0.995698
	2.0	43.120178	3.170023	0.999684	0.998417	0.993865
	2.5	38.529833	9.122146	0.999091	0.995629	0.981522
	3.0	36.404883	14.879627	0.998518	0.992445	0.974651
Barnfall	0.5	54.808111	0.214916	0.999966	0.999801	0.999011
	1.0	51.271809	0.485176	0.999923	0.999467	0.998049
	1.5	46.653800	1.405077	0.999776	0.998684	0.993094
	2.0	44.566359	2.272188	0.999640	0.997775	0.989764
	2.5	40.106131	6.345520	0.998995	0.993396	0.972860
	3.0	37.975950	10.362995	0.998369	0.988881	0.960132
Butrfly	0.5	52.256337	0.386764	0.999972	0.999828	0.995469
	1.0	49.814606	0.678609	0.999951	0.999587	0.995218
	1.5	45.453940	1.852193	0.999869	0.998951	0.987594
	2.0	43.006994	3.253725	0.999770	0.998153	0.983483
	2.5	38.323302	9.566434	0.999323	0.994824	0.959478
	3.0	36.205594	15.578330	0.998899	0.991521	0.948631

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Bobcat	0.5	52.033893	0.407090	0.999942	0.999688	0.749572
	1.0	49.543243	0.722363	0.999898	0.999113	0.750874
	1.5	45.812615	1.705370	0.999759	0.998420	0.742545
	2.0	42.731618	3.466718	0.999512	0.996680	0.734157
	2.5	38.468247	9.252426	0.998701	0.989505	0.714165
	3.0	36.114442	15.908751	0.997765	0.978119	0.701484
Bodie	0.5	54.468152	0.232415	0.999962	0.999851	0.982874
	1.0	51.291557	0.482975	0.999920	0.999644	0.979014
	1.5	46.599813	1.422653	0.999765	0.999059	0.974502
	2.0	44.210040	2.466472	0.999594	0.998164	0.970143
	2.5	40.344818	6.006182	0.998991	0.996141	0.956649
	3.0	38.222547	9.790968	0.998352	0.992489	0.948925
Bluheron	0.5	54.080286	0.254127	0.999968	0.999688	0.996459
	1.0	51.426837	0.468162	0.999942	0.999193	0.995028
	1.5	46.210697	1.556003	0.999809	0.998013	0.985959
	2.0	43.970313	2.606447	0.999681	0.996528	0.981108
	2.5	41.051854	5.103814	0.999375	0.994600	0.956595
	3.0	39.272646	7.688041	0.999058	0.991201	0.945713
Colomtn	0.5	53.537903	0.287932	0.999977	0.999732	0.985006
	1.0	50.855935	0.533933	0.999958	0.999428	0.984484
	1.5	46.030215	1.622029	0.999873	0.998431	0.977297
	2.0	43.722239	2.759665	0.999784	0.997228	0.972398
	2.5	39.680890	6.998279	0.999453	0.993494	0.952771
	3.0	37.681276	11.090539	0.999133	0.990122	0.940625
Desert	0.5	54.052085	0.255783	0.999963	0.999830	0.998898
	1.0	51.034300	0.512448	0.999927	0.999539	0.998291
	1.5	46.289013	1.528195	0.999781	0.998907	0.995657
	2.0	44.010951	2.582172	0.999630	0.997889	0.993172
	2.5	39.599920	7.129979	0.998969	0.994441	0.983607
	3.0	37.441277	11.720673	0.998292	0.990424	0.977950
Average Case	0.5	53.646769	0.286920	0.999975	0.999685	0.973499
	1.0	50.779430	0.549709	0.999953	0.999298	0.970332
	1.5	46.114570	1.598987	0.999865	0.998056	0.956438
	2.0	43.803550	2.730729	0.999768	0.996577	0.945265
	2.5	39.607410	7.251644	0.999384	0.991624	0.910601
	3.0	37.526240	11.732500	0.998998	0.986207	0.886576

A comparative study has been made among the 1 x 2 block based watermarking in G-lets D4 domain (WGD4_1x2) with Varsaki et al.'s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] in terms of peak signal to noise ratio (dB). The comparison has been done for five different color images such as "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat". The payloads of DPTHDI [88] and DGTDHS [129] techniques are 0.25 bpB and 1 bpB respectively which are fixed as well as considerably low. In contrast to DPTHDI [88], proposed WGD4_1x2 ensured equal or higher PSNR (dB) at 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for "Lena", 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Baboon", 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for "Pepper", 0.5, 1, 1.5 and 2 bpB of payloads for "Airplane" and 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for "Sailboat" respectively. Similarly, as compared to the DGTDHS [129], proposed WBT_1x2 ensured higher PSNR (dB) at 0.5 and 1 bpB of payloads for "Lena", 0.5 and 1 bpB of payloads for "Baboon", 0.5 and 1 bpB of payloads for "Pepper", 0.5 and 1 bpB of payloads for "Airplane" and 0.5 and 1 bpB of payloads for "Sailboat" respectively. Even at highest offered payload (i.e., 3 bpB), the PSNR values of all five images are above 30 dB; as a consequence, the obtained watermarked images retains good visual clarity.

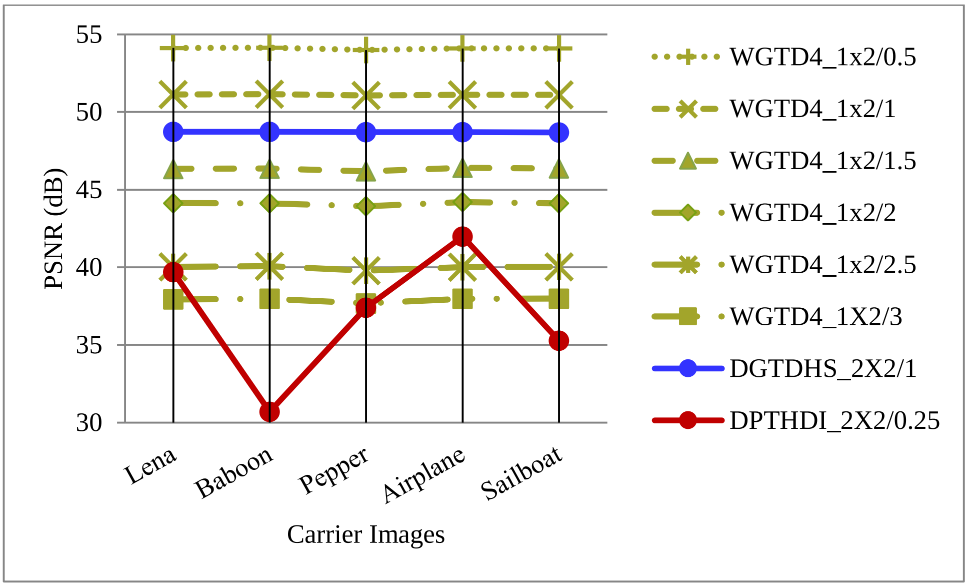


Fig. 6.7. Performance analysis of PSNR (dB) for variable payload based WGD4_1x2 and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Table 6.2 reveals that the average peak signal to noise ratio (PSNR) lies between 37.52 dB and 53.64 dB. Fig. 6.8 depicts the comparative analysis among WGD4_1x2, WGD4_2x2, WST_1x2, WBT_1x2, WLT_1x2, WDHT_1x2 and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes in terms of average PSNR at different payload values. In comparison with WGD4_2x2, WST_1x2, WBT_1x2, WLT_1x2 and WDHT_1x2, the WGD4_1x2 shows improvement in average PSNR values at 1 – 3 bpB of payload values. Hence, the WGD4_1x2 is superior in terms of watermarked image's quality as compared to WGD4_2x2, WST_1x2, WBT_1x2, WLT_1x2 and WDHT_1x2 respectively. The average PSNR for DPTHDI [88] is 37.40 dB as computed from "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" images at 0.25 bpB of payload. The average PSNR for DGTDHS [129] is 48.70 dB by taking the average of PSNR values for "Lighthouse", "Elaine", "Lenna", "Boat" and "F16" images at 1 bpB of payload. It is seen that the average PSNR for WGD4_1x2 scheme has been improved over DPTHDI [88] at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads however, compared to DGTDHS [129], the average PSNR for WGD4_1x2 scheme is improved with respect to 0.5 and 1 bpB of payload respectively.

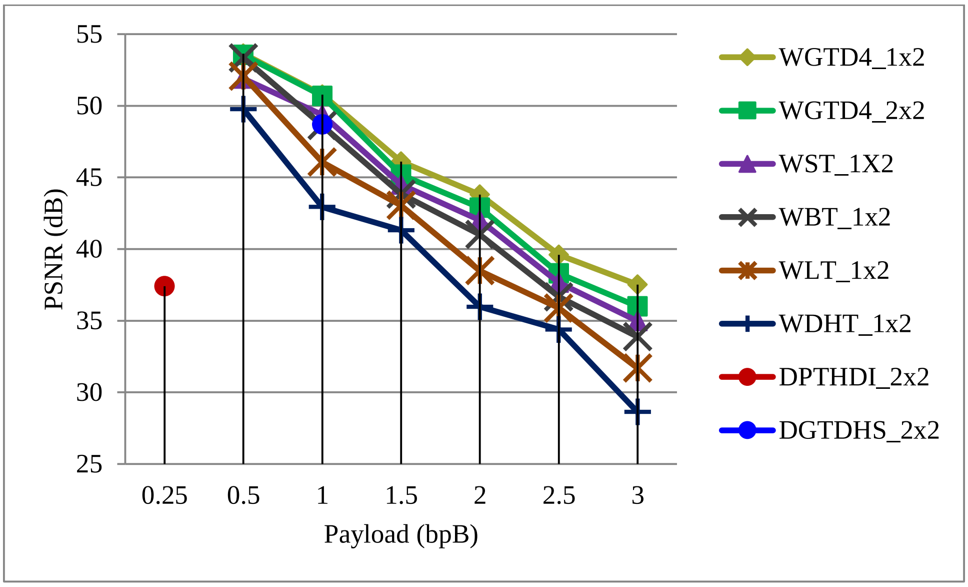


Fig. 6.8. Graphical representation of variation of average PSNR (dB) with respect to payload for WGD4_1X2, WGD4_2X2, WST_1X2, WBT_1X2, WLT_2x2, WDHT_1x2 and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

In fig. 6.9, the standard deviation (SD) analysis has been carried out for five color images such as "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively. The values of standard deviation (SD) in the watermarked images for the payload variation of 0.5 to 3 bpB are remains almost constant. It is to be noted that the payload value of 0 bpB designates the original image. The dispersion made in the standard deviation (SD) values for the watermarked images with respect to the original image are very minimum and hence, the WGD4_1x2 maintains minimum difference between source and watermarked images.

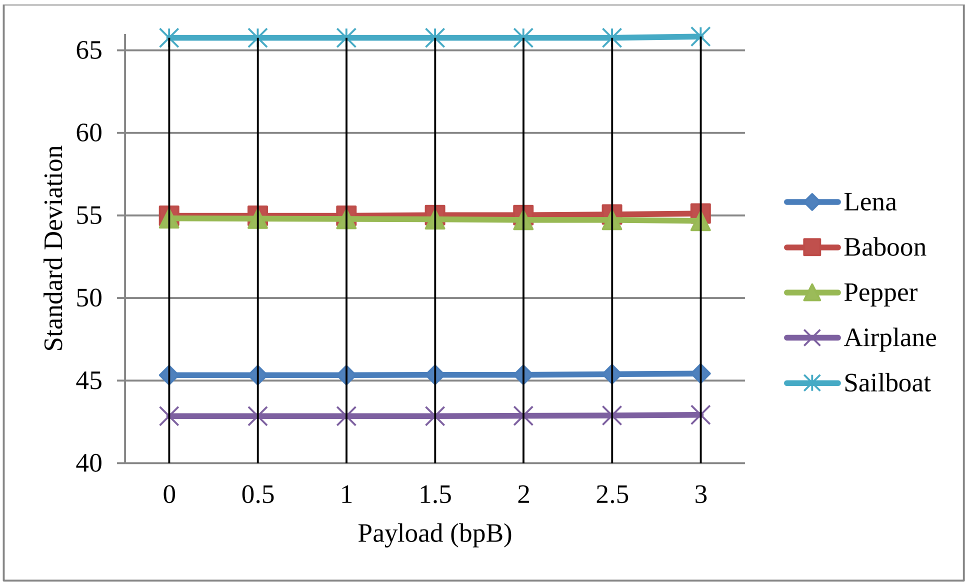


Fig. 6.9. Graphical representation of standard deviation (SD) for WGD4_1x2 with respect to 0, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

It is seen from fig 6.9 that the dispersion of standard deviation (SD) is almost constant with respect to the increasing values of payload for the spread from 0.5 to 3 bpB. To analyze the dispersion more clearly, a standard deviation error (SDE) analysis has been made. The standard deviation error (SDE) is computed as the absolute difference of standard deviation (SD) values between the original and the watermarked images. It can be analyzed from fig. 6.10 that the maximum dispersion of error is 0.2 for the watermarked images in comparison with the original image however, the standard deviation error (SDE) values are almost equal to zero for usual cases.

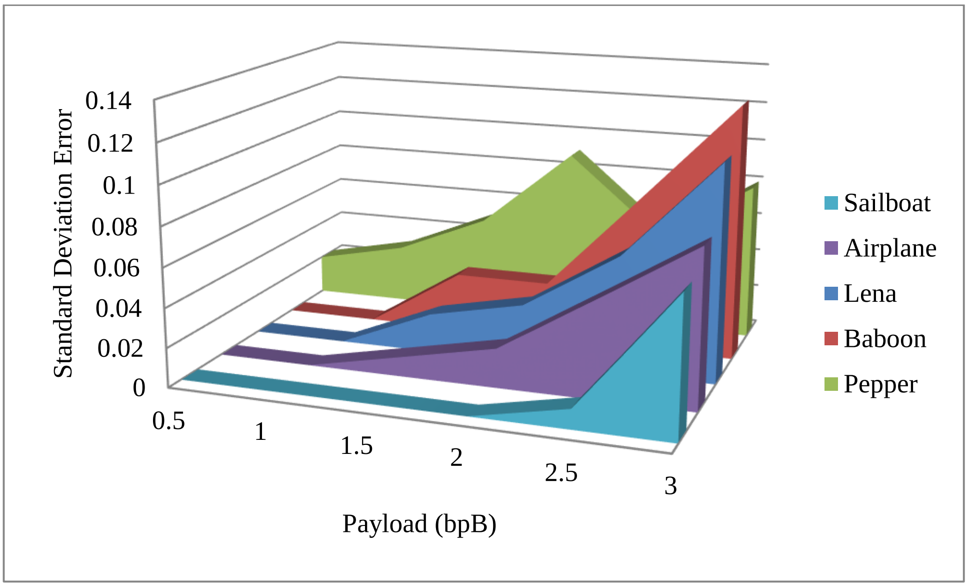


Fig. 6.10. Graphical representation of standard deviation error (SDE) for WGD4_1x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

The general characteristic of standard deviation error (SDE) states that as the payload increases, the error also increases. Fig. 6.10 depicts that the error is almost 0 for “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat”, up to the payload value of 3 bpB. However, the rate of increase of error has been minimized tremendously for the WGD4_1x2 scheme than the WGD4_2x2 scheme at varying payload which demonstrates the superiority of WGD4_1x2 scheme.

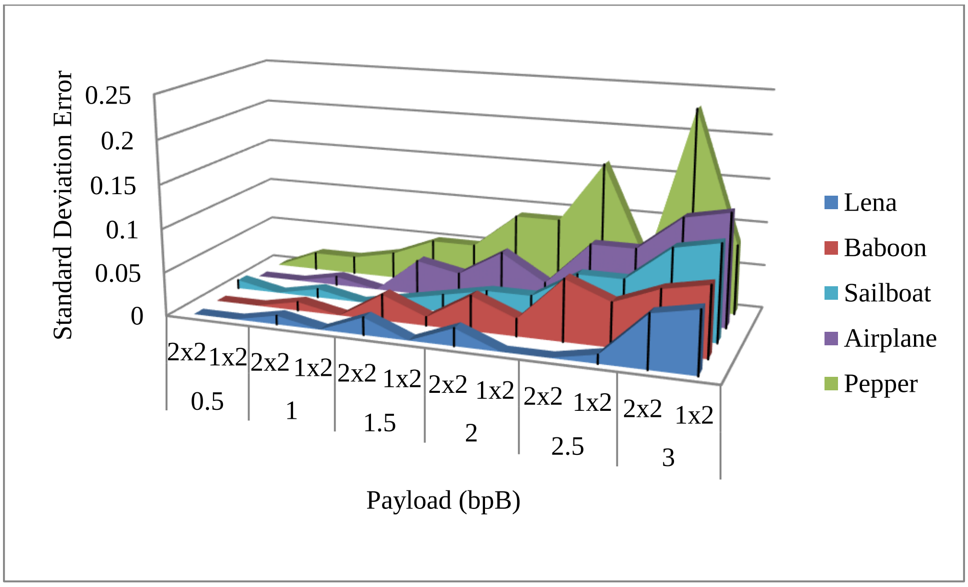


Fig. 6.11. Pictorial representation of variation of standard deviation error (SDE) between WGD4_1x2 and WGD4_2x2 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads

6.3 Salient Features

The WGD4_2x2 and WGD4_1x2 are two novel fragile watermarking schemes in G-lets D4 domain to avoid the drawbacks of Varsaki et al.’s schemes [88, 129] which includes high distortion, less payload, lacking in usage of color image as the cover and the high computational overhead. Both schemes offer a variable payload of up to 3 bpB by retaining perceptible visual imperceptibility.

Some of the relevant features of the proposed techniques are elaborated to highlight the effectiveness.

Both schemes perform a calculation based on integer which is a significant advantage of G-lets D4 in terms of computational complexity. No overflow and underflow conditions are arises and therefore, no need to incorporate any pixel adjustment scheme prior to embedding. It is seen that most of the existing schemes offers fixed payload which is relatively low; however, proposed WGD4_2x2 and WGD4_1x2 schemes are based on variable payloads (i.e., 0.5 to 3 bpB) by keeping perceptible quality of the watermarked image. In contrast to Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes, both WGD4_2x2 and WGD4_1x2 schemes ensure variable payloads with considerable quality degradation. The quality distortion can further be reduced by introducing a quality enhancement scheme which is to be discussed in great detail on chapter 7. Genetic Algorithm (GA) based optimization which is to be discussed in chapter 8 can also be an option for the minimization of quality degradation with a systematic manner. The valuable achievement of both quality improvement schemes is their ability of keeping the fabricated watermark bits intact.

Chapter 7

Quality Enhancement (QE)

7.1 Introduction

The major objective of proposed watermarking schemes of chapters 2 to 6 was to adopt variable payload with acceptable degradation in quality. However, some of the proposed schemes viz. WDHT_2x2, WDHT_1x2, WLT_2x2, WBT_2x2 and WST_2x2 suffers from low average PSNR (i.e., < 30 dB) at high payload (i.e., ≥ 2.5 bpB). As a result, the watermarked images generate perceptible visual distortion with respect to the original images [148]. To improve the quality of the watermarked images, a post-embedding quality enhancement scheme has been incorporated in transform domain. To preserve the integrity, the fabricated bits are kept unaffected. By introducing the quality enhancement scheme followed by (often following) embedding, the overall quality distortion in the watermarked images is reduced considerably. Therefore, the watermarking schemes proposed in chapters 2 to 6 are extended by incorporating the quality enhancement schemes.

7.2 Quality Enhancement

Quality enhancement method has been incorporated within the proposed watermarking techniques to minimize the quality distortion of the watermarked images. As a consequence of the incorporation of the enhancement technique, the degradation of quality has been significantly reduced for watermarking techniques proposed in chapters 2, 3, 4, 5 and 6 which are analyzed by means of peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ), respectively. Simulation results demonstrate that the techniques of chapters 2 to 6 have been improved in terms of visual clarity through the quality enhancement.

Section 7.2.1 deals with the adaptive quality enhancement scheme whereas, section 7.2.2 deals with the exploiting modification direction (EMD) based quality enhancement scheme.

7.2.1. Adaptive Quality Enhancement

The adaptive quality enhancement phase is introduced in between the fabrication and the inverse transform phases. In this phase, each embedded component is treated as k bits representation of which r bits from the least significant part are not altered. The absolute difference between the pre-embedded component (t) and the embedded component (t′) has been computed. If the difference is greater than 2^{r-1} then the proposed quality enhancement scheme take each embedded component (t′) as input and returns the quality enhanced component (t″) closest to the pre-embedded component (t) as output. The quality enhancement has been employed on the embedded component (t′) by considering all possible combinations of left most (k-r) bits i.e., 2^{k-r} alterations. The process is repeated until and unless the difference between the pre-embedded component (t) and the embedded component (t′) becomes less than or equal to 2^{r-1}. Therefore, if the difference is less than or equal to 2^{r-1} then the embedded component (t′) is kept unaltered and the embedded component (t′) itself is treated as the quality enhanced component (t″).

The proposed adaptive quality enhancement technique is given in algorithm 7.1.

Algorithm 7.1:

Input: Pre-embedded component (t) and embedded component (t′).

Output: Quality enhanced component (t″).

Method: The quality enhanced component (t″) ensures minimum difference between the embedded component (t′) and the pre-embedded component (t) by reassembling the left most (k-r) bits of the embedded component (t′). The process is repeated up to 2^{k-r} times without hampering the least significant r bits. Detailed process of the algorithm is described in a step-by-step manner:

Step 1. The bit-stream corresponding to the pre-embedded component (t), the embedded component (t′) and the quality enhanced component (t″) are considered as consisting of two segments i.e., (k-r) and r. Here, k is the length of the component (in bits) of which r bits starting from the least significant bit (LSB-0) position toward higher order bit position are kept unaffected.

Step 2. The decimal equivalent of the unaffected bits (r) corresponding to the embedded component (t′) is computed as given in equation (7.1).

$$s = t' \,\%\, 2^r \tag{7.1}$$

Step 3. The absolute difference (D) between the embedded component (t′) and the pre-embedded component (t) is computed in equation (7.2).

$$D = \begin{cases} t' - t : \; t' > t \\ t - t' : \; t' < t \end{cases} \tag{7.2}$$

Step 4. If the difference (D) becomes greater than 2^{r-1} then the embedded component (t′) to be altered (without affecting least *significant r bits) to minimize the difference between the embedded component (t') and the pre-embedded component (t). However, if the difference (D) becomes less than or equal to 2^{r-1} then no alteration is needed at all and as a consequence, the embedded component (t') itself is treated as the quality enhanced component (t''). The quality enhanced component (t'') can be derived as given in eqaution (7.3).*

$$t'' = \begin{cases} t' : D \leq 2^{r-1} \\ (2^r \times i + s) : D > 2^{r-1} \end{cases} \tag{7.3}$$

where, i is the iteration number, initialized as 0.

Step 5. Repeat steps 3 and 4 up to 2^{k-r} iterations (where, i ranges from $[0, 2^{k-r}]$ and is incremented by 1 at each iteration i.e., $0 \leq i \leq 2^{k-r}$) to find the suitable quality enhanced component (t'').

Step 6. Stop.

7.2.2. EMD based Quality Enhancement

The watermarking scheme based on exploiting modification direction in G-lets D4 domain (WGD4_1x2) has already been discussed in section 6.2.2. The WGD4_1x2 scheme ensured highest average PSNR (dB) amongst all of the proposed watermarking schemes discussed so far. However, the quality of the WGD4_1x2 scheme has further been improved by incorporating the exploiting modification direction (EMD) based quality enhancement scheme. The EMD based quality enhancement is introduced in the WGD4_1x2 prior to embedding. Initially, consider the pair of transformed components (t_i, t_{i+1}) from which $(2^{\lceil \lambda/2 \rceil} + 1)$ pairs of temporary values in the range of $(t_k - 2^{\lceil \lambda/2 \rceil -1})$ to $(t_k + 2^{\lceil \lambda/2 \rceil -1})$ are obtained. Each pair of temporary values is able to fabricate λ bit(s) (where, $1 \leq \lambda \leq 6$) of the secret bit-stream which results a pair of embedded components (t'_i, t'_{i+1}). Each pair of embedded components (t'_i, t'_{i+1}) is compared against the actual pair of transformed components (t_i, t_{i+1}). The process is repeated for $(2^{\lceil \lambda/2 \rceil} + 1)$ pairs of temporary values and then the pair of embedded components (t'_i, t'_{i+1}) which would ensure the minimum deviation with respect to the actual pair of transformed components (t_i, t_{i+1}) is considered to be the target pair of transformed components (t''_i, t''_{i+1}). On the other hand, if the actual pair of transformed components (t_i, t_{i+1}) itself offers minimum difference with respect to the pair of embedded components (t'_i, t'_{i+1}) then the target pair of transformed components (t''_i, t''_{i+1}) is to be assigned with the actual pair of transformed components (t_i, t_{i+1}). Finally, the target pair of transformed components is responsible for the fabrication of secret information.

The proposed EMD based quality enhancement technique is given in algorithm 7.2.

Algorithm 7.2:

Input: Actual pair of transformed components (t_i, t_{i+1}) of k bits representation and λ bits of secret data from the secret bit-stream (S).

Output: Target pair of transformed components (t''_i, t''_{i+1}) of k bits representation.

Method: The target pair of transformed components is chosen by introducing the EMD based quality enhancement feature into the WGD4_1x2 scheme of section 6.2.2 where, the fabrication of watermark bits ensures minimum quality degradation. The process in detail is discussed in a step-wise manner:

Step 1: Check the sign s_i and s_{i+1} for the actual pair of transformed components (t_i, t_{i+1}).

$$d = \begin{cases} (b_0 b_1 b_2 \dots b_{\lambda-1})_2 : s_i = s_{i+1} \\ (b_{\lambda-1} b_{\lambda-2} b_{\lambda-3} \dots b_0)_2 : s_i \neq s_{i+1} \end{cases} \qquad (7.4)$$

where, the secret data (d) corresponding to the binary stream ($b_0b_1b_2$...) or ($b_{\lambda-1}b_{\lambda-2}b_{\lambda-3}$...) is obtained by retrieving λ number of bits from the secret bit-stream (S) corresponding to the message digest (MD), size and content of the watermark.

Step 2: Consider the transformed components as positive integers during embedding i.e., (t_i, t_{i+1}) = ($|t_i|$, $|t_{i+1}|$).

Step 3: Assign $P_{VMA} = \lambda$ for VMA and n is obtained from P_{VMA} for generating the vector co-ordinates of VCA as per given rule:

$$n = \begin{cases} P_{VMA} - 1 : \lambda \; is \; even \\ P_{VMA} : \lambda \; is \; odd \end{cases} \qquad (7.5)$$

Step 4: Allocate the VCA evenly according to n, and generate n vectors denoted as (g_1, g_2, g_3,...,g_n).

Step 5: Allocate the VMA, consisting of λ bits into two halves i.e., λ_1 and λ_2 where, $\lambda_1 = \left\lfloor \frac{\lambda}{2} \right\rfloor$, $\lambda_2 = \lambda - \lambda_1$.

Step 6: Compute the reference value f by using equation (7.6) with respect to vectors (g_1, g_2, g_3, ... g_n) to embed λ number of secret bits.

$$f(g_1, g_2, \dots, g_n) = \sum_{i=0}^{n} g_i(2^i - 1) mod\ 2^{\lambda} \qquad (7.6)$$

Step 7: Calculate the summation value, $S_V = (d + f) \% 2^{\lambda}$.

Step 8: Obtain the pair of embedded components (t'_i, t'_{i+1}) by replacing the VMA with S_V as given below.

$$t'_i = \left(t_i \ \& \left(2^8 - 2^{\lambda_1}\right)\right) + \left\lfloor \frac{S_V}{2^{\left\lceil \frac{\lambda}{2} \right\rceil}} \right\rfloor$$

$$t'_{i+1} = \left(t_{i+1} \ \& \left(2^8 - 2^{\lambda_2}\right)\right) + (S_V \ \%2^{\left\lceil \frac{\lambda}{2} \right\rceil})$$

Here, '&' is the bitwise-AND operator and all calculations in this context are purely integer based.

Step 9: The sign s_i and s_{i+1} of the pair of transformed components (t_i, t_{i+1}) are re-assigned to the pair of embedded components (t'_i, t'_{i+1}).

Step 10: Repeat steps 1 to 9 for a set of ($2^{\lceil \lambda/2 \rceil}$ + 1) pairs of temporary values in the range of (t_k - $2^{\lceil \lambda/2 \rceil -1}$) to ($t_k$ + $2^{\lceil \lambda/2 \rceil -1}$) *to return the target pair of transformed components (t''_i, t''_{i+1}) as output. Amongst all pairs of temporary values, the pair of embedded components (t'_i, t'_{i+1}) having the minimum deviation with respect to the actual pair of transformed components (t_i, t_{i+1}) is chosen as the target pair of transformed components (t''_i, t''_{i+1}).*

Step 11: Stop.

7.3 Quality Enhancement of Proposed Watermarking Schemes

The incorporation of the quality enhancement method ensures high quality watermarked images without losing the fabricated secret bits. It is seen from chapters 2, 3, 4, 5 and 6 that the watermarking techniques degrade the quality of the watermarked images as the payload increases. The variation of quality with respect to increasing payload can be measured based on the standard quality metrics such as the peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ), respectively.

Section 7.3.1, 7.3.2, 7.3.3, 7.3.4 and 7.3.5 of this chapter deals with quality enhancement of Discrete Hartley Transform (DHT) based watermarking of chapter 2, Legendre Transform (LT) based watermarking of chapter 3, Binomial Transform (BT) based watermarking of chapter 4, Stirling Transform (ST) based watermarking of chapter 5 and group of linear transformations for dihedral group of order 4 (G-lets D4) based watermarking of chapter 6 respectively.

7.3.1 Quality Enhancement of Discrete Hartley Transform (DHT) based watermarking

Chapter 2 dealt with Discrete Hartley Transform (DHT) based watermarking which was further classified into two major sections: 2 x 2 block based watermark fabrication using two dimensional Discrete Hartley Transform (2D-DHT) or more specifically Separable Discrete Hartley Transform (SDHT) i.e., WDHT_2x2 and 1 x 2 block based watermark fabrication using one dimensional Discrete Hartley Transform (1D-DHT) i.e., WDHT_1x2.

Section 7.3.1.1 deals with the 2 x 2 block based watermark fabrication using two dimensional Discrete Hartley Transform (2D-DHT) or more specifically Separable Discrete Hartley Transform (SDHT) followed by adaptive quality enhancement scheme of section 7.2.1 i.e., WDHT_2x2_QE whereas in section 7.3.1.2, 1 x 2 block based watermark fabrication using one dimensional Discrete Hartley Transform (1D-DHT) followed by adaptive quality enhancement scheme of section 7.2.1 i.e., WDHT_1x2_QE has been discussed.

7.3.1.1 Quality Enhancement for 2 x 2 Block based Watermark Fabrication

The quality enhancement scheme of section 7.2.1 is introduced as the post-embedding operation in WDHT_2x2 scheme of section 2.2.1. Decompose the carrier image into 2 x 2 non-overlapping blocks of pixel components which in turn are adjusted based on the equation (2.8) of section 2.2.1.1 to avoid the occurrence of overflow/underflow. Separable Discrete Hartley Transform (SDHT) converts each 2 x 2 sub-matrix of pixel components into transform domain. Secret bits corresponding to the message digest, size and content of the watermark are fabricated on transformed components starting from the second bit position of the least significant part (i.e., LSB-2) toward higher order bit position. Adaptive quality enhancement of section 7.2.1 is applied on each embedded component to find the quality enhanced component corresponding to each embedded component to reduce the quality distortion. However, each quality enhanced component preserves two least significant bits along with the fabricated bits of the respective embedded component. Inverse Separable

Discrete Hartley Transform (ISDHT) is applied over each 2 x 2 sub-matrix of the quality enhanced components to obtain the pixel components in spatial domain. The process is repeated until and unless the entire secret information is concealed and the watermarked image is produced.

An example for the proposed technique is given in section 7.3.1.1.1. Simulation results, comparative analysis and comprehensive discussions for the same have been elaborated in section 7.3.1.1.2.

7.3.1.1.1 Example

The carrier image is partitioned into 2 x 2 non-overlapping blocks of red, green and blue channels in row major order. The pixel components are adjusted by applying the pixel adjustment rule as discussed in equation (2.8) of section 2.2.1.1. The 2 x 2 sub-matrices of adjusted pixel components for a payload value of 3 bpB are obtained as follows:

$$R_1 = \begin{bmatrix} 224 & 69 \\ 32 & 112 \end{bmatrix} G_1 = \begin{bmatrix} 92 & 202 \\ 32 & 51 \end{bmatrix} B_1 = \begin{bmatrix} 32 & 119 \\ 220 & 224 \end{bmatrix}$$

Separable Discrete Hartley Transform (SDHT) is applied over 2 x 2 sub-matrices of pixel components to obtain the 2 x 2 sub-matrices of transformed components such as T(R1), T(G1) and T(B1) as follows:

$$T(R_1) = \begin{bmatrix} 437 & 75 \\ 149 & 235 \end{bmatrix} T(G_1) = \begin{bmatrix} 377 & -129 \\ 211 & -91 \end{bmatrix} T(B_1) = \begin{bmatrix} 595 & -91 \\ -293 & -83 \end{bmatrix}$$

Secret bit stream of *"1010000101100000110011110110110000001"* is to be fabricated into the transformed components based on the embedding rule of WDHT_2x2 as mentioned in equation (2.9) of section 2.2.1.1. Suppose, three bits are fabricated (λ = 3) on each transformed component starting from LSB-2 toward higher order bit position then, the LSB-0 and LSB-1 of each embedded component are kept unaltered to ensure the non-fractional pixel components subsequent to inverse transform. The 2 x 2 sub-matrices of embedded components are obtained as follows:

$$T'(R_1) = \begin{bmatrix} 437 & 67 \\ 137 & 239 \end{bmatrix} T'(G_1) = \begin{bmatrix} 353 & -153 \\ 207 & -95 \end{bmatrix} T'(B_1) = \begin{bmatrix} 603 & -91 \\ -289 & -83 \end{bmatrix}$$

The adaptive quality enhancement of section 7.2.1 has been utilized to minimize the difference between the embedded component and the original component without losing the embedded bits. Basically, it is applied on each embedded component by taking the closest value of the pre-embedded component without hampering the two unaltered least significant bits along with three embedded bits i.e., LSB-0, LSB-1, LSB-2, LSB-3 and LSB-4 respectively. The 2 x 2 sub-matrices of quality enhanced components corresponding to RGB color channels are obtained as follows:

$$T''(R_1) = \begin{bmatrix} 437 & 67 \\ 137 & 239 \end{bmatrix} T''(G_1) = \begin{bmatrix} 385 & -121 \\ 207 & -95 \end{bmatrix} T'(B_1) = \begin{bmatrix} 603 & -91 \\ -289 & -83 \end{bmatrix}$$

Applying Inverse Separable Discrete Hartley Transform (ISDHT) on each 2 x 2 sub-matrix of quality enhanced components yields the 2 x 2 sub-matrix of pixel components as follows:

$$R'_1 = \begin{bmatrix} 220 & 67 \\ 32 & 118 \end{bmatrix} G'_1 = \begin{bmatrix} 94 & 202 \\ 38 & 51 \end{bmatrix} B'_1 = \begin{bmatrix} 35 & 122 \\ 221 & 225 \end{bmatrix}$$

The un-alteration of two least significant bits for each transformed component ensured that all re-computed pixel components corresponding to red, green and blue sub-matrices are non-fractional, non-negative and less than or equal to 255.

7.3.1.1.2 Results and Discussions

The WDHT_2x2_GAO scheme is nothing but the WDHT_2x2 scheme of section 2.2.1 followed by adaptive quality enhancement of section 7.2.1. The quality of WDHT_2x2_QE is extensively analyzed by means of peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) respectively. Twenty benchmark (BMP) images [130, 131] of dimension 512 x 512 along with the varying sizes of the fabricated watermark as given in fig. 1.1 are taken to compute results. The minimum and maximum values of PSNR obtained are 25.57 dB at 3 bpB of payload for the “Desert” image and 47.89 dB at 0.5 bpB of payload for the “San Diego” image respectively. Since, the PSNR of “Desert” image is less than 30 dB at 3 bpB of payload, the WDHT_2x2_QE scheme suffers from perceptible level of visual clarity [148] however, the WDHT_2x2_QE scheme supports variable payload that offers a spread from 0.5 to 3 bpB. The lowest MSE is 1.05 for “San Diego” at 0.5 bpB of payload while the highest MSE is 180.11 for “Desert” at 3 bpB of payload. The minimum values of IF, SSIM and UIQ are 0.972961 (Desert), 0.884292 (Bobcat) and 0.410075 (Splash) respectively whereas, the maximum values of IF, SSIM and UIQ are 0.999962 (Airplane), 0.999999 (Earth) and 0.996779 (San Diego) respectively. The average values are computed for various metrics of twenty carrier images at variable payload that offers a spread from 0.5 to 3 bpB to summarize the experimental results.

Table 7.1. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WDHT_2x2_QE scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	47.407641	1.181180	0.999927	0.999973	0.973613
	1.0	44.734959	2.185668	0.999863	0.996808	0.957305
	1.5	38.786919	8.597821	0.999488	0.997808	0.875112
	2.0	37.115939	12.63241	0.999260	0.991255	0.842236
	2.5	30.151189	62.800254	0.996351	0.979571	0.671367
	3.0	28.732823	87.055852	0.994867	0.955807	0.618730
Baboon	0.5	47.729109	1.096906	0.999942	0.999925	0.994269
	1.0	44.913212	2.097775	0.999889	0.998832	0.990719
	1.5	40.038899	6.444517	0.999661	0.998126	0.967685
	2.0	38.477328	9.233099	0.999515	0.995838	0.960026
	2.5	31.635000	44.625216	0.997673	0.987416	0.899631
	3.0	30.339634	60.133565	0.996859	0.978773	0.878582
Pepper	0.5	42.305850	3.823801	0.999629	0.983520	0.958071
	1.0	40.936259	5.241485	0.999503	0.979477	0.945116
	1.5	35.472861	18.441415	0.998239	0.969722	0.875616
	2.0	34.679115	22.139550	0.997930	0.963469	0.847007
	2.5	28.085008	101.060052	0.990346	0.934838	0.691251
	3.0	27.169507	124.775575	0.988396	0.912177	0.635053
Airplane	0.5	46.941701	1.314952	0.999962	0.999998	0.928282
	1.0	44.438064	2.340312	0.999933	0.996223	0.897211
	1.5	38.389446	9.4218406	0.999730	0.997982	0.765252
	2.0	36.634783	14.112435	0.999595	0.990199	0.727288
	2.5	30.906474	52.775569	0.998489	0.989865	0.581348
	3.0	29.129645	79.454003	0.997726	0.961418	0.535416
Sailboat	0.5	46.959658	1.309527	0.999934	0.999891	0.978710
	1.0	44.411404	2.354723	0.999881	0.997320	0.965363
	1.5	39.060173	8.073520	0.999594	0.997547	0.910958
	2.0	37.489262	11.591887	0.999415	0.992304	0.887870
	2.5	30.687894	55.499734	0.997227	0.978049	0.760780
	3.0	29.296692	76.455916	0.996167	0.958903	0.716743
Earth	0.5	47.743824	1.093195	0.999935	0.999999	0.988634
	1.0	44.900632	2.103861	0.999875	0.997560	0.979644
	1.5	39.676257	7.005750	0.999587	0.998513	0.932807
	2.0	37.834595	10.705841	0.999377	0.993497	0.908138

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	31.207602	49.240229	0.996900	0.987689	0.773856
	3.0	29.614950	71.053471	0.995635	0.969597	0.727436
San Diego	0.5	47.897953	1.055079	0.999960	0.999999	0.996779
	1.0	44.978641	2.066408	0.999923	0.999086	0.994316
	1.5	40.590317	5.676081	0.999789	0.999510	0.983461
	2.0	38.808829	8.554555	0.999683	0.997623	0.976754
	2.5	33.331694	30.193319	0.998885	0.997443	0.938470
	3.0	31.552879	45.477067	0.998324	0.990066	0.919266
Splash	0.5	43.288726	3.049353	0.999687	0.986118	0.900533
	1.0	41.783047	4.312961	0.999566	0.980952	0.867927
	1.5	36.040766	16.180938	0.998426	0.979118	0.718551
	2.0	34.870572	21.184738	0.998049	0.969539	0.669239
	2.5	28.509011	91.659873	0.991248	0.953598	0.463252
	3.0	27.309621	120.814272	0.988810	0.920714	0.410075
Oakland	0.5	45.339798	1.901518	0.999877	0.999119	0.992304
	1.0	43.355342	3.002936	0.999816	0.997173	0.987753
	1.5	38.186645	9.872241	0.999391	0.996538	0.963033
	2.0	37.024603	12.900900	0.999236	0.992724	0.950665
	2.5	30.828653	53.729778	0.996735	0.986747	0.870158
	3.0	29.595189	71.377518	0.995857	0.971783	0.833202
Foster City	0.5	47.654231	1.115982	0.999960	0.999999	0.967908
	1.0	44.776217	2.165003	0.999923	0.995607	0.945088
	1.5	39.301125	7.637791	0.999736	0.997691	0.843166
	2.0	37.457045	11.678197	0.999595	0.988728	0.805848
	2.5	31.781563	43.144359	0.998519	0.988621	0.662424
	3.0	29.939976	65.929950	0.997736	0.955558	0.606658
Anhinga	0.5	43.903242	2.647013	0.999797	0.999711	0.860142
	1.0	42.401148	3.740809	0.999713	0.998247	0.849703
	1.5	36.413196	14.851172	0.998864	0.997499	0.792635
	2.0	35.338098	19.022627	0.998543	0.993314	0.777727
	2.5	29.453971	73.736620	0.994357	0.985966	0.698655
	3.0	28.384028	94.336018	0.992773	0.970446	0.672307
Athens	0.5	45.872814	1.681894	0.999865	0.999999	0.944283
	1.0	43.364844	2.996373	0.999760	0.998599	0.929877
	1.5	38.773979	8.623476	0.999310	0.998879	0.843768
	2.0	37.157286	12.512720	0.999002	0.993807	0.818414

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	31.476935	46.279304	0.996291	0.992379	0.707134
	3.0	29.805452	68.004109	0.994559	0.974583	0.664984
Bardowl	0.5	44.362844	2.381200	0.999755	0.998586	0.989803
	1.0	42.578614	3.591028	0.999633	0.997484	0.986911
	1.5	36.840480	13.459603	0.998606	0.989146	0.942754
	2.0	35.927576	16.608206	0.998292	0.987419	0.934683
	2.5	28.454456	92.818531	0.990394	0.945990	0.820243
	3.0	27.618377	112.523376	0.988431	0.939110	0.794872
Barnfall	0.5	46.665185	1.401398	0.999773	0.999921	0.991770
	1.0	44.139956	2.506598	0.999598	0.997916	0.986069
	1.5	39.291557	7.654637	0.998721	0.997680	0.954944
	2.0	37.700119	11.042526	0.998180	0.993261	0.939133
	2.5	31.449835	46.568987	0.991935	0.978974	0.832835
	3.0	29.966805	65.523907	0.988946	0.963519	0.794218
Butrfly	0.5	46.534886	1.444081	0.999897	0.999999	0.985351
	1.0	43.914079	2.640416	0.999812	0.998612	0.978703
	1.5	39.664303	7.025060	0.999500	0.999081	0.941132
	2.0	37.921698	10.493260	0.999253	0.995147	0.919831
	2.5	32.067279	40.397305	0.997087	0.993281	0.811328
	3.0	30.194674	62.174585	0.995539	0.979172	0.760950
Bobcat	0.5	46.235915	1.546994	0.999782	0.999991	0.734985
	1.0	43.641806	2.811251	0.999604	0.997918	0.727944
	1.5	38.422680	9.350015	0.998682	0.997218	0.685157
	2.0	36.864316	13.385934	0.998111	0.991327	0.674880
	2.5	26.511285	145.195036	0.979439	0.899877	0.548967
	3.0	25.878475	167.970404	0.976242	0.884292	0.529517
Bodie	0.5	44.134201	2.509922	0.999555	0.998801	0.967046
	1.0	42.586546	3.584476	0.999373	0.997278	0.962158
	1.5	36.603500	14.214457	0.997452	0.987277	0.919055
	2.0	35.740201	17.340443	0.996938	0.984448	0.909128
	2.5	27.666503	111.283304	0.979752	0.911219	0.763941
	3.0	26.988645	130.081570	0.976614	0.901276	0.743963
Bluheron	0.5	46.036184	1.619801	0.999801	0.999999	0.982729
	1.0	43.674114	2.790415	0.999658	0.998055	0.974714
	1.5	38.685483	8.800999	0.998922	0.998641	0.919797
	2.0	37.033021	12.875918	0.998419	0.993854	0.891040

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	31.114467	50.307601	0.993840	0.989221	0.742274
	3.0	29.370426	75.168820	0.990794	0.969248	0.676173
Colomtn	0.5	45.203325	1.962220	0.999846	0.999846	0.973598
	1.0	43.393759	2.976490	0.999767	0.997948	0.965955
	1.5	37.858356	10.647429	0.999168	0.997974	0.923934
	2.0	36.640347	14.094369	0.998899	0.993635	0.907655
	2.5	30.676850	55.641047	0.995660	0.984576	0.824210
	3.0	29.471314	73.442746	0.994268	0.967571	0.793871
Desert	0.5	40.323197	6.036158	0.999055	0.993480	0.983388
	1.0	39.293657	7.650937	0.998812	0.991600	0.980269
	1.5	33.355740	30.026611	0.995383	0.977394	0.930877
	2.0	32.997537	32.608184	0.995018	0.974626	0.926066
	2.5	25.899586	1.6715589	0.974735	0.910103	0.810229
	3.0	25.575343	180.113403	0.972961	0.901899	0.800131
Average case	0.5	45.627015	2.008608	0.999797	0.997944	0.954610
	1.0	43.410815	3.157996	0.999695	0.995635	0.943637
	1.5	38.072630	11.100270	0.998912	0.993667	0.884485
	2.0	36.685610	14.735890	0.998616	0.988801	0.863681
	2.5	30.094760	62.431380	0.992793	0.968771	0.743618
	3.0	28.796720	91.593310	0.991075	0.951296	0.705607

Average PSNRs of both WDHT_2x2 and WDHT_2x2_QE are compared between themselves with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. Table 7.2 reveals the maximum and minimum values of average PSNR for WDHT_2x2 scheme as 45.62 dB and 27.82 dB respectively. Similarly, the maximum value of average PSNR for WDHT_2x2_QE scheme is 45.62 dB however, the minimum average PSNR becomes 28.79 dB. Therefore, the average PSNR for both schemes are identical up to 1 bpB and as the payload increases (up to 3 bpB), the value of average PSNR (dB) for WDHT_2x2_QE increases over the former one.

Table 7.2. Comparative analysis of obtained average PSNR values between WDHT_2x2 and WDHT_2x2_QE with respect to increasing payload

WDHT_2x2		WDHT_2x2_QE	
Payload (bpB)	PSNR (dB)	Payload (bpB)	PSNR (dB)
0.5	45.627015	0.5	45.627015
1.0	43.410815	1.0	43.410815
1.5	37.527065	1.5	38.072630

WDHT_2x2		WDHT_2x2_QE	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
2.0	35.680360	2.0	36.685610
2.5	29.564705	2.5	30.094760
3.0	27.828435	3.0	28.796720

The WDHT_2x2_QE scheme is compared against the Varsaki et al.’s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] in terms of PSNR (dB) and payload (bpB). Fig. 7.1 depicts an analysis of PSNR values for “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” respectively. The PSNR analysis of Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes is made with respect to 0.25 and 1 bpB of payloads respectively. The major limitation of the DPTHDI [88] and DGTDHS [129] is their fixed as well as low payload. Unlikely, the WDHT_2x2_QE is focused on variable payload that offers a spread from 0.5 to 3 bpB by retaining the PSNR value around 30 dB or more [148]. In contrast to DPTHDI [88], WDHT_2x2_QE ensured equal or higher PSNR (dB) at 0.5 and 1 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB for “Baboon”, 0.5 and 1 bpB of payloads for “Pepper”, 0.5 and 1 bpB of payloads for “Airplane” and 0.5, 1, 1.5 and 2 bpB of payloads for “Sailboat” respectively. Compared to DGTDHS [129], the PSNR values are lacking at 1 bpB for most of the images however, the WDHT_2x2_QE provides the payload variation [0.5 – 3 bpB].

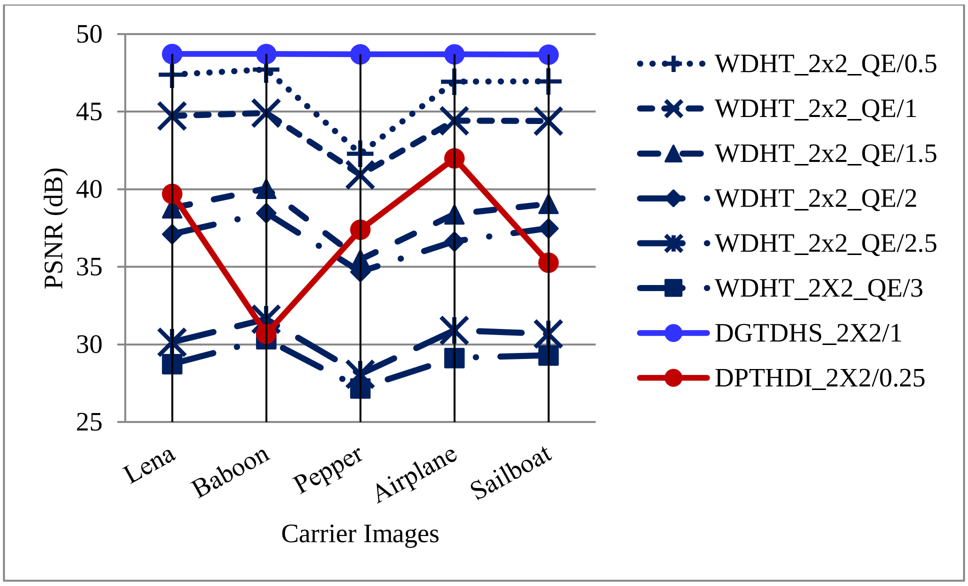


Fig. 7.1. Performance analysis of PSNR (dB) for variable payload based WDHT_2x2_QE and fixed payload based Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 7.2 illustrates the investigation of image quality for WDHT_2x2_QE, WDHT_2x2, DPTHDI [88] and DGTDHS [129] respectively. The payload of DPTHDI [88] and DGTDHS [129] are 0.25 and 1 bpB respectively however, the WDHT_2x2_QE and WDHT_2x2 offered payload values in the range [0.5 – 3 bpB]. Since, the peak signal to noise ratio (PSNR) is treated as the most prominent metric to evaluate the image quality, the average PSNR values as computed from twenty color images (fig. 1.1) at specified payload values have been considered for comparison purpose. In contrast to WDHT_2x2, the improvement of average PSNR in WDHT_2x2_QE can be visually perceived as the payload increases from 1 bpB. The numerical analysis ensured that the average PSNR is greater than or equal to 30 dB for the payload variation of 0.5 to 2.5 bpB which results perceptible image quality for the obtained watermarked images [148]. However, the average PSNR is slightly drifted from the acceptable level of PSNR value (i.e., < 30 dB) at 3 bpB. The average PSNR value for DPTHDI [88] is 37.40 dB that has been computed from “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images. In contrast to DPTHDI [88], the average PSNR value of WDHT_2x2_QE ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads. The average PSNR value for DGTDHS [129] is 48.70 dB that has been computed by taking the average of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” images. In contrast to fixed payload based DGTDHS [129], proposed WDHT_2x2_QE offered minor loss of average PSNR at 1 bpB however, the suitability of WDHT_2x2_QE scheme is its ability of providing variable payload for the spread of 0.5 to 3 bpB.

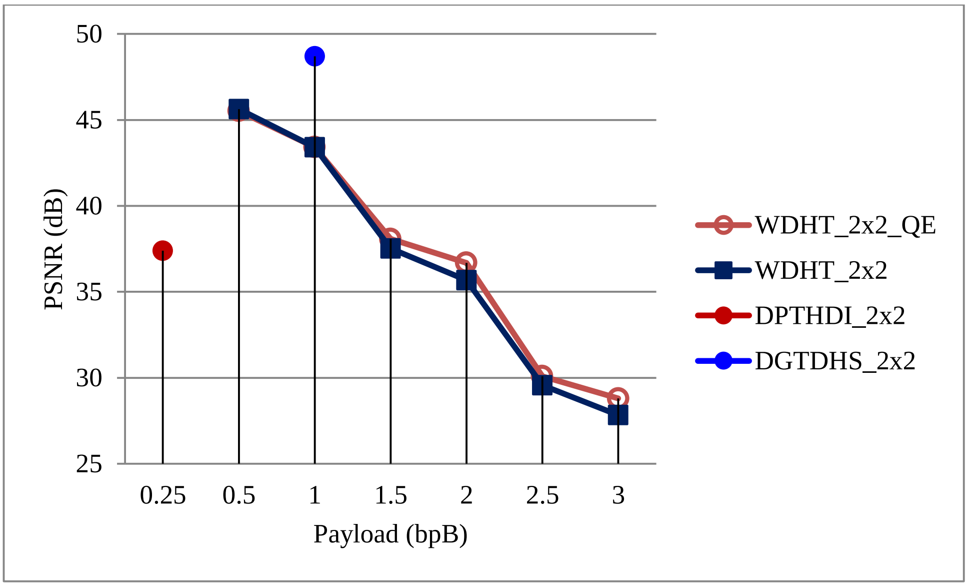


Fig. 7.2. Graphical representation of variation of average PSNR (dB) with respect to payload for WDHT_2x2_QE, WDHT_2x2 and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

7.3.1.2 Quality Enhancement for 1 x 2 Block based Watermark Fabrication

The WDHT_1x2_QE is the extension of WDHT_1x2 scheme of section 2.2.2 followed by quality enhancement of section 7.2.1. The carrier image is decomposed into 2 x 2 non-overlapping blocks where, the pixel components are adjusted prior to embedding as discussed in equation (2.17) of section 2.2.2.1. One dimensional Discrete Hartley Transform (1D-DHT) converts each 1 x 2 sub-matrix of pixel components into transform domain. The message digest, size and content of the watermark constitutes the secret bit-stream from which the subsequent bits are fabricated into the transformed components starting from the first bit position of the least significant part (i.e., LSB-1) toward higher order bit position. The subsequent difference between the pre-embedded transformed component and the embedded component has been minimized by introducing the adaptive quality enhancement of section 7.2.1. The quality enhanced component corresponding to each embedded component preserved the fabricated bits as well as the unaltered least significant bit (i.e., LSB-0). One dimensional inverse Discrete Hartley Transform (1D-DHT) is applied over 1 x 2 sub-matrices of quality enhanced components to re-compute the pixel components. The process is repeated till the embedding is completed and the resultant watermarked image is produced.

To illustrate the process, an example has been given in section 7.3.1.2.1 and that of results and discussions are elaborated in section 7.3.1.2.2.

7.3.1.2.1 Example

Divide the color cover image into 1 x 2 sub-matrices of pixel components where, each pixel component is adjusted prior to embedding at 3 bpB of payload as discussed in equation (2.17) of section 2.2.2.1. The 1 x 2 adjusted sub-matrices of red, green and blue channels become:

$$R_1 = [240\ 69] \quad G_1 = [92\ 202] \quad B_1 = [16\ 110]$$

Each 1 x 2 sub-matrix or pair of pixel components is converted into transform domain based on one dimensional Discrete Hartley Transform (1D-DHT). The 1 x 2 sub-matrices of transformed components such as T(R$_1$), T(G$_1$) and T(B$_1$) are obtained as follows:

$$T(R_1) = [309\ 171] \quad T(G_1) = [294\ -110] \quad T(B_1) = [126\ -94]$$

In this example, three bits in succession from the secret bit-stream *"101000010110000011"* are retrieved and are allotted for fabrication with 3: 3 bit embedding ratio (i.e., $\lambda_1 = 3$, $\lambda_2 = 3$) as derived from equation (2.18) of section 2.2.2.1. The bit replacement policy for each transformed component is starting from LSB-1 toward the higher order bit position. The 1 x 2 sub-matrices of embedded components are computed as follows:

$$T'(R_1) = [315\ 161] \quad T'(G_1) = [292\ -102] \quad T'(B_1) = [112\ -92]$$

The adaptive quality enhancement has been used to minimize the difference between the embedded component and the original component without losing the embedded bits. Hence, the quality enhanced component corresponding to each embedded component is obtained by taking the closest value of the pre-embedded component without hampering the least significant bit (LSB-0) along with three embedded bits (LSB-1, LSB-2 and LSB-3). Therefore, the 1 x 2 sub-matrices of quality enhanced components become:

$$T''(R_1) = [315\ 177] \quad T''(G_1) = [292\ -102] \quad T''(B_1) = [128\ -92]$$

On application of one dimensional inverse Discrete Hartley Transform (1D-IDHT) over 1 x 2 sub-matrices of quality enhanced components, the 1 x 2 sub-matrices of pixel components are obtained as follows:

$$R'_1 = [246\ 69] \ G'_1 = [95\ 197] \ B'_1 = [18\ 110]$$

7.3.1.2.2 Results and Discussions

The WDHT_1x2_QE is an extended version of WDHT_1x2 of section 2.2.2 which is effective in terms of watermarked image’s quality by incorporating the adaptive quality enhancement of section 7.2.1. The quality of the WDHT_1x2_QE is evaluated based on the following metrics: peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ). The results are computed for twenty benchmark (BMP) images [130, 131] of dimension 512 x 512 along with the varying sizes of the fabricated watermark as given in fig. 1.1 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. In WDHT_1x2_QE scheme, the maximum PSNR obtained is 51.07 dB at 0.5 bpB of payload for the “San Diego” image and that of the minimum value obtained is 27.85 dB at 3 bpB of payload for the “Desert” image. The low PSNR of “Desert” image is not a concern for this analysis since the average PSNR of WDHT_1x2_GAO is more than 30 dB [148]. The highest MSE obtained is 106.48 for “Desert” at 3 bpB of payload and that of the lowest MSE obtained is 0.5 for “San Diego” at 0.5 bpB of payload. The minimum values of other quality metrics such as the IF, SSIM and UIQ obtained at 3 bpB as 0.980144 (Desert), 0.935236 (Pepper) and 0.551902 (Splash) respectively; however, the maximum values are obtained at 0.5 bpB of payload and the corresponding values are 0.999984 (Airplane), 0.999851 (Baboon) and 0.999658 (Baboon) respectively. The computed values of IF, SSIM and UIQ are falling between 0 and 1; however, these values closed to one indicates a high similarity between the watermarked image and the original image. The average results at variable payload are computed to summarize the experimental results.

Table 7.3. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WDHT_1x2_QE scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	50.968356	0.520289	0.999967	0.999503	0.989080
	1.0	45.815900	1.704081	0.999879	0.998246	0.972989
	1.5	43.529893	2.884635	0.999829	0.997961	0.944514
	2.0	38.826281	8.520248	0.999387	0.992291	0.896768
	2.5	36.317638	15.181563	0.999144	0.990936	0.830486
	3.0	31.326526	47.910174	0.996566	0.970069	0.742341
Baboon	0.5	51.032210	0.512695	0.999971	0.999851	0.997658
	1.0	46.214702	1.554569	0.999911	0.999447	0.992896
	1.5	44.295230	2.4185625	0.999869	0.999293	0.987897
	2.0	39.574383	7.172028	0.999588	0.997260	0.973811
	2.5	37.475241	11.629370	0.999379	0.995954	0.954446
	3.0	31.974904	41.265764	0.997616	0.985706	0.916355
Pepper	0.5	48.386133	0.942902	0.999872	0.994562	0.984568
	1.0	43.630430	2.818625	0.999597	0.990957	0.966690
	1.5	42.080722	4.027245	0.999540	0.986875	0.944240
	2.0	36.687650	13.941685	0.998030	0.974966	0.893915
	2.5	35.083257	20.172261	0.997705	0.969448	0.845095
	3.0	29.274593	76.845951	0.989009	0.935236	0.752587
Airplane	0.5	50.821815	0.538144	0.999984	0.999355	0.969061
	1.0	45.066844	2.024864	0.999944	0.997828	0.922124
	1.5	43.050041	3.221633	0.999908	0.997507	0.8762239
	2.0	37.952333	10.419502	0.999714	0.990280	0.804984
	2.5	35.766863	17.234312	0.999512	0.989284	0.716152
	3.0	30.711302	55.201405	0.998486	0.966727	0.641431
Sailboat	0.5	50.898925	0.528673	0.999972	0.999564	0.989218
	1.0	45.538552	1.816457	0.999904	0.998476	0.966503
	1.5	43.724751	2.758069	0.999857	0.998274	0.955486
	2.0	38.668525	8.835432	0.999539	0.993137	0.916194

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	36.817881	13.529824	0.999298	0.992322	0.880739
	3.0	31.049396	51.067038	0.997333	0.971277	0.808534
Earth	0.5	51.016963	0.514498	0.999976	0.999637	0.994575
	1.0	45.862307	1.685969	0.999922	0.998655	0.983650
	1.5	43.817561	2.699754	0.999861	0.998493	0.971339
	2.0	38.841802	8.489851	0.999595	0.994084	0.933457
	2.5	36.718105	13.844261	0.999305	0.993475	0.891605
	3.0	31.353360	47.615061	0.997753	0.975969	0.811530
San Diego	0.5	51.071997	0.508019	0.999983	0.999790	0.998316
	1.0	46.166524	1.571910	0.999951	0.999315	0.994495
	1.5	44.285808	2.423815	0.999917	0.999251	0.992230
	2.0	39.468401	7.349202	0.999768	0.997073	0.977550
	2.5	37.666673	11.127894	0.999625	0.996804	0.969853
	3.0	32.308524	38.214472	0.998811	0.988297	0.926034
Splash	0.5	50.681726	0.555786	0.999953	0.999244	0.955520
	1.0	44.948857	2.080628	0.999789	0.997264	0.901751
	1.5	42.386343	3.753583	0.999691	0.988904	0.823766
	2.0	37.695630	11.053946	0.998861	0.980026	0.734655
	2.5	35.096524	20.110753	0.998388	0.973705	0.643004
	3.0	30.301658	60.661694	0.993835	0.943631	0.551902
Oakland	0.5	49.694129	0.697697	0.999960	0.999125	0.995369
	1.0	45.348813	1.897575	0.999916	0.998053	0.983627
	1.5	42.969328	3.282067	0.999815	0.997420	0.981625
	2.0	38.403758	9.390842	0.999600	0.991983	0.938134
	2.5	36.125213	15.869345	0.999112	0.990047	0.930570
	3.0	31.067714	50.852105	0.997910	0.969195	0.834066
Foster City	0.5	50.983241	0.518508	0.999983	0.999153	0.983734
	1.0	45.230526	1.949968	0.999944	0.997176	0.939958
	1.5	43.549495	2.871645	0.999902	0.996831	0.923289
	2.0	38.197154	9.848383	0.999725	0.988608	0.843212
	2.5	36.439924	14.760052	0.999498	0.986881	0.788444
	3.0	31.005595	51.584691	0.998576	0.960961	0.692949
Anhinga	0.5	48.873701	0.842772	0.999932	0.999760	0.929419
	1.0	43.727902	2.756069	0.999775	0.999151	0.903596
	1.5	41.507642	4.595324	0.999639	0.998648	0.831921
	2.0	37.380246	11.886545	0.999034	0.994949	0.800808

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	35.087926	20.150586	0.998417	0.993323	0.768152
	3.0	30.415425	59.093247	0.995190	0.976141	0.716037
Athens	0.5	48.885059	0.840571	0.999933	0.999815	0.974626
	1.0	44.403227	2.359161	0.999810	0.999307	0.947881
	1.5	42.475237	3.677533	0.999711	0.998888	0.910804
	2.0	38.197373	9.847885	0.999202	0.995178	0.870467
	2.5	36.019211	16.261446	0.998725	0.994170	0.810971
	3.0	30.859846	53.345239	0.995676	0.976619	0.744819
Bardowl	0.5	48.212958	0.981260	0.999897	0.999280	0.996669
	1.0	43.459361	2.931866	0.999689	0.998342	0.986964
	1.5	42.140463	3.972226	0.999590	0.998175	0.983367
	2.0	36.525319	14.472660	0.998461	0.988627	0.948272
	2.5	35.153307	19.849503	0.997938	0.987552	0.931102
	3.0	28.605903	89.637557	0.990446	0.952328	0.867475
Barnfall	0.5	50.558089	0.571835	0.999880	0.999616	0.996691
	1.0	44.750610	2.177806	0.999497	0.998444	0.986352
	1.5	42.871489	3.356845	0.999369	0.998369	0.979694
	2.0	38.164220	9.923350	0.997760	0.993033	0.948464
	2.5	36.194051	15.619789	0.997059	0.992207	0.922958
	3.0	30.841535	53.570632	0.987740	0.965147	0.840834
Butrfly	0.5	49.557670	0.719968	0.999941	0.999787	0.992709
	1.0	44.932340	2.088556	0.999822	0.999277	0.981754
	1.5	43.063078	3.211977	0.999757	0.999038	0.972430
	2.0	38.678093	8.815989	0.999246	0.996273	0.943050
	2.5	36.536725	14.43470	0.998905	0.995638	0.907983
	3.0	31.200530	49.320477	0.995738	0.982935	0.835638
Bobcat	0.5	49.193921	0.782867	0.999873	0.999582	0.832578
	1.0	44.432656	2.343228	0.999598	0.998606	0.820210
	1.5	42.536800	3.625770	0.999459	0.998079	0.717665
	2.0	38.049887	10.188063	0.998243	0.993676	0.697405
	2.5	35.816116	17.039966	0.997442	0.991670	0.668813
	3.0	28.881376	84.128412	0.985070	0.947042	0.608178
Bodie	0.5	49.352944	0.754720	0.999857	0.999536	0.982208
	1.0	43.999445	2.589022	0.999481	0.998389	0.972031
	1.5	42.281920	3.844928	0.999314	0.998266	0.957327
	2.0	37.305675	12.092409	0.997576	0.990951	0.932581

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	35.571472	18.027399	0.996745	0.989113	0.909863
	3.0	29.021932	81.449250	0.983353	0.942077	0.826238
Bluheron	0.5	49.534888	0.723754	0.999910	0.999656	0.992862
	1.0	44.462562	2.327148	0.999717	0.998745	0.978550
	1.5	42.525808	3.634958	0.999550	0.998581	0.965447
	2.0	37.963862	10.391880	0.998741	0.994146	0.925097
	2.5	35.750060	17.301123	0.997860	0.993646	0.874518
	3.0	30.730613	54.956493	0.993338	0.977484	0.782837
Colomtn	0.5	50.031172	0.645599	0.999950	0.999658	0.999658
	1.0	44.195509	2.474739	0.999811	0.998761	0.976359
	1.5	42.262694	3.861988	0.999701	0.998550	0.957797
	2.0	37.775643	10.852156	0.999172	0.994569	0.932197
	2.5	35.876017	16.806550	0.998702	0.993579	0.899350
	3.0	30.659955	55.857925	0.995746	0.971872	0.835726
Desert	0.5	45.674444	1.760499	0.999694	0.997437	0.994520
	1.0	41.263075	4.861527	0.999099	0.995495	0.987850
	1.5	39.716020	6.941899	0.998888	0.994216	0.980367
	2.0	34.835511	21.356456	0.996021	0.983298	0.960422
	2.5	33.069695	32.070873	0.994894	0.977302	0.933795
	3.0	27.857943	106.484395	0.980144	0.941597	0.876123
Average case	0.5	49.77152	0.723053	0.999924	0.999196	0.977452
	1.0	44.67251	2.300688	0.999753	0.997997	0.958312
	1.5	42.75352	3.553223	0.999658	0.997081	0.932871
	2.0	37.95959	10.742430	0.998863	0.991220	0.893572
	2.5	35.92909	17.051080	0.998383	0.989353	0.853895
	3.0	30.47243	60.453100	0.993417	0.965016	0.780582

The quality of WDHT_1x2 and WDHT_1x2_QE is analyzed in terms of average PSNR (dB) with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. The average PSNR for WDHT_1x2 scheme lie between 28.62 dB and 49.77 dB. In contrast, the average PSNR for WDHT_1x2_QE scheme ranges from 30.47 dB to 49.77 dB respectively. On application of adaptive quality enhancement, the average PSNR of latter scheme is improved over the former one while the payload exceeds 0.5 bpB of payload. The average PSNR of WDHT_1x2_QE retains acceptable values of average PSNR (i.e., ≥ 30 dB) for the payload range [0.5 – 3 bpB]. Thus, the quality of the watermarked images in average case is treated as well perceptible [148].

Table 7.4. Comparative analysis of obtained average PSNR values between WDHT_1x2 and WDHT_1x2_QE with respect to increasing payload

WDHT_1x2		**WDHT_1x2_QE**	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	49.77152	0.5	49.77152
1.0	42.93340	1.0	44.67251
1.5	41.31294	1.5	42.75352
2.0	35.96829	2.0	37.95959
2.5	34.38548	2.5	35.92909
3.0	28.62825	3.0	30.47243

In fig. 7.3, the comparative analysis among WDHT_1x2_QE scheme and Varsaki et al.'s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] as well as Discrete Gould Transform based data hiding scheme (DGTDHS) [129] is performed on the basis of PSNR (dB) and payload (bpB). Both DPTHDI [88] and DGTDHS [129] offered fixed/low payload of 0.25 bpB and 1 bpB respectively. In contrast to DPTHDI [88], WDHT_1x2_QE scheme ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for "Lena", 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Baboon", 0.5, 1 and 1.5 bpB of payloads for "Pepper", 0.5, 1 and 1.5 bpB of payloads for "Airplane" and 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for "Sailboat" respectively. Unlike to DGTDHS [129], the WDHT_1x2_QE ensured slight loss in PSNR values at 1 bpB however, the variable payload (0.5 – 3 bpB) is the significant achievement.

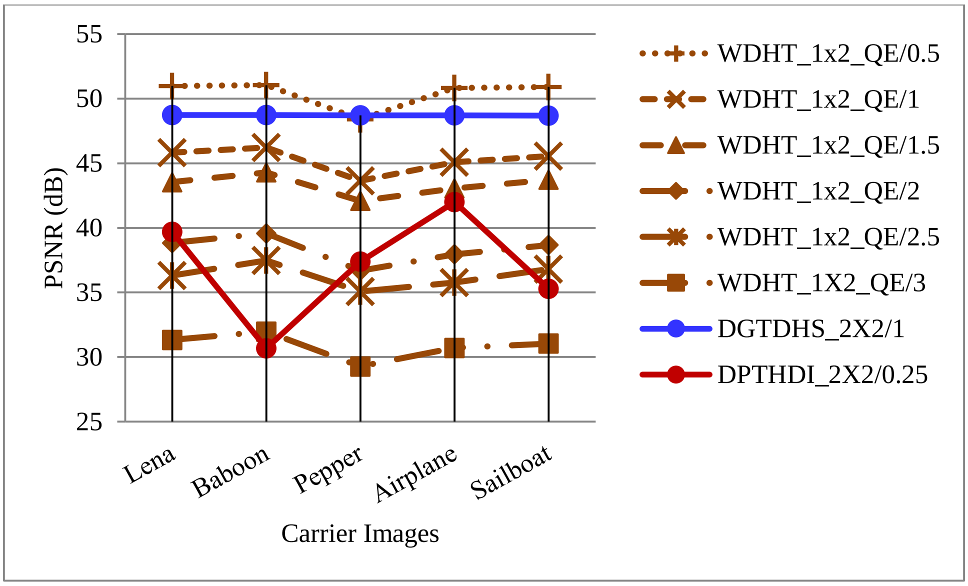


Fig. 7.3. Performance analysis of PSNR (dB) for variable payload based WDHT_1x2_QE and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 7.4 illustrates the variation of average PSNR values for WDHT_1x2_QE, WDHT_1x2, WDHT_2x2_QE, DPTHDI [88] and DGTDHS [129] respectively. In contrast to WDHT_1x2 and WDHT_2x2_QE, the WDHT_1x2_QE ensured higher average PSNR values while the payload increases from 0.5 bpB. It is also to be noted that the average PSNR computed at 3 bpB is above 30 dB which generates high-quality watermarked images [148]. The average of PSNR values for “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images in DPTHDI [88] is 37.40 dB at 0.25 bpB of payload. In comparison with DPTHDI [88], the average PSNR of WDHT_1x2_QE ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads. Again, the average of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat”, “F16” images in DGTDHS [129] is 48.70 dB at 1 bpB of payload. Compared to DGTDHS [129], the WDHT_1x2_QE lacking in terms of average PSNR at 1 bpB however, the payload values with variation 0.5 to 3 bpB is the key issue.

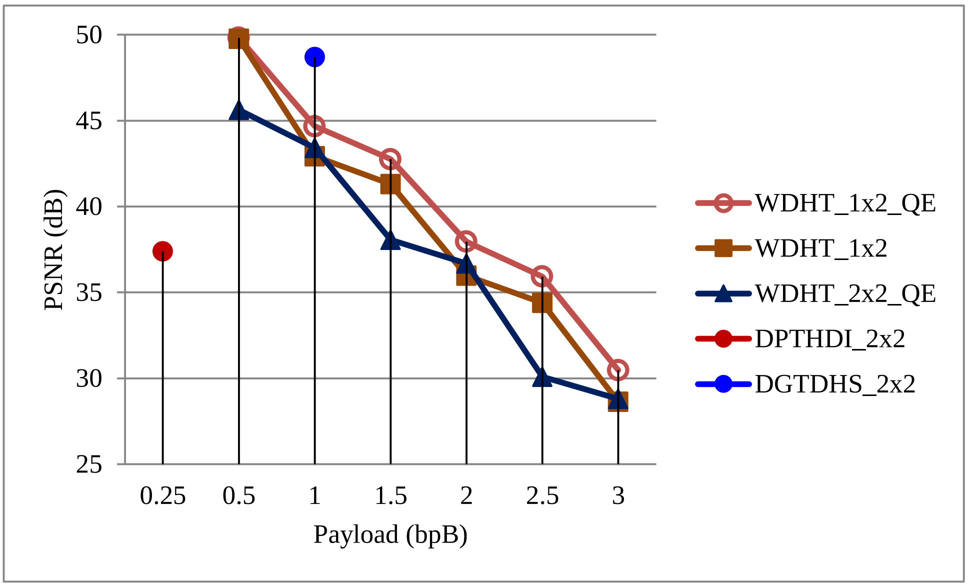


Fig. 7.4. Graphical representation of variation of average PSNR (dB) with respect to payload for WDHT_1x2_QE, WDHT_1x2, WDHT_2x2_QE and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

7.3.2 Quality Enhancement of Legendre Transform (LT) based watermarking

As per proposal of the thesis, Legendre Transform (LT) based watermarking has been divided into two classes: one consisting of 2 x 2 block based watermark fabrication using Legendre Transform (LT) (WLT_2x2) and the other is 1 x 2 block based watermark fabrication (WLT_1x2). Detailed discussions regarding both schemes are available in chapter 3.

Legendre Transform (LT) followed by adaptive quality enhancement of section 7.2.1 (WLT_2x2_QE) has been elaborated in section 7.3.2.1 while section 7.3.2.2 deals with the results, analysis and discussions on the 1 x 2 block based watermark fabrication using Legendre Transform (LT) followed by adaptive quality enhancement (WLT_1x2_QE).

7.3.2.1 Quality Enhancement for 2 x 2 Block based Watermark Fabrication

The WLT_2x2_QE works by decomposing the carrier image into 2 x 2 non-overlapping blocks of pixel components in row major order. The pixel components are adjusted prior to embedding as discussed in equation (3.7) of section 3.2.1.1. Each 2 x 2 block is converted into transform domain based on Legendre Transform (LT). Secret bits corresponding to the message digest, size and content of the watermark are fabricated on first/second/third transformed component in varying proportion starting from the first bit position of the least significant part (i.e., LSB-1) toward higher order bit position. The adaptive quality enhancement of section 7.2.1 is applied on each embedded component in transform domain to find the quality enhanced component. An additional re-adjustment operation has also been performed to avoid overflow/underflow. Inverse Legendre Transform (ILT) is applied on each block of enhanced components to obtain the pixel components. This process is repeated until and unless the entire bit-stream corresponding to the message digest (MD), watermark size and the content is embedded.

The different phases of embedding are explained with a suitable example as given in section 7.3.2.1.1. Experimental results, comparative analysis and discussions have been elaborated in section 7.3.2.1.2.

7.3.2.1.1 Example

Consider the 2 x 2 sub-matrices of pixel components corresponding to red, green and blue channels to fabricate the secret bit-stream. The pixel adjustment process is applied over the pixel components to adjust the values for the payload value of 3 bpB prior to embedding as discussed in equation (3.7) of section 3.2.1.1. The 2 x 2 sub-matrices of pixel components followed by quality enhancement are as follows:

$$R_1 = \begin{bmatrix} 224 & 69 \\ 32 & 112 \end{bmatrix} G_1 = \begin{bmatrix} 92 & 202 \\ 32 & 51 \end{bmatrix} B_1 = \begin{bmatrix} 32 & 119 \\ 220 & 224 \end{bmatrix}$$

Legendre Transform (LT) is applied on each 2 x 2 sub-matrix of pixel components to obtain the transformed components. The 2 x 2 sub-matrices of transformed components such as T(R1), T(G1) and T(B1) are obtained as follows:

$$T(R_1) = \begin{bmatrix} 224 & 362 \\ 830 & 4252 \end{bmatrix} T(G_1) = \begin{bmatrix} 92 & 496 \\ 1496 & 4496 \end{bmatrix} T(B_1) = \begin{bmatrix} 32 & 270 \\ 2066 & 12540 \end{bmatrix}$$

Suppose, the watermark bit stream *"1010000101100000110011110110110000001"* is to be fabricated into the transformed components. In this example, the payload value is 3 bits per Byte (bpB) which ensured the fabrication of four bits ($\lambda_1 = 4$, $\lambda_2 = 4$, $\lambda_3 = 4$) into first, second and third transformed components for each 2 x 2 sub-matrix starting from LSB-1 toward higher order bit position by not affecting the LSB-0. The 2 x 2 sub-matrices of embedded components become:

$$T'(R_1) = \begin{bmatrix} 234 & 368 \\ 812 & 4252 \end{bmatrix} T'(G_1) = \begin{bmatrix} 64 & 486 \\ 1502 & 4496 \end{bmatrix} T'(B_1) = \begin{bmatrix} 44 & 262 \\ 2064 & 12540 \end{bmatrix}$$

On application of adaptive quality enhancement, the difference between the embedded component and the original component has been minimized since each enhanced component is nothing but the closest value of the pre-embedded component that preserved the least significant bit along with four fabricated bits of the embedded component (LSB-0, LSB-1, LSB-2, LSB-3 and LSB-4). The 2 x 2 sub-matrices of quality enhanced components corresponding to RGB color channels become:

$$T''(R_1) = \begin{bmatrix} 234 & 368 \\ 844 & 4252 \end{bmatrix} T''(G_1) = \begin{bmatrix} 96 & 486 \\ 1502 & 4496 \end{bmatrix} T'(B_1) = \begin{bmatrix} 44 & 262 \\ 2064 & 12540 \end{bmatrix}$$

Inverse Legendre Transform (ILT) is applied on each 2 x 2 sub-matrices of quality enhanced components to obtain 2 x 2 sub-matrices of pixel components. The redefined sub-matrices namely R'$_1$, G'$_1$ and B'$_1$ corresponding to red, green and blue channels are obtained as follows:

$$R'_1 = \begin{bmatrix} 234 & 67 \\ 40 & 108 \end{bmatrix} G'_1 = \begin{bmatrix} 96 & 195 \\ 50 & 44 \end{bmatrix} B'_1 = \begin{bmatrix} 44 & 109 \\ 233 & 217 \end{bmatrix}$$

In this example, no additional re-adjustment operation has been performed since the pixel components are non-fractional, non-negative and less than or equal to the maximum possible value of a pixel component i.e., 255.

7.3.2.1.2 Results and Discussions

The image quality performance of WLT_2X2_QE is analyzed with respect to the payload values in the range of 0.5 to 3 bpB. Standard metrics such as peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) are computed to measure the image quality. Twenty benchmark (BMP) images [130, 131] of dimension 512 x 512 and the varying sizes of the fabricated watermark as given in fig. 1.1 has been considered to compute results as given in table 7.5. For this experiment, the minimum PSNR is 25.16 dB at 3 bpB of payload for the

"Desert" image and the maximum PSNR is 49.26 dB at 0.5 bpB of payload for the "Bluheron" image. The minimum PSNR obtained for "Desert" implies massive distortion in quality but, still the scheme is effective due to the ability of providing variable payload of up to 3 bpB. The lowest MSE obtained is 0.77 for "Bluheron" at 0.5 bpB and that of highest value is 198.00 for "Desert" at 3 bpB. The computed values of IF, SSIM and UIQ lie between 0 and 1, the closer the IF, SSIM and UIQ to one, watermarked image is more similar to the original one. It is also observed from table 7.5 that the IF, SSIM and UIQ are maximum at 0.5 bpB and minimum at 3 bpB. Maximum values of IF, SSIM and UIQ are 0.999973 (Airplane), 0.999904 (San Diego) and 0.997381 (San Diego) respectively and that of the minimum values are 0.970239 (Desert), 0.884094 (Bobcat) and 0.378854 (Splash) respectively. The IF, SSIM and UIQ values are very close to one even at highest payload. These values ensured perceptible fidelity, high level of pixel's independence and acceptable range of structural distortion respectively. The average results of twenty carrier images at variable payload have been computed to summarize the experimental results.

Table 7.5. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WLT_2x2_QE scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	48.597498	0.898111	0.999943	0.999667	0.979761
	1.0	45.461848	1.848823	0.999884	0.999274	0.960674
	1.5	42.366922	3.770406	0.999762	0.998326	0.928404
	2.0	38.133533	9.993715	0.999378	0.994896	0.855348
	2.5	32.050538	40.553330	0.997591	0.974712	0.708631
	3.0	29.603674	71.238194	0.995652	0.962673	0.589563
Baboon	0.5	48.591863	0.899278	0.999952	0.999877	0.996092
	1.0	45.422723	1.865554	0.999901	0.999673	0.991190
	1.5	42.332744	3.800195	0.999799	0.999340	0.984054
	2.0	38.058933	10.166863	0.999464	0.997094	0.959470
	2.5	32.129419	39.823403	0.997920	0.985461	0.906521
	3.0	29.554769	72.044920	0.996217	0.980960	0.861552
Pepper	0.5	45.895704	1.673053	0.999845	0.990208	0.968212
	1.0	41.654114	4.442924	0.999578	0.983048	0.947976
	1.5	40.138782	6.297991	0.999426	0.982231	0.924593
	2.0	34.738589	21.838429	0.997943	0.966096	0.852798
	2.5	28.584724	90.075763	0.991238	0.929568	0.714734
	3.0	27.329985	120.249092	0.988823	0.918606	0.593899

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	48.538964	0.910298	0.999973	0.999601	0.944582
	1.0	45.460914	1.849221	0.999947	0.999159	0.903693
	1.5	42.357023	3.779009	0.999891	0.998032	0.843717
	2.0	38.292504	9.634515	0.999724	0.994568	0.742013
	2.5	33.830161	26.919330	0.999228	0.984832	0.612846
	3.0	30.119893	63.254442	0.998190	0.968706	0.500685
Sailboat	0.5	48.314324	0.958623	0.999951	0.999706	0.983467
	1.0	45.059465	2.028307	0.999897	0.999331	0.968068
	1.5	42.188704	3.928347	0.999801	0.998555	0.944198
	2.0	37.784009	10.831269	0.999455	0.995138	0.892304
	2.5	31.651378	44.457242	0.997786	0.973640	0.771835
	3.0	29.275852	76.823674	0.996142	0.962899	0.679805
Earth	0.5	48.593479	0.898943	0.999946	0.999748	0.990599
	1.0	45.531802	1.819282	0.999892	0.999467	0.981072
	1.5	42.395549	3.745634	0.999777	0.998736	0.962861
	2.0	38.365636	9.473636	0.999436	0.996325	0.916701
	2.5	32.805467	34.082670	0.997761	0.983373	0.804982
	3.0	30.000514	65.017298	0.995985	0.975126	0.698264
San Diego	0.5	48.604975	0.896567	0.999966	0.999904	0.997381
	1.0	45.537232	1.817008	0.999932	0.999798	0.994801
	1.5	42.400407	3.741447	0.999860	0.999522	0.989966
	2.0	38.391148	9.418148	0.999648	0.998678	0.977723
	2.5	34.012771	25.810904	0.999036	0.996098	0.949934
	3.0	30.392935	59.400054	0.997785	0.992109	0.903949
Splash	0.5	46.648097	1.406923	0.999862	0.989492	0.919325
	1.0	42.745347	3.455776	0.999650	0.985233	0.873533
	1.5	40.845153	5.352602	0.999486	0.983851	0.806064
	2.0	35.927701	16.607725	0.998342	0.974638	0.687380
	2.5	30.104038	63.485788	0.993458	0.946580	0.498525
	3.0	28.621411	89.318055	0.991242	0.932239	0.378854
Oakland	0.5	47.586542	1.133511	0.999933	0.999430	0.994396
	1.0	43.939937	2.624741	0.999838	0.998740	0.988851
	1.5	41.586618	4.512513	0.999738	0.998198	0.979976
	2.0	36.889087	13.309803	0.999187	0.994710	0.9534309
	2.5	31.422535	46.862648	0.996940	0.983730	0.893583
	3.0	29.108551	79.840868	0.995242	0.975589	0.811309

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	48.612614	0.894991	0.999968	0.999545	0.974108
	1.0	45.544356	1.814030	0.999935	0.999029	0.948491
	1.5	42.402652	3.739513	0.999866	0.997719	0.904646
	2.0	38.421540	9.352470	0.999667	0.993734	0.814464
	2.5	34.118873	25.187962	0.999102	0.982691	0.677094
	3.0	30.204657	62.031833	0.997827	0.962109	0.542673
Anhinga	0.5	47.728449	1.097072	0.999915	0.999712	0.8726902
	1.0	43.676971	2.788580	0.999786	0.999186	0.864299
	1.5	41.781123	4.314872	0.999669	0.998974	0.849285
	2.0	36.713638	13.858507	0.998938	0.996171	0.808390
	2.5	31.004667	51.595709	0.996051	0.981003	0.736665
	3.0	29.004890	81.769498	0.993735	0.973588	0.674875
Athens	0.5	48.558146	0.906286	0.999927	0.999750	0.960121
	1.0	46.706752	1.388050	0.999889	0.999463	0.951222
	1.5	43.721007	2.760448	0.999779	0.999275	0.932109
	2.0	39.141304	7.924098	0.999367	0.997147	0.873220
	2.5	34.227177	24.567592	0.998033	0.986351	0.780572
	3.0	30.634331	56.188471	0.995503	0.977878	0.670921
Bardowl	0.5	47.339213	1.199938	0.999878	0.999446	0.995658
	1.0	44.047008	2.560822	0.999737	0.998381	0.989511
	1.5	42.024657	4.079572	0.999586	0.998223	0.984636
	2.0	36.330195	15.137732	0.998440	0.987718	0.943151
	2.5	28.987786	82.092163	0.991447	0.942406	0.847843
	3.0	27.814774	107.548152	0.988939	0.940440	0.804168
Barnfall	0.5	48.949717	0.828149	0.999866	0.999781	0.994941
	1.0	45.610890	1.786451	0.999709	0.999438	0.990437
	1.5	42.279819	3.846789	0.999382	0.998924	0.979759
	2.0	38.193079	9.857626	0.998365	0.995661	0.954838
	2.5	32.518997	36.406641	0.993441	0.976404	0.868526
	3.0	29.748186	68.906745	0.988457	0.968712	0.778153
Butrfly	0.5	48.670968	0.883046	0.999937	0.999787	0.990622
	1.0	46.305868	1.522276	0.999891	0.999527	0.984927
	1.5	43.381585	2.984845	0.999787	0.999280	0.974388
	2.0	38.894548	8.387364	0.999403	0.997599	0.939452
	2.5	34.104753	25.269986	0.998158	0.989601	0.868090
	3.0	30.600036	56.633925	0.995940	0.982353	0.764531

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	48.596957	0.898223	0.999873	0.999582	0.743136
	1.0	46.486256	1.460342	0.999793	0.999157	0.737645
	1.5	43.562549	2.863026	0.999597	0.998721	0.726672
	2.0	38.819152	8.534245	0.998795	0.995076	0.696673
	2.5	26.754760	137.279054	0.980559	0.887367	0.566266
	3.0	26.263812	153.708932	0.978274	0.884094	0.529917
Bodie	0.5	46.899420	1.327817	0.999766	0.999528	0.972632
	1.0	43.354855	3.003273	0.999469	0.998529	0.964690
	1.5	41.190815	4.943093	0.999144	0.998180	0.954904
	2.0	35.458097	18.504212	0.996690	0.984820	0.912549
	2.5	27.813281	1.075851	0.980229	0.907274	0.778247
	3.0	26.813373	135.438788	0.975706	0.904522	0.727557
Bluheron	0.5	49.264767	0.770200	0.999905	0.999835	0.991207
	1.0	45.663201	1.765062	0.999784	0.999396	0.981450
	1.5	42.021566	4.082476	0.999500	0.998783	0.960256
	2.0	37.896155	10.555160	0.998708	0.996524	0.911468
	2.5	33.382134	29.844675	0.996338	0.985607	0.818467
	3.0	30.321499	60.385191	0.992601	0.976480	0.697682
Colomtn	0.5	48.019783	1.025892	0.999919	0.999758	0.979110
	1.0	44.100237	2.529627	0.999802	0.999390	0.971830
	1.5	41.767339	4.328589	0.999661	0.998886	0.961100
	2.0	37.070747	12.764554	0.999003	0.996276	0.929695
	2.5	31.560186	45.400610	0.996463	0.980771	0.862805
	3.0	29.178797	78.559838	0.993868	0.972995	0.781227
Desert	0.5	44.542082	2.284926	0.999641	0.997288	0.992252
	1.0	39.932415	6.604483	0.998961	0.993369	0.982220
	1.5	38.767726	8.635901	0.998684	0.993181	0.977007
	2.0	32.706036	34.871985	0.994642	0.975195	0.930987
	2.5	25.644833	177.254390	0.973082	0.905930	0.810896
	3.0	25.164132	198.000872	0.970239	0.905150	0.782532
Average case	0.5	47.927680	1.089592	0.999898	0.998582	0.962015
	1.0	44.612110	2.448732	0.999764	0.997429	0.948829
	1.5	41.975640	4.275363	0.9996100	0.996847	0.928430
	2.0	37.311280	13.051100	0.998730	0.991403	0.877603
	2.5	31.335420	52.402290	0.993693	0.964170	0.773853
	3.0	28.987800	87.817940	0.991318	0.955861	0.688606

Average PSNR values are computed as well as compared for WLT_2x2 and WLT_2x2_QE with respect to the payload values in the range [0.5 – 3 bpB]. Table 7.6 reveals the minimum and maximum values of average PSNR as 26.35 and 47.92 dB respectively for WLT_2x2 whereas, the average PSNR of WLT_2x2_QE lie in between 28.98 dB and 47.92 dB respectively. Therefore, the quality enhancement ensured improvement in average PSNR values for WLT_2x2_QE over WLT_2x2 as the payload exceeds 0.5 bpB of payload.

Table 7.6. Comparative analysis of obtained average PSNR values between WLT_2x2 and WLT_2x2_QE with respect to increasing payload

WLT_2x2		**WLT_2x2_QE**	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	47.92768	0.5	47.92768
1.0	43.95488	1.0	44.61211
1.5	39.38779	1.5	41.97564
2.0	35.95718	2.0	37.31128
2.5	29.59656	2.5	31.33542
3.0	26.35943	3.0	28.98780

Fig. 7.5 depicts the average PSNR analysis among the proposed 2 x 2 block based watermarking (WLT_2X2_QE), Varsaki et al.’s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] as well as Discrete Gould Transform based data hiding scheme (DGTDHS) [129] with respect to payload (bpB). Five benchmark color images [130, 131] viz. “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” are taken as the cover and the varying sizes of the “Gold-Coin” have been considered as the secret watermark as given in fig. 1.1. The PSNR of the watermarked images for DPTHDI [88] and DGTDHS [129] are computed with respect to 0.25 and 1 bpB of payloads respectively. The major limitation of the DPTHDI [88] and DGTDHS [129] are their fixed as well as low fabrication density. Unlikely, WLT_2X2_QE is focused on variable payload that offers a spread from 0.5 to 3 bpB by retaining the PSNR value around 30 dB or more [148]. In contrast to DPTHDI [88], proposed WLT_2X2_QE ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2 and 2.5 bpB for “Baboon”, 0.5, 1 and 1.5 bpB of payloads for “Pepper”, 0.5, 1 and 1.5 bpB of payloads for “Airplane” and 0.5, 1, 1.5 and 2 bpB of payloads for “Sailboat” respectively. Compared to DGTDHS [129], proposed WLT_2x2_QE lower PSNR values at 1 bpB; however, the payload variation in the range 0.5 to 3 bpB is offered with minor loss of PSNR values.

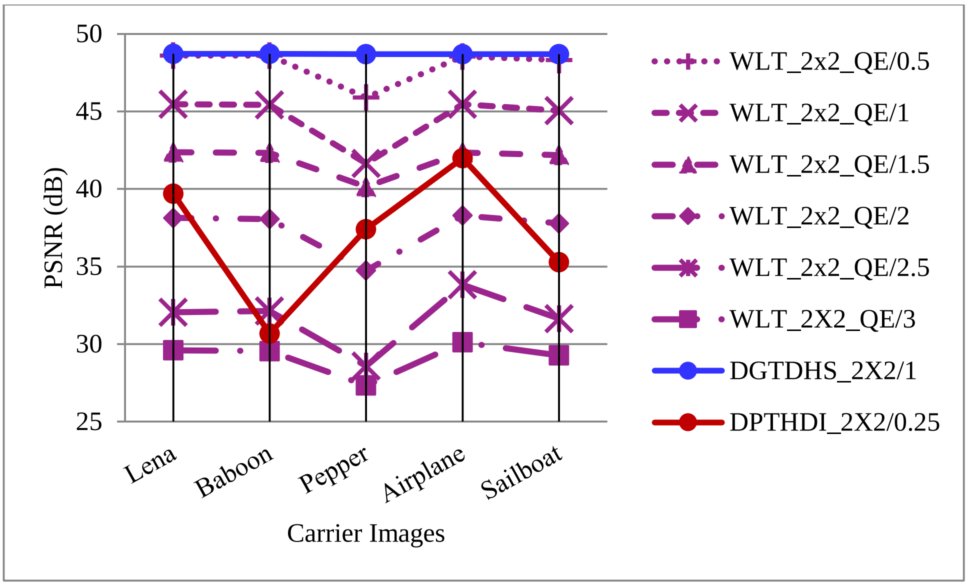


Fig. 7.5. Performance analysis of PSNR (dB) for variable payload based WLT_2x2_QE and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

In fig. 7.6, a comparative analysis is made in terms of average PSNR among WLT_2X2_QE, WLT_2X2, WDHT_2X2_QE, DPTHDI [88] and DGTDHS [129] respectively. It is to be noted that the first three schemes supports variable payload however, the rest two schemes which are proposed by Varsaki et al. dealt with fixed payload values. Unlike to WLT_2x2, the average PSNR of WLT_2x2_QE shows a clear improvement as the payload increases from 0.5 bpB and as a consequence, the perceptible fidelity in the watermarked images (i.e., ≥ 30 dB) is achieved up to 2.5 bpB [148]. In comparison with WDHT_2X2_QE, the WLT_2X2_QE is improved in terms of average PSNR with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. The average PSNR for DPTHDI [88] is 37.40 dB which is obtained by taking the average of PSNR values for "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" images at 0.25 bpB of payload. Compared to DPTHDI [88], the average PSNR of WLT_2x2_QE ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB by taking the average of PSNR values for "Lighthouse", "Elaine", "Lenna", "Boat", "F16" images at 1 bpB of payload. In contrast to fixed payload based DGTDHS [129], proposed WLT_2x2_QE offered slightly low average PSNR at 1 bpB but, still the importance of WLT_2x2_QE exist since the payload has been achieved in the range of 0.5 to 3 bpB. The average PSNR analysis ensured that the decrease in block size from 2 x 2 to 1 x 2 causes enhancement of average PSNR values for the payload range (0.5 – 3 bpB) which leads to massive quality enhancement in the obtained watermarked images.

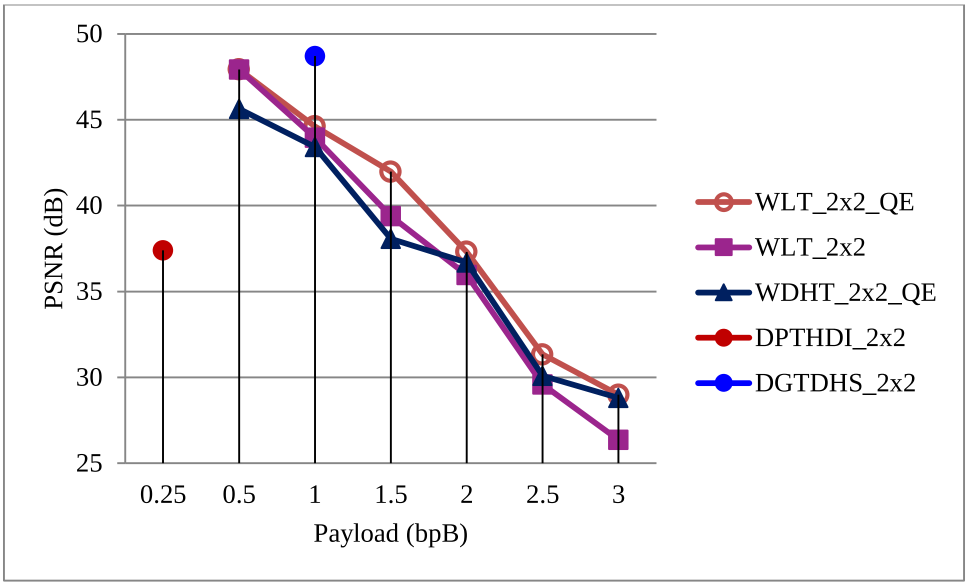


Fig. 7.6. Graphical representation of variation of average PSNR (dB) with respect to payload for WLT_2x2_QE, WLT_2X2, WDHT_2X2_QE and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

7.3.2.2 Quality Enhancement for 1 x 2 Block based Watermark Fabrication

Watermarking in the domain of Legendre Transform (LT) becomes more effective due to the incorporation of the quality enhancement process. The carrier image is partitioned into 1 x 2 non-overlapping blocks where, the pixel components are adjusted prior to embedding based on the pixel adjustment method as discussed in equation (3.17) of section 3.2.2.1. Each 1 x 2 sub-matrices of pixel components are successively converted into transform domain based on the Legendre Transform (LT) to fabricate the secret bits corresponding to the message digest, size and content of the watermark. The fabrication of secret bits is started from the first bit position of the least significant part (i.e., LSB-1) toward higher order bit position of the transformed component. Adaptive quality enhancement scheme of section 7.2.1 has been applied on each embedded component to minimize the deviation between pre-embedded component and the embedded component respectively. The quality enhanced component preserves the least significant bit (i.e., LSB-0) as well as the fabricated bits of the embedded component. Inverse Legendre Transform (ILT) converts 1 x 2 sub-matrices of quality enhanced components into the pixel components. The process is repeated till the secret bits are concealed and the watermarked image is produced.

Section 7.3.2.2.1 explains the watermarking process (WLT_1x2_QE) with a suitable example that describes the different phases of embedding. Simulation results, comparative analysis and detailed discussions are given in section 7.3.2.2.2.

7.3.2.2.1 *Example*

Consider 1 x 2 sub-matrices of pixel components corresponding to red, green and blue channel to fabricate the secret bit-stream. To achieve the payload value of 3 bpB, the pixel components are adjusted prior to embedding as discussed in equation (3.17) of section 3.2.2.1. The 1 x 2 sub-matrices of pixel components are as follows:

$$R_1 = [240\ 69] \quad G_1 = [92\ 202] \quad B_1 = [16\ 110]$$

Legendre Transform (LT) is applied on each 1 x 2 sub-matrix / pair of pixel components to obtain 1 x 2 sub-matrix or pair of transformed components. The 1 x 2 sub-matrices of transformed components corresponding to red, green and blue channels i.e., $T(R_1)$, $T(G_1)$ and $T(B_1)$ are obtained as follows:

$$T(R_1) = [240\ 378] \quad T(G_1) = [92\ 496] \quad T(B_1) = [16\ 236]$$

Secret bit-stream *"101000010110000111"* is to be fabricated into the transformed components based on the embedding rule of equation (3.17) of section 3.2.2.1. For example, three bits are fabricated ($\lambda = \lambda_1 = \lambda_2 = 3$) on transformed components starting from LSB-1 toward the higher order bit position. Hence, the 1 x 2 sub-matrices of embedded components becomes:

$$T'(R_1) = [250\ 368] \quad T'(G_1) = [84\ 502] \quad T'(B_1) = [16\ 238]$$

The adaptive quality enhancement of section 7.2.1 has been used to ensure the enhancement of quality without losing the embedded bits. Basically, it is applied on each embedded component by taking the closest value of the pre-embedded component without hampering the least significant bit (i.e., LSB-0) along with the three embedded bits i.e., LSB-0, LSB-1, LSB-2 and LSB-3 respectively. The 1 x 2 sub-matrices after quality enhancement are as follows:

$$T''(R_1) = [234\ 384] \quad T''(G_1) = [84\ 502] \quad T''(B_1) = [16\ 238]$$

Applying inverse Legendre Transform (ILT) on 1 x 2 sub-matrices of embedded components yields the 1 x 2 sub-matrices of pixel components. The re-computed pixel components of identical block sizes are as follows:

$$R'_1= [234\ 75] \ G'_1= [84\ 209] \ B'_1= [16\ 111]$$

All re-computed pixel components are non-negative, non-fractional and less than or equal to maximum possible value (i.e., ≤ 255).

7.3.2.2.2 Results and Discussions

Table 7.7 is filled up with the data values as obtained on embedding the varying sizes of watermark (“Gold Coin”) into twenty color images (fig 1.1). In proposed WLT_1x2_QE, the peak signal to noise ratio (PSNR) value ranges from 30.16 dB to 54.22 dB of which the lower limit specify the PSNR of “Desert” at 3 bpB and that of upper limit denote the PSNR of “Barnfall” as obtained at 0.5 bpB respectively. Since, an inverse relationship is exist between the peak signal to noise ratio (PSNR) and the mean squared error (MSE), the highest PSNR ensures the lowest MSE of 0.24 whereas, the lowest PSNR points out the highest MSE of 62.66 respectively. The lowest values of image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) are 0.991610 (Desert), 0.954743 (Pepper) and 0.529837 (Splash) respectively while the highest values of these metrics are 0.999992 (Airplane), 0.999898 (San Diego) and 0.99933 (San Diego) respectively. The IF, SSIM and UIQ metrics falls into the range [0, 1] for the given payload variation of 0.5 to 3 bpB. The average results are also computed to summarize the experimental results.

Table 7.7. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WLT_1x2_QE scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.126733	0.251424	0.999984	0.999645	0.994757
	1.0	46.673650	1.398670	0.999911	0.999211	0.970898
	1.5	44.680665	2.213165	0.999864	0.998032	0.957679
	2.0	40.010497	6.486801	0.999609	0.996623	0.895047
	2.5	37.874292	10.608430	0.999360	0.991255	0.858269
	3.0	33.279022	30.561739	0.998181	0.985367	0.732040
Baboon	0.5	54.063045	0.255138	0.999986	0.999870	0.998966
	1.0	46.674475	1.398404	0.999925	0.999697	0.994214
	1.5	44.629040	2.239630	0.999883	0.999183	0.990739
	2.0	40.009369	6.488485	0.999664	0.998650	0.976144
	2.5	37.901798	10.541454	0.999455	0.996219	0.962944
	3.0	33.374507	29.897140	0.998466	0.994169	0.922979
Pepper	0.5	48.030595	1.023342	0.999900	0.990116	0.977738
	1.0	44.546889	2.282398	0.999791	0.989079	0.961660
	1.5	40.956181	5.217497	0.999562	0.981147	0.945342
	2.0	38.189180	9.866481	0.999256	0.979245	0.898797
	2.5	34.125536	25.149344	0.997952	0.960808	0.852984
	3.0	31.382530	47.296314	0.996559	0.954743	0.734404

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	53.973755	0.260438	0.999992	0.999580	0.983352
	1.0	46.642680	1.408679	0.999959	0.999063	0.924539
	1.5	44.601714	2.253766	0.999935	0.997693	0.900291
	2.0	39.964530	6.555824	0.999810	0.996016	0.800080
	2.5	37.949970	10.425173	0.999698	0.990342	0.751500
	3.0	33.408469	29.664250	0.999139	0.983906	0.612346
Sailboat	0.5	53.335572	0.301663	0.999984	0.999693	0.995550
	1.0	46.465891	1.467206	0.999925	0.999326	0.976773
	1.5	44.222695	2.459295	0.999876	0.998288	0.968393
	2.0	39.813056	6.788514	0.999660	0.997132	0.925200
	2.5	37.295481	12.120825	0.999396	0.992089	0.895727
	3.0	33.066944	32.091197	0.998400	0.987825	0.796843
Earth	0.5	54.137597	0.250796	0.999985	0.999730	0.997664
	1.0	46.681633	1.396101	0.999917	0.999408	0.986370
	1.5	44.704989	2.200804	0.999866	0.998506	0.980197
	2.0	40.049893	6.428223	0.999603	0.997468	0.944900
	2.5	37.990276	10.328867	0.999354	0.993227	0.916366
	3.0	33.368604	29.937803	0.998109	0.988788	0.812604
San Diego	0.5	54.122957	0.251642	0.999990	0.999898	0.999330
	1.0	46.684078	1.395315	0.999948	0.999778	0.996197
	1.5	44.702162	2.202237	0.999916	0.999494	0.994780
	2.0	40.043142	6.438224	0.999751	0.999152	0.985998
	2.5	38.087536	10.100123	0.999609	0.997877	0.980167
	3.0	33.459498	29.317738	0.998858	0.996404	0.951322
Splash	0.5	49.266754	0.769847	0.999917	0.989394	0.963196
	1.0	45.155668	1.983871	0.999809	0.988233	0.895883
	1.5	41.526942	4.574947	0.999538	0.982111	0.865450
	2.0	38.446035	9.299869	0.999135	0.979405	0.752744
	2.5	34.959912	20.753393	0.997930	0.967053	0.694115
	3.0	31.751431	43.444740	0.996052	0.958001	0.529837
Oakland	0.5	51.286666	0.483519	0.999968	0.999417	0.997710
	1.0	45.888116	1.675979	0.999903	0.999101	0.992149
	1.5	42.860479	3.365366	0.999785	0.997906	0.989115
	2.0	39.209286	7.801025	0.999524	0.997154	0.974837
	2.5	36.041633	16.177702	0.998936	0.992372	0.960890
	3.0	32.406778	37.359624	0.997665	0.989219	0.909972

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	54.153834	0.249860	0.999991	0.999519	0.993394
	1.0	46.686176	1.394641	0.999950	0.998943	0.963030
	1.5	44.701377	2.202635	0.999919	0.997488	0.951016
	2.0	40.072606	6.394692	0.999759	0.995839	0.875665
	2.5	38.132137	9.996929	0.999622	0.989528	0.830281
	3.0	33.566009	28.607467	0.998901	0.982664	0.680336
Anhinga	0.5	50.833323	0.536720	0.999958	0.999809	0.882627
	1.0	45.879051	1.679481	0.999870	0.999594	0.869365
	1.5	42.602179	3.571596	0.999727	0.998792	0.877814
	2.0	38.533580	9.114278	0.999312	0.998148	0.826018
	2.5	36.002337	16.324751	0.998762	0.994339	0.809853
	3.0	32.277468	38.488719	0.997106	0.988979	0.739764
Athens	0.5	51.884716	0.421316	0.999966	0.999864	0.971366
	1.0	46.182203	1.566246	0.999874	0.999663	0.956086
	1.5	43.909096	2.643447	0.999789	0.999111	0.958864
	2.0	38.671380	8.829625	0.999300	0.998548	0.891646
	2.5	37.036086	12.866834	0.998981	0.994641	0.868368
	3.0	32.913757	33.243338	0.997370	0.986008	0.763961
Bardowl	0.5	50.122462	0.632170	0.999934	0.999465	0.997727
	1.0	45.519226	1.824558	0.999815	0.999160	0.994449
	1.5	42.871065	3.357173	0.999660	0.998051	0.989741
	2.0	38.615707	8.943543	0.999118	0.997417	0.974807
	2.5	35.542374	18.148591	0.998167	0.986580	0.939692
	3.0	32.285814	38.414826	0.996200	0.983318	0.897949
Barnfall	0.5	54.225905	0.245747	0.999961	0.999762	0.998852
	1.0	46.822993	1.351390	0.999782	0.999447	0.992625
	1.5	44.661765	2.222817	0.999663	0.998629	0.990632
	2.0	39.856823	6.720444	0.999039	0.997818	0.971203
	2.5	37.764385	10.880324	0.998389	0.993009	0.956245
	3.0	33.435054	29.483220	0.995916	0.989175	0.893693
Butrfly	0.5	52.613555	0.356225	0.999974	0.999852	0.995605
	1.0	46.341627	1.509793	0.999892	0.999602	0.988149
	1.5	44.100897	2.529243	0.999827	0.999025	0.986132
	2.0	38.924056	8.330568	0.999454	0.998346	0.953044
	2.5	37.217131	12.341478	0.999188	0.995421	0.937675
	3.0	32.816718	33.994490	0.997792	0.989634	0.847609

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	52.199342	0.391873	0.999944	0.999695	0.749617
	1.0	46.239258	1.545804	0.999782	0.999105	0.740411
	1.5	43.950217	2.618536	0.999647	0.997653	0.742616
	2.0	38.756178	8.658896	0.998888	0.995740	0.707301
	2.5	36.753449	13.732051	0.998232	0.988766	0.695719
	3.0	32.112932	39.974872	0.994974	0.979041	0.633106
Bodie	0.5	50.407332	0.592034	0.999893	0.999533	0.981783
	1.0	45.415760	1.868548	0.999675	0.999275	0.970274
	1.5	42.647540	3.534486	0.999395	0.997817	0.964231
	2.0	39.391279	7.480874	0.998763	0.996967	0.950131
	2.5	35.094462	20.120285	0.996538	0.980213	0.914337
	3.0	32.077027	40.306729	0.993385	0.976543	0.867005
Bluheron	0.5	54.188895	0.247851	0.999969	0.999756	0.996671
	1.0	47.132107	1.258547	0.999846	0.999437	0.987203
	1.5	44.767892	2.169157	0.999734	0.998402	0.979925
	2.0	40.299192	6.069614	0.999254	0.997469	0.950232
	2.5	38.154648	9.945246	0.998777	0.993556	0.927518
	3.0	33.192825	31.174378	0.996163	0.988583	0.812856
Colomtn	0.5	52.498062	0.365825	0.999971	0.999762	0.985223
	1.0	46.275919	1.532810	0.999880	0.999486	0.976688
	1.5	43.408735	2.966243	0.999768	0.998709	0.972986
	2.0	39.571410	7.176939	0.999436	0.997891	0.953840
	2.5	36.974500	13.050594	0.998977	0.994407	0.936535
	3.0	33.075644	32.026976	0.997480	0.990452	0.870666
Desert	0.5	45.918758	1.664196	0.999731	0.997251	0.994053
	1.0	43.383755	2.983353	0.999535	0.996765	0.990937
	1.5	39.592435	7.142279	0.998936	0.992418	0.982504
	2.0	37.242832	12.268660	0.998295	0.991534	0.969657
	2.5	32.297017	38.315862	0.994467	0.970116	0.931466
	3.0	30.160209	62.669953	0.991610	0.965179	0.890736
Average case	0.5	52.06949	0.477581	0.99995	0.998581	0.972759
	1.0	46.06456	1.64609	0.999849	0.998169	0.956395
	1.5	43.5049	3.084216	0.999715	0.996423	0.949422
	2.0	39.2835	7.807079	0.999332	0.995328	0.908865
	2.5	36.65975	15.09641	0.99859	0.988091	0.881033
	3.0	32.67056	35.89778	0.996916	0.9829	0.795001

The effect of quality enhancement in WLT_1x2_QE is better analyzed by making a comparison between the WLT_1x2 and WLT_1x2_QE in terms of average PSNR. Table 7.8 demonstrates that the average PSNR is computed with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads which is obtained from twenty benchmark images [130, 131] as given in fig. 1.1. The average PSNR (in terms of dB) of WLT_1x2 falls into the range [31.68, 52.06] and that of values for WLT_1x2_QE lies in between 32.67 to 52.06 dB. Therefore, it is evident that the incorporation of adaptive quality enhancement in WLT_1x2 makes it improved when the payload exceeds 1 bpB of payload.

Table 7.8. Comparative analysis of obtained average PSNR values between WLT_1x2 and WLT_1x2_QE with respect to increasing payload

WLT_1x2		**WLT_1x2_QE**	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	52.06949	0.5	52.06949
1.0	46.06456	1.0	46.06456
1.5	43.04556	1.5	43.50490
2.0	38.48890	2.0	39.28350
2.5	35.88123	2.5	36.65975
3.0	31.68995	3.0	32.67056

The computed results of WLT_1X2_QE (especially, the average PSNR (dB) and the payload (bpB)) are compared against two existing embedding schemes: Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129]. The PSNR values of DPTHDI [88] and DGTDHS [129] schemes are computed at 0.25 and 1 bpB of payload respectively. The major limitation of the DPTHDI [88] and DGTDHS [129] schemes are their fixed (as well as low) payload. In contrast, the basic advantage of WLT_1X2_QE is its ability of supporting variable payload for a spread from 0.5 to 3 bpB. The WLT_1X2_QE also provides perceptible visual clarity at respective payload since, the PSNR values exceeds the level of acceptability (i.e., > 30 dB) [148]. In contrast to DPTHDI [88], the obtained PSNR of WLT_1x2_QE is greater than or equal to that of values for “Lena” at 0.5, 1, 1.5 and 2 bpB, for “Baboon” at 0.5, 1, 1.5, 2, 2.5 and 3 bpB, for “Pepper” at 0.5, 1, 1.5 and 2 bpB, for “Airplane” at 0.5, 1 and 1.5 bpB and for “Sailboat” at 0.5, 1, 1.5, 2 and 2.5 bpB respectively. In contrast to DGTDHS [129], WLT_1x2_QE ensured variable payload (i.e., up to 3 bpB) with minor loss of PSNR values at 1 bpB.

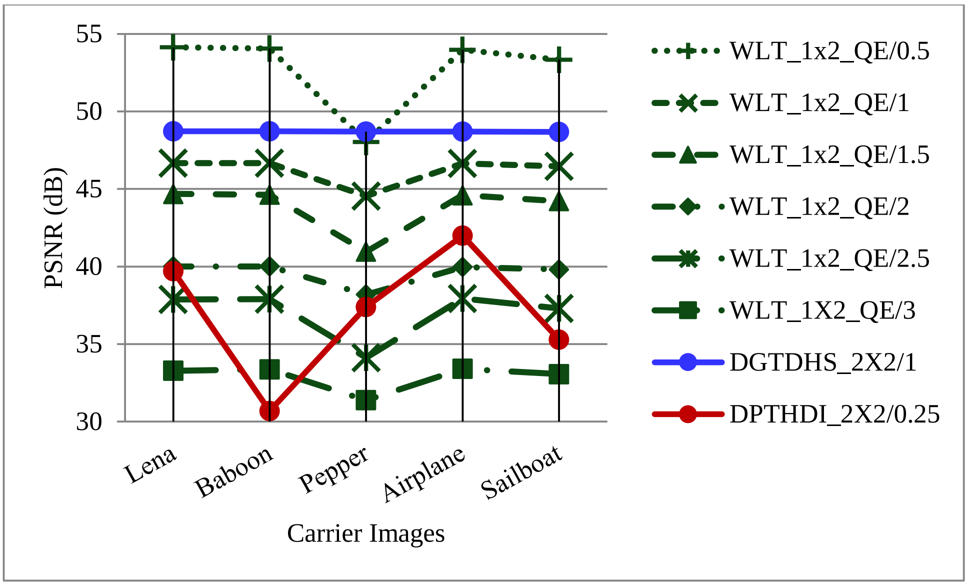


Fig. 7.7. Performance analysis of PSNR (dB) for variable payload based WLT_1x2_QE and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

In fig 7.8, the average PSNR variation is observed for WLT_1x2_QE, WLT_1x2, WLT_2x2_QE, WDHT_1x2_QE, DPTHDI [88] and DGTDHS [129] respectively. The payload is another important parameter which is fixed for DPTHDI [88] as well as DGTDHS [129] and variable for WLT_1x2_QE, WLT_1x2, WLT_2x2_QE as well as WDHT_1x2_QE respectively. In contrast to WLT_1x2 method, the improvement of quality of images in terms of average PSNR for WLT_1X2_QE is observed as the payload increases from 1 bpB. In contrast to WDHT_1X2_QE, WLT_2X2_QE and WLT_1x2, the WLT_1X2_QE method ensured less degradation in quality with respect to variable payload. The average PSNR for DPTHDI [88] is 37.40 dB by considering the average of PSNR values for "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" images at 0.25 bpB of payload. Compared to DPTHDI [88], the average PSNR of WLT_1x2_QE has been enhanced with respect to 0.5, 1, 1.5 and 2 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB by taking the average of PSNR values for "Lighthouse", "Elaine", "Lenna", "Boat", "F16" images at 1 bpB of payload. Compared to DGTDHS [129], the average PSNR of WLT_1x2_QE provides 2.5 dB of lesser value at 1 bpB however, the ability of providing variable payload for a spread from 0.5 to 3 bpB, makes WLT_1x2_QE a relevant choice of embedding by retaining the perceptible level of quality [148]. In addition, the block size as 1 x 2 makes WLT_1x2_QE as the superior technique over 2 x 2 block based WLT_2x2 since the 1 x 2 block based scheme offered better average PSNR values with respect to the specified payload values.

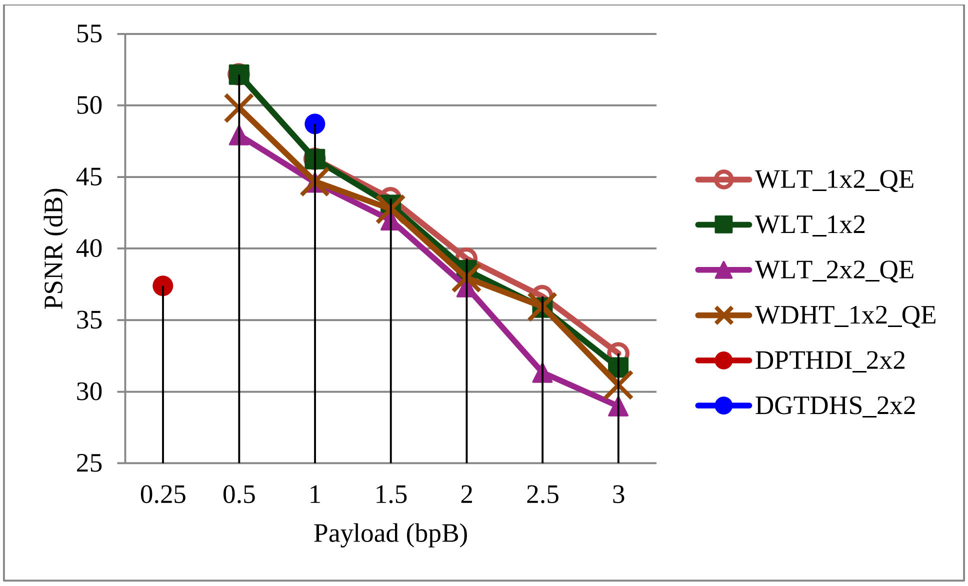


Fig. 7.8. Graphical representation of variation of average PSNR (dB) with respect to payload for WLT_1x2_QE, WLT_1X2, WLT_2X2_QE, WDHT_1x2_QE and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

7.3.3 Quality Enhancement of Binomial Transform (BT) based watermarking

Binomial Transform (BT) based watermarking has been proposed using specifications comprising of 2 x 2 and 1 x 2 blocks respectively. Both schemes such as the WBT_2x2 and WBT_1x2 are available in chapter 4.

Section 7.3.3.1 deals with the 2 x 2 block based watermark fabrication using Binomial Transform (BT) followed by adaptive quality enhancement of section 7.2.1 i.e., WBT_2X2_QE whereas, 1 x 2 block based watermark fabrication followed by adaptive quality enhancement of section 7.2.1 i.e., WBT_1X2_QE is discussed in section 7.3.3.2.

7.3.3.1 Quality Enhancement for 2 x 2 Block based Watermark Fabrication

In WBT_2x2_QE scheme, the quality enhancement of section 7.2.1 is introduced as the post-embedding operation in WBT_2x2 of section 4.2.1. The carrier image is partitioned into 2 x 2 non-overlapping blocks where, the pixel components are adjusted prior to embedding based on the pixel value adjustment strategy as discussed in equation (4.6) of section 4.2.1.1. Binomial Transform (BT) converts each 2 x 2 sub-matrix of pixel components into transform domain. Secret bits consisting of the message digest, size and content of the watermark are fabricated into the transformed components starting from the least significant bit position (i.e., LSB-0) toward higher order bit positions. The adaptive quality enhancement of section 7.2.1 has been applied on each embedded component to reduce the quality distortion by keeping the fabricated bits unaffected. Inverse Binomial Transform (IBT) is applied over each 2 x 2 sub-matrices of quality enhanced components to obtain the pixel components in spatial domain. The process is repeated until and unless the secret information is concealed.

The process of carrier image decomposition as well as pixel adjustment, application of transformation, embedding of secret information, quality enhancement and the inverse transformation are clearly explained with a suitable example as given in section 7.3.3.1.1. Results, comparative analysis and discussions have been elaborated in section 7.3.3.1.2.

7.3.3.1.1 Example

Decompose the cover image into 2 x 2 non-overlapping blocks corresponding to red, green and blue channels. For example, a random payload value of 3 bpB has been chosen for embedding information and therefore, the pixel components are adjusted prior to embedding as discussed in equation (4.6) of section 4.2.1.1. The adjusted 2 x 2 sub-matrices of pixel components are obtained as follows:

$$R_1 = \begin{bmatrix} 224 & 72 \\ 32 & 155 \end{bmatrix} G_1 = \begin{bmatrix} 62 & 215 \\ 56 & 32 \end{bmatrix} B_1 = \begin{bmatrix} 111 & 172 \\ 224 & 32 \end{bmatrix}$$

Binomial Transform (BT) has been used to convert 2 x 2 sub-matrices of pixel components into transform domain. The 2 x 2 sub-matrices of transformed components such as T(R$_1$), T(G$_1$) and T(B$_1$) are obtained as follows:

$$T(R_1) = \begin{bmatrix} 224 & 152 \\ 112 & -51 \end{bmatrix} T(G_1) = \begin{bmatrix} 62 & -153 \\ -312 & -447 \end{bmatrix} T(B_1) = \begin{bmatrix} 111 & -61 \\ -9 & 235 \end{bmatrix}$$

Let the secret bit-stream *"0101110101100100110011110110110000000"* is to be fabricated into the transformed components based on the embedding rule of equation (4.7) of section 4.2.1.1. This example deals with the payload value of 3 bpB which ensured three, two, three and four bits of secret information fabrication (λ_1 = 3, λ_2 = 2, λ_3 = 3, λ_4 = 4) on first, second, third and fourth transformed components starting from LSB-0 toward higher order bit positions. Hence, the 2 x 2 sub-matrices of embedded components are as here under:

$$T'(R_1) = \begin{bmatrix} 226 & 155 \\ 117 & -54 \end{bmatrix} T'(G_1) = \begin{bmatrix} 58 & -154 \\ -313 & -447 \end{bmatrix} T'(B_1) = \begin{bmatrix} 110 & -62 \\ -9 & 224 \end{bmatrix}$$

The quality of embedded components is improved based on the adaptive quality enhancement used in this context. The enhancement process does not hamper the embedded bits. Basically, it is applied on each embedded coefficient by taking the closest value of the pre-embedded component without affecting the embedded bits. The 2 x 2 sub-matrices of quality enhanced components are computed as follows:

$$T''(R_1) = \begin{bmatrix} 226 & 155 \\ 117 & -54 \end{bmatrix} T''(G_1) = \begin{bmatrix} 58 & -154 \\ -313 & -447 \end{bmatrix} T'(B_1) = \begin{bmatrix} 110 & -62 \\ -9 & 240 \end{bmatrix}$$

Applying one dimensional inverse Binomial Transform (IBT) on 2 x 2 sub-matrices of quality enhanced components yields the generation of 2 x 2 sub-matrices of pixel components in spatial domain as given below:

$$R'_1 = \begin{bmatrix} 226 & 71 \\ 33 & 166 \end{bmatrix} G'_1 = \begin{bmatrix} 58 & 212 \\ 53 & 28 \end{bmatrix} B'_1 = \begin{bmatrix} 110 & 172 \\ 225 & 29 \end{bmatrix}$$

Re-adjustment operation is no longer used in this example since no violation in pixel component representation is occurred during inverse Binomial Transform (IBT) phase. As a consequence, all re-generated pixel components become non-fractional and are lies in the range of 0 to 255.

7.3.3.1.2 Results and Discussions

The WBT_2X2_QE scheme is an improvement of WBT_2X2 scheme of section 4.2.1 followed by adaptive quality enhancement of section 7.2.1. Table 7.9 deals with the quality of the WBT_2X2_QE is evaluated based on the following metrics: peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ). The results are computed for twenty benchmark (BMP) images [130, 131] of dimension 512 x 512 along with the varying sizes of the fabricated watermark as given in fig. 1.1 with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. In proposed WBT_2x2_QE, the minimum value of PSNR is 25.90 dB at 3 bpB of payload for the “Desert” image whereas, maximum value of PSNR is 50.22 dB at 0.5 bpB of payload for the “Foster City” image. It is seen that the PSNR of “Desert” at 3 bpB falls below the acceptable level of visual clarity (i.e., < 30 dB) however, the average PSNR value as obtained from twenty benchmark images [130, 131] gives well perceptible quality as evident from table 7.9 (i.e., ≥ 30 dB) [148]. The lowest MSE is 0.61 for “Foster City” at payload of 0.5 bpB and the highest MSE is 166.78 for “Desert” at 3 bpB of payload. Usually the IF, SSIM and UIQ lie between 0 and 1, the closer the IF, SSIM and UIQ to one, watermarked image is more similar to the original image. The minimum and maximum values of remaining metrics such as the IF, SSIM and UIQ falls into the range [0.974906 (Desert), 0.999981 (Airplane)], [0.815983 (Splash), 0.957760 (San Diego)] and [0.424371 (Splash), 0.998400 (San Diego)] respectively. The average values are also computed for various metrics of twenty cover images with respect to the payload variation from 0.5 to 3 bpB for the summarization of the experimental results.

Table 7.9. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WBT_2x2_QE scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	50.170722	0.625184	0.999960	0.997008	0.987649
	1.0	45.386183	1.881317	0.999882	0.994955	0.963883
	1.5	42.156215	3.957845	0.999751	0.984144	0.930079
	2.0	38.359392	9.487266	0.999421	0.972160	0.859673
	2.5	35.194905	19.660283	0.998799	0.927994	0.777900
	3.0	30.683498	55.555939	0.996698	0.881090	0.641219
Baboon	0.5	50.123143	0.632071	0.999966	0.998939	0.997615
	1.0	45.343037	1.900100	0.999899	0.998155	0.991792
	1.5	42.106585	4.003334	0.999788	0.994186	0.984289
	2.0	38.542892	9.094757	0.999522	0.989422	0.961138
	2.5	35.564697	18.055544	0.999050	0.973596	0.939299
	3.0	31.123046	50.208319	0.997381	0.953671	0.888084
Pepper	0.5	46.481090	1.462080	0.999862	0.987135	0.973145
	1.0	41.868835	4.228602	0.999603	0.979568	0.949959
	1.5	40.578819	5.691129	0.999491	0.971059	0.927635
	2.0	35.177781	19.737958	0.998170	0.945697	0.856572
	2.5	33.963260	26.106839	0.997662	0.909341	0.791451
	3.0	28.146054	99.649462	0.990668	0.836608	0.639797
Airplane	0.5	50.072236	0.639523	0.999981	0.996426	0.963779
	1.0	45.333210	1.904404	0.999945	0.993990	0.909617
	1.5	42.067337	4.039676	0.999884	0.981225	0.844767
	2.0	38.436511	9.320285	0.999733	0.966677	0.743989
	2.5	35.049479	20.329766	0.999418	0.912421	0.643382
	3.0	31.812749	42.835650	0.998774	0.871750	0.551397
Sailboat	0.5	49.734037	0.691315	0.999965	0.997561	0.989584
	1.0	44.935943	2.086825	0.999894	0.995717	0.968871
	1.5	41.983120	4.118776	0.999792	0.987019	0.943910
	2.0	38.176738	9.894788	0.999502	0.976600	0.892381
	2.5	35.312123	19.136744	0.999034	0.942404	0.838775
	3.0	30.578456	56.916047	0.997149	0.894696	0.717158
Earth	0.5	50.211011	0.619411	0.999963	0.997722	0.994337
	1.0	45.436824	1.859507	0.999890	0.996125	0.982788
	1.5	42.193974	3.923583	0.999767	0.987718	0.964036
	2.0	38.734924	8.701375	0.999482	0.978945	0.919838

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	35.553660	18.101491	0.998929	0.946123	0.862358
	3.0	31.447881	46.589949	0.997120	0.910885	0.744137
San Diego	0.5	50.199054	0.621119	0.999976	0.999143	0.998400
	1.0	45.513612	1.826918	0.999931	0.998581	0.995415
	1.5	42.219038	3.901004	0.999854	0.995318	0.990272
	2.0	39.069827	8.055594	0.999700	0.992342	0.979797
	2.5	35.902814	16.703169	0.999379	0.979605	0.963165
	3.0	32.562613	36.042842	0.998662	0.967145	0.930813
Splash	0.5	47.457569	1.167678	0.999882	0.985263	0.941465
	1.0	42.856725	3.368277	0.999663	0.979048	0.880062
	1.5	41.047743	5.108648	0.999524	0.964269	0.809011
	2.0	36.142813	15.805164	0.998434	0.940479	0.690513
	2.5	34.147866	25.020369	0.997708	0.876529	0.574362
	3.0	29.290658	76.562220	0.992376	0.815983	0.424371
Oakland	0.5	48.711909	0.874760	0.999946	0.997839	0.996125
	1.0	44.044938	2.562043	0.999844	0.996276	0.989936
	1.5	41.620613	4.477329	0.999746	0.989878	0.980873
	2.0	37.505178	11.549482	0.999302	0.982066	0.957767
	2.5	35.135767	19.929828	0.998882	0.956312	0.924835
	3.0	30.624288	56.318553	0.996556	0.925091	0.855414
Foster City	0.5	50.223314	0.617659	0.999978	0.995890	0.984041
	1.0	45.384716	1.881952	0.999933	0.992950	0.951803
	1.5	42.159227	3.955101	0.999859	0.977993	0.905974
	2.0	38.581844	9.013549	0.999683	0.961619	0.814623
	2.5	35.374659	18.863159	0.999336	0.902636	0.717258
	3.0	31.782457	43.135476	0.998492	0.844469	0.591342
Anhinga	0.5	48.518520	0.914594	0.999929	0.998785	0.880234
	1.0	43.245345	3.079964	0.999764	0.997130	0.869721
	1.5	40.215099	6.188287	0.999525	0.992256	0.849972
	2.0	36.806812	13.564352	0.998960	0.986679	0.829577
	2.5	33.768748	27.302692	0.997912	0.966133	0.781278
	3.0	30.305261	60.611391	0.995361	0.940675	0.723035
Athens	0.5	48.616577	0.894175	0.999928	0.998795	0.968605
	1.0	46.100794	1.595882	0.999872	0.996943	0.956766
	1.5	41.598655	4.500024	0.999640	0.992058	0.933402
	2.0	39.943610	6.587479	0.999474	0.987054	0.909522

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	35.344399	18.995049	0.998478	0.963431	0.833598
	3.0	33.442918	29.429882	0.997644	0.942910	0.768146
Bardowl	0.5	47.654216	1.115985	0.999886	0.998745	0.996966
	1.0	43.460276	2.931248	0.999700	0.997075	0.989359
	1.5	41.090608	5.058474	0.999490	0.994332	0.984215
	2.0	36.804052	13.572977	0.998603	0.982683	0.946972
	2.5	34.616668	22.460194	0.997729	0.973124	0.928683
	3.0	28.879789	84.159163	0.991286	0.919157	0.828981
Barnfall	0.5	49.699070	0.696904	0.999893	0.998321	0.997215
	1.0	45.139822	1.991123	0.999687	0.996706	0.991625
	1.5	41.946143	4.153995	0.999337	0.990275	0.982732
	2.0	38.781135	8.609279	0.998577	0.983377	0.962902
	2.5	34.971373	20.698696	0.996630	0.956413	0.922686
	3.0	31.689975	44.063889	0.992392	0.920771	0.844351
Butrfly	0.5	49.015265	0.815743	0.999942	0.998879	0.994139
	1.0	44.981576	2.065012	0.999852	0.996944	0.983467
	1.5	41.899642	4.198712	0.999702	0.992944	0.974351
	2.0	39.065005	8.064544	0.999427	0.987447	0.948384
	2.5	35.511315	18.278846	0.998706	0.968052	0.906709
	3.0	32.495607	36.603246	0.997375	0.947228	0.825320
Bobcat	0.5	49.066413	0.806193	0.999886	0.998370	0.748072
	1.0	45.974652	1.642915	0.999767	0.996511	0.741588
	1.5	41.576817	4.522708	0.999364	0.987272	0.726668
	2.0	39.390351	7.482474	0.998944	0.980844	0.707033
	2.5	35.237320	19.469209	0.997270	0.948886	0.674034
	3.0	27.197981	123.960173	0.982466	0.848488	0.552868
Bodie	0.5	48.151928	0.995147	0.999826	0.998603	0.976576
	1.0	43.478243	2.919147	0.999487	0.997047	0.967810
	1.5	40.887612	5.300528	0.999083	0.992128	0.957622
	2.0	36.321396	15.168432	0.997292	0.978663	0.924840
	2.5	34.214162	24.641328	0.995739	0.964265	0.898173
	3.0	27.919454	104.986859	0.980904	0.881864	0.775014
Bluheron	0.5	49.847640	0.673467	0.999917	0.998128	0.993673
	1.0	46.205767	1.557771	0.999809	0.996926	0.985988
	1.5	42.161785	3.952772	0.999516	0.990264	0.969313
	2.0	39.679375	7.000720	0.999141	0.984366	0.941976

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	34.205770	24.688991	0.996980	0.949822	0.869943
	3.0	32.402909	37.392920	0.995415	0.927818	0.798530
Colomtn	0.5	49.506922	0.728430	0.999943	0.998396	0.983158
	1.0	43.986857	2.596537	0.999797	0.996952	0.976866
	1.5	40.899335	5.286239	0.999587	0.991299	0.965292
	2.0	37.481388	11.612922	0.999093	0.984837	0.945107
	2.5	34.328348	24.001892	0.998123	0.957121	0.898612
	3.0	30.778066	54.359287	0.995761	0.929234	0.839214
Desert	0.5	44.838152	2.134347	0.999661	0.996140	0.992907
	1.0	40.125487	6.317301	0.999019	0.992246	0.982882
	1.5	39.069097	8.056948	0.998785	0.989932	0.979908
	2.0	33.375037	29.893491	0.995416	0.970657	0.940649
	2.5	32.258657	38.655792	0.994253	0.95776 0	0.922757
	3.0	25.909278	166.783262	0.974906	0.880760	0.809591
Average case	0.5	48.914940	0.886289	0.999915	0.996804	0.967884
	1.0	44.440140	2.509842	0.999762	0.994492	0.951510
	1.5	41.473870	4.719756	0.999574	0.987278	0.930216
	2.0	37.818800	11.610840	0.998894	0.976631	0.886663
	2.5	34.782800	22.054990	0.998001	0.946598	0.833463
	3.0	30.453650	65.308230	0.993369	0.902015	0.737439

Table 7.10 shows the numerical comparison between WBT_2x2_QE and WBT_2x2 in terms of average PSNR values with respect to payload variation (0.5 – 3 bpB). Both schemes offered identical PSNR values at 0.5 bpB however, the WBT_2x2_QE is distinguishable from WBT_2x2 for its ability of providing higher average PSNR values at 1, 1.5, 2, 2.5 and 3 bpB respectively. The minimum values of average PSNR which yields the maximum quality distortion for WBT_2x2 and WBT_2x2_QE are also computed as 28.70 and 30.45 dB respectively. Therefore, the minimum and maximum expansions of average PSNR values for WBT_2x2_QE are 0.28 and 1.75 dB respectively.

Table 7.10. Comparative analysis of obtained average PSNR values between WBT_2x2 and WBT_2x2_QE against increasing payload

WBT_2x2		WBT_2x2_QE	
Payload (bpB)	PSNR (dB)	Payload (bpB)	PSNR (dB)
0.5	48.91494	0.5	48.91494
1.0	44.16739	1.0	44.44014

WBT_2x2		WBT_2x2_QE	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
1.5	39.93230	1.5	41.47387
2.0	36.53638	2.0	37.81880
2.5	32.92258	2.5	34.78280
3.0	28.70732	3.0	30.45365

In fig. 7.9, the PSNR (dB) analysis has been made for five color images such as "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively. The existing watermarking schemes such as DPTHDI [88] and DGTDHS [129] deals with the computation of PSNR (dB) values for the fixed payload of 0.25 and 1 bpB respectively. The payload of both schemes is considered to be low as far as the 2 x 2 block based watermarking schemes are concerned. However, the Binomial Transform (BT) based watermarking of section 4.2.1 followed by the adaptive quality enhancement (WBT_2x2_QE) ensured the usage of varying payload within 0.5 to 3 bpB. In comparison to DPTHDI [88], the PSNR values of proposed WBT_2x2_QE is equal or higher up to 1.5 bpB of payload for "Lena", up to 3 bpB of payload for "Baboon", up to 1.5 bpB of payload for "Pepper", up to 1.5 bpB of payload for "Airplane" and up to 2.5 bpB of payload for "Sailboat" respectively. Compared to DGTDHS [129], the offered PSNR values in WBT_2x2_QE is slightly less however, the variable payload makes WBT_2x2_QE as an effective choice for embedding with permissible range of payload [from 0.5 to 3 bpB].

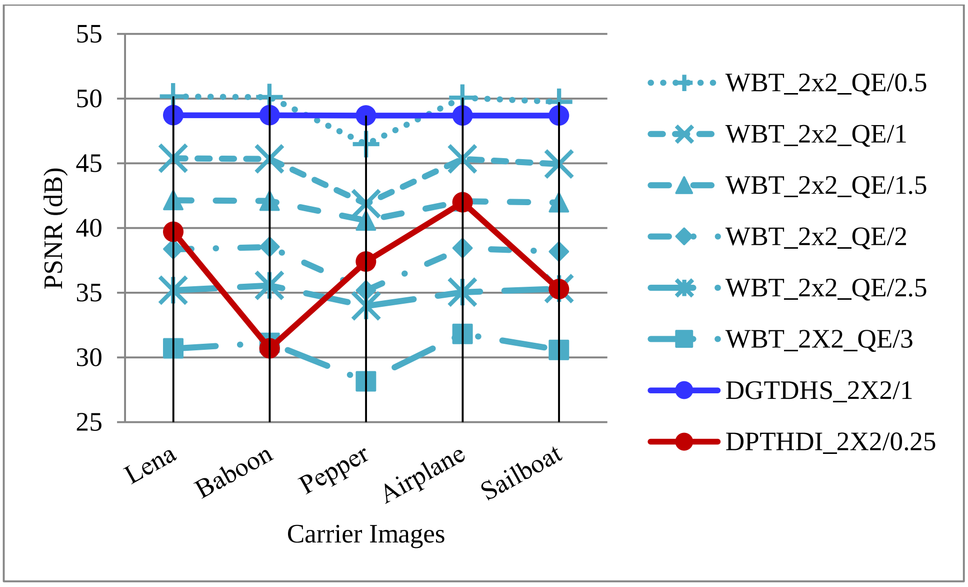


Fig. 7.9. Performance analysis of PSNR (dB) for variable payload based WBT_2x2_QE and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes against five color images

Fig. 7.10 depicts the variation of average peak signal to noise ratio (PSNR) for proposed 2 x 2 block based scheme (WBT_2x2_QE) in comparison with variable payload (WBT_2x2, WLT_2x2_QE and WDHT_2x2_QE) and fixed payload (DPTHDI [88] and DGTDHS [129]) based schemes, respectively. Six color images (as shown in fig. 1.1) such as “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” are taken to compute the average PSNR of 37.40 dB for DPTHDI [88] at 0.25 bpB of payload. On the other hand, the images such as “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” respectively are taken to compute the average PSNR of 48.70 dB for DGTDHS [129] at 1 bpB of payload. In contrast to WBT_2x2 technique, the improvement of average PSNR in WBT_2X2_QE technique can be visually perceived as the payload increases from 0.5 bpB. Compared to WLT_2x2_QE and WDHT_2x2_QE, the superiority of the WBT_2x2_QE with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads have also been observed from the line chart. In comparison with DPTHDI [88], the average PSNR of WBT_2x2_QE ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads. In comparison with DGTDHS [129], the WBT_2x2_QE is lacking in terms of average PSNR at 1 bpB of payload however, the ability of offering variable payload for a spread of 0.5 to 3 bpB is considered to be a significant achievement. It is to be noted that the WBT_2x2_QE offers average PSNR values of greater than or equal to 30 dB with respect to the variable payload of 0.5 to 3 bpB. As a consequence, the incorporation of quality enhancement ensured the improvement of average PSNR values for payload range (1– 3 bpB) which results well perceptible watermarked images [148].

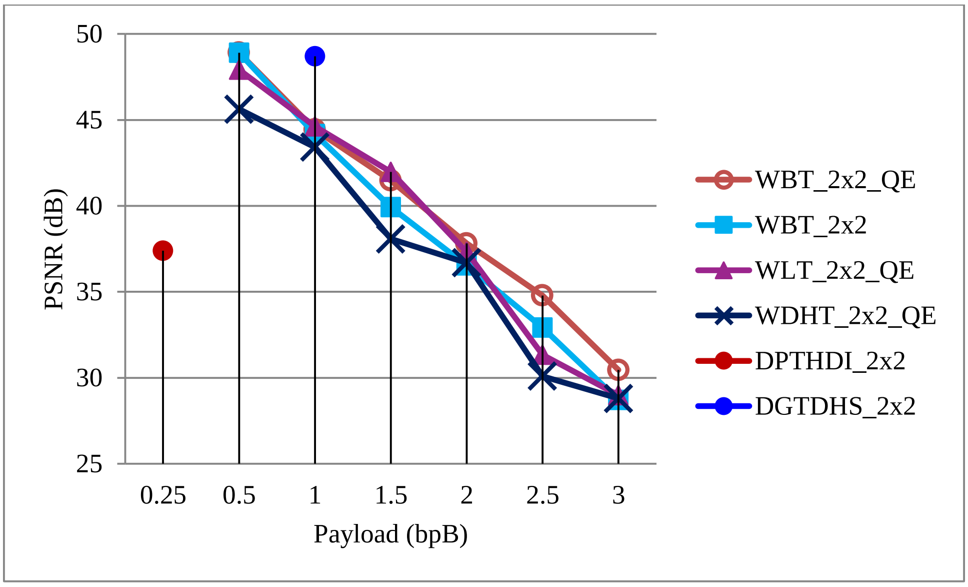


Fig. 7.10. Graphical representation of variation of average PSNR (dB) with respect to payload for WBT_2x2_QE, WBT_2x2, WLT_2x2_QE, WDHT_2x2_QE and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

7.3.3.2 Quality Enhancement for 1 x 2 Block based Watermark Fabrication

In section 4.2.2, the WBT_1x2 has already been discussed; however, the scheme is extended as WBT_1x2_QE by introducing the adaptive quality enhancement scheme of section 7.2.1 as the post-embedding operation. The carrier image is partitioned into 1 x 2 non-overlapping blocks and then, the pixel components are adjusted to avoid overflow/underflow based on a pixel adjustment process as discussed in equation (4.16) of section 4.2.2.1. Binomial Transform (BT) converts each 1 x 2 sub-matrix of pixel components into transform domain. Secret bits corresponding to the message digest, size and content of the watermark are fabricated on each transformed component starting from the least significant bit position (i.e., LSB-0) toward higher order bit position. The adaptive quality enhancement ensured the reduced difference between the pre-embedded component and the embedded component without altering the embedded bits. Inverse Binomial Transform (IBT) is applied over 1 x 2 sub-matrices of quality enhanced components to obtain the 1 x 2 sub-matrices of pixel components in spatial domain. The process is repeated till the entire secret information is concealed and the watermarked image is produced.

Section 7.3.3.2.1 dealt with an example of embedding followed by quality enhancement. The experimental results are computed, analyzed and compared to assess the validity of this scheme. The results and discussions are summarized in section 7.3.3.2.2.

7.3.3.2.1 Example

Consider the 1 x 2 sub-matrices of red, green and blue channel for watermark fabrication. Suppose, the embedding payload is 3 bpB and the pixel components are adjusted prior to embedding based on equation (4.16) of section 4.2.2.1. The 1 x 2 sub-matrices are as follows:

$$R_1 = [240\ 78] \quad G_1 = [119\ 217] \quad B_1 = [16\ 130]$$

Binomial Transform (BT) converts each 1 x 2 sub-matrix or pair of pixel components into transform domain. The 1 x 2 sub-matrices of transformed components such as T(R_1), T(G_1) and T(B_1) are obtained as follows:

$$T(R_1) = [240\ 162] \quad T(G_1) = [119\ -98] \quad T(B_1) = [16\ -114]$$

Consider the watermark bit stream *"111101110010110000"* which is to be fabricated into the transformed component based on the embedding rule given in equation (4.17) of section 4.2.2.1. In this example, three bits are fabricated ($\lambda = \lambda_1 = \lambda_2 = 3$) on each transformed component starting from LSB-0 toward the higher order bit position. Therefore, the 1 x 2 sub-matrices of embedded components corresponding to RGB color channels are obtained as follows:

$$T'(R_1) = [248\ 165] \quad T'(G_1) = [115\ -98] \quad T'(B_1) = [19\ -112]$$

The adaptive quality enhancement of section 7.2.1 has been used to ensure the enhancement of quality without losing the embedded bits. Basically, it is applied on each embedded component by taking the closest value of the pre-embedded component without hampering the three embedded bits i.e., LSB-0, LSB-1 and LSB-2 respectively. Hence, the 1 x 2 sub-matrices on quality enhancement become:

$$T''(R_1) = [239\ 165] \quad T'(G_1) = [115\ -98] \quad T'(B_1) = [19\ -112]$$

Applying inverse Binomial Transform (IBT) on 1 x 2 sub-matrices of embedded components yields the 1 x 2 sub-matrices of pixel components in spatial domain as follows:

$$R'_1 = [239\ 74] \quad G'_1 = [115\ 213] \quad B'_1 = [19\ 131]$$

The re-computed pixel components are non-fractional and the usual values are lies between 0 and 255.

7.3.3.2.2 Results and Discussions

This section represents the results and discussions of the WBT_1X2_QE scheme. Experiment deals with twenty 512 × 512 color images and varying sizes of the secret “Gold-Coin” as given in fig. 1.1 to compute the results. It is apparent from table 7.11 that payload is inversely proportional to peak signal to noise ratio (PSNR) and directly proportional to mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) respectively. The PSNR ranges between 54.15 (Foster City) to 31.54 (Desert) dB for the payload (bpB) range [0.5, 3]. The minimum MSE obtained is 0.24 for “Foster City” at 0.5 bpB and that of maximum value obtained is 45.56 for “Desert” at 3 bpB. Similarly, IF ranges between [0.999992 (Airplane) - 0.993754 (Desert)], SSIM ranges between [0.999871 (San Diego), 0.923728 (Splash)] and UIQ ranges between [0.999255 (San Diego) - 0.628142 (Splash)] respectively. In general, lower the PSNR and higher MSE, IF, SSIM and UIQ with respect to varying payload ensures higher perceptibility of the watermarked images. Since, the WBT_1X2_QE is heterogeneous with wide scale of variability and hence, conclusions are drawn on the basis of appropriate images. The average values are also computed against the above mentioned metrics at variable payload to summarize the experimental results.

Table 7.11. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WBT_1x2_QE scheme

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Lena	0.5	54.145421	0.250344	0.999984	0.999567	0.994298
	1.0	49.422982	0.742646	0.999953	0.997964	0.984533
	1.5	45.435007	1.860285	0.999884	0.996522	0.963332
	2.0	42.502457	3.654556	0.999778	0.991507	0.935129
	2.5	38.358550	9.489106	0.999432	0.982354	0.868211
	3.0	35.703722	17.486708	0.998943	0.963741	0.815873
Baboon	0.5	54.142422	0.250517	0.999986	0.999850	0.998909
	1.0	49.322226	0.760077	0.999959	0.999290	0.996941
	1.5	45.520421	1.824055	0.999904	0.998674	0.991608
	2.0	42.490234	3.664855	0.999810	0.996826	0.985749
	2.5	38.551545	9.076653	0.999530	0.992977	0.962495
	3.0	35.852948	16.896060	0.999129	0.986362	0.949688
Pepper	0.5	53.876715	0.266323	0.999977	0.999142	0.988760
	1.0	46.422549	1.481922	0.999862	0.988717	0.971649
	1.5	42.116044	3.994623	0.999652	0.982054	0.951360
	2.0	40.196147	6.215349	0.999514	0.976348	0.931294
	2.5	35.387672	18.806720	0.998397	0.957892	0.867860
	3.0	35.387672	18.806720	0.998397	0.957892	0.867860
Airplane	0.5	54.138654	0.250734	0.999992	0.999486	0.982639
	1.0	49.423420	0.742571	0.999978	0.997591	0.955813
	1.5	45.210806	1.958843	0.999943	0.995753	0.908061
	2.0	42.368889	3.768698	0.999891	0.989879	0.859009
	2.5	38.079958	10.117762	0.999709	0.976556	0.753247
	3.0	35.558079	18.083080	0.999478	0.955148	0.693007
Sailboat	0.5	54.110007	0.252394	0.999987	0.999685	0.995188
	1.0	49.095326	0.800843	0.999959	0.998367	0.987344
	1.5	45.045569	2.034807	0.999897	0.997199	0.970370
	2.0	42.193501	3.924011	0.999803	0.993173	0.951641
	2.5	38.112927	10.041245	0.999497	0.985650	0.902647
	3.0	35.380402	18.838230	0.999059	0.970526	0.866289
Earth	0.5	54.155713	0.249752	0.999985	0.999672	0.997370
	1.0	49.456888	0.736871	0.999956	0.998448	0.992869
	1.5	45.562043	1.806658	0.999891	0.997366	0.982804
	2.0	42.530015	3.631439	0.999777	0.993553	0.967947

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	38.702135	8.767319	0.999466	0.986735	0.925858
	3.0	35.801090	17.099021	0.998936	0.971993	0.886154
San Diego	0.5	54.137201	0.250818	0.999990	0.999871	0.999255
	1.0	49.422528	0.742724	0.999972	0.999407	0.997982
	1.5	45.638023	1.775325	0.999932	0.999090	0.995444
	2.0	42.548258	3.616217	0.999861	0.997785	0.991760
	2.5	38.939289	8.301401	0.999682	0.995533	0.981178
	3.0	35.965774	16.462771	0.999363	0.990430	0.970813
Splash	0.5	54.000320	0.258850	0.999977	0.998787	0.965256
	1.0	47.299330	1.211008	0.999879	0.987145	0.932784
	1.5	42.638653	3.541726	0.999644	0.981135	0.878412
	2.0	40.507256	5.785683	0.999442	0.972816	0.821446
	2.5	35.815189	17.043604	0.998344	0.953023	0.703651
	3.0	33.655747	28.022420	0.997331	0.923728	0.628142
Oakland	0.5	54.064019	0.255081	0.999986	0.999721	0.998422
	1.0	48.289669	0.964080	0.999942	0.998432	0.995446
	1.5	43.925624	2.633406	0.999837	0.997338	0.990496
	2.0	41.349281	4.765979	0.999706	0.994647	0.984639
	2.5	37.132205	12.585189	0.999214	0.988554	0.960999
	3.0	34.569153	22.707272	0.998578	0.977916	0.941792
Foster City	0.5	54.156509	0.249706	0.999991	0.999422	0.992571
	1.0	49.501043	0.729417	0.999974	0.997207	0.980486
	1.5	45.362868	1.891443	0.999932	0.995284	0.953022
	2.0	42.457757	3.692365	0.999863	0.988844	0.921662
	2.5	38.507402	9.169384	0.999670	0.976319	0.831993
	3.0	35.813524	17.050139	0.999366	0.952509	0.768612
Anhinga	0.5	52.081071	0.402692	0.999969	0.999698	0.882230
	1.0	48.553767	0.907201	0.999930	0.999048	0.878763
	1.5	42.096374	4.012757	0.999689	0.997646	0.853355
	2.0	40.504523	5.789325	0.999555	0.995977	0.846341
	2.5	35.570508	18.031402	0.998616	0.988848	0.796408
	3.0	33.759829	27.358824	0.997919	0.980477	0.785326
Athens	0.5	51.881060	0.421671	0.999966	0.999551	0.970407
	1.0	49.219801	0.778216	0.999937	0.998957	0.966760
	1.5	43.427115	2.953716	0.999763	0.997002	0.934497
	2.0	40.944049	5.232092	0.999581	0.995306	0.925277

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	36.800734	13.583351	0.998917	0.985785	0.848380
	3.0	33.783735	27.208635	0.997843	0.977868	0.833153
Bardowl	0.5	52.530426	0.363109	0.999963	0.999768	0.998743
	1.0	47.790922	1.081404	0.999890	0.998969	0.996695
	1.5	42.917713	3.321306	0.999663	0.997446	0.987350
	2.0	40.603659	5.658671	0.999436	0.996135	0.983928
	2.5	35.934682	16.581053	0.998326	0.986298	0.940099
	3.0	33.409324	29.658410	0.997064	0.980399	0.929953
Barnfall	0.5	53.104854	0.318122	0.999950	0.999696	0.998398
	1.0	49.326944	0.759251	0.999880	0.998758	0.996527
	1.5	44.518158	2.297547	0.999618	0.997595	0.988677
	2.0	42.134568	3.977621	0.999373	0.994487	0.981777
	2.5	37.533952	11.473213	0.998156	0.986902	0.947680
	3.0	35.299630	19.191870	0.997058	0.975722	0.929779
Butrfly	0.5	52.514547	0.364439	0.999974	0.999703	0.995016
	1.0	49.101758	0.799658	0.999943	0.999161	0.992913
	1.5	43.944819	2.621793	0.999811	0.997778	0.979531
	2.0	41.189277	4.944844	0.999660	0.995876	0.972219
	2.5	37.157337	12.512571	0.999135	0.989010	0.925682
	3.0	34.083210	25.395650	0.998322	0.982247	0.910287
Bobcat	0.5	52.161319	0.395319	0.999944	0.99954	0.749370
	1.0	49.393287	0.747741	0.999894	0.998862	0.747211
	1.5	43.609700	2.832111	0.999588	0.996480	0.731360
	2.0	41.125139	5.018413	0.999304	0.993409	0.725839
	2.5	36.845700	13.443438	0.998119	0.984379	0.687809
	3.0	34.008016	25.839176	0.996580	0.969668	0.677063
Bodie	0.5	53.142233	0.315396	0.999946	0.999759	0.979354
	1.0	47.720609	1.099054	0.999807	0.998815	0.975527
	1.5	42.970078	3.281500	0.999424	0.997113	0.962939
	2.0	40.980414	5.188465	0.999116	0.994822	0.956949
	2.5	35.692436	17.532213	0.996977	0.981386	0.915865
	3.0	33.935035	26.277061	0.995582	0.970653	0.900881
Bluheron	0.5	52.656006	0.352760	0.999956	0.999649	0.995693
	1.0	49.623876	0.709075	0.999913	0.998578	0.992728
	1.5	43.676377	2.788962	0.999659	0.996887	0.977700
	2.0	41.678973	4.417565	0.999460	0.993780	0.967755

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	36.704722	13.886988	0.998297	0.984942	0.911029
	3.0	35.321056	19.097422	0.997660	0.974911	0.887589
Colomtn	0.5	53.079946	0.319952	0.999974	0.999730	0.984703
	1.0	49.026124	0.813706	0.999936	0.998820	0.982228
	1.5	43.139789	3.155741	0.999754	0.997553	0.968905
	2.0	41.275642	4.847480	0.999621	0.994711	0.961700
	2.5	36.491217	14.586752	0.998860	0.987824	0.921934
	3.0	34.738936	21.836686	0.998290	0.978674	0.908653
Desert	0.5	52.888865	0.334344	0.999952	0.999715	0.998172
	1.0	44.970361	2.070351	0.999675	0.996624	0.992990
	1.5	40.040329	6.442395	0.999023	0.992868	0.981407
	2.0	38.763757	8.643798	0.998741	0.990751	0.978198
	2.5	33.028415	32.377168	0.995275	0.972206	0.933545
	3.0	31.544109	45.568987	0.993754	0.964575	0.926171
Average case	0.5	53.348370	0.306116	0.999972	0.999601	0.973238
	1.0	48.619170	0.933891	0.999912	0.997458	0.965909
	1.5	43.839780	2.851450	0.999725	0.995439	0.947532
	2.0	41.417190	4.821871	0.999565	0.992032	0.932513
	2.5	36.967330	13.87033	0.998681	0.982159	0.879329
	3.0	34.678550	22.94426	0.997933	0.970272	0.853854

Table 7.12 reveals the average PSNR computations against varying payload (0.5 – 3 bpB) to measure the image quality subsequent to embedding. The average PSNR values of WBT_1x2 lies in between 33.88 to 53.34 dB and that of average PSNR of WBT_1x2_QE lies in the range of 34.67 – 53.34 dB respectively. The average PSNR is almost identical with respect to 0.5, 1 and 1.5 bpB respectively however, on exceeding this payload limit, the improvement of average PSNR is clearly observed. The average PSNR value is above 30 dB at 3 bpB which revealed that the quality of the watermarked image is highly perceptible [148].

Table 7.12. Comparative analysis of obtained average PSNR values between WBT_1x2 and WBT_1x2_QE with respect to increasing payload

WBT_1x2		WBT_1x2_QE	
Payload (bpB)	PSNR (dB)	Payload (bpB)	PSNR (dB)
0.5	53.34837	0.5	53.34837
1.0	48.61917	1.0	48.61917
1.5	43.83774	1.5	43.83978

WBT_1x2		**WBT_1x2_QE**	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
2.0	41.00855	2.0	41.41719
2.5	36.66372	2.5	36.96733
3.0	33.88552	3.0	34.67855

Two existing embedding schemes namely, Discrete Pascal Transform based data hiding scheme (abbreviated as DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (abbreviated as DGTDHS) [129] are analyzed in terms of PSNR values however, the major limitation of both schemes are their fixed payloads of 0.25 and 1 bpB respectively. To address the issue of fixed payload (which is also treated as very low), the WBT_1X2_QE is designed with an elegant way of supporting variable payload for a spread from 0.5 to 3 bpB. It is also to be noted that the PSNR values obtained throughout the experiments always resides above 30 dB and hence, an acceptable level of visual imperceptibility is achieved [148]. In contrast to DPTHDI [88], the WBT_1x2_QE ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads for "Lena", 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Baboon", 0.5, 1, 1.5 and 2 bpB of payloads for "Pepper", 0.5, 1, 1.5 and 2 bpB of payloads for "Airplane" and 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for "Sailboat" respectively. In contrast to DGTDHS [129], proposed WBT_1x2_QE ensured better PSNR values at 1 bpB of payload for "Lena", "Baboon", "Airplane" and "Sailboat" images and supports variable payload (i.e., up to 3 bpB).

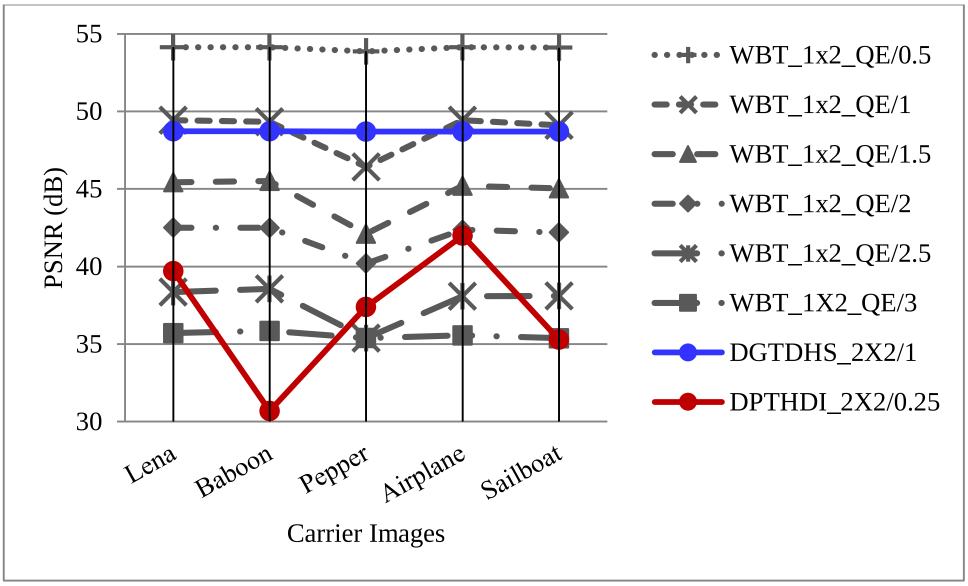


Fig. 7.11. Performance analysis of PSNR (dB) for variable payload based WBT_1x2_QE and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

On application of adaptive quality enhancement in WBT_1x2 of section 4.2.2, the concept of WBT_1x2_QE is emerged. Fig. 7.12 deals with the analysis of average PSNR values for WBT_1x2_QE, WBT_1x2, WBT_2x2_QE, WLT_1x2_QE, WDHT_1x2_QE, DPTHDI [88], and DGTDHS [129] respectively. The average PSNR of WBT_1x2_QE is gradually improving over WBT_1x2 while the payload exceeds 1.5 bpB; though, the improvement is really very less. The average PSNR values as obtained from WBT_1x2_QE is also higher than WDHT_1x2_QE, WLT_1x2_QE, WBT_2x2_QE and WBT_1x2 techniques with respect to 0.5 to 3 bpB of payload respectively. In DPTHDI [88], the average PSNR value of 37.40 dB is computed at 0.25 bpB of payload from six color images viz. “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” respectively. Similarly, the average PSNR value of 48.70 dB is computed at 1 bpB of payload for DGTDHS [129] which is obtained by considering the benchmark images viz. “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” respectively. Compared to DPTHDI [88], the WBT_1x2_QE scheme offers higher PSNR values for the payload variation of 0.5, 1, 1.5 and 2 bpB respectively. In contrast to DGTDHS [129], the WBT_1x2_QE offered almost identical PSNR (dB) at 1 bpB while the latter scheme also supports fabrication of secret information with variable payload. Since, the average PSNR values of WBT_1x2_QE scheme are above 30 dB for the given benchmark images of fig 1.1, the watermarked images as obtained throughout the experiments are retaining good visual clarity [148].

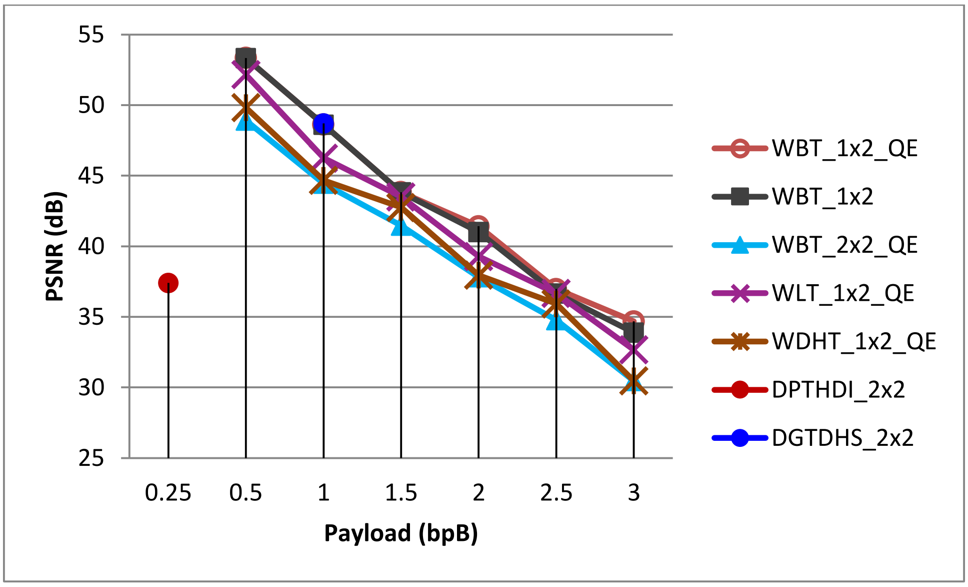


Fig. 7.12. Graphical representation of variation of average PSNR (dB) with respect to payload for WBT_1x2_QE, WBT_1x2, WBT_2x2_QE, WLT_1x2_QE, WDHT_1x2_QE and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

7.3.4. Quality Enhancement of Stirling Transform (ST) based watermarking

As discussed in chapter 5, the Stirling Transform (ST) based watermarking has been classified into two categories: 2 x 2 block based watermark fabrication (WST_2x2) and 1 x 2 block based watermark fabrication (WST_1x2).

Section 7.3.4.1 deals with the 2 x 2 block based watermark fabrication followed by quality enhancement (WST_2x2_QE) whereas in section 7.3.4.2, the 1 x 2 block based watermark fabrication followed by quality enhancement (WST_1x2_QE) has been discussed.

7.3.4.1 Quality Enhancement for 2 x 2 Block based Watermark Fabrication

The adaptive quality enhancement of section 7.2.1 has been introduced as the post-embedding operation in WST_2x2 scheme (section 5.2.1). The 1 x 2 block based decomposition is carried out for embedding secret information with a pre-specified embedding rule. However, to address the problem of overflow and underflow, the pixel components of each block are adjusted based on the pre-embedding adjustment policy as discussed in equation (5.7) of section 5.2.1.1. Stirling Transform (ST) converts each 2 x 2 sub-matrix of pixel components into transform domain. Secret bits corresponding to the message digest, size and content of the watermark bits are fabricated in varying proportion on first/third/fourth transformed component of a 2 x 2 sub-matrix starting from the least significant bit position (i.e., LSB-0) toward higher order bit position. To minimize the quality distortion, the adaptive quality enhancement of section 7.2.1 has been used in such way that the embedded bits of the transformed components are kept unaffected. Inverse Stirling Transform (IST) is applied over each 2 x 2 sub-matrices of quality enhanced components to obtain the pixel components in spatial domain. The process is repeated until and unless the entire secret information is concealed and the watermarked image is produced.

Section 7.3.4.1.1 describes the WST_2x2_QE process with an example. Simulation results, performance analysis and the discussions have been elaborated in section 7.3.4.1.2.

7.3.4.1.1 Example

Consider the 2 x 2 sub-matrices of red, green and blue channel for watermark fabrication. Suppose, the payload value is 3 bpB and the pixel components are pre-adjusted as discussed in equation (5.7) of section 5.2.1.1. The 2 x 2 adjusted sub-matrices are as follows:

$$R_1 = \begin{bmatrix} 212 & 198 \\ 32 & 45 \end{bmatrix} G_1 = \begin{bmatrix} 97 & 32 \\ 224 & 32 \end{bmatrix} B_1 = \begin{bmatrix} 156 & 166 \\ 118 & 65 \end{bmatrix}$$

Stirling Transform (ST) is applied on each 2 x 2 sub-matrix to convert it from spatial domain into transform domain. The 2 x 2 sub-matrices of transformed components such as T(R_1), T(G_1) and T(B_1) are obtained as follows:

$$T(R_1) = \begin{bmatrix} 212 & 410 \\ 838 & 1835 \end{bmatrix} T(G_1) = \begin{bmatrix} 97 & 129 \\ 417 & 1697 \end{bmatrix} T(B_1) = \begin{bmatrix} 156 & 322 \\ 772 & 2091 \end{bmatrix}$$

Let the watermark bit stream *"010000010110010011111001011011010101"* is to be fabricated into the transformed components based on the embedding rule as given in equation (5.8) of section 5.2.1.1. In this example, the payload is 3 bits per Byte which means four bits are fabricated (λ_1 = 4, λ_2 = 4, λ_3 = 4) on first/third/fourth transformed component starting from LSB-0 toward higher order bit position. It has been observed that a small perturbation on the first transformed component affects other components significantly. Therefore, an additional re-adjustment operation is performed on the second transformed component by adding the difference of the pre-embedding and post-embedding value of the first component. Hence, the 2 x 2 sub-matrices of embedded components are obtained as follows:

$$T'(R_1) = \begin{bmatrix} 210 & 408 \\ 840 & 1830 \end{bmatrix} T'(G_1) = \begin{bmatrix} 98 & 130 \\ 431 & 1705 \end{bmatrix} T'(B_1) = \begin{bmatrix} 150 & 316 \\ 779 & 2090 \end{bmatrix}$$

The adaptive quality enhancement of section 7.2.1 has been used to ensure the enhancement of quality without hampering the embedded bits. Basically, it is applied on each embedded component by taking the closest value of the pre-embedded component without affecting the embedded bits. The 2 x 2 sub-matrices after quality enhancement become:

$$T''(R_1) = \begin{bmatrix} 210 & 408 \\ 840 & 1830 \end{bmatrix} T''(G_1) = \begin{bmatrix} 98 & 130 \\ 415 & 1705 \end{bmatrix} T''(B_1) = \begin{bmatrix} 150 & 316 \\ 779 & 2090 \end{bmatrix}$$

The application of Inverse Stirling Transform (ST) on 2 x 2 sub-matrices of embedded components yields the 2 x 2 sub-matrices of pixel components as given below:

$$R'_1 = \begin{bmatrix} 210 & 198 \\ 36 & 18 \end{bmatrix} G'_1 = \begin{bmatrix} 98 & 32 \\ 221 & 7 \end{bmatrix} B'_1 = \begin{bmatrix} 150 & 166 \\ 131 & 8 \end{bmatrix}$$

It is observed that the modified pixel components are non-fractional, non-negative, less than or equal to 255. The re-adjustment operation has not been used in this example.

7.3.4.1.2 Results and Discussions

The WST_2x2_QE scheme analyzes the performance of watermarked quality in terms of the following metrics: peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ). The experiment deals with the fabrication variable sizes of Gold-Coin into twenty benchmark (BMP) images [130, 131] of dimension 512 × 512. In proposed WST_2x2_QE scheme, the minimum value of PSNR is 25.24 dB at 3 bpB of payload for the "Desert" image and that of the maximum value of PSNR is obtained as 54.55 dB at 0.5 bpB of payload for the "Bluheron" image. On contrary, the maximum value of MSE is 181.40 for "Desert" at 3 bpB and that of minimum value of 0.22 is achieved for "Bluheron" at payload of 0.5 bpB respectively. In order to represent the minimum and maximum values of IF, SSIM and UIQ, the permissible ranges of values as obtained from the experimental results of twenty images are summarized as follows: [0.972702 (Desert), 0.999992 (Airplane)], [0.874988 (Bobcat), 0.99998 (Athens)] and [0.410209 (Splash), 0.999331 (San Diego)] respectively. Experimental results given in table 7.13 ensure that the IF, SSIM and UIQ are minimum at 3 bpB and minimum at 0.5 bpB. Since, the values are very close to one, high similarity between the watermarked and original images are observed. The average values are computed for various metrics of twenty carrier images at variable payload to summarize the experimental results.

Table 7.13. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WST_2x2_QE scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.164743	0.249233	0.999984	0.999684	0.994767
	1.0	47.606940	1.128200	0.999928	0.999065	0.975113
	1.5	43.655541	2.802374	0.999823	0.997313	0.944830
	2.0	38.472595	9.243166	0.999427	0.989939	0.862841
	2.5	34.660643	22.233918	0.998602	0.982496	0.760561
	3.0	30.377307	59.614189	0.996425	0.943190	0.625750
Baboon	0.5	54.143899	0.250432	0.999986	0.999885	0.999015
	1.0	47.598109	1.130496	0.999940	0.999667	0.995116
	1.5	43.619941	2.825440	0.999850	0.998989	0.987788
	2.0	38.369371	9.465492	0.999501	0.995398	0.961294
	2.5	34.583443	22.632678	0.998804	0.992757	0.933175
	3.0	30.323337	60.359629	0.996839	0.974384	0.877444
Pepper	0.5	53.611422	0.283098	0.999975	0.998541	0.987707
	1.0	45.834190	1.696919	0.999847	0.990865	0.966292
	1.5	40.859082	5.335463	0.999506	0.981247	0.936526
	2.0	35.002534	20.550715	0.998059	0.961591	0.858718
	2.5	32.918886	33.204100	0.997010	0.954647	0.767010
	3.0	27.967882	103.822650	0.990230	0.899309	0.632939

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	54.124779	0.251537	0.999992	0.999625	0.983309
	1.0	47.606157	1.128403	0.999967	0.998888	0.933771
	1.5	43.650468	2.805650	0.999919	0.996822	0.873359
	2.0	38.677709	8.816767	0.999747	0.988599	0.751729
	2.5	34.700781	22.029374	0.999369	0.979784	0.634117
	3.0	31.458040	46.481091	0.998668	0.946490	0.534571
Sailboat	0.5	54.096070	0.253205	0.999987	0.999741	0.995591
	1.0	47.606157	1.128403	0.999967	0.998888	0.933771
	1.5	43.650468	2.805650	0.999919	0.996822	0.873359
	2.0	38.677709	8.816767	0.999747	0.988599	0.751729
	2.5	34.700781	22.029374	0.999369	0.979784	0.634117
	3.0	31.458040	46.481091	0.998668	0.946490	0.534571
Earth	0.5	54.167048	0.2491010	0.999985	0.999763	0.997650
	1.0	47.614754	1.126172	0.999933	0.999287	0.988308
	1.5	43.692068	2.778903	0.999835	0.997964	0.972378
	2.0	38.733780	8.703667	0.999482	0.992459	0.921579
	2.5	34.703945	22.013333	0.998693	0.986671	0.843943
	3.0	30.935115	52.428663	0.996728	0.958902	0.733019
San Diego	0.5	54.167580	0.249070	0.999990	0.999911	0.999331
	1.0	47.623290	1.123961	0.999958	0.999732	0.996761
	1.5	43.700215	2.773695	0.999896	0.999232	0.992542
	2.0	38.760058	8.651163	0.999677	0.997219	0.979112
	2.5	34.745386	21.804279	0.999187	0.995005	0.955384
	3.0	31.515186	45.873483	0.998289	0.985884	0.920403
Splash	0.5	53.781897	0.272201	0.999975	0.997960	0.965398
	1.0	46.358052	1.504094	0.999857	0.989555	0.907801
	1.5	41.719053	4.376984	0.999571	0.982329	0.839964
	2.0	36.232999	15.480336	0.998451	0.967405	0.697914
	2.5	33.574108	28.554167	0.997313	0.956315	0.557933
	3.0	29.124220	79.553328	0.992109	0.901439	0.410209
Oakland	0.5	53.978528	0.260152	0.999985	0.999776	0.998472
	1.0	46.981197	1.303048	0.999925	0.999147	0.993423
	1.5	42.620753	3.556354	0.999789	0.997579	0.984600
	2.0	37.204543	12.377302	0.999241	0.991821	0.956242
	2.5	34.039998	25.649594	0.998539	0.987158	0.909725
	3.0	29.995311	65.095236	0.996027	0.963307	0.839016

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Foster City	0.5	54.174437	0.248677	0.999991	0.999581	0.993350
	1.0	47.618453	1.125213	0.999959	0.998700	0.967249
	1.5	43.694612	2.777276	0.999901	0.996274	0.926938
	2.0	38.785216	8.601193	0.999693	0.986796	0.823027
	2.5	34.893329	21.074019	0.999243	0.976976	0.702599
	3.0	31.626582	44.711797	0.998401	0.937088	0.582447
Anhinga	0.5	53.822131	0.269691	0.999979	0.999952	0.883977
	1.0	47.497999	1.156859	0.999911	0.999661	0.873021
	1.5	42.987611	3.268280	0.999749	0.998854	0.865023
	2.0	36.879956	13.337814	0.998979	0.993705	0.816559
	2.5	33.747012	27.439683	0.997893	0.985932	0.755968
	3.0	29.799344	68.099811	0.994787	0.964348	0.700139
Athens	0.5	54.102421	0.252835	0.999979	0.999980	0.972700
	1.0	49.100087	0.799966	0.999935	0.999736	0.960673
	1.5	45.864738	1.685025	0.999865	0.999187	0.952753
	2.0	39.813246	6.788216	0.999457	0.995000	0.888902
	2.5	34.518053	22.976033	0.998165	0.984873	0.781586
	3.0	32.256709	38.673135	0.996905	0.968062	0.723337
Bardowl	0.5	53.810332	0.270425	0.999972	0.999913	0.999162
	1.0	47.397139	1.184040	0.999880	0.999482	0.995687
	1.5	43.234332	3.087785	0.999684	0.998070	0.987958
	2.0	36.697878	13.908891	0.998565	0.986800	0.945339
	2.5	33.642329	28.109133	0.997154	0.983542	0.911960
	3.0	28.435207	93.230836	0.990371	0.935096	0.819891
Barnfall	0.5	54.282906	0.242543	0.999962	0.999855	0.998867
	1.0	47.219026	1.233609	0.999803	0.999340	0.992750
	1.5	43.648786	2.806737	0.999545	0.998138	0.984718
	2.0	38.448778	9.293996	0.998458	0.992833	0.956698
	2.5	34.570415	22.700674	0.996353	0.987305	0.902423
	3.0	30.757878	54.612553	0.990674	0.954452	0.806652
Butrfly	0.5	54.125218	0.251511	0.999982	0.999940	0.996048
	1.0	48.550518	0.907880	0.999935	0.999658	0.989900
	1.5	45.203438	1.962169	0.999860	0.999047	0.982385
	2.0	39.459596	7.364117	0.999476	0.995681	0.947836
	2.5	34.618833	22.448998	0.998396	0.989410	0.872436
	3.0	32.057226	40.490922	0.997094	0.974588	0.812653

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	54.068113	0.254840	0.999964	0.999913	0.750419
	1.0	48.839988	0.849339	0.999880	0.999486	0.743493
	1.5	45.534373	1.818205	0.999744	0.998480	0.736392
	2.0	39.220494	7.780919	0.998901	0.991635	0.699693
	2.5	35.063720	20.263215	0.997151	0.985278	0.661569
	3.0	26.804387	135.719300	0.980812	0.874988	0.545352
Bodie	0.5	54.100674	0.252937	0.999957	0.999898	0.980116
	1.0	46.321950	1.516649	0.999736	0.999337	0.971582
	1.5	42.177965	3.938073	0.999309	0.997715	0.960156
	2.0	35.819449	17.026892	0.996966	0.983279	0.914658
	2.5	33.557218	28.665435	0.995048	0.980353	0.884113
	3.0	27.500389	115.622252	0.979113	0.897048	0.748646
Bluheron	0.5	54.552339	0.227953	0.999972	0.999860	0.996625
	1.0	47.592680	1.1319109	0.999861	0.999306	0.987634
	1.5	44.404528	2.358454	0.999711	0.998244	0.976203
	2.0	38.479582	9.228308	0.998870	0.992972	0.920191
	2.5	35.039042	20.378684	0.997506	0.988169	0.853488
	3.0	31.507495	45.954795	0.994367	0.962549	0.745324
Colomtn	0.5	54.155071	0.249788	0.999980	0.999863	0.985545
	1.0	47.172128	1.247002	0.999902	0.999471	0.978875
	1.5	42.938386	3.305534	0.999741	0.998432	0.970385
	2.0	37.331260	12.021378	0.999060	0.993107	0.932870
	2.5	33.995021	25.916613	0.997975	0.986988	0.879366
	3.0	30.085361	63.759403	0.995025	0.961163	0.815657
Desert	0.5	53.083555	0.319686	0.999953	0.999696	0.998276
	1.0	44.489379	2.312822	0.999646	0.997611	0.991855
	1.5	39.472506	7.342258	0.998863	0.992941	0.981555
	2.0	33.001206	32.580649	0.994992	0.974110	0.934102
	2.5	31.701561	43.946489	0.993397	0.971159	0.914800
	3.0	25.544412	181.400756	0.972702	0.896627	0.795102
Average case	0.5	54.02566	0.257946	0.999978	0.999667	0.973816
	1.0	47.33141	1.236749	0.999889	0.998344	0.957154
	1.5	43.31644	3.220515	0.999704	0.996184	0.936491
	2.0	37.7034	12.00189	0.998837	0.987947	0.876052
	2.5	34.19873	25.20349	0.997758	0.98173	0.805814
	3.0	29.97647	72.09925	0.992712	0.94227	0.710156

A comparative analysis is made between WST_2x2 and WST_2x2_QE based on two primary metrics: payload (bpB) and average PSNR (dB). The average PSNR of WST_2x2 ranges from 28.57 to 54.02 dB for a spread of 0.5 to 3 bpB however, the WST_2x2_QE deals with equal or improved values of average PSNR for that of variable payload as evident from table 7.14. Therefore, the adaptive quality enhancement ensures improvement in PSNR for the proposed quality enhanced technique (WST_2x2_QE) over the WST_2x2 technique in terms of average PSNR (dB) as the payload exceeds 0.5 bpB. Since, the PSNR values above 30 dB is treated as the good quality, the watermarked images obtained in WST_2x2_QE is also considered as highly perceptible [148].

Table 7.14. Comparative analysis of obtained average PSNR values between WST_2x2 and WST_2x2_QE with respect to increasing payload

WST_2x2		WST_2x2_QE	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	54.02566	0.5	54.02566
1.0	46.24363	1.0	47.33141
1.5	42.08165	1.5	43.31644
2.0	36.99384	2.0	37.70340
2.5	33.30990	2.5	34.19873
3.0	28.57277	3.0	29.97647

An analysis has been made among the proposed 2 x 2 block based watermarking WST_2x2_QE, Varsaki et al.'s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] as well as Discrete Gould Transform based data hiding scheme (DGTDHS) [129] in terms of peak signal to noise ratio (dB) and payload (bpB). The comparison has been done for five different color images such as "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat". It is observed from table 7.13 that the PSNR of the watermarked images for Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes are computed with respect to 0.25 and 1 bpB of payloads respectively. The major limitation of the DPTHDI [88] and DGTDHS [129] schemes are its fixed as well as low payload. Unlikely, the WST_2x2_QE scheme is focused on variable payload and achieves acceptable visual imperceptibility of higher than 30 dB [148] for "Lena", "Baboon", "Airplane" and "Sailboat" respectively. In contrast to DPTHDI [88], proposed WST_2x2_QE scheme ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for "Lena", 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Baboon", 0.5, 1 and 1.5 bpB of payloads for "Pepper", 0.5, 1 and 1.5 bpB of payloads for "Airplane" and 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for “Sailboat” respectively. In contrast to DGTDHS [129], the PSNR values of the respective images for WST_2x2_QE is lacking at 1 bpB however, the latter one supports variable payload (bpB) for the range (0.5 – 3).

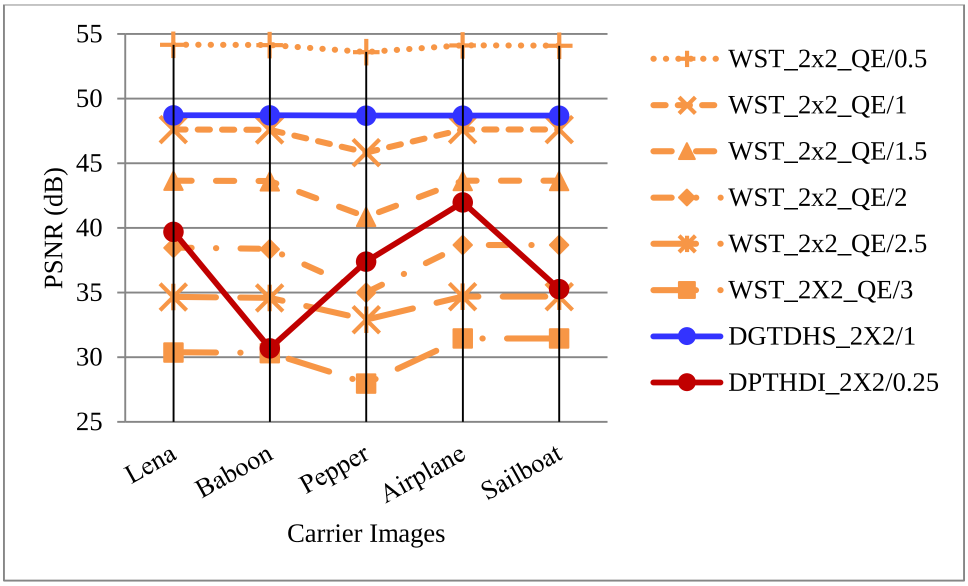


Fig. 7.13. Performance analysis of PSNR (dB) for variable payload based WST_2x2_QE and fixed payload based Varsaki et al.’s DPTHDI [88] and DGTDHS [129] schemes with respect to five color images

The line chart of fig. 7.14 is analyzed to show the variation of average PSNR for WST_2x2_QE, WBT_2x2_QE, WLT_2x2_QE, WDHT_2x2_QE, DPTHDI [88] and DGTDHS [129] respectively. In contrast to WST_2x2, the improvement of PSNR (in actual sense quality) for WST_2x2_QE can be visually perceived as the payload increases from 0.5 bpB. In contrast to WBT_2x2_QE, WLT_2x2_QE and WDHT_2x2_QE, the superiority of the WST_2x2_QE has also been observed from the line chart. The average PSNR value 37.40 dB is obtained for DPTHDI [88] by averaging the PSNR values of “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images at 0.25 bpB of payload. In contrast to DPTHDI [88], the average PSNR of WST_2x2_QE ensures equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads. The average PSNR for DGTDHS [129] scheme is computed as 48.70 dB at 1 bpB which is obtained by considering “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” respectively. The average PSNR of WST_2x2_QE slightly suffers than DGTDHS [129] at 1 bpB of payload, however, the ability of providing variable payload that offers a spread from 0.5 to 3 bpB is considered to be the significant improvement. The overall quality distortion of WST_2x2_QE scheme at varying payload (i.e., 0.5 to 3 bpB) falls above the perceptible quality level [148].

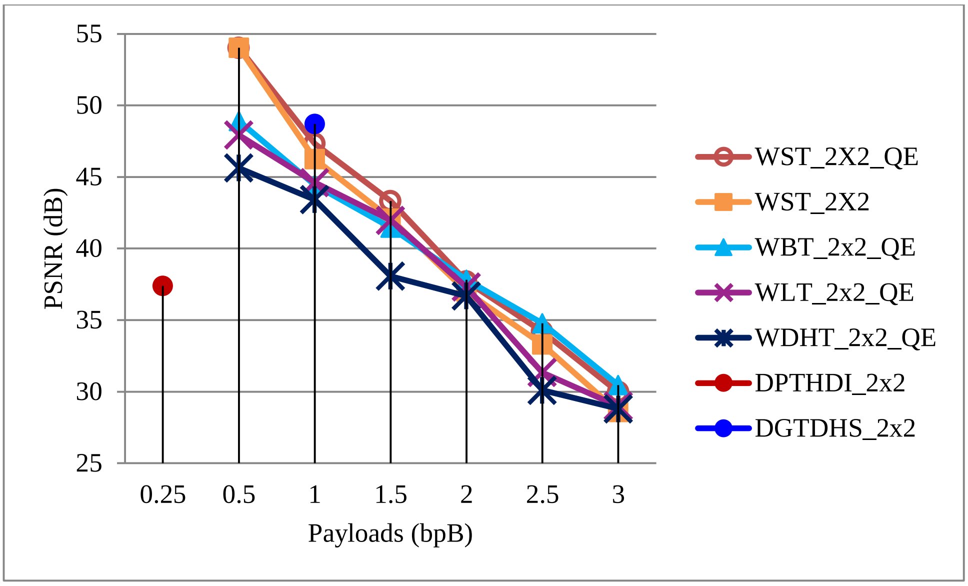


Fig. 7.14. Graphical representation of variation of average PSNR (dB) with respect to payload for WST_2x2_QE, WST_2x2, WBT_2x2_QE, WLT_2x2_QE, WDHT_2x2_QE and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

7.3.4.2 Quality Enhancement for 1 x 2 Block based Watermark Fabrication

Decomposition of carrier image into 1 x 2 overlapping blocks yields the pair of pixel components which in turn are adjusted through the pixel adjustment process. On application of Stirling Transform (ST) over each 1 x 2 sub-matrix of pixel components, transformed components are obtained with identical block sizes. Secret bits corresponding to the message digest, size and content of the watermark is scrambled based on Arnold's cat map and are fabricated into the transformed components of 1 x 2 sub-matrices starting from the least significant bit position (i.e., LSB-0) toward higher order bit position. The WST_1x2_QE is different from that of WST_1x2 since, an adaptive quality enhancement of section 7.2.1 has been applied on the embedded components to minimize the quality degradation. The fabricated bits of the enhanced components are kept unaltered during the quality enhancement process. Inverse Stirling Transform (IST) is applied over 1 x 2 sub-matrices of enhanced components to obtain the embedded 1 x 2 sub-matrices of pixel components in spatial domain. Successively the entire secret information is concealed into the cover image in transform domain and the watermarked image is produced. The usage of Arnold's cat map ensured an additional level of security for the fabricated watermark information.

Proposed scheme has been explained with a suitable example in section 7.3.4.2.1. Experimental results along with the comparative analysis and discussions are elaborated in section 7.3.4.2.2.

7.3.4.2.1 Example

Consider the 1 x 2 sub-matrices of red, green and blue channel for watermark fabrication. Suppose, the payload is 3 bpB and the pixel components are adjusted prior to embedding as discussed in equation (5.22) of section 5.2.2.2. The 1 x 2 sub-matrices are as given below:

$$R_1 = [164\ 16] \quad G_1 = [240\ 57] \quad B_1 = [71\ 31]$$

Stirling Transform (ST) is applied on each 1 x 2 sub-matrix or pair of pixel components to convert it from spatial domain into transform domain. Hence, the 1 x 2 sub-matrices of transformed components viz. T(R_1), T(G_1) and T(B_1) are obtained as follows:

$$T(R_1) = [164\ 180] \quad T(G_1) = [240\ 297] \quad T(B_1) = [71\ 102]$$

Consider the watermark bit stream of *"011101110011100010"* which is to be fabricated into the pairs of transformed components based on the embedding rule of equation (5.25) of section 5.2.2.1. For the payload value of 3 bpB, three bits are fabricated ($\lambda = \lambda_1 = \lambda_2 = 3$) on both components starting from LSB-0 toward the higher order bit positions. Hence, the 1 x 2 sub-matrices of embedded components in transform domain becomes:

$$T'(R_1) = [166\ 181] \quad T'(G_1) = [243\ 302] \quad T'(B_1) = [65\ 98]$$

The adaptive quality enhancement of section 7.2.1 is applied on each embedded component by taking the closest value of the pre-embedded component without hampering the embedded bits i.e., LSB-0, LSB-1 and LSB-2. The 1 x 2 enhanced sub-matrices are computed as follows:

$$T''(R_1) = [166\ 181] \quad T''(G_1) = [243\ 294] \quad T''(B_1) = [73\ 98]$$

Applying inverse Stirling Transform (IST) on each 1 x 2 sub-matrices of quality enhanced components yields the 1 x 2 sub-matrices of pixel components corresponding to red, green and blue channels as follows:

$$R'_1 = [166\ 15] \quad G'_1 = [243\ 51] \quad B'_1 = [73\ 21]$$

The pixel components are bound to the range [0, 255] and no fractional components are obtained at any circumstances.

7.3.4.2.2 Results and Discussions

Table 7.15 reveals the results of standard quality metrics against the variation of payload for a range from 0.5 to 3 bpB. Proposed WST_1x2_QE deals with peak signal to noise ratio

(PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) respectively. Varying sizes of the secret watermark (i.e., “Gold Coin”) are embedded into twenty 512 × 512 color images [130, 131] which yields the PSNR value (in terms of dB) falling into the range [33.07 (Desert) – 54.15 (Foster City)]. The PSNR of greater than 30 dB is the indication of high perceptible watermarked images [148]. The MSE is inversely proportional to the PSNR and hence, lower MSE values ensured higher PSNR. The minimum MSE is 0.24 for “Foster City” at payload of 0.5 bpB and that of the maximum MSE is 32.00 for “Desert” at 3 bpB. The minimum values of IF and UIQ are 0.995108 (Desert), 0.703987 (Bobcat) as obtained at 3 bpB and that of SSIM is 0.966799 (Pepper) at 2.5 bpB respectively. Unlikely, the maximum values of IF and UIQ are 0.999992 (Airplane) and 0.999333 (San Diego) as obtained at 0.5 bpB and that of SSIM is 0.999919 (Athens) at 1 bpB. The performance of evaluated results is also summarized in the form of average values of the above mentioned metrics for twenty color images with respect to the payload variation of 0.5 to 3 bpB.

Table 7.15. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WST_1x2_QE scheme

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Lena	0.5	54.126228	0.251453	0.999984	0.999646	0.994776
	1.0	50.038663	0.644486	0.999959	0.999683	0.985716
	1.5	49.182325	0.784960	0.999951	0.998925	0.983656
	2.0	45.053844	2.030934	0.999872	0.998933	0.958708
	2.5	41.480019	4.624645	0.999719	0.995138	0.921545
	3.0	38.803292	8.565469	0.999470	0.995161	0.870191
Baboon	0.5	54.041626	0.256399	0.999986	0.999870	0.998967
	1.0	50.027607	0.646129	0.999965	0.999871	0.997203
	1.5	49.092479	0.801368	0.999957	0.999553	0.995882
	2.0	45.030490	2.041885	0.999892	0.999563	0.990902
	2.5	41.233878	4.894321	0.999743	0.997205	0.974603
	3.0	38.663875	8.844898	0.999534	0.997221	0.963191
Pepper	0.5	48.034773	1.022357	0.999900	0.990122	0.977761
	1.0	46.665162	1.401406	0.999868	0.990282	0.972085
	1.5	43.012197	3.249829	0.999681	0.983120	0.963254
	2.0	41.701740	4.394467	0.999587	0.983263	0.946878
	2.5	36.151043	15.775241	0.998463	0.966799	0.905508
	3.0	35.229077	19.506198	0.998153	0.966839	0.865417

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	53.971148	0.260594	0.999992	0.999579	0.983422
	1.0	49.968752	0.654945	0.999981	0.999628	0.958382
	1.5	49.137706	0.793066	0.999977	0.998757	0.954725
	2.0	45.059882	2.028113	0.999941	0.998757	0.899332
	2.5	41.959552	4.141189	0.999881	0.994790	0.841307
	3.0	38.993159	8.199067	0.999765	0.994796	0.760206
Sailboat	0.5	53.339840	0.301367	0.999984	0.999694	0.995547
	1.0	49.683212	0.699453	0.999964	0.999722	0.988372
	1.5	48.309740	0.959635	0.999951	0.999059	0.986245
	2.0	44.753676	2.176269	0.999890	0.999066	0.966514
	2.5	40.854602	5.340969	0.999732	0.995373	0.938650
	3.0	38.467954	9.253050	0.999534	0.995405	0.901103
Earth	0.5	54.155713	0.249752	0.999985	0.999732	0.997668
	1.0	50.043001	0.643843	0.999961	0.999767	0.993535
	1.5	49.384758	0.749211	0.999955	0.999198	0.992655
	2.0	45.119531	2.000447	0.999881	0.999202	0.979855
	2.5	42.034801	4.070054	0.999756	0.996485	0.959156
	3.0	39.102509	7.995201	0.999524	0.996466	0.927026
San Diego	0.5	54.121904	0.251703	0.999990	0.999899	0.999333
	1.0	50.037489	0.644660	0.999976	0.999910	0.998168
	1.5	49.361433	0.753246	0.999971	0.999700	0.997938
	2.0	45.112445	2.003714	0.999925	0.999698	0.994461
	2.5	42.083991	4.024215	0.999849	0.998744	0.989517
	3.0	39.107507	7.986005	0.999702	0.998748	0.980483
Splash	0.5	49.270565	0.769172	0.999917	0.989397	0.963107
	1.0	47.535985	1.146784	0.999885	0.989492	0.935045
	1.5	44.458163	2.329507	0.999751	0.984851	0.930092
	2.0	42.664714	3.520537	0.999649	0.984936	0.867810
	2.5	37.775819	10.851715	0.998851	0.974976	0.799631
	3.0	36.405696	14.876841	0.998507	0.974867	0.708034
Oakland	0.5	51.278461	0.484433	0.999968	0.999417	0.997711
	1.0	48.69399	0.878377	0.999947	0.999427	0.995700
	1.5	46.322267	1.516539	0.999899	0.998570	0.994363
	2.0	43.716516	2.763304	0.999832	0.998567	0.988286
	2.5	39.250362	7.727589	0.999490	0.994904	0.975985
	3.0	37.463116	11.661883	0.999281	0.994866	0.958944

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	54.156509	0.249706	0.999991	0.999518	0.993383
	1.0	50.058037	0.641618	0.999977	0.999580	0.982171
	1.5	49.385554	0.749074	0.999973	0.998559	0.9792545
	2.0	45.114207	2.002901	0.999928	0.998568	0.945416
	2.5	42.107491	4.002498	0.999857	0.993972	0.901040
	3.0	39.135286	7.935087	0.999717	0.993977	0.832464
Anhinga	0.5	50.822348	0.538078	0.999958	0.999807	0.882629
	1.0	49.533637	0.723963	0.999944	0.999856	0.881614
	1.5	45.002375	2.055146	0.999842	0.999120	0.876641
	2.0	43.366814	2.995014	0.999770	0.999310	0.864182
	2.5	38.412005	9.373026	0.999283	0.996071	0.846795
	3.0	37.224579	12.320331	0.999057	0.995682	0.822270
Athens	0.5	51.881060	0.421671	0.999966	0.999864	0.971365
	1.0	51.1353861	0.500658	0.999960	0.999919	0.970370
	1.5	49.676317	0.700565	0.999944	0.999409	0.965759
	2.0	46.252485	1.541103	0.999876	0.999628	0.951279
	2.5	43.427323	2.953575	0.999764	0.997383	0.931376
	3.0	40.869172	5.323081	0.999574	0.996973	0.897297
Bardowl	0.5	50.094251	0.636290	0.999934	0.999464	0.997727
	1.0	49.020251	0.814807	0.999916	0.999472	0.997147
	1.5	45.869211	1.683291	0.999824	0.998417	0.992595
	2.0	43.916581	2.638895	0.999729	0.998453	0.989394
	2.5	38.169622	9.911014	0.998961	0.988079	0.955838
	3.0	37.103446	12.668806	0.998686	0.988052	0.949004
Barnfall	0.5	53.104854	0.318122	0.999950	0.999784	0.998572
	1.0	50.239177	0.615407	0.999899	0.999812	0.996867
	1.5	49.346371	0.755863	0.999875	0.999242	0.996360
	2.0	44.830247	2.138235	0.999654	0.999267	0.987759
	2.5	41.447815	4.659065	0.999195	0.995747	0.978107
	3.0	38.606275	8.962988	0.998505	0.995706	0.958025
Butrfly	0.5	52.514547	0.364439	0.999974	0.999853	0.995615
	1.0	50.882130	0.530722	0.999962	0.999869	0.993905
	1.5	49.685431	0.699096	0.999950	0.999492	0.992314
	2.0	46.003199	1.632151	0.999883	0.999589	0.984845
	2.5	42.958190	3.290495	0.999766	0.997896	0.973611
	3.0	40.424834	5.896535	0.999581	0.997689	0.957285

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	52.161319	0.395319	0.999944	0.999696	0.749616
	1.0	51.007421	0.515630	0.999927	0.999725	0.748645
	1.5	49.607109	0.711818	0.999900	0.999097	0.745794
	2.0	46.127967	1.585928	0.999775	0.999234	0.737210
	2.5	42.290387	3.837440	0.999457	0.995538	0.719907
	3.0	40.105867	6.345905	0.999106	0.995214	0.703987
Bodie	0.5	50.196920	0.621424	0.999888	0.999554	0.978414
	1.0	48.106353	1.005645	0.999822	0.999581	0.975564
	1.5	45.527169	1.821224	0.999669	0.998513	0.971786
	2.0	43.049536	3.222009	0.999430	0.998545	0.962559
	2.5	37.321803	12.047584	0.997792	0.985578	0.933671
	3.0	35.977433	16.418632	0.997074	0.985666	0.915463
Bluheron	0.5	52.656006	0.352760	0.999956	0.999772	0.996039
	1.0	50.119309	0.632629	0.999922	0.999779	0.992999
	1.5	49.667859	0.701930	0.999914	0.999234	0.992126
	2.0	45.156172	1.983641	0.999757	0.999250	0.979666
	2.5	42.589158	3.582321	0.999561	0.996878	0.967055
	3.0	37.927677	10.478825	0.998716	0.996703	0.908932
Colomtn	0.5	52.128962	0.398276	0.999968	0.999769	0.984849
	1.0	49.555255	0.720368	0.999943	0.999805	0.982616
	1.5	46.392429	1.492235	0.999883	0.999180	0.981911
	2.0	43.779013	2.723823	0.999787	0.999248	0.972661
	2.5	39.515006	7.270758	0.999432	0.996504	0.962703
	3.0	37.625756	11.233231	0.999122	0.996389	0.936836
Desert	0.5	45.961851	1.647764	0.999735	0.997282	0.993767
	1.0	45.142962	1.989683	0.999685	0.997345	0.993226
	1.5	40.870682	5.321230	0.999157	0.993606	0.985443
	2.0	39.966217	6.553278	0.998984	0.993743	0.981910
	2.5	33.790021	27.169282	0.995777	0.975711	0.946543
	3.0	33.078297	32.007418	0.995108	0.975451	0.933366
Average case	0.5	51.90094	0.489554	0.999949	0.998586	0.972513
	1.0	49.37469	0.802561	0.999923	0.998626	0.966967
	1.5	47.46458	1.431442	0.999851	0.99728	0.96394
	2.0	44.28876	2.598832	0.999752	0.997341	0.947481
	2.5	40.24264	7.477350	0.999216	0.991689	0.921127
	3.0	38.01574	11.32397	0.998886	0.991594	0.887476

Table 7.16 shows that both WST_1x2 and WST_1x2_QE are compared with each other in terms of average PSNR (dB) values with respect to the payload variation of 0.5 to 3 bpB. The WST_1x2_QE scheme ensured the inclusion of the adaptive quality enhancement into the WST_1x2 of section 5.2.2 for improving average PSNR (dB) which is reflected above 1 bpB of payload. Numerical analysis summarizes the average PSNR improvement in the range of 2 to 4 dB against 1.5, 2, 2.5 and 3 bpB of payloads. All average PSNR values at varying payload are more than 30 dB which signifies high level of perceptibility for the watermarked images [148].

Table 7.16. Comparative analysis of obtained average PSNR values between WST_1x2 and WST_1x2_QE with respect to increasing payload

WST_1x2		**WST_1x2_QE**	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	51.90094	0.5	51.90094
1.0	49.37469	1.0	49.37469
1.5	44.50645	1.5	47.46458
2.0	42.00782	2.0	44.28876
2.5	37.66123	2.5	40.24264
3.0	34.97197	3.0	38.01574

Fig. 7.15 depicts the PSNR analysis among the WST_1x2_QE, Discrete Pascal Transform based data hiding scheme (abbreviated as DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (abbreviated as DGTDHS) [129] respectively, since the peak signal to noise ratio (PSNR) is the dominant metric to analyze the quality of watermarked images. In general, increasing values of PSNR ensures the decreasing of payload values and vice-versa. However, the trade-off between this two is maintained by setting the minimum acceptable level of PSNR value as 30 dB since it denotes perceptible visual clarity of watermarked images [148]. But, the PSNR analysis of WST_1x2_QE discloses another interesting point unlike the DPTHDI [88] and DGTDHS [129] i.e., the variability of payload. Therefore, the PSNR values of DPTHDI [88] and DGTDHS [129] has been computed at 0.25 and 1 bpB respectively and that of WST_1x2_QE computes the PSNR values against the variable payload for a spread from 0.5 to 3 bpB. All computations have been carried out with reference to the following five images: “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat”. In contrast to DPTHDI [88], the WST_1x2_QE ensured equal or higher PSNR (dB) at 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB for

"Baboon", 0.5, 1, 1.5 and 2 bpB of payloads for "Pepper", 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for "Airplane" and 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for "Sailboat" respectively. In comparison with the DGTDHS [129], the WST_1x2_QE ensured higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for "Lena", 0.5, 1 and 1.5 bpB of payloads for "Baboon", 0.5 bpB of payloads for "Pepper", 0.5, 1 and 1.5 bpB of payloads for "Airplane" and 0.5, 1 and 1.5 bpB of payloads for "Sailboat" respectively. It is also evident from fig. 7.15 that the watermarked images retain good visual clarity since the PSNR values are above 30 dB.

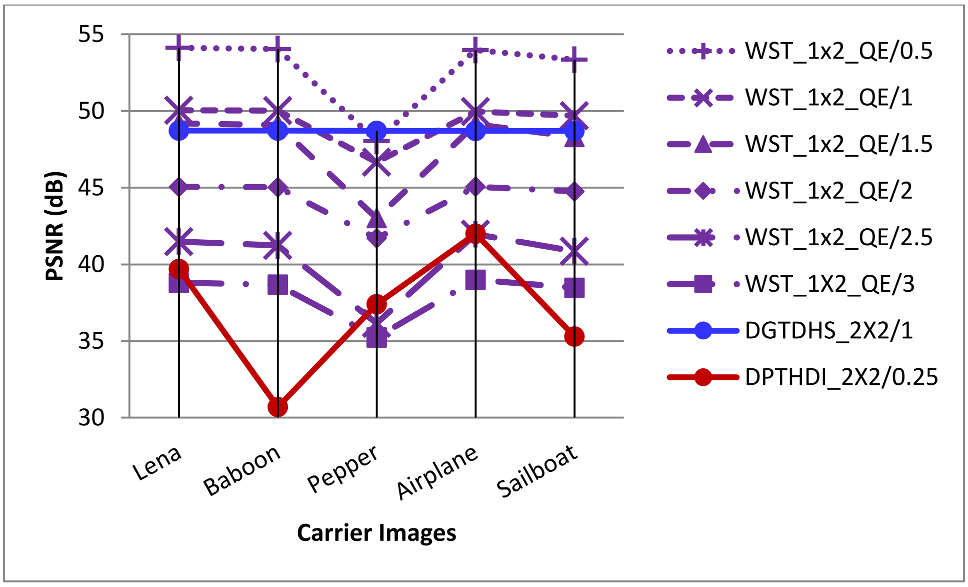


Fig. 7.15. Performance analysis of PSNR (dB) for variable payload based WST_1x2_QE and fixed payload based Varsaki et al.'s DPTHDI [88] and DGTDHS [129] schemes with respect to five color images

Some critical analysis is made in fig. 7.16 which depicts how the average PSNR varies with respect to the payload values for WST_1x2_QE, WST_1x2, WST_2x2_QE, WBT_1x2_QE, WLT_1x2_QE, WDHT_1x2_QE, DPTHDI [88] and DGTDHS [129] respectively. Compared to WDHT_1x2_QE, WLT_1x2_QE, WBT_1x2_QE, WST_2x2_QE and WST_1x2 the average PSNR values of WST_1x2_QE is significantly high as far as the payload variation of more than 1 bpB is concerned. Since, the PSNR values of WST_1x2_QE never dropped below 30 dB for the payload range [0.5 – 3 bpB], the quality of the watermarked images are also treated as elevated. The average PSNR for DPTHDI [88] is 37.40 dB which is obtained by taking the averages of PSNR values for "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" images at 0.25 bpB of payload. In contrast to DPTHDI [88], the average PSNR of WST_1x2_QE ensures equal or higher PSNR (dB) at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB as obtained by taking the averages of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” at 1 bpB of payload. Compared to DGTDHS [129] scheme, the average PSNR of WST_1x2_QE offers higher PSNR (dB) at 1 bpB while the latter scheme also supports fabrication of secret information with variable payload.

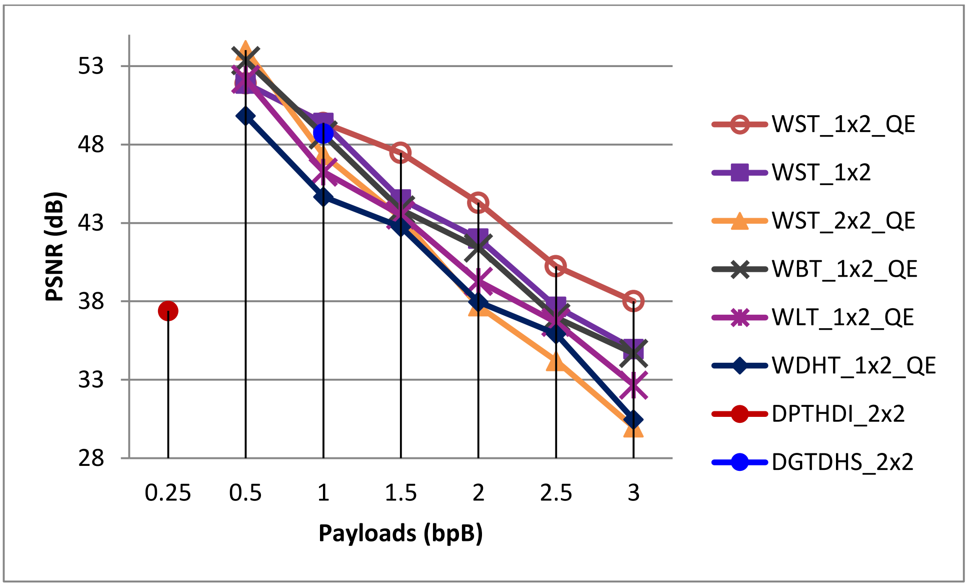


Fig. 7.16. Graphical representation of variation of average PSNR (dB) with respect to payload for WDHT_1X2_QE, WLT_1X2_QE, WBT_1X2_QE, WST_2X2_QE, WST_1x2 and WST_1X2_QE and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

7.3.5 Quality Enhancement of G-lets D4 domain based watermarking

The watermarking based on group of linear transformations for dihedral group of order 4 (G-lets D4) is applicable for 2 x 2 as well as 1 x 2 blocks. These two schemes based on different block sizes are abbreviated as WGD4_2x2 and WGD4_1x2 respectively.

On application of quality enhancement, the 2 x 2 and 1 x 2 blocks based watermarking in G-lets D4 domain are significantly enhanced in terms of quality of the watermarked images which are discussed in sections 7.3.5.1 and 7.3.5.2 respectively.

7.3.5.1 Quality Enhancement for 2 x 2 Block based Watermark Fabrication

The primary step of WGD4_2x2_QE is the decomposition of the host image into non-overlapping blocks of size 2 x 2. To address the issue of overflow/underflow, apply an adjustment process based on the embedding payload by keeping the pixel components within the valid range as discussed in equation (2.8) of section 2.2.1.1. Forward transform in G-lets D4 domain converts each 2 x 2 sub-matrix of pixel components into the transformed components of identical block size. It is already discussed in section 6.2.1 that the secret bits for the message digest, size and the content of the watermark are embedded on transformed components starting from the least significant bit position (i.e., LSB-0) toward higher order bit position. Adaptive quality enhancement enhances the quality of the watermarked images without affecting the embedded bits of the transformed components. Inverse transform in G-lets D4 domain is applied over 2 x 2 sub-matrices of enhanced components to re-compute the pixel components in spatial domain. The process is repeated till the secret information is concealed and the watermarked image is produced.

Section 7.3.5.1.1 explains the decomposition, adjustment, embedding, quality enhancement and the inverse transform phases with a suitable example. The simulated results are computed, analyzed and validated with short discussions in section 7.3.5.1.2.

7.3.5.1.1 Example

The carrier image is partitioned into 2 x 2 non-overlapping blocks in row major order. Three 2 x 2 sub-matrices namely R$_1$, G$_1$ and B$_1$ corresponding to red, green and blue channels have been taken for watermark fabrication. The 2 x 2 sub-matrices of pixel components are represented as follows:

$$R_1 = \begin{bmatrix} 230 & 72 \\ 17 & 155 \end{bmatrix} G_1 = \begin{bmatrix} 62 & 215 \\ 56 & 22 \end{bmatrix} B_1 = \begin{bmatrix} 111 & 172 \\ 251 & 7 \end{bmatrix}$$

Consider the secret key (K) = “skghosal” from which the pseudo random number r = (7809650151167322995 % 8) +1 = 4 is obtained. Since, r = 4, G$_4$ i.e., rotation matrix R$_3$ is multiplied with the 2 x 2 sub-matrix of pixel components by applying forward transform in G-lets D4 domain. The 2 x 2 sub-matrices of transformed components are obtained as follows:

$$T(R_1) = \begin{bmatrix} 17 & 155 \\ -230 & -72 \end{bmatrix} T(G_1) = \begin{bmatrix} 56 & 22 \\ -62 & -215 \end{bmatrix} T(B_1) = \begin{bmatrix} 251 & 7 \\ -111 & -172 \end{bmatrix}$$

Let the watermark bit stream *“011001100110010011001111011011000110*” is to be fabricated into the transformed components for the payload value of 3 bits per Byte. In this example, three bits are fabricated ($\lambda_1 = 3$, $\lambda_2 = 3$, $\lambda_3 = 3$, $\lambda_4 = 3$) on each transformed component. The order of embedding may vary for each component based on its sign. Hence, the 2 x 2 sub-matrices of embedded components are obtained as follows:

$$T'(R_1) = \begin{bmatrix} 22 & 156 \\ -228 & -78 \end{bmatrix} T'(G_1) = \begin{bmatrix} 58 & 22 \\ -57 & -215 \end{bmatrix} T'(B_1) = \begin{bmatrix} 254 & 6 \\ -104 & -174 \end{bmatrix}$$

The adaptive quality enhancement of section 7.2.1 has been used to ensure the enhancement of quality without hampering the embedded bits. Basically, it is applied on each embedded component by taking the closest value of the pre-embedded component without affecting the embedded bits. Thus the new 2 x 2 sub-matrices after adaptive quality enhancement become:

$$T''(R_1) = \begin{bmatrix} 14 & 156 \\ -228 & -70 \end{bmatrix} T''(G_1) = \begin{bmatrix} 58 & 22 \\ -65 & -215 \end{bmatrix} T''(B_1) = \begin{bmatrix} 254 & 6 \\ -112 & -174 \end{bmatrix}$$

Inverse transform in G-lets D4 domain is applied on 2 x2 sub-matrices of quality enhanced components corresponding to red, green and blue channels. The 2 x 2 sub-matrices of pixel components in spatial domain are obtained as follows:

$$R'_1 = \begin{bmatrix} 14 & 156 \\ 228 & 70 \end{bmatrix} G'_1 = \begin{bmatrix} 58 & 22 \\ 65 & 215 \end{bmatrix} B'_1 = \begin{bmatrix} 254 & 6 \\ 112 & 174 \end{bmatrix}$$

It has been observed that the modified pixel components for each 2 x 2 matrices are non-fractional, non-negative and less than or equal to 255.

7.3.5.1.2 Results and Discussions

The quality of the 2 x 2 block based quality enhanced scheme (WGD4_2x2_QE) is extensively analyzed against twenty 512 x 512 benchmark images [130, 131] in which the varying sizes of the watermark are concealed. The WGD4_2x2_QE scheme offered a minimum peak signal to noise ratio (PSNR) value of 39.17 dB at 3 bpB of payload for the “Desert” image and that of maximum value is 54.19 dB at 0.5 bpB for the “Lena” image. The lowest and highest mean squared error (MSE) values are 0.24 (Lena) and 7.86 (Desert) where, the term lowest and highest specifies the obtained MSE values at 0.5 and 3 bpB respectively. The minimum values of image fidelity (IF), structural similarity index (SSIM) and universal imae quality index (UIQ) are 0.998844 (Desert), 0.975434 (Splash) and 0.72012 (Bobcat) at 3 bpB whereas, the maximum values are 0.999992 (Airplane), 0.999911 (San Diego) and 0.999334 (San Diego) at 3 bpB. The average values are computed for various metrics of twenty different carrier images at variable payloads to summarize the experimental results. Since, the PSNR values at highest payload (i.e., at 3 bpB) is more than 30 dB as well as the IF, SSIM and UIQ are very close to one for payload variation of 0.5 – 3 bpB, the quality distortion in the watermarked image is very less as compared to the 2 x 2 block based schemes discussed so far. Moreover, the obtained watermarked images preserved a good visual clarity [148].

Table 7.17. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WGD4_2x2_QE scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.196500	0.247417	0.999984	0.999686	0.994808
	1.0	51.180434	0.495492	0.999968	0.999177	0.990166
	1.5	49.359601	0.753564	0.999952	0.998953	0.984142
	2.0	46.357402	1.504319	0.999905	0.997921	0.972454
	2.5	42.697472	3.494082	0.999779	0.994644	0.937069
	3.0	40.743904	5.478856	0.999653	0.991251	0.907221
Baboon	0.5	54.156000	0.249735	0.999986	0.999886	0.999022
	1.0	51.150645	0.498902	0.999973	0.999666	0.998044
	1.5	49.388194	0.748619	0.999960	0.999633	0.996991
	2.0	46.365096	1.501656	0.999920	0.999256	0.994609
	2.5	42.680266	3.507952	0.999814	0.998097	0.986896
	3.0	40.735206	5.489840	0.999709	0.996951	0.979875
Pepper	0.5	54.034890	0.256797	0.999978	0.999440	0.990092
	1.0	51.056199	0.509871	0.999957	0.998449	0.985853
	1.5	48.993407	0.819859	0.999930	0.997894	0.979764
	2.0	46.004001	1.631849	0.999860	0.994718	0.969343
	2.5	42.070984	4.036285	0.999654	0.987714	0.939243
	3.0	39.964750	6.5554924	0.999432	0.980187	0.912959
Airplane	0.5	54.120917	0.251761	0.999992	0.999626	0.983152
	1.0	51.115621	0.502942	0.999985	0.999045	0.969501
	1.5	49.380080	0.750019	0.999978	0.998761	0.955033
	2.0	46.371106	1.499580	0.999957	0.997502	0.927075
	2.5	42.676960	3.510623	0.999899	0.993656	0.858082
	3.0	40.720998	5.507830	0.999842	0.989165	0.809246
Sailboat	0.5	54.114845	0.252113	0.999987	0.999740	0.995608
	1.0	51.125217	0.501832	0.999974	0.999321	0.992255
	1.5	49.353588	0.754608	0.999961	0.999162	0.98674
	2.0	46.349429	1.507083	0.999923	0.998296	0.977254
	2.5	42.627953	3.550463	0.999820	0.995646	0.950357
	3.0	40.685122	5.553517	0.999719	0.992907	0.928954
Earth	0.5	54.1948266	0.247512	0.999985	0.999764	0.997682
	1.0	51.165703	0.497175	0.999970	0.999380	0.995612
	1.5	49.373025	0.751238	0.999955	0.999202	0.992675
	2.0	46.363780	1.502112	0.999910	0.998393	0.986881

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	42.685846	3.503448	0.999792	0.995871	0.967694
	3.0	40.751313	5.469518	0.999675	0.993389	0.950071
San Diego	0.5	54.188650	0.247865	0.999990	0.999911	0.999334
	1.0	51.168158	0.496894	0.999981	0.999770	0.998738
	1.5	49.377695	0.750431	0.999972	0.999700	0.997952
	2.0	46.368558	1.500460	0.999944	0.999399	0.996381
	2.5	42.687405	3.502190	0.999869	0.998441	0.991224
	3.0	40.739213	5.484778	0.999795	0.997508	0.986418
Splash	0.5	54.058783	0.255388	0.999977	0.999176	0.966074
	1.0	51.077490	0.507377	0.999955	0.997618	0.949486
	1.5	49.079090	0.803843	0.999928	0.996681	0.933322
	2.0	46.116363	1.590171	0.999858	0.992490	0.901719
	2.5	42.221683	3.898629	0.999649	0.983746	0.823980
	3.0	40.230107	6.166938	0.999443	0.975434	0.767990
Oakland	0.5	54.129326	0.251274	0.999986	0.999804	0.998596
	1.0	51.128024	0.501508	0.999973	0.999479	0.997538
	1.5	49.239046	0.774775	0.999957	0.999338	0.996053
	2.0	46.208210	1.556894	0.999915	0.998624	0.993107
	2.5	42.479881	3.673603	0.999799	0.996580	0.983421
	3.0	40.500507	5.794681	0.999682	0.994413	0.974100
Foster City	0.5	54.186913	0.247964	0.999991	0.999578	0.993367
	1.0	51.160064	0.497821	0.999982	0.998923	0.988152
	1.5	49.375135	0.750873	0.999973	0.998554	0.979214
	2.0	46.352105	1.506155	0.999946	0.997105	0.963493
	2.5	42.711631	3.482709	0.999875	0.992618	0.915768
	3.0	40.774251	5.440705	0.999806	0.988079	0.875629
Anhinga	0.5	52.448813	0.369997	0.999971	0.999895	0.882825
	1.0	49.741529	0.690124	0.999947	0.999654	0.885574
	1.5	49.673283	0.701054	0.999946	0.999484	0.877636
	2.0	46.567729	1.433202	0.999889	0.998915	0.874258
	2.5	43.260728	3.069075	0.999764	0.997258	0.862224
	3.0	41.461676	4.644219	0.999643	0.995715	0.840816
Athens	0.5	51.855675	0.424143	0.999966	0.999907	0.971388
	1.0	49.354203	0.754501	0.999939	0.999683	0.974436
	1.5	49.946667	0.658284	0.999947	0.999478	0.965994
	2.0	46.768227	1.368540	0.999890	0.998933	0.962458

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	43.746681	2.744177	0.999780	0.997193	0.941412
	3.0	42.113790	3.996697	0.999680	0.995807	0.919565
Bardowl	0.5	52.469129	0.368270	0.999963	0.999885	0.999054
	1.0	49.895188	0.666133	0.999933	0.999614	0.998920
	1.5	49.487218	0.731742	0.999926	0.999649	0.997669
	2.0	46.390834	1.492783	0.999850	0.999182	0.995913
	2.5	42.966655	3.284088	0.999671	0.998183	0.989914
	3.0	41.181177	4.954074	0.999503	0.997129	0.984167
Barnfall	0.5	53.432516	0.295004	0.999955	0.999812	0.998833
	1.0	50.605045	0.565686	0.999913	0.999444	0.998217
	1.5	49.199476	0.781866	0.999874	0.999256	0.995755
	2.0	46.175749	1.568575	0.999750	0.998377	0.993053
	2.5	42.624285	3.553462	0.999429	0.996713	0.984492
	3.0	40.667103	5.576606	0.999105	0.994104	0.976465
Butrfly	0.5	52.335775	0.379754	0.999972	0.999887	0.995565
	1.0	49.745587	0.689479	0.999950	0.999630	0.995474
	1.5	49.199476	0.781866	0.999874	0.999256	0.995755
	2.0	46.175749	1.568575	0.999750	0.998377	0.993053
	2.5	42.624285	3.553462	0.999429	0.996713	0.984492
	3.0	40.667103	5.576606	0.999105	0.994104	0.976465
Bobcat	0.5	52.050556	0.405531	0.999942	0.999737	0.749630
	1.0	49.505535	0.728663	0.999897	0.999076	0.750896
	1.5	49.910751	0.663750	0.999906	0.999142	0.746175
	2.0	46.750476	1.374145	0.999805	0.998005	0.742745
	2.5	43.622647	2.823680	0.999603	0.995887	0.742392
	3.0	41.926621	4.172710	0.999413	0.994221	0.720120
Bodie	0.5	54.002689	0.258708	0.999957	0.999882	0.980279
	1.0	51.081650	0.506891	0.999916	0.999619	0.979328
	1.5	48.905717	0.836582	0.999857	0.999464	0.976134
	2.0	45.935791	1.657681	0.999717	0.998759	0.973368
	2.5	42.275069	3.850999	0.999351	0.997402	0.961860
	3.0	40.308638	6.056427	0.998980	0.995167	0.954009
Bluheron	0.5	53.637012	0.281435	0.999965	0.999752	0.996472
	1.0	50.769326	0.544687	0.999933	0.999141	0.995349
	1.5	49.719181	0.693684	0.999915	0.999171	0.992167
	2.0	46.736056	1.378715	0.999831	0.998086	0.987382

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	41.957383	4.143258	0.999492	0.996282	0.964293
	3.0	39.639020	7.066075	0.999135	0.989794	0.948091
Colomtn	0.5	53.872445	0.266585	0.999979	0.999826	0.985477
	1.0	50.917212	0.526452	0.999958	0.999502	0.985171
	1.5	49.367375	0.752216	0.999941	0.999410	0.982111
	2.0	46.298770	1.524766	0.999880	0.998758	0.979510
	2.5	42.692357	3.498199	0.999726	0.996880	0.964653
	3.0	40.711918	5.519357	0.999567	0.994083	0.953610
Desert	0.5	53.587786	0.284643	0.999959	0.999841	0.998602
	1.0	50.670424	0.557234	0.999920	0.999565	0.998393
	1.5	48.543887	0.909267	0.999866	0.999433	0.996932
	2.0	45.468829	1.845854	0.999729	0.998510	0.995000
	2.5	41.367798	4.745702	0.999305	0.996558	0.986182
	3.0	39.175165	7.862557	0.998844	0.993217	0.980392
Average case	0.5	53.553700	0.292095	0.999974	0.999752	0.973793
	1.0	50.680660	0.561983	0.999951	0.999288	0.971355
	1.5	49.343590	0.758407	0.999931	0.999081	0.966411
	2.0	46.306210	1.525656	0.999861	0.997980	0.958953
	2.5	42.633900	3.571304	0.999675	0.995304	0.936782
	3.0	40.684880	5.618374	0.999487	0.992131	0.917308

The adaptive quality enhancement does not make any impact on average PSNR for WGD4_2x2 at 0.5 and 1 bpB respectively. However, the enhanced scheme (WGD4_2x2_QE) differs from the former one at higher payload (1.5 – 3 bpB) as evident from table 7.18. The minimum values of average PSNR for WGD4_2x2_QE and WGD4_2x2 are obtained as 40.68 and 35.99 dB respectively. The WGD4_2x2_QE also shows the average PSNR enhancement of 4.15, 3.43, 4.34 and 4.69 dB more over WGD4_2x2 with respect to 1.5, 2, 2.5 and 3 bpB respectively. Since, the PSNR values obtained at varying payload (0.5 – 3 bpB) are greater than 30 dB, the watermarked images retains high transparency [148].

Table 7.18. Comparative analysis of average PSNR between WGD4_2x2 and WGD4_2x2_QE with respect to increasing payload

WGD4_2x2		WGD4_2x2_QE	
Payload (bpB)	PSNR (dB)	Payload (bpB)	PSNR (dB)
0.5	53.55370	0.5	53.55370
1.0	50.68066	1.0	50.68066

WGD4_2x2		WGD4_2x2_QE	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
1.5	45.19256	1.5	49.34359
2.0	42.87705	2.0	46.30621
2.5	38.29444	2.5	42.63390
3.0	35.99861	3.0	40.68488

Fig. 7.17 depicts an analysis of PSNR values which is made over “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” respectively. Varsaki et al.’s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] offered average PSNR values of greater than or equal to 30 dB but, both of them provides low as well as fixed payload values of 0.25 and 1 bpB respectively. In contrast to DPTHDI [88], the WGD4_2x2_QE ensured equal or higher PSNR (dB) at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB for “Baboon”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for “Pepper”, 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for “Airplane” and 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for “Sailboat” respectively. Compared to the DGTDHS [129], the WGD4_2x2_QE ensured higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for “Lena”, 0.5, 1 and 1.5 bpB of payloads for “Baboon”, 0.5, 1 and 1.5 bpB of payloads for “Pepper”, 0.5, 1 and 1.5 bpB of payloads for “Airplane” and 0.5, 1 and 1.5 bpB of payloads for “Sailboat” respectively. The WGD4_2X2_QE provides the variation of payload (0.5 – 3 bpB) by retaining an acceptable visual quality (PSNR ≥ 30 dB).

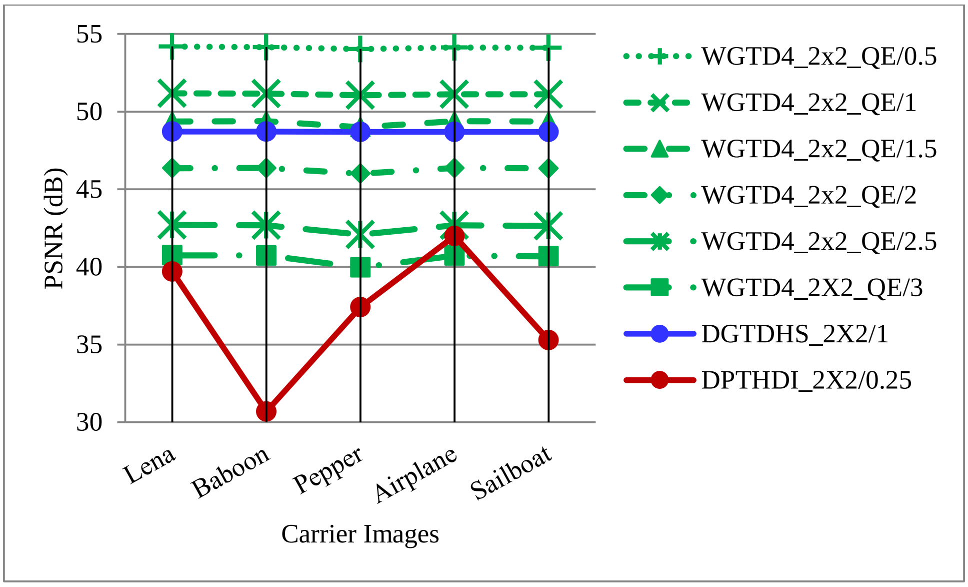


Fig. 7.17. Performance analysis of PSNR (dB) for variable payload based WGD4_2x2_QE and fixed payload based Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 7.18 illustrates a graphical analysis of average PSNR with respect to payload. Two existing schemes proposed by Varsaki et al. i.e., DPTHDI [88] and DGTDHS [129] are providing average PSNR values of 37.4 and 48.7 dB respectively. But, the available payload of both schemes are fixed as well as consisting of very low fabrication density i.e., 0.25 and 1 bpB respectively. In order to avoid the issue of fixed payload, the concept of WGD4_2x2_QE has been emerged which supports payload variation of 0.5 to 3 bpB by retaining an acceptable level of quality by means of average PSNR (i.e., ≥ 30 dB) [148]. In contrast to WGD4_2x2, the dispersion of PSNR value is clearly observed for WGD4_2X2_QE as the payload increases from 1 bpB. In comparison with other quality enhanced schemes viz. WST_2X2_QE, WBT_2X2_QE, WLT_2X2_QE and WDHT_2X2_QE, the WGD4_2X2_QE offered tremendous improvement in terms of average PSNR values with respect to variable payload (0.5 – 3 bpB). Except DPTHDI [88] and DGTDHS [129], the average PSNR for all schemes are computed from twenty color images as given in fig. 1.1. On contrary, "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" images are considered for DPTHDI [88] and "Lighthouse", "Elaine", "Lenna", "Boat" and "F16" images are considered for DGTDHS [129] respectively. In contrast to DPTHDI [88], the average PSNR of WGD4_2x2_QE ensured equal or higher PSNR (dB) at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. Compared to DGTDHS [129], the average PSNR of WGD4_2x2_QE ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads.

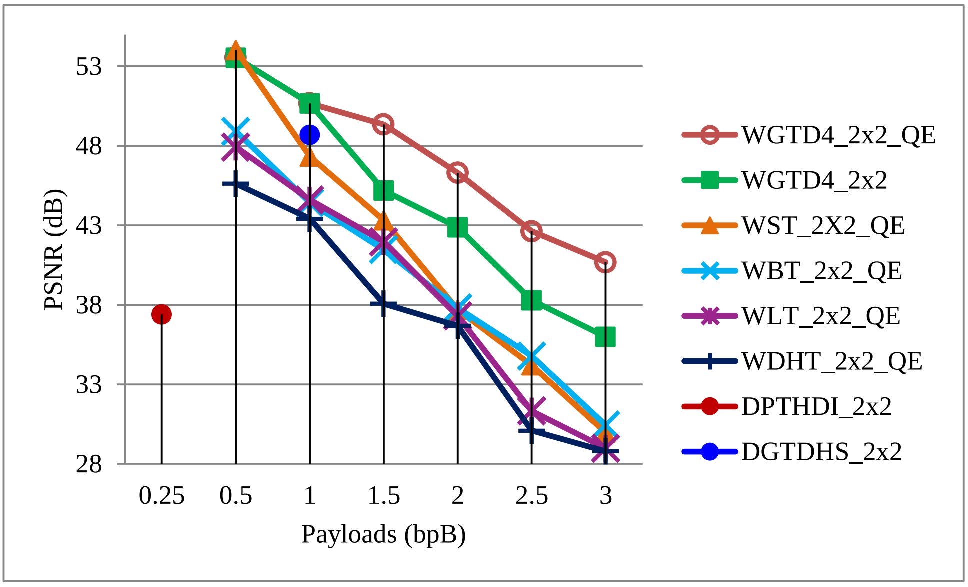


Fig. 7.18. Graphical representation of variation of average PSNR (dB) with respect to payload for WGD4_2x2_QE, WGD4_2x2, WST_2x2_QE, WBT_2x2_QE, WLT_2x2_QE, WDHT_2x2_QE and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

7.3.5.2 Quality Enhancement for 1 x 2 Block based Watermark Fabrication

Decomposition of carrier images into non-overlapping blocks of size 1 x 2 yields pairs of pixel components which are successively converted into transform domain based on forward transform in G-lets D4 domain. To minimize the quality degradation, the EMD based quality enhancement of section 7.2.2 has been applied on the actual pair of transformed components to obtain the target pair of transformed components in which the fabrication is actually to be done. It has been discussed in section 6.2.2 that the message digest, size and content of the watermark bits are fabricated on each transformed component at the vector modification area (VMA). Each transformed component is capable of fabricating a maximum of three watermark bits. Inverse transform in G-lets D4 domain is applied over each 1 x 2 sub-matrices of quality enhanced components to obtain the pixel components in spatial domain. The process is repeated until and unless the entire secret information is concealed and the watermarked image is produced.

The proposed embedding methodology is clearly explained with the example given in section 7.3.5.2.1. Comparative analysis, results and discussions have been elaborated in section 7.3.5.2.2.

7.3.5.2.1 Example

Consider the pair of pixel components (174, 105) for a given cover image that can fabricate the secret data 14. The embedding and extraction steps for the payload value of 2 bpB are described in the following section.

Embedding: - The following steps are followed to embed the secret data:

1) Pair of pixel components $(p_0, p_1) = (174, 105)$, secret data (d) = 14, no. of bits to be embedded $(\lambda) = 4$ and secret key (K) = “skghosal”.
2) Obtain the pseudo random number, r = (7809650151167322995 % 8) +1 = 4. Since, r = 4, G_4 i.e., rotation matrix R_3 is multiplied with the pair of pixel components based on the forward transform in G-lets D4 domain. Therefore, the pair of transformed components is obtained as $T_4 = (-105, 174)$.
3) Since, sign s_0 is –ve and s_1 is +ve, the bit-stream of the secret data (d) is considered in reverse order i.e., $d = (0111)_2 = 7$. Consequently, the transformed components are considered as positive integers i.e., $(t_0, t_1) = (|-105|, |174|) = (105,174)$.

4) EMD based quality enhancement of section 7.2.2 has been used to obtain the ideal pair of transformed components (t_0, t_1) from a set of five pairs of transformed components, which are (103,172), (104, 173), (105, 174), (106,175) and (107,176). It can be examined that the ideal pair for embedding is (t_0, t_1) = (105, 174).

5) $P_{VMA} = 4$, n = (4 – 1) = 3, (t_0, t_1) = (105, 174) = $(01101001, 10101110)_2$.

6) Generate 3 vectors (g_1, g_2, g_3) = $(0110, 1010, 1011)_2$ = (6, 10, 11).

7) Compute, f (g_1, g_2, g_3) = $(6 \text{ x } 1 + 10 \text{ x } 3 + 11 \text{ x } 7) \ \% \ 2^4 = 113 \ \% \ 16 = 1 = (0001)_2$.

8) Calculate the summation value, $S_V = (7 + 1) \ \% \ 2^4 = 8 = (1000)_2$

9) Since, VMA = 01 || 10, $\lambda_1 = \lfloor 4 / 2 \rfloor = 2$, $\lambda_2 = 4 – 2 = 2$. Replace it by S_V as given below:

$t'_i = (105 \ \& \ (2^8 – 2^2)) + \lfloor 8 / 2^{\lceil 4/2 \rceil} \rfloor = (105 \ \& \ (256 – 4) + \lfloor 8/4 \rfloor$

$= (105 \ \& \ 252) + 2$

$= 106.$

$t'_{i+1} = (174 \ \& \ (2^8 – 2^2)) + 8 \ \% \ 2^{\lceil 4/2 \rceil} = (174 \ \& \ (256 – 4)) + 8 \ \% \ 4$

$= (174 \ \& \ 252) + 0$

$= 172.$

10) Embedded pair of transformed components, (t'_0, t'_1) = (106, 172) = $(01101010, 10101100)_2$.

11) Since, sign s_0 was –ve and s_1 was +ve, so by re-assigning the sign, the embedded components yields, (t'_0, t'_1) = (-106, 172).

12) Inverse transform is applied for r = 4 which re-generate the pair of pixel components, (p'_0, p'_1) = (172, - (-106)) = (172, 106).

Extraction: - The following steps are followed to recover the secret data:

1) Pair of pixel components (p_0, p_1) = (172, 106), no. of bits to be extracted (λ) = 4 and secret key (K) = “skghosal”.

2) Obtain the pseudo random number r = (7809650151167322995 % 8) +1 = 4. Since, r = 4, G_4 i.e., rotation matrix R_3 is multiplied with the pair of pixel components based on the forward transform in G-lets D4 domain. Therefore, the pair of transformed components is obtained as T_4 = (-106, 172).

3) Since, sign s_0 is –ve and s_1 is +ve, (t_0, t_1) = (|-106|, 172) = (106,172).

4) P_{VMA} = 4, n = (4 – 1) = 3, (t_0, t_1) = (106, 172) = $(01101010, 10101100)_2$ and d = $(1000)_2$ = 8.

5) Generate 3 vectors (g_1, g_2, g_3) = $(0110, 1010, 1011)_2$ = (6, 10, 11).

6) Compute, f = (6 x 1 + 10 x 3 + 11 x 7) % 2^4 = 1.

7) The recovered secret data, d' = (2^4 – 1 + 8) % 2^4 = (16 – 1 + 8) % 16 = 7 = $(0111)_2$.

8) Since, sign s_0 was –ve and s_1 was +ve, so re-assigning the signs to the embedded components yields, (t'_0, t'_1) = (-106, 172) and secret data is retrieved in reverse order, d' = $(1110)_2$ = 14.

9) Inverse transform is applied for r = 4 which re-generates the pair of pixel components, (p'_0, p'_1) = (172, - (-106)) = (172, 106).

7.3.5.2.2 Results and Discussions

This section represents the results of the standard quality metrics for WGD4_1x2_QE scheme with respect to 0.5, 1.0, 1.5, 2.0, 2.5 and 3 bits per Byte (bpB) of payloads. Twenty benchmark (BMP) color images [130, 131] as given in fig. 1.1 are taken to compute results. Variable sizes of Gold-Coin (i.e., the authenticating watermark data) are embedded into the source benchmark images of dimension 512 × 512 based on the embedding proportion as derived from the payload offered. In proposed WGD4_1x2_QE, the minimum value of peak signal to noise ratio (PSNR) is 36.64 dB at 3 bpB of payload for the “Athens” image and that of the maximum value is 54.80 dB at 0.5 bpB of payload for the “Barnfall” image. The minimum value of obtained mean squared error (MSE) is 0.21 for “Barnfall” at 0.5 bpB of payload and that of the maximum value is 14.09 for “Athens” at 3 bpB of payload. Usually, the image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) are ranges from [0, 1]. In this experiment, the obtained values of IF, SSIM and UIQ belongs to the range [0.998260 (Bobcat), 0.999992 (Airplane)], [0.966667 (Splash), 0.999881 (San Diego)] and [0.687538 (Bobcat), 0.999232 (San Diego)] respectively. The minimum and maximum values of IF, SSIM and UIQ are obtained at 3 and 0.5 bpB respectively. Since, the minimum value of PSNR for the above mentioned twenty images is greater than 30 dB as well as IF, SSIM and UIQ are close to one, the quality of the watermarked image is well perceptible [148]. The average values of above mentioned metrics are also computed from twenty benchmark images with respect to variable payload for the spread of 0.5 to 3 bpB to summarize the experimental results.

Table 7.19. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WGD4_1x2_QE scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.122957	0.251642	0.999984	0.999580	0.994002
	1.0	51.131130	0.501149	0.999968	0.999168	0.988280
	1.5	46.934496	1.317136	0.999916	0.997659	0.971720
	2.0	44.986636	2.062608	0.999869	0.996153	0.957535
	2.5	41.454702	4.651682	0.999705	0.991296	0.916314
	3.0	39.181341	7.851384	0.999505	0.985038	0.875352
Baboon	0.5	54.136782	0.250843	0.999986	0.999854	0.998885
	1.0	51.145136	0.499535	0.999973	0.999710	0.997788
	1.5	46.939107	1.315738	0.999930	0.999180	0.994457
	2.0	44.994091	2.059070	0.999891	0.998640	0.991463
	2.5	41.506808	4.596206	0.999756	0.997003	0.982371
	3.0	39.266012	7.699793	0.999592	0.994870	0.972166
Pepper	0.5	53.996503	0.259077	0.999978	0.999374	0.989625
	1.0	51.068759	0.508398	0.999957	0.998630	0.984935
	1.5	46.715784	1.385166	0.999882	0.995607	0.970288
	2.0	44.719431	2.193497	0.999814	0.992468	0.958633
	2.5	41.065216	5.088136	0.999567	0.984208	0.925171
	3.0	38.805961	8.560206	0.999269	0.975153	0.891002
Airplane	0.5	54.091841	0.253452	0.999992	0.999493	0.980928
	1.0	51.113820	0.503150	0.999985	0.998986	0.964849
	1.5	47.014743	1.293022	0.999962	0.997301	0.926839
	2.0	45.070901	2.022973	0.999942	0.995441	0.897430
	2.5	41.423114	4.685639	0.999865	0.990017	0.821883
	3.0	39.233465	7.757714	0.999777	0.982960	0.764788
Sailboat	0.5	54.093148	0.253376	0.999987	0.999661	0.994812
	1.0	51.111340	0.503438	0.999974	0.999330	0.990131
	1.5	46.963244	1.308446	0.999933	0.998158	0.977176
	2.0	45.009047	2.051991	0.999896	0.996934	0.966298
	2.5	41.462745	4.643076	0.999765	0.993232	0.934968
	3.0	39.269225	7.694099	0.999611	0.988644	0.907597
Earth	0.5	54.139491	0.250686	0.999985	0.999682	0.997291
	1.0	51.144086	0.499656	0.999970	0.999366	0.994636
	1.5	46.911748	1.324053	0.999921	0.998189	0.986110
	2.0	44.964210	2.073286	0.999876	0.997018	0.978800

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	41.352984	4.761918	0.999717	0.992958	0.953369
	3.0	39.182496	7.849295	0.999532	0.988815	0.929016
San Diego	0.5	54.125503	0.251495	0.999990	0.999881	0.999232
	1.0	51.130127	0.501265	0.999981	0.999761	0.998490
	1.5	46.950967	1.312150	0.999951	0.999326	0.996280
	2.0	44.993919	2.059151	0.999923	0.998888	0.994264
	2.5	41.456178	4.650101	0.999826	0.997481	0.987763
	3.0	39.238105	7.749430	0.999711	0.995770	0.980918
Splash	0.5	54.049560	0.255931	0.999977	0.999030	0.964135
	1.0	51.085170	0.506481	0.999955	0.997791	0.945965
	1.5	46.825867	1.350496	0.999879	0.993922	0.902046
	2.0	44.774738	2.165740	0.999805	0.988970	0.865363
	2.5	41.215449	4.915134	0.999562	0.979187	0.785668
	3.0	38.786688	8.598279	0.999239	0.966667	0.711260
Oakland	0.5	54.106179	0.252616	0.999986	0.999746	0.998429
	1.0	51.123478	0.502033	0.999973	0.999483	0.997114
	1.5	46.883524	1.332686	0.999928	0.998557	0.992979
	2.0	44.942367	2.083740	0.999887	0.997580	0.989392
	2.5	41.397218	4.713662	0.999743	0.994709	0.978061
	3.0	39.141880	7.923048	0.999571	0.990993	0.964102
Foster City	0.5	54.159628	0.249526	0.999991	0.999424	0.992236
	1.0	51.146496	0.499379	0.999982	0.998854	0.984754
	1.5	47.041282	1.285144	0.999953	0.996862	0.963938
	2.0	45.069338	2.023701	0.999927	0.994801	0.945434
	2.5	41.643595	4.453698	0.999840	0.988834	0.894620
	3.0	39.312458	7.617886	0.999728	0.980479	0.840709
Anhinga	0.5	52.331008	0.380171	0.999970	0.999836	0.882762
	1.0	49.769127	0.685752	0.999947	0.999623	0.885525
	1.5	46.487973	1.459765	0.999887	0.998038	0.877124
	2.0	44.047515	2.560523	0.999803	0.996142	0.857092
	2.5	39.221181	7.779688	0.999406	0.985006	0.814428
	3.0	37.181427	12.443359	0.999047	0.976661	0.790102
Athens	0.5	51.861877	0.423538	0.999966	0.999864	0.971347
	1.0	49.368925	0.751948	0.999940	0.999667	0.974375
	1.5	45.932868	1.658798	0.999867	0.997738	0.955090
	2.0	43.286756	3.050736	0.999756	0.995544	0.936320

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	38.892677	8.390978	0.999329	0.987437	0.882569
	3.0	36.640344	14.094379	0.998873	0.973691	0.836150
Bardowl	0.5	52.484000	0.367012	0.999963	0.999858	0.999006
	1.0	50.012786	0.648338	0.999935	0.999610	0.998831
	1.5	46.222279	1.551859	0.999845	0.998987	0.995552
	2.0	43.840648	2.685440	0.999732	0.997978	0.991639
	2.5	39.702350	6.963784	0.999306	0.994719	0.978164
	3.0	37.467697	11.649588	0.998840	0.990413	0.965521
Barnfall	0.5	54.808111	0.214916	0.999966	0.999801	0.999011
	1.0	51.271809	0.485176	0.999923	0.999467	0.998049
	1.5	47.024471	1.290129	0.999794	0.998717	0.993306
	2.0	45.123572	1.998587	0.999683	0.997782	0.990059
	2.5	41.114044	5.031250	0.999201	0.993377	0.974168
	3.0	38.905384	8.366462	0.998677	0.988164	0.959346
Butrfly	0.5	52.256337	0.386764	0.999972	0.999828	0.995469
	1.0	49.814606	0.678609	0.999951	0.999587	0.995218
	1.5	46.290604	1.527636	0.999891	0.998505	0.987228
	2.0	43.788037	2.718170	0.999807	0.996843	0.979004
	2.5	39.467950	7.349965	0.999480	0.990956	0.949526
	3.0	37.237722	12.283103	0.999133	0.983467	0.923642
Bobcat	0.5	52.033893	0.407090	0.999942	0.999688	0.749572
	1.0	49.543243	0.722363	0.999898	0.999113	0.750874
	1.5	46.889226	1.330937	0.999812	0.997739	0.742716
	2.0	43.430456	2.951445	0.999584	0.994152	0.727149
	2.5	39.441229	7.395327	0.998964	0.983891	0.704846
	3.0	37.200344	12.389275	0.998260	0.969490	0.687538
Bodie	0.5	54.468152	0.232415	0.999962	0.999851	0.982874
	1.0	51.291557	0.482975	0.999920	0.999644	0.979014
	1.5	46.977533	1.304148	0.999784	0.999093	0.974842
	2.0	44.826406	2.140127	0.999647	0.998078	0.969692
	2.5	41.263812	4.860703	0.999177	0.996480	0.959254
	3.0	39.053951	8.085096	0.998634	0.993325	0.949577
Bluheron	0.5	54.080286	0.254127	0.999968	0.999680	0.996459
	1.0	51.426837	0.468162	0.999942	0.999193	0.995028
	1.5	46.394424	1.491550	0.999817	0.997991	0.986187
	2.0	44.414878	2.352840	0.999712	0.996568	0.977846

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	41.757889	4.338019	0.999468	0.995076	0.960984
	3.0	39.637399	7.068714	0.999134	0.990868	0.945290
Colomtn	0.5	53.537903	0.287932	0.999977	0.999732	0.985006
	1.0	50.855935	0.533933	0.999958	0.999428	0.984484
	1.5	46.641049	1.409208	0.999889	0.998428	0.977653
	2.0	44.537144	2.287525	0.999821	0.996902	0.970749
	2.5	40.926795	5.252920	0.999589	0.992867	0.953735
	3.0	38.811260	8.549767	0.999331	0.988266	0.937789
Desert	0.5	54.052085	0.255783	0.999963	0.999830	0.998898
	1.0	51.034300	0.512448	0.999927	0.999539	0.998291
	1.5	46.563793	1.434501	0.999794	0.998874	0.995712
	2.0	44.469911	2.323214	0.999666	0.997782	0.992718
	2.5	40.405538	5.922793	0.999142	0.994261	0.983178
	3.0	38.143991	9.969680	0.998549	0.989997	0.975890
Average case	0.5	53.646760	0.286920	0.999975	0.999685	0.973499
	1.0	50.779430	0.549709	0.999953	0.999298	0.970332
	1.5	46.730250	1.384128	0.999882	0.997944	0.958362
	2.0	44.564500	2.293218	0.999802	0.996233	0.946844
	2.5	40.808570	5.522234	0.999520	0.991150	0.917052
	3.0	38.584860	9.210028	0.999201	0.984687	0.890388

Performance comparison of WGD4_1x2 and WGD4_1x2_QE are made in terms of average PSNR at varying payload as evident from table 7.20. The WGD4_1x2_QE ensured higher average PSNR values over WGD4_1x2 as the payload exceeds 1 bpB. Therefore, the dispersion average PSNR values are to be observed for the payload range [1.5 – 3 bpB]. The minimum values of average PSNR for WGD4_1x2 and WGD4_1x2_QE are 37.52 and 38.58 dB respectively. Hence, all watermarked images at maximum payload also retained perceptible quality since the average PSNR is above 30 dB [148].

Table 7.20. Comparative analysis of obtained average PSNR values between WGD4_1x2 and WGD4_1x2_QE with respect to increasing payload

WGD4_1x2		WGD4_1x2_QE	
Payload (bpB)	PSNR (dB)	Payload (bpB)	PSNR (dB)
0.5	53.64676	0.5	53.64676
1.0	50.77943	1.0	50.77943
1.5	46.11457	1.5	46.73025

WGD4_1x2		**WGD4_1x2_QE**	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
2.0	43.80355	2.0	44.56450
2.5	39.60741	2.5	40.80857
3.0	37.52624	3.0	38.58486

The image quality of WGD4_1x2_QE has been analyzed (fig. 7.19) for “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” images in terms of PSNR (dB) with respect to varying level of payload (bpB). Varsaki et al.’s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] offered acceptable PSNR of greater than or equal to 30 dB [148] however, both of them are computed at the fixed payload values of 0.25 and 1 bpB respectively. To address the problem, the WGD4_1x2_QE has been designed and implemented which is focused on variable payload with a permissible level of visual acceptability (PSNR ≥ 30 dB) [148]. In comparison with DPTHDI [88], the WGD4_1x2_QE ensured higher PSNR (dB) at 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB for “Baboon”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for “Pepper”, 0.5, 1, 1.5 and 2 bpB of payloads for “Airplane” and 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for “Sailboat” respectively. Compared to DGTDHS [129], the WGD4_1x2_QE ensured higher PSNR (dB) at 0.5 and 1 bpB of payloads for “Lena”, 0.5 and 1 bpB of payloads for “Baboon”, 0.5 and 1 bpB of payloads for “Pepper”, 0.5 and 1 bpB of payloads for “Airplane” and 0.5 and 1 bpB of payloads for “Sailboat” respectively.

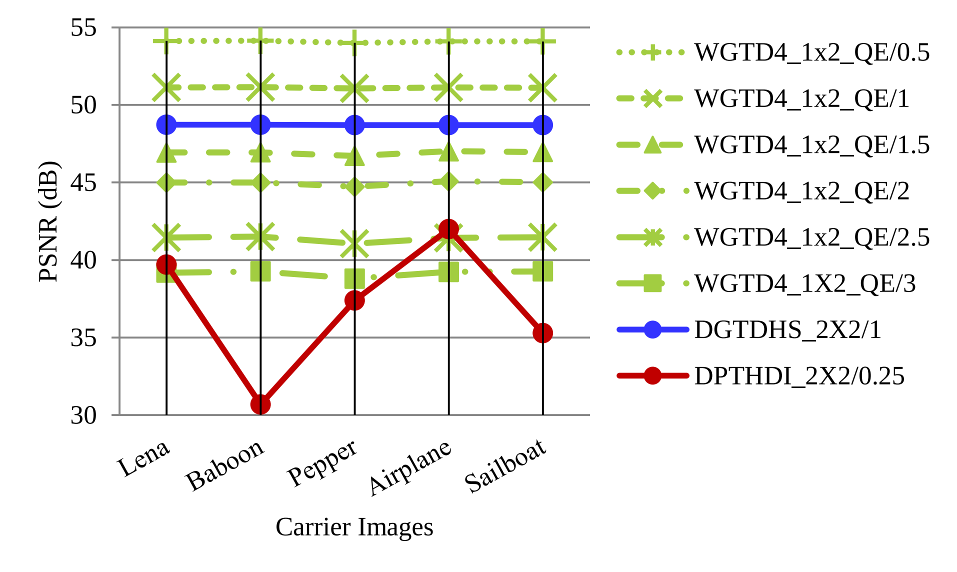


Fig. 7.19. Performance analysis of PSNR (dB) for variable payload based WGD4_1x2_QE and fixed payload based Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 7.20 illustrates the variations of average PSNR for WGD4_1x2_QE, WGD4_1x2, WGD4_2x2_QE, WST_1x2_QE, WBT_1x2_QE, WLT_1x2_QE, WDHT_1x2_QE, DPTHDI [88] and DGTDHS [129] respectively. In contrast to WGD4_1x2 scheme, the improvement of PSNR in WGD4_1x2_QE method can be visually perceived as the payload increases from 1 bpB. In contrast to WDHT_1x2_QE, WLT_1x2_QE, WBT_1x2_QE and WST_1x2_QE methods, the WGD4_1x2_QE ensured less degradation in fidelity with respect to variable payload though WGD4_2x2_QE offered outstanding results. The average PSNR for DPTHDI [88] scheme is 37.40 dB as the average of PSNR values for “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images at 0.25 bpB of payload. In contrast, the average PSNR of WGD4_1x2_QE scheme ensures equal or higher PSNR (dB) at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB which is the average of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” images at 1 bpB of payload. In contrast to DGTDHS [129], the average PSNR of WGD4_1x2_QE scheme offers higher PSNR (dB) at 1 bpB while the latter one also supports fabrication of secret information with variable payload. In general, the PSNR usually decreases while the payload increases. It is to be noted that the average PSNR values obtained at varying payload for proposed WGD4_1x2_QE is much greater than 30 dB and hence, the watermarked images retains high transparency [148]. In addition, no visual dissimilarity between the original and the watermarked images are observed.

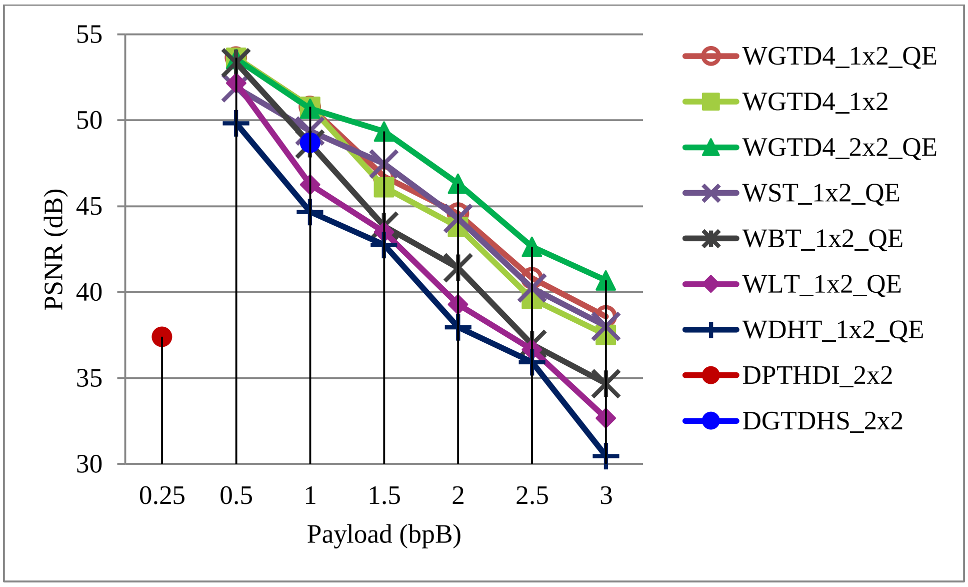


Fig. 7.20. Graphical representation of variation of average PSNR (dB) with respect to payload for WGD4_1X2_QE, WGD4_1X2, WGD4_2X2_QE, WST_1X2_QE, WBT_1X2_QE, WLT_1x2_QE, WDHT_1x2_QE and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

A comparative analysis has also been made between the hybrid GEMD [14] and the WGD4_1x2_QE techniques to check the improvement of image quality in terms of PSNR. The experiment is carried out for "Lena" image for payload variation of 1.5 – 3 bpB since, no PSNR dispersion is observed for both the schemes up to 1 bpB of payload. The PSNR values even at the highest payload (i.e., at 3 bpB) also crossed acceptable level and hence, the quality of the watermarked images provides good visual Imperceptibility [148].

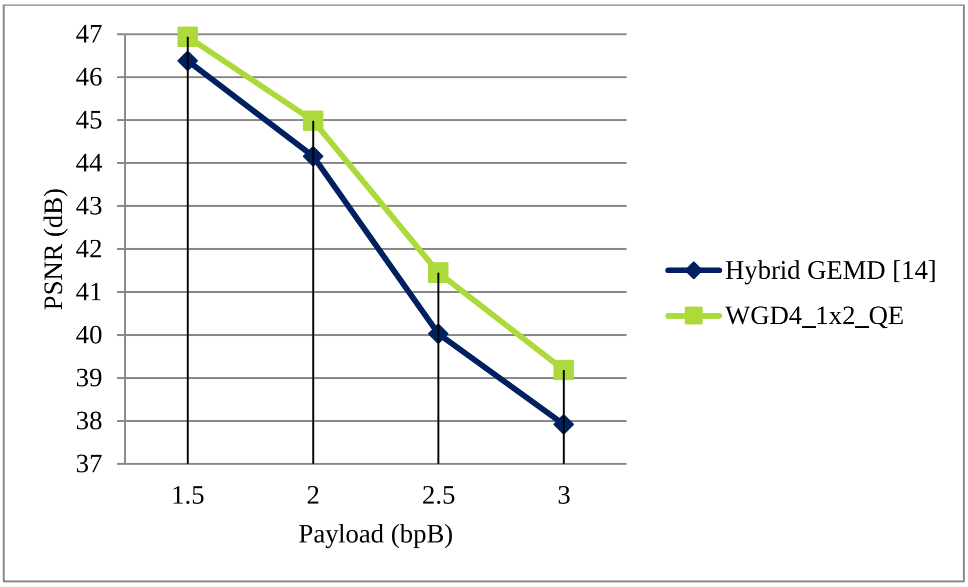


Fig. 7.21. Graphical representation to measure the improvement of PSNR (dB) with respect to variable payload (bpB) in WGD4_1x2_QE over hybrid GEMD method [14]

7.4. Salient Features

The quality enhancement techniques of sections 7.2.1 and 7.2.2 offered high quality watermarked images. Perceptual distortion is reduced considerably for majority of the proposed watermarking schemes as the payload exceeds 1 bpB of payload. The schemes are applied on each embedded component in such way that the concealed information is kept unaffected. The adaptive quality enhancement scheme of section 7.2.1 modifies the embedded components to reduce the differences between the original and embedded components. Unlikely, EMD based quality enhancement scheme of section 7.2.2 is applied prior to embedding to select the target pair of transformed components in which the secret bits are actually fabricated.

Chapter 8

Genetic Algorithm based Optimization (GAO)

8.1. Introduction

The effectiveness of a watermarking scheme is evaluated based on the characteristics such as imperceptibility, payload, robustness and bit error rate. A digital watermark is called “fragile” if it fails to be detectable after a bit modification. Fragile watermarking is an excellent choice of authenticating digital media wherein, the imperceptibility and payload plays the primary role. In general, increasing payload might reduce the imperceptibility and robustness whereas, decreasing payload might enhance the imperceptibility and robustness. As the characteristics are conflicting among themselves, the watermarking is considered to be an optimization problem. In this pretext, Genetic Algorithm (GA) has been used to solve the optimization problem due to several advantages such as its effective way of finding solution, parallelism and easy implementation etc. Therefore, the distortion of the watermarked images followed by GA optimization is minimized superbly over the proposed watermarking schemes (i.e., chapter 2 to chapter 6) prior to optimization. The processes have been designed in such way that the optimization does not affect the fabricated secret bits of each component.

8.2. Genetic Algorithm based Optimization

Proposed Genetic Algorithm (GA) begins with an initial population consisting of eight chromosomes. Each of the chromosomes preserves the secret bits into the least significant part. The fitness value for each of the chromosome is evaluated based on a fitness function and two parent chromosomes having the highest fitness values of each population are to be selected. Single point crossover is introduced into the middle of the bit-stream of each chromosome to generate new off-springs (children) from the parents. The new off-spring has been mutated by flipping a random bit position if only if, the fitness value becomes superior. Therefore, solutions from one population are taken and are used to form a new population by generating off-springs. Promising candidates are kept and allowed to reproduce by a hope, that the new population will be better than the old one. The process is repeated up to eight generations; however, a termination condition is specified if an optimized solution is produced.

Let, the embedded component is consisting of k bits of which r bits starting from the least significant bit (LSB-0) toward higher order bit position(s) are to be kept unaltered. The difference between the pre-embedded component (t) and the embedded component (t′) has been computed. If the difference is greater than $\pm 2^{r-1}$ then the optimization scheme take each embedded component (t′) as input and a component closest to the pre-embedded component

(t) is returned as the optimized component (t″) in output. However, if the optimized component (t″) is not found up to eighth generation as well as none of the termination condition is met, and then the embedded component (t′) itself is considered to be the optimized component (t″).

Algorithm 8.1:

Input: k bits of pre-embedded component (t) and embedded component (t′) respectively.

Output: k bits of optimized component (t″).

Method: The optimized component (t″) has been obtained by repeating the basic steps of Genetic Algorithm (GA) such as initial population, selection, cross-over and mutation. The optimized component (t″) does not lose the least significant r bits of each embedded component (t′). The steps are discussed as follows:

Step 1: The bit-stream corresponding to pre-embedded component (t), embedded component (t') and optimized component (t'') are considered to be partitioned into two segments i.e., (k-r) and r bits respectively. Here, k is the length of the component (in bits) of which r bits starting from the least significant bit (LSB-0) toward higher order bit position(s) are kept intact.

Step 2: Compute the difference between the pre-embedded component (t) and the embedded component (t'). If the difference is greater than $\pm 2^{r-1}$ *then proceed to next step to find the optimized component (t'') otherwise, the embedded component (t') is treated as the optimized component (t'') and go to step 11 to stop the searching.*

Step 3: An initial population is obtained in the first generation (i.e., g = 0) by generating random population of n chromosomes to optimize the embedded component (t'). The optimized solution can be obtained by choosing population size of eight i.e., n = 8.

Step 4: Each chromosome is a k-bit representation in the range of 0 to $2^k - 1$ *i.e., min = 0 and max =* $2^k - 1$. *To preserve the watermark information, least r bits of each chromosome are replaced by the least r bits of each embedded component (t').*

Step 5: Evaluate the fitness f(x) of each chromosome x in the population as given in equation (8.1).

$$f(x) = \frac{1}{|t - R_i| + 1} \qquad (8.1)$$

where, t is the pre-embedded component and R_i is the i^{th} chromosome of the population respectively.

Step 6: Select two parent chromosomes from the population of eight chromosomes according to their fitness (the better fitness, the bigger chance to be selected). According to the principle of steady-state selection, two chromosomes having less fitness are replaced by the pair of new off-springs though rest of them are survived.

Step 7: Single point crossover is applied on the pair of selected parents to form new off-springs (children).

Step 8: Randomly mutate the children at selected position(s) if and only if the mutated children produce higher fitness value than the as before.

Step 9: Generate new population by placing new off-springs. The value of min and max are updated at each generation.

Step 10: Repeat steps 4 to 9 up to eighth generation (i.e., $0 \leq g < 8$) to find the optimized component (t"). However, if the fitness value f(x) of any chromosome in new population is greater than or equal to a pre-specified threshold value (T) then it can be considered as a termination condition and jumps to step 11. The termination condition has been specified by the formula given in equation (8.2).

$$f(x) \geq T \qquad (8.2)$$

$$\text{where,} \quad T = \begin{cases} 1 : r = 1 \\ 0.5 : r = 2 \\ 0.25 : r = 3 \\ 0.125 : r = 4 \\ 0.0625 : r = 5 \end{cases}$$

Step 11: Stop.

8.3. Genetic Algorithm based Optimization in Watermarking

The incorporation of the genetic algorithm based optimization ensures improvement of quality in the watermarked image by keeping the fabricated information intact. The optimization can be applied over watermarking techniques of chapters 2, 3, 4, 5 and 6 (except section 6.2.2). The quality of the watermarked images is analyzed by means of the standard quality metrics such as the peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) respectively. Simulation results ensured that the techniques of chapters 2, 3, 4, 5 and 6 have been tremendously improved in terms of visual clarity.

Section 8.3.1 to 8.3.5 of this chapter deals with the optimizations of Discrete Hartley Transform (DHT) (chapter 2), Legendre Transform (LT) (chapter 3), Binomial Transform (BT) (chapter 4), Stirling Transform (ST) (chapter 5) and group of linear transformations for dihedral group of order 4 (G-lets D4) (chapter 6) based watermarking respectively.

8.3.1. Optimization of Discrete Hartley Transform (DHT) based watermarking

Discrete Hartley Transform (DHT) based watermarking has been implemented into two categories: one is applicable for 2 x 2 blocks (WDHT_2x2) and the other is the 1 x 2 block based watermark fabrication in Discrete Hartley Transform (DHT) domain (WDHT_1x2). Detailed discussions regarding both schemes are described in chapter 2.

Separable Discrete Hartley Transform (SDHT) followed by genetic algorithm based optimization (WDHT_2x2_GAO) has been elaborated in section 8.3.1.1, while section 8.3.1.2 deals with the results, analysis and discussions on 1 x 2 block based watermark fabrication using one dimensional Discrete Hartley Transform (1D-DHT) followed by genetic algorithm based optimization (WDHT_1x2_GAO).

8.3.1.1. Optimization of 2 x 2 Block based Watermark Fabrication

The WDHT_2x2_GAO highlights the WDHT_2x2 scheme of section 2.2.1 followed by GA optimization of section 8.2. In this scheme, the carrier image gets partitioned into 2 x 2 non-overlapping blocks and the pixel components are adjusted prior to embedding using equation (2.8) of section 2.2.1.1. Separable Discrete Hartley Transform (SDHT) converts each 2 x 2 sub-matrix of pixel components into transform domain. Secret bits for the message digest, size and the content of the watermark are fabricated on each transformed component starting from the second bit position of the least significant part (i.e., LSB-2) toward higher order bit position. Optimized component corresponding to each embedded component is computed based on the GA optimization of section 8.2. The optimized component preserved the two least significant bits along with the fabricated bits of the embedded component. Inverse Separable Discrete Hartley Transform (ISDHT) is applied over each 2 x 2 sub-matrices of optimized components to obtain pixel components. The process is repeated till the entire secret information is concealed in transform domain and the watermarked image is produced in spatial domain.

An example has been given in section 8.3.1.1.1. Results and discussions have been elaborated in section 8.3.1.1.2.

8.3.1.1.1. Example

Consider the 2 x 2 sub-matrices of red, green and blue channels to embed the secret information. Suppose, the payload value is 3 bpB and the pixel components are adjusted prior to embedding as discussed in equation (2.8) of section 2.2.1.1. The 2 x 2 sub-matrices are as given below:

$$R_1 = \begin{bmatrix} 224 & 69 \\ 32 & 112 \end{bmatrix} G_1 = \begin{bmatrix} 92 & 202 \\ 32 & 51 \end{bmatrix} B_1 = \begin{bmatrix} 32 & 119 \\ 220 & 224 \end{bmatrix}$$

Separable Discrete Hartley Transform (SDHT) is applied on 2 x 2 sub-matrices of pixel components to obtain the 2 x 2 sub-matrices of transformed components, T(R1), T(G1) and T(B1) as follows:

$$T(R_1) = \begin{bmatrix} 437 & 75 \\ 149 & 235 \end{bmatrix} T(G_1) = \begin{bmatrix} 377 & -129 \\ 211 & -91 \end{bmatrix} T(B_1) = \begin{bmatrix} 595 & -91 \\ -293 & -83 \end{bmatrix}$$

Secret bit stream of *“101000010110000001100111101101100000 1”* is to be fabricated into the transformed components of red, green and blue sub-matrices based on the WDHT_2x2 scheme of section 2.2.1. In this example, three bits are fabricated (λ = 3) on each transformed component starting from LSB-2 toward higher order bits position. The two least significant bits i.e., LSB-0 and LSB-1 are unaltered to generate non-fractional pixel components subsequent to inverse transform. Hence, the 2 x 2 sub-matrices of embedded components are as follows:

$$T'(R_1) = \begin{bmatrix} 437 & 67 \\ 137 & 239 \end{bmatrix} T'(G_1) = \begin{bmatrix} 353 & -153 \\ 207 & -95 \end{bmatrix} T'(B_1) = \begin{bmatrix} 603 & -91 \\ -289 & -83 \end{bmatrix}$$

Optimized component is obtained corresponding to each embedded component based on the genetic algorithm (GA) based optimization of section 8.2. The least two bits along with three fabricated bits (i.e., LSB-0, LSB-1, LSB-2, LSB-3 and LSB-4) of the embedded components are kept unaltered in the optimized component as well. The 2 x 2 sub-matrices of optimized components are as follows:

$$T''(R_1) = \begin{bmatrix} 437 & 67 \\ 137 & 239 \end{bmatrix} \; T''(G_1) = \begin{bmatrix} 385 & -121 \\ 207 & -95 \end{bmatrix} \; T'(B_1) = \begin{bmatrix} 603 & -91 \\ -289 & -83 \end{bmatrix}$$

Applying inverse Separable Discrete Hartley Transform (ISDHT) on each 2 x 2 sub-matrices of embedded components yields the 2 x 2 sub-matrices of pixel components as follows:

$$R'_1 = \begin{bmatrix} 220 & 67 \\ 32 & 118 \end{bmatrix} \; G'_1 = \begin{bmatrix} 94 & 202 \\ 38 & 51 \end{bmatrix} \; B'_1 = \begin{bmatrix} 35 & 122 \\ 221 & 225 \end{bmatrix}$$

The obtained 2 x 2 sub-matrices corresponding to RGB color channels are consisting of non-fractional and non-negative pixel components that lie between 0 and 255.

8.3.1.1.2. Results and Discussions

The WDHT_2x2_GAO scheme is the genetic algorithm based optimization (discussed in section 8.2) of the WDHT_2x2 scheme (discussed in section 2.2.1). The quality of the WDHT_2x2_GAO scheme is extensively analyzed in terms of peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) respectively. Twenty benchmark (BMP) images [130, 131] of dimension 512 x 512 along with the varying sizes of the fabricated watermark as given in fig. 1.1 are taken to derive results. The minimum and maximum values of PSNR are observed as 25.56 dB at 3 bpB of payload for the “Desert” image and 47.89 dB at 0.5 bpB of payload for the “San Diego” image respectively. The quality of the “Desert” (along with most of the images) is not above the perceptible level since the PSNR is below 30 dB [148]. In spite of this inability, proposed scheme is useful since it provides payload in the range [0.5 – 3 bpB]. The lowest MSE is 1.05 for “San Diego” at 0.5 bpB of payload and the highest MSE is 180.66 for “Desert” at 3 bpB of payload respectively. The minimum values of IF, SSIM and UIQ are 0.972828 (Desert), 0.881421 (Bobcat) and 0.410117 (Splash) respectively and that of the maximum values obtained are 0.999962 (Airplane), 0.999999 (Earth) and 0.996779 (San Diego) respectively. The IF, SSIM and UIQ values lie between 0 and 1 and that of values are maximum at 0.5 bpB and minimum at 3 bpB respectively. The average values are computed for various metrics of twenty images at variable payload that offers a spread from 0.5 to 3 bpB to summarize the experimental results.

Table 8.1. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WDHT_2x2_GAO scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	47.407641	1.181180	0.999927	0.999973	0.973613
	1.0	44.734959	2.185668	0.999863	0.996808	0.957305
	1.5	38.914849	8.348248	0.999499	0.996503	0.878756
	2.0	37.227911	12.310882	0.999276	0.989983	0.844857
	2.5	30.171458	62.507845	0.996325	0.978139	0.673343
	3.0	28.740007	86.911972	0.994858	0.954472	0.619212
Baboon	0.5	47.729109	1.096906	0.999942	0.999925	0.994269
	1.0	44.913212	2.097775	0.999889	0.998832	0.990719
	1.5	40.285076	6.089374	0.999679	0.997845	0.968974
	2.0	38.620533	8.933610	0.999530	0.995374	0.961050
	2.5	31.754915	43.409904	0.997733	0.987015	0.901211
	3.0	30.402321	59.271809	0.996902	0.978107	0.879601
Pepper	0.5	42.305850	3.823801	0.999629	0.983520	0.958071
	1.0	40.936259	5.241485	0.999503	0.979477	0.945116
	1.5	35.061643	20.272907	0.998046	0.966646	0.878714
	2.0	34.431401	23.439060	0.997787	0.960841	0.849959
	2.5	27.931599	104.693668	0.989966	0.932761	0.693056
	3.0	27.121431	126.164511	0.988245	0.910817	0.636374
Airplane	0.5	46.941701	1.314952	0.999962	0.999998	0.928282
	1.0	44.438064	2.340312	0.999933	0.996223	0.897211
	1.5	38.434484	9.324637	0.999733	0.996396	0.766766
	2.0	36.671808	13.992633	0.999599	0.988668	0.726782
	2.5	30.911568	52.713700	0.998491	0.988725	0.581898
	3.0	29.129895	79.449426	0.997726	0.959499	0.535361
Sailboat	0.5	46.959658	1.309527	0.999934	0.999891	0.978710
	1.0	44.411404	2.354723	0.999881	0.997320	0.965363
	1.5	39.107468	7.986077	0.999599	0.996295	0.913781
	2.0	37.531815	11.478860	0.999421	0.991118	0.890582
	2.5	30.620592	56.366504	0.997184	0.976679	0.762749
	3.0	29.269526	76.935672	0.996143	0.958070	0.718148
Earth	0.5	47.743824	1.093195	0.999935	0.999999	0.988634
	1.0	44.900632	2.103861	0.999875	0.997560	0.979644
	1.5	39.789042	6.826154	0.999598	0.997468	0.936329
	2.0	37.942710	10.442616	0.999394	0.992481	0.910701

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	31.140534	50.006556	0.996828	0.986177	0.775751
	3.0	29.613976	71.069402	0.995628	0.968258	0.727860
San Diego	0.5	47.897953	1.055079	0.999960	0.999999	0.996779
	1.0	44.978641	2.066408	0.999923	0.999086	0.994316
	1.5	40.700562	5.533808	0.999795	0.999129	0.984047
	2.0	38.898355	8.380015	0.999690	0.997255	0.977441
	2.5	33.401971	29.708670	0.998902	0.997146	0.939823
	3.0	31.593776	45.050818	0.998339	0.989590	0.920388
Splash	0.5	43.288726	3.049353	0.999687	0.986118	0.900533
	1.0	41.783047	4.312961	0.999566	0.980952	0.867927
	1.5	35.738106	17.348810	0.998290	0.976398	0.720489
	2.0	34.655598	22.259760	0.997924	0.966770	0.669617
	2.5	28.446135	92.996554	0.991062	0.951245	0.463881
	3.0	27.289759	121.368064	0.988737	0.918634	0.410117
Oakland	0.5	45.339798	1.901518	0.999877	0.999119	0.992304
	1.0	43.355342	3.002936	0.999816	0.997173	0.987753
	1.5	38.027976	10.239592	0.999356	0.995401	0.964372
	2.0	36.912857	13.237153	0.999202	0.991512	0.951981
	2.5	30.777240	54.369620	0.996674	0.985848	0.872446
	3.0	29.594706	71.385453	0.995847	0.970890	0.835002
Foster City	0.5	47.654231	1.115982	0.999960	0.999999	0.967908
	1.0	44.776217	2.165003	0.999923	0.995607	0.945088
	1.5	39.397298	7.470514	0.999742	0.995872	0.848600
	2.0	37.539598	11.458307	0.999603	0.986965	0.808123
	2.5	31.809775	42.865001	0.998530	0.987280	0.664319
	3.0	29.952106	65.746051	0.997743	0.953722	0.607114
Anhinga	0.5	43.903242	2.647013	0.999797	0.999711	0.860142
	1.0	42.401148	3.740809	0.999713	0.998247	0.849703
	1.5	36.759677	13.712373	0.998951	0.997407	0.798675
	2.0	35.544869	18.138167	0.998611	0.990698	0.778292
	2.5	29.577955	71.661323	0.994513	0.983414	0.700730
	3.0	28.450233	92.908833	0.992881	0.969171	0.673983
Athens	0.5	45.872814	1.681894	0.999865	0.999999	0.944283
	1.0	43.364844	2.996373	0.999760	0.998599	0.929877
	1.5	38.773795	8.623842	0.999310	0.998829	0.853411
	2.0	37.151228	12.530186	0.999001	0.991725	0.818516

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	31.494261	46.095039	0.996306	0.990329	0.709163
	3.0	29.843294	67.414123	0.994606	0.973652	0.667597
Bardowl	0.5	44.362844	2.381200	0.999755	0.998586	0.989803
	1.0	42.578614	3.591028	0.999633	0.997484	0.986911
	1.5	36.531600	14.451745	0.998499	0.987355	0.943918
	2.0	35.666457	17.637401	0.998180	0.985048	0.934033
	2.5	28.337573	95.350513	0.990124	0.944000	0.821458
	3.0	27.599686	113.008658	0.988375	0.938645	0.797048
Barnfall	0.5	46.665185	1.401398	0.999773	0.999921	0.991770
	1.0	44.139956	2.506598	0.999598	0.997916	0.986069
	1.5	39.330547	7.586222	0.998726	0.996648	0.957798
	2.0	37.743418	10.932978	0.998190	0.992575	0.941203
	2.5	31.439522	46.679705	0.991841	0.977764	0.837336
	3.0	29.983749	65.268760	0.988922	0.962392	0.797729
Butrfly	0.5	46.534886	1.444081	0.999897	0.999999	0.985351
	1.0	43.914079	2.640416	0.999812	0.998612	0.978703
	1.5	39.726833	6.924636	0.999507	0.998731	0.945367
	2.0	37.967051	10.384251	0.999260	0.994353	0.921624
	2.5	32.096216	40.129035	0.997099	0.992289	0.813875
	3.0	30.225880	61.729436	0.995567	0.978291	0.762456
Bobcat	0.5	46.235915	1.546994	0.999782	0.999991	0.734985
	1.0	43.641806	2.811251	0.999604	0.997918	0.727944
	1.5	38.421383	9.352807	0.998679	0.996908	0.689051
	2.0	36.845032	13.445504	0.998101	0.989050	0.674250
	2.5	26.228573	154.961221	0.978065	0.892258	0.549615
	3.0	25.788562	171.484198	0.975742	0.881421	0.531311
Bodie	0.5	44.134201	2.509922	0.999555	0.998801	0.967046
	1.0	42.586546	3.584476	0.999373	0.997278	0.962158
	1.5	36.192179	15.626525	0.997182	0.984118	0.919650
	2.0	35.512543	18.273681	0.996760	0.982223	0.909936
	2.5	27.457571	116.767832	0.978720	0.908179	0.764286
	3.0	26.879729	133.385139	0.975989	0.900298	0.745616
Bluheron	0.5	46.036184	1.619801	0.999801	0.999999	0.982729
	1.0	43.674114	2.790415	0.999658	0.998055	0.974714
	1.5	38.671988	8.828389	0.998919	0.997906	0.923437
	2.0	37.024301	12.901796	0.998416	0.992975	0.892749

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	31.087027	50.626469	0.993801	0.988479	0.744023
	3.0	29.364601	75.269711	0.990782	0.968763	0.677037
Colomtn	0.5	45.203325	1.962220	0.999846	0.999846	0.973598
	1.0	43.393759	2.976490	0.999767	0.997948	0.965955
	1.5	38.195069	9.853112	0.999230	0.997345	0.929047
	2.0	36.866238	13.380013	0.998955	0.992692	0.910310
	2.5	30.777818	54.362380	0.995760	0.983416	0.827712
	3.0	29.532960	72.407630	0.994349	0.967166	0.796130
Desert	0.5	40.323197	6.036158	0.999055	0.993480	0.983388
	1.0	39.293657	7.650937	0.998812	0.991600	0.980269
	1.5	33.120756	31.696019	0.995095	0.973915	0.934317
	2.0	32.901206	33.339548	0.994892	0.972486	0.925964
	2.5	25.795079	171.227058	0.974095	0.908083	0.811576
	3.0	25.562156	180.661112	0.972828	0.901693	0.802486
Average case	0.5	45.627010	2.008609	0.999797	0.997944	0.954610
	1.0	43.410820	3.157996	0.999695	0.995635	0.943637
	1.5	38.059020	11.304790	0.998872	0.992356	0.887775
	2.0	36.682750	14.844820	0.998590	0.987240	0.864899
	2.5	30.062870	71.874930	0.992601	0.966961	0.745413
	3.0	28.796920	91.844540	0.991010	0.950178	0.707029

It is seen from table 8.2 that WDHT_2x2 and WDHT_2x2_GAO are compared in terms of average PSNR with respect to the payload variation of 0.5 to 3 bpB; the values are ranges from 27.82 to 45.62 dB for WDHT_2x2 and from 28.79 to 45.62 dB for WDHT_2x2_GAO respectively. The average PSNR values for both schemes are identical for 0.5 and 1 bpB since, no optimization is applicable at that payload values. On the contrary, the latter one ensured improvement in PSNR over the former one for the payload range of 1.5 to 3 bpB. The average PSNR is above 30 dB for the payload values 0.5 – 2.5 bpB. On exceeding this range, the PSNR become less than 30 dB and create significant quality degradation [148].

Table 8.2. Comparative analysis of obtained average PSNR values between WDHT_2x2 and WDHT_2x2_GAO with respect to increasing payload

WDHT_2x2		WDHT_2x2_GAO	
Payload (bpB)	PSNR (dB)	Payload (bpB)	PSNR (dB)
0.5	45.627015	0.5	45.62701
1.0	43.410815	1.0	43.41082

WDHT_2x2		WDHT_2x2_GAO	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
1.5	37.527065	1.5	38.05902
2.0	35.680360	2.0	36.68275
2.5	29.564705	2.5	30.06287
3.0	27.828435	3.0	28.79692

The WDHT_2x2_GAO is compared with Varsaki et al.'s DPTHDI [88] and DGTDHS [129] in terms of PSNR (dB). Five color images viz. "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively (fig. 8.1) are taken to accomplish the analysis. The PSNR (dB) of the watermarked images DPTHDI [88] and DGTDHS [129] are obtained at 0.25 and 1 bpB respectively. The payload is fixed as well as low, compared to the WDHT_2x2_ GAO which is focused on variable payload that offers a spread from 0.5 to 3 bpB. The average PSNR value of WDHT_2x2_GAO is above 30 dB for the payload range (0.5 – 2.5 bpB) however, at 3 bpB significant quality loss (i.e., PSNR < 30 dB) is observed [148]. In contrast to DPTHDI [88], the WDHT_2x2_ GAO ensured equal or higher PSNR (dB) at 0.5 and 1 bpB of payloads for "Lena", 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Baboon", 0.5 and 1 bpB of payloads for "Pepper", 0.5 and 1 bpB of payloads for "Airplane" and 0.5, 1, 1.5 and 2 bpB of payloads for "Sailboat" respectively. Compared to DGTDHS [129], the WDHT_2x2_ GAO ensured variable payload (0.5 – 3 bpB) with slight loss in PSNR values.

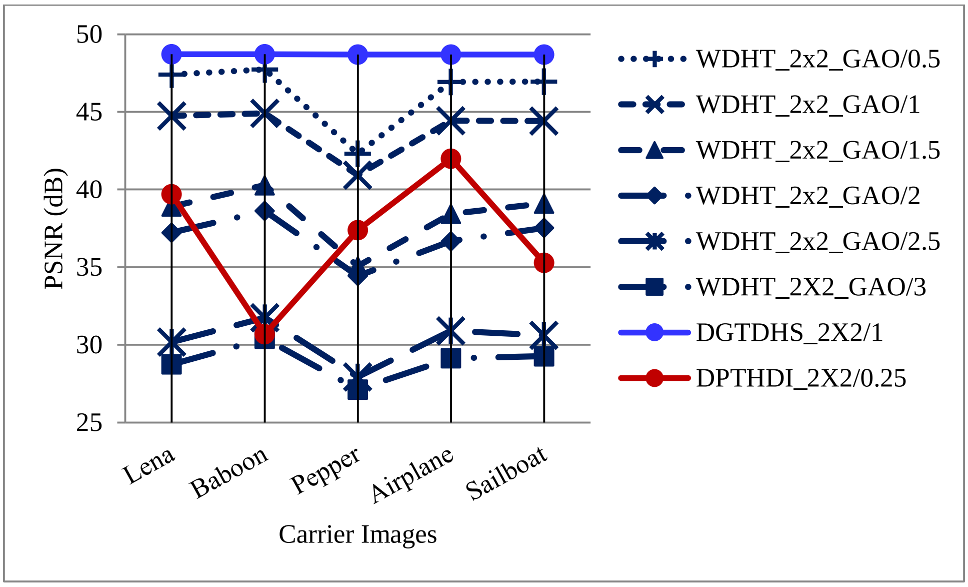


Fig. 8.1. Performance analysis of PSNR (dB) for variable payload based WDHT_2x2_GAO and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 8.2 illustrates the variation of average PSNR for WDHT_2x2_GAO, WDHT_2x2, DPTHDI [88] and DGTDHS [129] with respect to increasing payload. In contrast to WDHT_2x2 scheme, the average PSNR of WDHT_2x2_GAO scheme is gradually increasing as the payload increases from 1 bpB. Numerical analysis ensured that the average PSNR is greater than or equal to 30 dB for the payload values into the range (0.5 – 2.5 bpB) however, the average PSNR at 3 bpB is around 29 dB and hence, the quality of the watermarked images in average case slightly falls below the perceptible quality level [148]. The average PSNR for DPTHDI [88] is 37.40 dB as computed from the average of PSNR values for “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” respectively at 0.25 bpB of payload. In contrast to DPTHDI [88], the average PSNR of WDHT_2x2_GAO ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB which is computed from the average of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” images at 1 bpB of payload. In comparison with the fixed payload based DGTDHS [129], proposed WDHT_2x2_GAO offered variable payload with a minor loss of average PSNR.

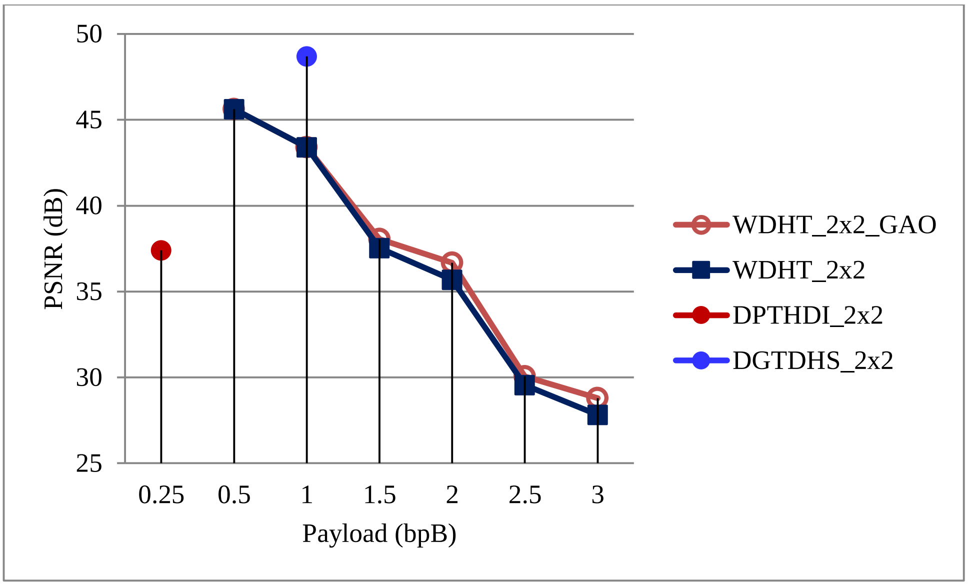


Fig. 8.2. Graphical representation of variation of average PSNR (dB) with respect to payload for WDHT_2x2_GAO, WDHT_2x2 and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

8.3.1.2. Optimization of 1 x 2 Block based Watermark Fabrication

The optimization scheme of section 8.2 is introduced as the post-embedding operation in WDHT_1x2 scheme of section 2.2.2. The carrier image is decomposed into 2 x 2 non-overlapping blocks where, the pixel components are adjusted prior to embedding as discussed in equation (2.17) of section 2.2.2.1. One dimensional Discrete Hartley Transform (1D-DHT) converts each 1 x 2 sub-matrix of pixel components into transform domain. The message digest, size and content of the watermark constitutes the secret bit-stream which are fabricated into the transformed components starting from the first bit position of the least significant part (i.e., LSB-1) toward higher order bit position. Genetic Algorithm (GA) based optimization of section 8.2 is applied on each embedded component in transform domain to find the optimized component. One dimensional inverse Discrete Hartley Transform (1D-DHT) is applied over 1 x 2 sub-matrices of optimized components to re-compute the pixel components in spatial domain. The process is repeated until and unless the fabrication of secret information is done and the resultant watermarked image is produced.

An example has been given in section 8.3.1.2.1 to illustrate the WDHT_1x2_GAO. Experimental results, analysis and discussions have been elaborated in section 8.3.1.2.2.

8.3.1.2.1. Example

Embedding of secret bit-stream into the 1 x 2 sub-matrices of pixel components is done subsequent to the pixel adjustment process described in equation (2.17) of section 2.2.2.1. The upper as well as the lower bounds are re-computed based on the offered payload of 3 bpB. The adjusted 1 x 2 sub-matrices corresponding to red, green and blue channels are obtained as follows:

$$R_1 = [240\ 69] \quad G_1 = [92\ 202] \quad B_1 = [16\ 110]$$

Each 1 x 2 sub-matrix or pair of pixel components is converted into transform domain based on one dimensional Discrete Hartley Transform (1D-DHT). The 1 x 2 sub-matrices of transformed components such as T(R_1), T(G_1) and T(B_1) are obtained as follows:

$$T(R_1) = [309\ 171] \quad T(G_1) = [294\ -110] \quad T(B_1) = [126\ -94]$$

Let the secret bit-stream *"101000010110000011"* is to be fabricated into the transformed components based on the proposed embedding strategy. In this example, three bits are fabricated (i.e., λ_1 =3, λ_2 = 3) on each transformed component starting from LSB-1 toward the higher order bit position. Hence, the 1 x 2 sub-matrices of embedded components are as follows:

$$T'(R_1) = [315\ 161] \quad T'(G_1) = [292\ -102] \quad T'(B_1) = [112\ -92]$$

Optimized component is obtained corresponding to each embedded component based on the genetic algorithm (GA) based optimization of section 8.2. The optimized component preserved the least significant bit along with three embedded bits in LSB-0, LSB-1, LSB-2 and LSB-3 of the embedded components. The 1 x 2 sub-matrices of optimized components are as given below:

$$T''(R_1) = [315\ 177] \quad T''(G_1) = [292\ -102] \quad T''(B_1) = [128\ -92]$$

Applying one dimensional inverse Discrete Hartley Transform (1D-IDHT) on each 1 x 2 sub-matrices of embedded components yields the 1 x 2 sub-matrices of pixel components in spatial domain as follows:

$$R'_1 = [246\ 69]\ G'_1 = [95\ 197]\ B'_1 = [18\ 110]$$

All re-generated pixel components preserved the hidden information besides keeping the values of pixel components in the range of 0 to 255.

8.3.1.2.2. Results and Discussions

The WDHT_1x2_GAO scheme is an extended version of WDHT_1x2 scheme of section 2.2.2 which offered better quality watermarked images due to the inclusion of genetic algorithm based optimization. The results are computed and the quality are analyzed for twenty benchmark (BMP) images [130, 131] of dimension 512 x 512, and the varying sizes of the fabricated watermark (fig. 1.1) with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. In WDHT_1x2_GAO, the minimum value of peak signal to noise ratio (PSNR) is 27.78 dB at 3 bpB of payload for the “Desert” image whereas, the maximum value of PSNR is 51.07 dB at 0.5 bpB of payload for the “San Diego” image. On the contrary, the minimum value of mean squared error (MSE) is 0.5 for “San Diego” at 0.5 bpB of payload and that of maximum value is 108.17 for “Desert” at 3 bpB of payload. The image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) lie in between 0 and 1 for ideal case but, the WDHT_1x2_GAO redefines the range as [0.999984 (Airplane), 0.979784 (Desert)] for IF, [0.999851 (Baboon), 0.934005 (Pepper)] for SSIM and [0.999658 (Baboon), 0.551887 (Splash)] for UIQ corresponding to payload values of 0.5 and 3 bpB respectively. The average experimental results at variable payload have also been computed as well as summarized to represent the experimental results. Since, the average PSNR is greater than or equal to 30 dB, the values of IF, SSIM and UIQ are closer to one and the WDHT_1x2_GAO provides variable payload for the range of (0.5 – 3 bpB), the quality of the watermarked images ensured acceptable visual transparency and higher similarity compared to the original images [148].

Table 8.3. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WDHT_1x2_GAO technique

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	50.968356	0.520289	0.999967	0.999503	0.989080
	1.0	45.929593	1.660049	0.999883	0.997784	0.972581
	1.5	43.722918	2.759234	0.999836	0.997326	0.944237
	2.0	38.937897	8.304061	0.999401	0.991618	0.896957
	2.5	36.414339	14.847263	0.999163	0.990318	0.831777
	3.0	31.327377	47.900784	0.9965633	0.968904	0.742683
Baboon	0.5	51.032210	0.512695	0.999971	0.999851	0.997658
	1.0	46.293562	1.5265960	0.999913	0.999291	0.992692
	1.5	44.429940	2.344694	0.999873	0.999069	0.987800
	2.0	39.693400	6.9781506	0.999599	0.997015	0.973893
	2.5	37.629757	11.222886	0.999400	0.995773	0.955117
	3.0	32.006639	40.965329	0.9976316	0.985383	0.916950
Pepper	0.5	48.386133	0.942902	0.999872	0.994562	0.984568
	1.0	45.118099	2.001107	0.999725	0.993856	0.970669
	1.5	41.772867	4.323083	0.999507	0.984128	0.943791
	2.0	36.550541	14.388854	0.997966	0.972656	0.894548
	2.5	35.010681	20.512196	0.997649	0.967921	0.847270
	3.0	29.214872	77.909987	0.988848	0.934005	0.754461
Airplane	0.5	50.821815	0.538144	0.999984	0.999355	0.969061
	1.0	45.111406	2.004193	0.999945	0.997335	0.919635
	1.5	43.136354	3.158238	0.999910	0.996733	0.871723
	2.0	37.997681	10.311271	0.999717	0.989428	0.801333
	2.5	35.797989	17.111235	0.999515	0.988363	0.715219
	3.0	30.703909	55.295450	0.998484	0.965219	0.640695
Sailboat	0.5	50.898925	0.528673	0.999972	0.999564	0.989218
	1.0	45.748234	1.730840	0.999909	0.998140	0.966383
	1.5	43.844957	2.682777	0.999861	0.997741	0.955255
	2.0	38.729947	8.711353	0.999545	0.992833	0.916541
	2.5	36.893590	13.296009	0.999310	0.991824	0.881793
	3.0	31.008813	51.546479	0.997307	0.970105	0.809606
Earth	0.5	51.016963	0.514498	0.999976	0.999637	0.994575
	1.0	45.748234	1.730840	0.999909	0.998140	0.966383
	1.5	43.844957	2.682777	0.999861	0.997741	0.955255
	2.0	38.729947	8.711353	0.999545	0.992833	0.916541

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	36.893590	13.296009	0.999310	0.991824	0.881793
	3.0	31.008813	51.546479	0.997307	0.970105	0.809606
San Diego	0.5	51.071997	0.508019	0.999983	0.999790	0.998316
	1.0	46.235591	1.547110	0.999952	0.999192	0.994439
	1.5	44.412342	2.354214	0.999919	0.999070	0.992168
	2.0	39.590280	7.145823	0.999775	0.996846	0.977857
	2.5	37.800980	10.789026	0.999636	0.996606	0.970360
	3.0	32.335894	37.974400	0.998819	0.988053	0.926529
Splash	0.5	50.681726	0.555786	0.999953	0.999244	0.955520
	1.0	44.978857	2.066305	0.999790	0.996538	0.896105
	1.5	42.119449	3.991493	0.999663	0.985465	0.819747
	2.0	37.608080	11.279043	0.998833	0.976681	0.731656
	2.5	35.046466	20.343877	0.998360	0.972214	0.642786
	3.0	30.270528	61.098065	0.993785	0.942137	0.551887
Oakland	0.5	49.694129	0.697697	0.999960	0.999125	0.995369
	1.0	45.194371	1.966270	0.999908	0.997632	0.983712
	1.5	42.746117	3.455163	0.999798	0.996803	0.981507
	2.0	38.448116	9.295413	0.999601	0.991534	0.939041
	2.5	36.098957	15.965577	0.999097	0.989462	0.931438
	3.0	31.048662	51.075677	0.997901	0.968227	0.834520
Foster City	0.5	50.983241	0.518508	0.999983	0.999153	0.983734
	1.0	45.308110	1.915443	0.999945	0.996628	0.939799
	1.5	43.694439	2.777386	0.999906	0.995955	0.923328
	2.0	38.262363	9.701614	0.999731	0.988173	0.842630
	2.5	36.511830	14.517682	0.999507	0.986242	0.788608
	3.0	31.006097	51.578730	0.998578	0.959990	0.694774
Anhinga	0.5	48.873701	0.842772	0.999932	0.999760	0.929419
	1.0	43.951777	2.617595	0.999786	0.997938	0.903337
	1.5	41.821673	4.274772	0.999664	0.998180	0.832805
	2.0	37.499109	11.565633	0.999059	0.994525	0.801275
	2.5	35.220517	19.544680	0.998464	0.992644	0.768404
	3.0	30.442762	58.722438	0.995218	0.975023	0.715393
Athens	0.5	48.885059	0.840571	0.999933	0.999815	0.974626
	1.0	44.371508	2.376454	0.999809	0.997981	0.946819
	1.5	42.454958	3.694745	0.999709	0.998399	0.911652
	2.0	38.188563	9.867884	0.999201	0.994971	0.870777

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	36.013879	16.281425	0.998723	0.993464	0.810151
	3.0	30.841269	53.573917	0.995657	0.975730	0.743566
Bardowl	0.5	48.212958	0.981260	0.999897	0.999280	0.996669
	1.0	43.393590	2.976605	0.999684	0.997964	0.986936
	1.5	42.107092	4.002866	0.999586	0.997886	0.983400
	2.0	36.437072	14.769748	0.998429	0.987814	0.947826
	2.5	35.096691	20.109961	0.997910	0.986760	0.930688
	3.0	28.544951	90.904464	0.990309	0.951880	0.867600
Barnfall	0.5	50.558089	0.571835	0.999880	0.999616	0.996691
	1.0	44.737727	2.184276	0.999495	0.998264	0.986850
	1.5	42.870660	3.357486	0.999368	0.997998	0.980203
	2.0	38.230199	9.773732	0.997796	0.992556	0.949627
	2.5	36.320091	15.172991	0.997132	0.991585	0.925367
	3.0	30.833446	53.670506	0.987683	0.964934	0.841708
Butrfly	0.5	49.557670	0.719968	0.999941	0.999787	0.992709
	1.0	44.926125	2.091547	0.999821	0.998712	0.981205
	1.5	43.069575	3.207176	0.999758	0.998688	0.972598
	2.0	38.706347	8.758819	0.999251	0.996030	0.943385
	2.5	36.576338	14.303637	0.998914	0.995029	0.907870
	3.0	31.189378	49.447288	0.995724	0.982027	0.835108
Bobcat	0.5	49.193921	0.782867	0.999873	0.999582	0.832578
	1.0	44.406341	2.357470	0.999595	0.997612	0.820028
	1.5	42.529374	3.631975	0.999458	0.997637	0.718001
	2.0	38.050503	10.186618	0.998243	0.992800	0.697607
	2.5	35.815828	17.041095	0.997442	0.991208	0.668493
	3.0	28.791130	85.894892	0.984741	0.945123	0.607693
Bodie	0.5	49.352944	0.754720	0.999857	0.999536	0.982208
	1.0	43.908857	2.643592	0.999469	0.998227	0.972674
	1.5	42.219871	3.900257	0.999302	0.997915	0.958066
	2.0	37.176568	12.457288	0.997497	0.990033	0.932889
	2.5	35.507547	18.294713	0.996678	0.988166	0.910788
	3.0	28.913329	83.511709	0.982909	0.942039	0.827759
Bluheron	0.5	49.534888	0.723754	0.999910	0.999656	0.992862
	1.0	44.444451	2.336873	0.999716	0.998481	0.978717
	1.5	42.511963	3.646565	0.999549	0.998152	0.965440
	2.0	37.959736	10.401757	0.998740	0.993878	0.924170

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	35.745535	17.319159	0.997857	0.993535	0.873552
	3.0	30.711811	55.194937	0.993309	0.975639	0.781964
Colomtn	0.5	50.031172	0.645599	0.999950	0.999658	0.999658
	1.0	44.435306	2.341799	0.999821	0.998286	0.977217
	1.5	42.543328	3.620324	0.999720	0.998101	0.959588
	2.0	37.915287	10.508762	0.999199	0.993826	0.932774
	2.5	36.006643	16.308574	0.998740	0.992864	0.900159
	3.0	30.679522	55.606826	0.995766	0.971847	0.836382
Desert	0.5	45.674444	1.760499	0.999694	0.997437	0.994520
	1.0	41.091814	5.057069	0.999056	0.994583	0.987908
	1.5	39.505811	7.286168	0.998824	0.993089	0.980341
	2.0	34.697578	22.045630	0.995881	0.982647	0.960308
	2.5	32.930192	33.117768	0.994723	0.976407	0.933579
	3.0	27.789489	108.176116	0.979784	0.940552	0.876176
Average case	0.5	49.771520	0.723053	0.999924	0.999196	0.977452
	1.0	44.766680	2.256602	0.999757	0.997629	0.957204
	1.5	42.767930	3.557570	0.999654	0.996304	0.931845
	2.0	37.970460	10.758140	0.998850	0.990435	0.892582
	2.5	35.966570	16.969790	0.998377	0.988610	0.853761
	3.0	30.433430	61.079720	0.993316	0.963846	0.780753

Genetic algorithm (GA) based optimization makes the difference between proposed WDHT_1x2_GAO and WDHT_1x2 in terms of average PSNR for the range of payload [1 – 3 bpB]. Since, the optimization process does not make any impact on the average PSNR at 0.5 bpB, identical values of average PSNR are achieved for both schemes. The permissible range of average PSNR values for WDHT_1x2_GAO and WDHT_1x2 are belonging to the following ranges: (28.62 – 49.77 dB) and (30.43 – 49.77 dB) respectively. The average PSNR value of WDHT_1x2_GAO at 3 bpB ensured an average PSNR of greater than 30 dB and hence, the obtained watermarked images are perceived as high-quality images [148].

Table 8.4. Comparative analysis of average PSNR between WDHT_1x2 and WDHT_1x2_GAO with respect to increasing payload

WDHT_1x2		WDHT_1X2_GAO	
Payload (bpB)	PSNR (dB)	Payload (bpB)	PSNR (dB)
0.5	49.77152	0.5	49.77152
1.0	42.93340	1.0	44.76668

WDHT_1x2		WDHT_1X2_GAO	
Payload (bpB)	PSNR (dB)	Payload (bpB)	PSNR (dB)
1.5	41.31294	1.5	42.76793
2.0	35.96829	2.0	37.97046
2.5	34.38548	2.5	35.96657
3.0	28.62825	3.0	30.43343

Varsaki et al.'s Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] provides acceptable PSNR values (i.e, ≥ 30 dB) for "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat" respectively (fig. 8.3). Considering the block size as 2 x 2, the obtained payload values for both schemes are computed as 0.25 and 1 bpB respectively. However, these fixed payloads are treated as significantly low as far as the variable payload (0.5 – 3 bpB) based WDHT_1x2_GAO is concerned. In contrast to DPTHDI [88], the WDHT_1x2_GAO ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for "Lena", 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Baboon", 0.5, 1 and 1.5 bpB of payloads for "Pepper", 0.5, 1 and 1.5 bpB of payloads for "Airplane" and 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for "Sailboat" respectively. Compared to DGTDHS [129], the PSNR values for the test images are slightly lacking for WDHT_1x2_GAO at 1 bpB. The WDHT_1x2_GAO offered PSNR of 30 dB or more at varying payloads and hence, results well perceptible watermarked images [148].

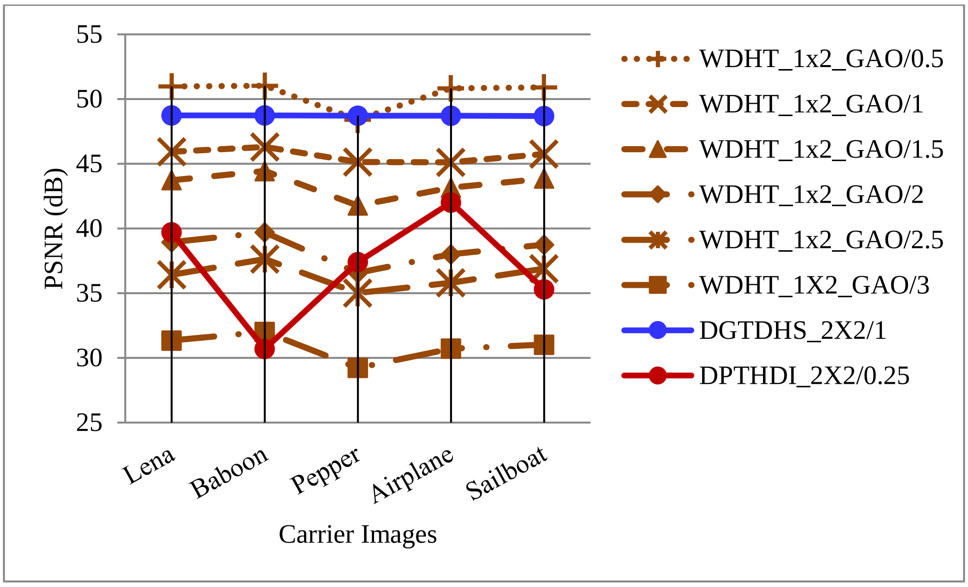


Fig. 8.3. Performance analysis of PSNR (dB) for variable payload based WDHT_1x2_GAO and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 8.4 illustrates the average peak signal to noise ratio (PSNR) analysis with respect to the variable payload (WDHT_1x2_GAO, WDHT_1x2 and WDHT_2x2_GAO) and fixed payload (DPTHDI [88] and DGTDHS [129]) based watermarking techniques, respectively. The average PSNR values of WDHT_1x2_GAO are obtained by averaging the PSNR values of twenty benchmark images (fig. 1.1) for the payload range of 0.5 to 3 bpB. In contrast to WDHT_1x2 and WDHT_2x2_GAO, the average PSNR of WDHT_1x2_GAO is increasing in a consistent manner while the payload value increases from 0.5 bpB. The average of PSNR values for “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images is 37.40 dB at 0.25 bpB of payload which is treated as the average PSNR for DPTHDI [88]. Compared to DPTHDI [88], the average PSNR of WDHT_1x2_GAO ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads. Again, the average of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat”, “F16” images is 48.70 dB at 1 bpB of payload and the same is treated as the average PSNR for DGTDHS [129]. In comparison with DGTDHS [129], the average PSNR of WDHT_1x2_GAO is lacking at 1 bpB however, the ability of providing variable payload for a spread from 0.5 to 3 bpB is considered to be a major accomplishment. In general, the average PSNR values decreases while the payload values increases however, the trade-off between these two metrics are to be maintained. The WDHT_1x2_GAO provides at least an average PSNR of 30 dB for the payload variation of 0.5 to 3 bpB. As a consequence, the obtained watermarked images preserved good visual clarity as far as the average PSNR values are concerned [148].

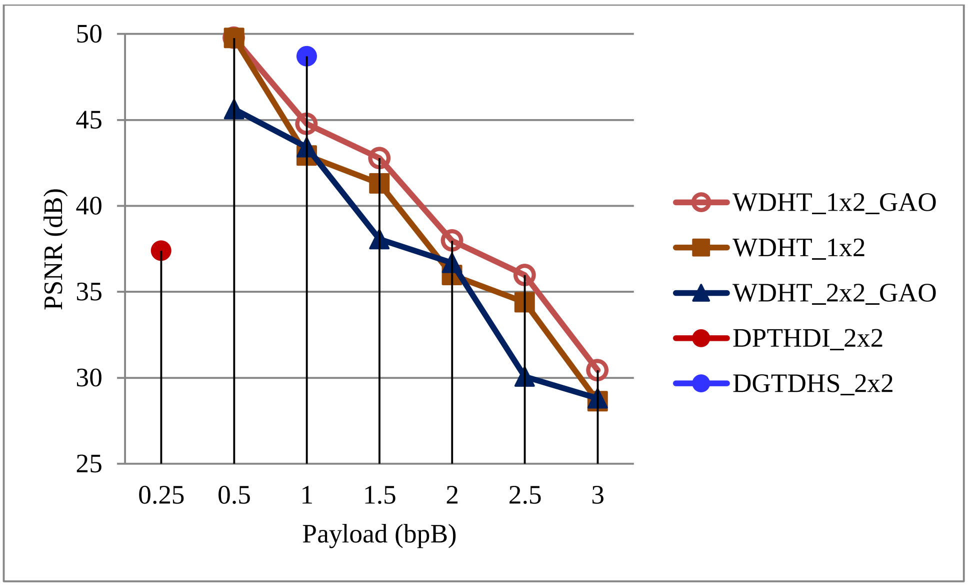


Fig. 8.4. Graphical representation of variation of average PSNR (dB) with respect to payload for WDHT_1x2_GAO,WDHT_1x2, WDHT_2x2_GAO and Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes

8.3.2. Optimization of Legendre Transform (LT) based watermarking

Two watermarking schemes operated in Legendre Transform (LT) are distinguished from each other based on their block sizes as discussed in chapter 3. Both the 2 x 2 block based scheme (WLT_2x2) and the 1 x 2 block based scheme (WLT_1x2) ensured the fabrication of secret information for the variable payload that offers a spread from 0.5 to 3 bpB respectively.

Section 8.3.2.1 deals with the 2 x 2 block based watermark fabrication using Legendre Transform (LT) followed by Genetic Algorithm based optimization (WLT_2x2_GAO) and that of 1 x 2 block based watermark fabrication followed by Genetic Algorithm based optimization (WLT_1x2_GAO) has been elaborated in section 8.3.2.2.

8.3.2.1. Optimization of 2 x 2 Block based Watermark Fabrication

The WLT_2x2_GAO takes the carrier image as the input, decomposes the image into 2 x 2 non-overlapping blocks and then the pixel components are adjusted prior to embedding as discussed in equation (3.7) of section 3.2.1.1. Each 2 x 2 block is converted into transform domain based on Legendre Transform (LT). Secret bits corresponding to the message digest, size and content of the watermark are fabricated on first/second/third transformed component starting from the first bit position of the least significant part (i.e., LSB-1) toward higher order bit position. Genetic Algorithm (GA) optimizes embedded components to keep the transparency closer to the original. An additional re-adjustment operation is also incorporated to avoid overflow/underflow. Inverse Legendre Transform (ILT) is applied on each 2 x 2 block of optimized components to obtain the pixel components. This process is repeated until and unless the entire bit-stream corresponding to the message digest (MD), watermark size and the content is embedded.

In section 8.3.2.1.1, an example is given to describe the process of embedding along with the quality enhancement through optimization. Simulation results are computed, analyzed and discussed by making comparisons with the existing techniques as given in section 8.3.2.1.2.

8.3.2.1.1. Example

The source image is partitioned into 2 x 2 non-overlapping blocks corresponding to RGB color channels. The adjustment of pixel components for the payload value of 3 bpB is carried out based on the pixel adjustment process as discussed in equation (3.7) of section 3.2.1.1. The 2 x 2 sub-matrices are consisting of adjusted pixel components corresponding to red/green/blue channel as follows:

$$R_1 = \begin{bmatrix} 224 & 69 \\ 32 & 112 \end{bmatrix} G_1 = \begin{bmatrix} 92 & 202 \\ 32 & 51 \end{bmatrix} B_1 = \begin{bmatrix} 32 & 119 \\ 220 & 224 \end{bmatrix}$$

Each 2 x 2 sub-matrix of pixel components is converted into transform domain based on Legendre Transform (LT). The 2 x 2 sub-matrices of transformed components such as T(R1), T(G1) and T(B1) are obtained as given below:

$$T(R_1) = \begin{bmatrix} 224 & 362 \\ 830 & 4252 \end{bmatrix} T(G_1) = \begin{bmatrix} 92 & 496 \\ 1496 & 4496 \end{bmatrix} T(B_1) = \begin{bmatrix} 32 & 270 \\ 2066 & 12540 \end{bmatrix}$$

The secret bit-stream *"10100001011000001100111101101100001"* is considered for fabrication into the transformed components. The payload value of 3 bpB ensured the following embedding distributions on first, second and third transformed components: N_1 = 4, N_2 = 4 and N_3 = 4 respectively. On embedding, the LSB-0 of these components are not hampered since the fabrication starts from the first least significant bit position (LSB-1) toward higher order bit position. Hence, the 2 x 2 sub-matrices of embedded components are as given below:

$$T'(R_1) = \begin{bmatrix} 234 & 368 \\ 812 & 4252 \end{bmatrix} T'(G_1) = \begin{bmatrix} 64 & 486 \\ 1502 & 4496 \end{bmatrix} T'(B_1) = \begin{bmatrix} 44 & 262 \\ 2064 & 12540 \end{bmatrix}$$

Optimized component is found corresponding to each embedded component in terms of the Genetic Algorithm (GA) based optimization of section 8.2. The optimized component has preserved the unaltered least significant bit and four embedded bits i.e., LSB-0, LSB-1, LSB-2, LSB-3 and LSB-4 respectively. The optimized 2 x 2 sub-matrices are obtained as follows:

$$T''(R_1) = \begin{bmatrix} 234 & 368 \\ 844 & 4252 \end{bmatrix} T''(G_1) = \begin{bmatrix} 96 & 486 \\ 1502 & 4496 \end{bmatrix} T'(B_1) = \begin{bmatrix} 44 & 262 \\ 2064 & 12540 \end{bmatrix}$$

Inverse Legendre Transform (ILT) is applied on 2 x 2 sub-matrices of optimized components to obtain 2 x 2 sub-matrices of pixel components namely R'$_1$, G'$_1$ and B'$_1$ corresponding to red, green and blue channels as follows:

$$R'_1 = \begin{bmatrix} 234 & 67 \\ 40 & 108 \end{bmatrix} G'_1 = \begin{bmatrix} 96 & 195 \\ 50 & 44 \end{bmatrix} B'_1 = \begin{bmatrix} 44 & 109 \\ 233 & 217 \end{bmatrix}$$

8.3.2.1.2. Results and Discussions

Standard metrics such as peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) are computed for WLT_2X2_GAO scheme. Table 8.5 shows that twenty benchmark (BMP) images [130, 131] of dimension 512 x 512 and the varying sizes of the fabricated watermark (fig. 1.1) has been considered to compute results. The ranges of PSNR (in terms of dB) and MSE values are derived with respect to minimum (i.e., 0.5 bpB) and maximum (i.e., 3 bpB) payload as [49.26 (Bluheron), 24.53 (Desert)] and [0.77 (Bluheron), 228.89 (Desert)] respectively. The minimum PSNR value corresponding to the "Desert" (as well as the PSNR of other images) at 3 bpB falls below the acceptable level and therefore, the quality of the watermarked images is distorted. Since, the average PSNR value of 28.26 dB ≈ 30 dB, and the variable payload is achieved, the WLT_2x2_GAO proves its effectiveness. The computed values of IF, SSIM and UIQ lie between 0 and 1, the closer the IF, SSIM and UIQ to one, watermarked image is more similar to the original one. Table 8.5 also reveals that the minimum values of the IF, SSIM and UIQ are obtained at 3 bpB where, the values are 0.965733 (Desert), 0.877702 (Bobcat) and 0.359286 (Splash) respectively. On the contrary, the highest values are obtained for the same metrics at 0.5 bpB of payload and the corresponding values are 0.999973 (Airplane), 0.999904 (San Diego) and 0.997381 (San Diego) respectively. The average experimental results of twenty carrier images at variable payload have also been computed to summarize the experimental results.

Table 8.5. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WLT_2x2_GAO scheme

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Lena	0.5	48.597498	0.898111	0.999943	0.999667	0.979761
	1.0	45.132935	1.994283	0.999873	0.999098	0.959802
	1.5	41.577018	4.522499	0.999715	0.997839	0.920315
	2.0	37.248444	12.252815	0.999243	0.993739	0.841584
	2.5	31.646705	44.505105	0.997264	0.974116	0.695497
	3.0	29.006159	81.745606	0.994863	0.960599	0.571682
Baboon	0.5	48.591863	0.899278	0.999952	0.999877	0.996092
	1.0	45.099157	2.009854	0.999893	0.999613	0.991218
	1.5	41.540447	4.560742	0.999758	0.999140	0.981516
	2.0	37.132478	12.584398	0.999336	0.996853	0.953994
	2.5	31.816138	42.802238	0.997755	0.985940	0.902275
	3.0	29.121291	79.606988	0.995802	0.981365	0.854115
Pepper	0.5	45.895704	1.673053	0.999845	0.990208	0.968212
	1.0	40.942759	5.233646	0.999502	0.981532	0.945641
	1.5	39.524125	7.255507	0.999346	0.981231	0.918081
	2.0	34.196365	24.742515	0.997704	0.964514	0.838936
	2.5	28.142354	99.734395	0.990330	0.927976	0.693651
	3.0	26.700452	139.006501	0.987136	0.913823	0.572788

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	48.538964	0.910298	0.999973	0.999601	0.944582
	1.0	45.018482	2.047538	0.999941	0.998994	0.900542
	1.5	41.483915	4.620498	0.999867	0.997417	0.830175
	2.0	37.275676	12.176225	0.999651	0.992824	0.726404
	2.5	33.169236	31.344168	0.999101	0.983254	0.599376
	3.0	29.416736	74.371522	0.997872	0.964925	0.486183
Sailboat	0.5	48.314324	0.958623	0.999951	0.999706	0.983467
	1.0	44.587799	2.260999	0.999886	0.999159	0.967330
	1.5	41.426239	4.682270	0.999763	0.998186	0.939175
	2.0	36.917693	13.222421	0.999334	0.994080	0.881605
	2.5	31.007062	51.567268	0.997429	0.972778	0.755661
	3.0	28.577564	90.224371	0.995471	0.960961	0.662416
Earth	0.5	48.593479	0.898943	0.999946	0.999748	0.990599
	1.0	45.169969	1.977349	0.999882	0.999335	0.980607
	1.5	41.534885	4.566588	0.999729	0.998373	0.958837
	2.0	37.317265	12.060179	0.999291	0.995250	0.905494
	2.5	32.019307	40.846007	0.997289	0.982282	0.792952
	3.0	29.034815	81.207998	0.994944	0.970252	0.680534
San Diego	0.5	48.604975	0.896567	0.999966	0.999904	0.997381
	1.0	45.187262	1.969491	0.999926	0.999744	0.994648
	1.5	41.563373	4.536730	0.999831	0.999371	0.988900
	2.0	37.329441	12.026416	0.999554	0.998258	0.975066
	2.5	33.383715	29.833813	0.998882	0.995682	0.946998
	3.0	29.639409	70.654429	0.997369	0.990517	0.895446
Splash	0.5	46.648097	1.406923	0.999862	0.989492	0.919325
	1.0	42.053995	4.052106	0.999584	0.984311	0.869646
	1.5	40.181911	6.235757	0.999409	0.983091	0.788203
	2.0	35.294647	19.213902	0.998131	0.973162	0.662292
	2.5	29.607098	71.182048	0.992667	0.945844	0.472664
	3.0	27.946675	104.330874	0.989909	0.930291	0.359286
Oakland	0.5	47.586542	1.133511	0.999933	0.999430	0.994396
	1.0	43.422098	2.957130	0.999816	0.998467	0.988401
	1.5	40.837865	5.361592	0.999692	0.997802	0.977408
	2.0	36.055058	16.127777	0.999028	0.993615	0.946626
	2.5	30.960369	52.124680	0.996602	0.982773	0.888920
	3.0	28.445748	93.004829	0.994465	0.972130	0.798155

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Foster City	0.5	48.612614	0.894991	0.999968	0.999545	0.974108
	1.0	45.189175	1.968624	0.999930	0.998810	0.947870
	1.5	41.547689	4.553143	0.999838	0.997040	0.898126
	2.0	37.323400	12.043156	0.999570	0.991696	0.798803
	2.5	33.471215	29.238751	0.998957	0.981367	0.661667
	3.0	29.480568	73.286420	0.997427	0.955840	0.523338
Anhinga	0.5	47.728449	1.097072	0.999915	0.999712	0.872690
	1.0	43.261313	3.068662	0.999764	0.999271	0.857900
	1.5	40.258976	6.126080	0.999530	0.998309	0.825140
	2.0	35.765527	17.239615	0.998678	0.995927	0.798242
	2.5	30.699210	55.355314	0.995760	0.983442	0.722780
	3.0	28.547480	90.851547	0.993019	0.975625	0.655189
Athens	0.5	48.558146	0.906286	0.999927	0.999750	0.960121
	1.0	44.877145	2.115269	0.999830	0.999508	0.941960
	1.5	40.892945	5.294022	0.999577	0.998540	0.891393
	2.0	37.477108	11.624373	0.999075	0.996839	0.859731
	2.5	33.332309	30.189046	0.997577	0.988955	0.764249
	3.0	29.588175	71.492890	0.994290	0.979069	0.643558
Bardowl	0.5	47.339213	1.199938	0.999878	0.999446	0.995658
	1.0	43.080252	3.199301	0.999672	0.997949	0.987682
	1.5	40.516820	5.772956	0.999418	0.997970	0.979255
	2.0	35.366143	18.900184	0.998062	0.987162	0.935033
	2.5	28.399182	94.007409	0.990236	0.941326	0.832492
	3.0	26.974798	130.497004	0.986612	0.937894	0.781835
Barnfall	0.5	48.949717	0.828149	0.999866	0.999781	0.994941
	1.0	45.333442	1.904303	0.999688	0.999379	0.989849
	1.5	41.289143	4.832434	0.999228	0.998738	0.975303
	2.0	37.005538	12.957656	0.997874	0.995541	0.939676
	2.5	31.873749	42.238198	0.992437	0.975597	0.852275
	3.0	29.173027	78.664282	0.986763	0.966422	0.766607
Butrfly	0.5	48.670968	0.883046	0.999937	0.999787	0.990622
	1.0	45.019275	2.047164	0.999854	0.999569	0.981499
	1.5	41.205586	4.926310	0.999650	0.998878	0.960536
	2.0	37.460826	11.668033	0.999170	0.997167	0.928369
	2.5	33.206131	31.079012	0.997724	0.990884	0.849029
	3.0	29.638817	70.664062	0.994922	0.983152	0.740509

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Bobcat	0.5	48.596957	0.898223	0.999873	0.999582	0.743136
	1.0	45.192245	1.967233	0.999723	0.998900	0.737329
	1.5	41.574994	4.524607	0.999365	0.997969	0.717569
	2.0	37.422279	11.772057	0.998335	0.993987	0.687194
	2.5	26.040010	1.618375	0.977076	0.884188	0.550065
	3.0	25.192065	1.967314	0.972181	0.877702	0.508067
Bodie	0.5	46.899420	1.327817	0.999766	0.999528	0.972632
	1.0	42.758522	3.445308	0.999385	0.998258	0.963156
	1.5	40.667888	5.575598	0.999036	0.997923	0.953683
	2.0	35.037779	20.384607	0.996372	0.983853	0.912947
	2.5	27.337540	120.040102	0.977946	0.905508	0.764612
	3.0	26.209755	155.634115	0.972047	0.900450	0.712640
Bluheron	0.5	49.264767	0.770200	0.999905	0.999835	0.991207
	1.0	45.856847	1.688090	0.999793	0.999468	0.982931
	1.5	41.231224	4.897313	0.999400	0.998510	0.956234
	2.0	37.428546	11.755082	0.998559	0.995913	0.909060
	2.5	32.796008	34.156982	0.995808	0.984579	0.808378
	3.0	29.374897	75.091478	0.990802	0.974435	0.669690
Colomtn	0.5	48.019783	1.025892	0.999919	0.999758	0.979110
	1.0	44.048032	2.560218	0.999799	0.999365	0.972361
	1.5	41.014947	5.147373	0.999597	0.998639	0.956419
	2.0	36.678228	13.971964	0.998908	0.996012	0.935078
	2.5	31.221347	49.084636	0.996177	0.981273	0.852400
	3.0	28.773202	86.250207	0.993266	0.971562	0.770039
Desert	0.5	44.542082	2.284926	0.999641	0.997288	0.992252
	1.0	39.425545	7.422083	0.998823	0.992200	0.981072
	1.5	38.256971	9.713667	0.998524	0.992726	0.974466
	2.0	32.393095	37.477516	0.994239	0.974231	0.932169
	2.5	25.260969	193.634778	0.970698	0.904067	0.797688
	3.0	24.534361	228.899185	0.965733	0.900417	0.762753
Average case	0.5	47.927680	1.089592	0.999898	0.998582	0.962015
	1.0	44.032810	2.794433	0.999728	0.997147	0.947072
	1.5	40.906350	5.385284	0.999514	0.996385	0.919537
	2.0	36.406280	15.71004	0.998506	0.990531	0.868415
	2.5	30.769480	57.22912	0.992886	0.963592	0.760181
	3.0	28.268800	82.54207	0.989745	0.953372	0.670742

Relative performance are measured between the 2 x 2 block based watermark fabrication (WLT_2x2) of section 3.2.1 and the 2 x 2 block based watermark fabrication using Legendre Transform (LT) followed by genetic algorithm (GA) based optimization (WLT_2X2_GAO) in terms of average PSNR at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. Table 8.6 demonstrated that the average PSNR of WLT_2x2 lies into the range [47.92 – 26.35 dB] for the extreme payload values 0.5 and 3 bpB respectively. Compared to WLT_2x2, the WLT_2x2_GAO is providing the identical value of average PSNR at 0.5 bpB; however, the average PSNR increases as the payload increase from 0.5 bpB and it becomes 28.26 dB at 3 bpB which is a bit shorter than the acceptable level (i.e., < 30 dB) at 3 bpB of payload. Hence, the quality of the watermarked images is severely distorted.

Table 8.6. Comparative analysis of average PSNR between WLT_2x2 and WLT_2x2_GAO with respect to increasing payload

WLT_2x2		**WLT_2X2_GAO**	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	47.92768	0.5	47.92768
1.0	43.95488	1.0	44.03281
1.5	39.38779	1.5	40.90635
2.0	35.95718	2.0	36.40628
2.5	29.59656	2.5	30.76948
3.0	26.35943	3.0	28.26880

Fig. 8.5 depicts the average PSNR analysis among the 2 x 2 block based watermarking in Legendre Transform (LT) domain followed by genetic algorithm based optimization (WLT_2X2_GAO), Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] respectively. Five color images [130, 131] such as “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” are considered as the cover while the varying sizes of the “Gold-Coin” are taken as the secret watermark as given in fig. 1.1. Both DPTHDI [88] and DGTDHS [129] offered PSNR values with respect to 0.25 and 1 bpB of payloads. The major limitation of DPTHDI [88] and DGTDHS [129] are their fixed as well as low payload. On the contrary, the PSNR values of the majority of the images are above 30 dB for the payload range (0.5 – 2.5 bpB) in WLT_2X2_GAO however, the severe quality distortion (i.e., PSNR < 30 dB) is observed at 3 bpB [148]. In spite of the perceptible quality degradation at 3 bpB, the WLT_2X2_GAO is applied to fabricate the secret information since the payload offered (0.5 – 3 bpB) is variable

in nature. In contrast to DPTHDI [88], proposed WLT_2X2_GAO ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2 and 2.5 bpB for “Baboon”, 0.5, 1 and 1.5 bpB of payloads for “Pepper”, 0.5, 1 and 1.5 bpB of payloads for “Airplane” and 0.5, 1, 1.5 and 2 bpB of payloads for “Sailboat” respectively. Compared to DGTDHS [129], the WLT_2X2_GAO ensured payload variation from 0.5 to 3 bpB with minor loss of PSNR at 1 bpB.

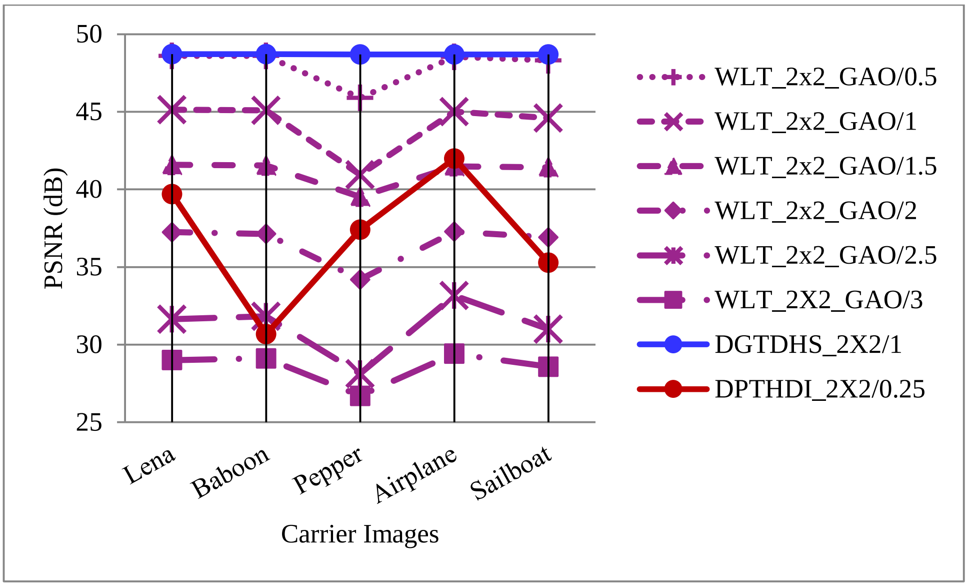


Fig. 8.5. Performance analysis of PSNR (dB) for variable payload based WLT_2x2_GAO and fixed payload based Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 8.6 illustrates that the improvement of average PSNR for WLT_2x2_GAO over the WLT_2x2 can be visually perceived as the payload increases from 0.5 bpB however, the quality improvement is perceived as the payload increases from 1 bpB since the average PSNR values of both schemes are very close to each other at 1 bpB. The average PSNR values for the payload range [0.5 – 2.5 bpB] is greater than or equal to 30 dB and hence, the obtained watermarked images preserved high transparency [148]. Compared to WDHT_2X2_GAO, the superiority of WLT_2X2_GAO is clearly expressed in terms of average PSNR for payload range 0.5 to 2.5 bpB. The average PSNR for DPTHDI [88] is 37.40 dB is obtained by taking the averages of PSNR values for “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images at 0.25 bpB of payload. In contrast to DPTHDI [88], the average PSNR of WLT_2x2_GAO ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB as obtained by averaging the PSNR values of “Lighthouse”, “Elaine”, “Lenna”, “Boat”, “F16” images at 1 bpB of payload. In contrast to DGTDHS [129], the WLT_2x2_GAO offered variable payload with minor loss of average PSNR at 1 bpB.

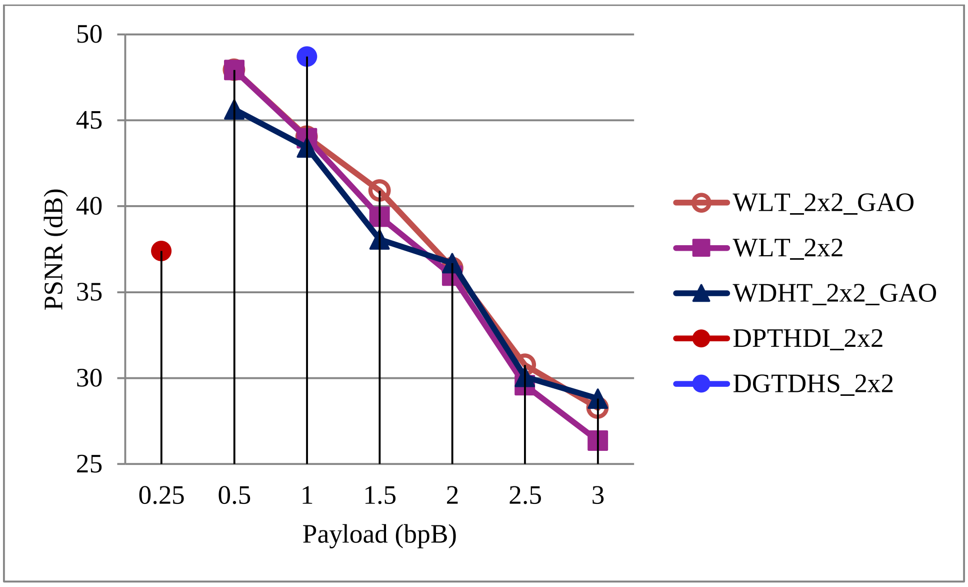


Fig. 8.6. Graphical representation of variation of average PSNR (dB) with respect to payload for WLT_2x2_GAO, WLT_2X2, WDHT_2X2_GAO and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

8.3.2.2. Optimization of 1 x 2 Block based Watermark Fabrication

The cover image is divided into blocks of 1 x 2 pixel components. Adjusts the pixel components of the blocks based on the pre-embedding pixel adjustment process and then the Legendre Transform (LT) is applied on all blocks as discussed in equation (3.17) of section 3.2.2.1. Subsequently, the message digest, size and the content of the watermark are embedded into the transformed components corresponding to the RGB color channels starting from the first bit position of the least significant part (i.e., LSB-1) toward higher order bit position. Optimized component is found against each embedded component based on the genetic algorithm (GA) based optimization to improve the watermarked image's quality. Inverse Legendre Transform (ILT) has been applied over the 1 x 2 sub-matrices of optimized components to get back the pixel components of identical block sizes in spatial domain. The process is repeated till complete secret information is concealed and the watermarked image is produced.

An example has been given in section 8.3.2.2.1 to demonstrate the different steps of the watermarking process (WLT_1x2_GAO) such as the decomposition, pixel adjustment, transformation, fabrication, optimization, and at last inverse transformation. Experimental results, comparative analysis and discussions have been elaborated in section 8.3.2.2.2.

8.3.2.2.1. Example

Consider the 1 x 2 sub-matrices of pixel components corresponding to red, green and blue channel to fabricate the secret bit-stream. To achieve a payload of 3 bpB, the pixel components are adjusted prior to embedding as discussed in equation (3.17) of section 3.2.2.1. Consider the 1 x 2 sub-matrices of pixel components as follows:

$$R_1 = [240\ 69] \quad G_1 = [92\ 202] \quad B_1 = [16\ 110]$$

Legendre Transform (LT) is applied on each 1 x 2 sub-matrix / pair of pixel components to obtain 1 x 2 sub-matrix or pair of transformed co-efficients. The 1 x 2 sub-matrices of transformed components namely T(R_1), T(G_1) and T(B_1) are obtained as given below:

$$T(R_1) = [240\ 378] \quad T(G_1) = [92\ 496] \quad T(B_1) = [16\ 236]$$

Secret bit-stream *"101000010110000111"* is to be concealed into both transformed components starting from LSB-1 toward the higher order bit position as per the following embedding rule: $\lambda = \lambda_1 = \lambda_2 = 3$. On embedding, the modified components become:

$$T'(R_1) = [250\ 368] \quad T'(G_1) = [84\ 502] \quad T'(B_1) = [16\ 238]$$

Genetic algorithm (GA) based optimization of section 8.2 has been introduced to obtain the optimized component corresponding to each embedded component. The optimized component has preserved the unaltered least significant bit along with three embedded bits i.e., LSB-0, LSB-1, LSB-2 and LSB-3 respectively. The sub-matrices of optimized components are as follows:

$$T''(R_1) = [234\ 384] \quad T''(G_1) = [84\ 502] \quad T''(B_1) = [16\ 238]$$

Applying inverse Legendre Transform (ILT) on each sub-matrix of embedded components yields the sub-matrix of pixel components as given below:

$$R'_1 = [234\ 75]\ G'_1 = [84\ 209]\ B'_1 = [16\ 111]$$

Since, the re-computed pixel components are lies in the range 0 to 255, all values are non-fractional as well as non-negative.

8.3.2.2.2. Results and Discussions

The WLT_1x2_GAO embeds watermark information into the color images of dimension 512 x 512 (fig. 1.1) for the payload variation from 0.5 to 3 bpB. Besides the payload value, the quality of the watermarked images is another important issue. The quality of the watermarked images has been measured in terms of peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) respectively as given table 8.7. The average PSNR values obtained in WLT_1x2_GAO lies between [54.22 dB (Barnfall), 0.5 bpB] and [32.20 dB (Desert), 3 bpB] respectively. Since, the permissible value of PSNR at varying payload (0.5 – 3 bpB) is greater than or equal to 30 dB, the quality of the watermarked images is considered to be well perceptible. The MSE values are falling into the range [0.24 dB (Barnfall), 0.5 bpB] to [39.11 (Desert), 3bpB]. Usual values of IF, SSIM and UIQ lie between 0 and 1 however, the ranges are redefined for minimum and maximum payload values (0.5 bpB, 3 bpB) as follows: [0.999992 (Airplane) - 0.994106 (Desert)], [0.999898 (San Diego) - 0.960874 (Pepper)] and [0.99933 (San Diego) - 0.596191 (Splash)] respectively. Simulation results ensured that the IF, SSIM and UIQ are maximum at 0.5 bpB and minimum at 3 bpB. The average results of twenty different carrier images at variable payload have also been computed to summarize the experimental results.

Table 8.7. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WLT_1x2_GAO scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.126733	0.251424	0.999984	0.999645	0.994757
	1.0	46.673650	1.398670	0.999911	0.999211	0.970898
	1.5	44.575181	2.267578	0.999857	0.997879	0.956595
	2.0	41.322320	4.795659	0.999698	0.997143	0.916347
	2.5	38.639731	8.894207	0.999448	0.991659	0.871900
	3.0	35.417407	18.678398	0.998826	0.988139	0.783579
Baboon	0.5	54.063045	0.255138	0.999986	0.999870	0.998966
	1.0	46.674475	1.398404	0.999925	0.999697	0.994214
	1.5	44.528496	2.292085	0.999878	0.999148	0.990203
	2.0	41.306691	4.812948	0.999745	0.998867	0.980989
	2.5	38.533425	9.114604	0.999520	0.996162	0.964304
	3.0	35.323741	19.085617	0.998993	0.994889	0.940561
Pepper	0.5	48.030595	1.023342	0.999900	0.990116	0.977738
	1.0	44.546889	2.282398	0.999791	0.989079	0.961660
	1.5	40.917815	5.263792	0.999506	0.980931	0.944240
	2.0	39.393673	7.476751	0.999327	0.980661	0.915734
	2.5	35.079899	20.187867	0.998113	0.963629	0.869849
	3.0	33.445465	29.412626	0.997351	0.960874	0.791192

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	53.973755	0.260438	0.999992	0.999580	0.983352
	1.0	46.642680	1.408679	0.999959	0.999063	0.924539
	1.5	44.463042	2.326891	0.999933	0.997443	0.894693
	2.0	41.288420	4.833239	0.999861	0.996565	0.824031
	2.5	38.862443	8.449597	0.999758	0.990783	0.766040
	3.0	35.533276	18.186650	0.999479	0.986675	0.657531
Sailboat	0.5	53.335572	0.301663	0.999984	0.999693	0.995550
	1.0	46.465891	1.467206	0.999925	0.999326	0.976773
	1.5	44.200825	2.471711	0.999875	0.998201	0.965880
	2.0	41.199428	4.933300	0.999751	0.997611	0.935297
	2.5	38.389366	9.422013	0.999526	0.992834	0.902826
	3.0	35.280301	19.277477	0.999028	0.990158	0.841475
Earth	0.5	54.137597	0.250796	0.999985	0.999730	0.997664
	1.0	46.681633	1.396101	0.999917	0.999408	0.986370
	1.5	44.460373	2.328322	0.999863	0.998345	0.978982
	2.0	41.276140	4.846925	0.999713	0.997804	0.956289
	2.5	38.585919	9.005096	0.999473	0.993558	0.925302
	3.0	35.313269	19.131692	0.998871	0.990526	0.860909
San Diego	0.5	54.122957	0.251642	0.999990	0.999898	0.999330
	1.0	46.684078	1.395315	0.999948	0.999778	0.996197
	1.5	44.467439	2.324536	0.999913	0.999370	0.994206
	2.0	41.291570	4.829734	0.999819	0.999176	0.988250
	2.5	38.703824	8.763910	0.999672	0.997618	0.980434
	3.0	35.425428	18.643931	0.999303	0.996531	0.961238
Splash	0.5	49.266754	0.769847	0.999917	0.989394	0.963196
	1.0	45.155668	1.983871	0.999809	0.988233	0.895883
	1.5	42.091769	4.017014	0.999598	0.983073	0.863980
	2.0	40.122360	6.321852	0.999403	0.982155	0.784413
	2.5	36.531343	14.452602	0.998555	0.971562	0.725810
	3.0	34.315557	24.072690	0.997731	0.966489	0.596191
Oakland	0.5	51.286666	0.483519	0.999968	0.999417	0.997710
	1.0	45.888116	1.675979	0.999903	0.999101	0.992149
	1.5	42.976143	3.276921	0.999800	0.997724	0.987329
	2.0	40.575054	5.696065	0.999674	0.997390	0.976200
	2.5	37.081105	12.734146	0.999219	0.992566	0.958264
	3.0	34.586807	22.615158	0.998701	0.990317	0.920778

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	54.153834	0.249860	0.999991	0.999519	0.993394
	1.0	46.686176	1.394641	0.999950	0.998943	0.963030
	1.5	44.473941	2.321058	0.999916	0.996990	0.944513
	2.0	41.307730	4.811796	0.999827	0.996062	0.890218
	2.5	38.790580	8.590577	0.999688	0.988926	0.830937
	3.0	35.592173	17.941675	0.999350	0.984511	0.722493
Anhinga	0.5	50.833323	0.536720	0.999958	0.999809	0.882627
	1.0	45.879051	1.679481	0.999870	0.999594	0.869365
	1.5	42.575703	3.593437	0.999724	0.998595	0.861629
	2.0	40.115380	6.332021	0.999512	0.997521	0.824727
	2.5	36.747842	13.749790	0.998942	0.994186	0.806807
	3.0	34.606236	22.514212	0.998269	0.991934	0.769197
Athens	0.5	51.884716	0.421316	0.999966	0.999864	0.971366
	1.0	46.182203	1.566246	0.999874	0.999663	0.956086
	1.5	43.560898	2.864115	0.999771	0.999016	0.959599
	2.0	40.342011	6.010065	0.999519	0.997925	0.887852
	2.5	37.832045	10.712130	0.999147	0.994043	0.852632
	3.0	35.246474	19.428213	0.998451	0.993077	0.798734
Bardowl	0.5	50.122462	0.632170	0.999934	0.999465	0.997727
	1.0	45.519226	1.824558	0.999815	0.999160	0.994449
	1.5	42.602539	3.571300	0.999636	0.997855	0.986544
	2.0	40.409262	5.917715	0.999403	0.997617	0.979714
	2.5	36.442775	14.750367	0.998481	0.987088	0.945317
	3.0	34.511898	23.008617	0.997660	0.986141	0.925792
Barnfall	0.5	54.225905	0.245747	0.999961	0.999762	0.998852
	1.0	46.822993	1.351390	0.999782	0.999447	0.992625
	1.5	44.798653	2.153848	0.999657	0.998621	0.989787
	2.0	41.157341	4.981339	0.999204	0.998184	0.974648
	2.5	38.848901	8.475986	0.998624	0.994442	0.961790
	3.0	35.652584	17.693832	0.997188	0.991926	0.923467
Butrfly	0.5	52.613555	0.356225	0.999974	0.999852	0.995605
	1.0	46.341627	1.509793	0.999892	0.999602	0.988149
	1.5	43.835745	2.688474	0.999810	0.998875	0.983392
	2.0	40.610071	5.650321	0.999602	0.998262	0.958496
	2.5	38.122683	10.018714	0.999301	0.995474	0.935345
	3.0	35.414193	18.692223	0.998684	0.994364	0.892276

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	52.199342	0.391873	0.999944	0.999695	0.749617
	1.0	46.239258	1.545804	0.999782	0.999105	0.740411
	1.5	43.851287	2.678869	0.999623	0.998451	0.726393
	2.0	41.315953	4.802694	0.999326	0.997159	0.715222
	2.5	39.001823	8.182725	0.998853	0.993755	0.701357
	3.0	36.150269	15.778054	0.997785	0.990482	0.675177
Bodie	0.5	50.407332	0.592034	0.999893	0.999533	0.981783
	1.0	45.415760	1.868548	0.999675	0.999275	0.970274
	1.5	42.650779	3.531851	0.999378	0.997774	0.964630
	2.0	40.484287	5.816364	0.998995	0.997551	0.953343
	2.5	36.110543	15.923039	0.997143	0.984323	0.923342
	3.0	34.168744	24.900377	0.995664	0.983313	0.894927
Bluheron	0.5	54.188895	0.247851	0.999969	0.999756	0.996671
	1.0	47.132107	1.258547	0.999846	0.999437	0.987203
	1.5	44.567990	2.271335	0.999722	0.998489	0.978050
	2.0	40.882778	5.306430	0.999350	0.997733	0.954753
	2.5	38.731668	8.707901	0.998933	0.993993	0.934304
	3.0	35.582041	17.983582	0.997798	0.990914	0.868150
Colomtn	0.5	52.498062	0.365825	0.999971	0.999762	0.985223
	1.0	46.275919	1.532810	0.999880	0.999486	0.976688
	1.5	43.537843	2.879360	0.999774	0.998705	0.973027
	2.0	40.661378	5.583962	0.999563	0.998051	0.954659
	2.5	37.500077	11.563055	0.999096	0.994700	0.935933
	3.0	34.996698	20.578347	0.998391	0.992226	0.897842
Desert	0.5	45.918758	1.664196	0.999731	0.997251	0.994053
	1.0	43.383755	2.983353	0.999535	0.996765	0.990937
	1.5	39.354855	7.543880	0.998815	0.992450	0.981427
	2.0	38.143969	9.969730	0.998490	0.992516	0.974273
	2.5	33.089607	31.924168	0.995073	0.974707	0.938649
	3.0	32.206891	39.119314	0.994106	0.973690	0.924202
Average case	0.5	52.069490	0.477581	0.99995	0.998581	0.972759
	1.0	46.064560	1.646090	0.999849	0.998169	0.956395
	1.5	43.424570	3.133319	0.999702	0.996397	0.946255
	2.0	40.660290	5.686446	0.999489	0.995798	0.917273
	2.5	37.581280	12.181120	0.998828	0.989300	0.886557
	3.0	34.938470	21.337130	0.998081	0.986859	0.832286

The average PSNR of 1 x 2 block based watermark fabrication (WLT_1x2) and the window of identical size based watermark fabrication using Legendre Transform (LT) followed by genetic algorithm based optimization (WLT_1X2_GAO) is measured as well as compared with each other with respect to the payload variation from 0.5 to 3 bpB. Table 8.8 demonstrated the permissible ranges of average PSNR values for WLT_1x2 and WLT_1x2_GAO schemes are summarized for payload values 0.5 and 3 bpB as follows: [52.06 - 31.68 dB] and [52.06 – 34.93 dB]. The enhancement of average PSNR in WLT_1x2_GAO is better compared to WLT_1x2 as the payload value exceeds 1 bpB.

Table 8.8. Comparative analysis of obtained average PSNR values between WLT_1x2 and WLT_1x2_GAO with respect to increasing payload

WLT_1x2		WLT_1X2_GAO	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	52.06949	0.5	52.06949
1.0	46.06456	1.0	46.06456
1.5	43.04556	1.5	43.42457
2.0	38.48890	2.0	40.66029
2.5	35.88123	2.5	37.58128
3.0	31.68995	3.0	34.93847

The comparison of PSNR values has been made for five color images such as “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” respectively. Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] are two novel embedding schemes proposed by Varsaki et al. in the year 2010 and 2015 respectively. Both schemes provided at least 30 dB of PSNR values for the given image sets at 0.25 and 1 bpB respectively. In spite of well perceptible quality of the watermarked images, obtained payload values are considered as fixed and significantly low. The above problem can be avoided by analyzing the performance of WLT_1X2_GAO for the payload range [0.5 – 3 bpB]. It is apparent from fig. 8.7 that the PSNR values for all five tested images are higher than 30 dB at maximum payload of 3 bpB and thus, well perceptible watermarked images with high transparency is obtained [148]. In contrast to DPTHDI [88], the WLT_1X2_GAO scheme ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB for “Baboon”, 0.5, 1, 1.5 and 2 bpB of payloads for “Pepper”, 0.5, 1 and 1.5 bpB of payloads for “Airplane” and 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for “Sailboat” respectively. In comparison with DGTDHS

[129], the WLT_1X2_GAO ensured variable payload that offers a spread from 0.5 to 3 bpB however, the PSNR values at 1 bpB is slightly reduced.

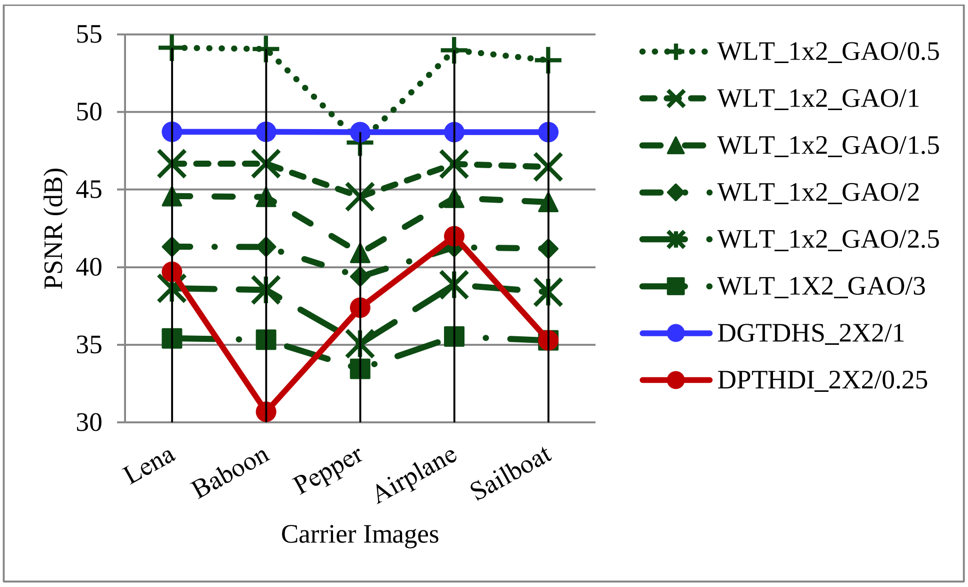


Fig. 8.7. Performance analysis of PSNR (dB) for variable payload based WLT_1x2_GAO and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

In fig 8.8, the average PSNR variation is observed for WLT_1x2_GAO, WLT_1x2, WLT_2x2_GAO, WDHT_1x2_GAO, DPTHDI [88] and DGTDHS [129] respectively with respect to payload values. In contrast to WLT_1x2 method, the average PSNR value in WLT_1X2_GAO scheme is statistically improved as the payload increases from 1 bpB. In contrast to WDHT_1X2_GAO, WLT_2X2_GAO and WLT_1x2 methods, the WLT_1X2_GAO method ensures less degradation in fidelity with respect to variable payload. The average PSNR for DPTHDI [88] scheme is 37.40 dB as computed from the average of PSNR values of “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images at 0.25 bpB of payload. In contrast to DPTHDI [88] scheme, the average PSNR of WLT_1x2_GAO scheme ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads. The average PSNR for DGTDHS [129] scheme is 48.70 dB as computed from the average of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat”, “F16” images at 1 bpB of payload. Compared to DGTDHS [129] scheme, the WLT_1x2_GAO scheme is considering minor loss in the average PSNR at 1 bpB. However, the overall quality distortion of WLT_1x2_GAO for the payload variation from 0.5 to 3 bpB has been minimized over the other mentioned schemes.

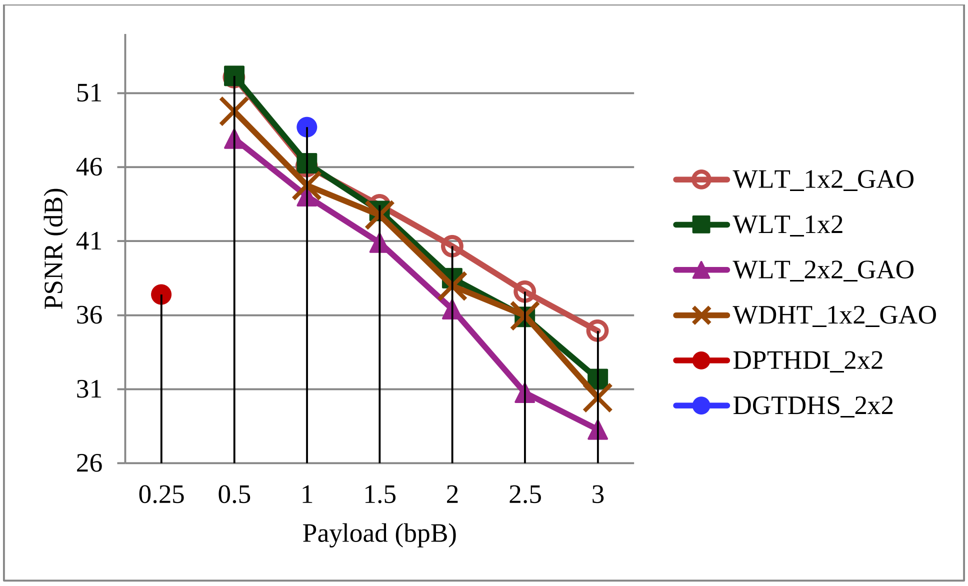


Fig. 8.8. Graphical representation of variation of average PSNR (dB) with respect to payload for WLT_1x2_GAO, WLT_1X2, WLT_2X2_GAO, WDHT_1x2_GAO and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

8.3.3. Optimization of Binomial Transform (BT) based watermarking

Two variants of Binomial Transform (BT) based watermarking has been proposed in chapter 4 as discussed. These are 2 x 2 block based watermark fabrication (WBT_2x2) and 1 x 2 block based watermark fabrication (WBT_1x2).

Section 8.3.3.1 deals with the 2 x 2 block based watermark fabrication followed by genetic algorithm (GA) based optimization of section 8.2 (WBT_2X2_GAO) that of 1 x 2 block based watermark fabrication using Binomial Transform (BT) followed by genetic algorithm (GA) based optimization (WBT_1X2_GAO) has been discussed in section 8.3.3.2.

8.3.3.1. Optimization of 2 x 2 Block based Watermark Fabrication

In WBT_2x2_GAO, divide the cover image into 2 x 2 non-overlapping blocks. The pixel components of each block are adjusted based on the adjustment process as discussed in equation (4.6) of section 4.2.1.1 and then apply Binomial Transform (BT) to obtain transformed components. Embedding secret bits into the transformed components (message digest, size and content of the watermark) is started from the least significant bit position (i.e., LSB-0) toward higher order bit position. Genetic algorithm (GA) optimizes the embedded component for a wide range of values and then generate optimized component that will preserve the fabricated bits. Inverse Binomial Transform (IBT) is applied over 2 x 2 sub-matrices of optimized components to obtain the pixel components. Repeat above steps till entire secret information is concealed and the watermarked image is obtained.

An example has been given in section 8.3.3.1.1. Results and discussions have been elaborated in section 8.3.3.1.2.

8.3.3.1.1. Example

Let three consecutive 2 x 2 sub-matrices of pixel components of the cover image corresponding to RGB color channels which in turn are adjusted for payload value of 3 bpB based on the pixel adjustment process described in equation (4.6) of section 4.2.1.1. The sub-matrices namely R1, G1 and B1 are obtained as follows:

$$R_1 = \begin{bmatrix} 224 & 72 \\ 32 & 155 \end{bmatrix} G_1 = \begin{bmatrix} 62 & 215 \\ 56 & 32 \end{bmatrix} B_1 = \begin{bmatrix} 111 & 172 \\ 224 & 32 \end{bmatrix}$$

Binomial Transform (BT) has been introduced to convert each 2 x 2 sub-matrix of pixel components into transform domain. The 2 x 2 sub-matrices of transformed components such as T(R1), T(G1) and T(B1) are obtained as follows:

$$T(R_1) = \begin{bmatrix} 224 & 152 \\ 112 & -51 \end{bmatrix} T(G_1) = \begin{bmatrix} 62 & -153 \\ -312 & -447 \end{bmatrix} T(B_1) = \begin{bmatrix} 111 & -61 \\ -9 & 235 \end{bmatrix}$$

Secret bit-stream *“01011101011001001100111101101100000 0”* is to be allocated into 3: 2: 3: 4 (λ1 = 3, λ2 = 2, λ3 = 3, λ4 = 4) ratio for embedding across the first, second, third and fourth components corresponding to each sub-matrix as given in equation (4.7) of section 4.2.1.1. Hence, the 2 x 2 sub-matrices of embedded components are obtained as follows:

$$T'(R_1) = \begin{bmatrix} 226 & 155 \\ 117 & -54 \end{bmatrix} T'(G_1) = \begin{bmatrix} 58 & -154 \\ -313 & -447 \end{bmatrix} T'(B_1) = \begin{bmatrix} 110 & -62 \\ -9 & 224 \end{bmatrix}$$

Genetic algorithm (GA) based optimization of section 8.2 has been introduced to obtain the optimized component corresponding to each embedded component. The optimized component has preserved the least three embedded bits i.e., LSB-0, LSB-1 and LSB-2 respectively. The 2 x 2 sub-matrices of optimized components are as follows:

$$T''(R_1) = \begin{bmatrix} 226 & 155 \\ 117 & -54 \end{bmatrix} T''(G_1) = \begin{bmatrix} 58 & -154 \\ -313 & -447 \end{bmatrix} T'(B_1) = \begin{bmatrix} 110 & -62 \\ -9 & 240 \end{bmatrix}$$

Applying inverse Binomial Transform (IBT) on each 2 x 2 sub-matrices of optimized components yields the 2 x 2 sub-matrices of pixel components in spatial domain as follows:

$$R'_1 = \begin{bmatrix} 226 & 71 \\ 33 & 166 \end{bmatrix} G'_1 = \begin{bmatrix} 58 & 212 \\ 53 & 28 \end{bmatrix} B'_1 = \begin{bmatrix} 110 & 172 \\ 225 & 29 \end{bmatrix}$$

It is observed that the modified pixel components are non-fractional, non-negative and less than or equal to 255.

8.3.3.1.2. Results and Discussions

Table 8.9 summarized the optimized results of WBT_2x2_GAO for twenty 512 x 512 color images (fig. 1.1). Average values of standard quality metrics are also computed for the payload range of 0.5 to 3 bpB. The maximum value of peak signal to noise ratio (PSNR) is 50.22 dB at 0.5 bpB of payload for the "Foster City" and that of the minimum value is 25.72 dB at 3 bpB of payload for the "Desert" which in turn is falling below the acceptable level (i.e., < 30 dB). However, the average PSNR of 30.56 dB (i.e., ≥ 30 dB) yields the perceptible quality in the watermarked images [148]. The minimum value of mean squared error (MSE) obtained is 0.61 for "FosterCity" at payload of 0.5 bpB and that of the maximum value is 173.86 for "Desert" at 3 bpB. The usual range of image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) belongs to [0 – 1] but, the WBT_2x2_GAO offered the minimum values as 0.973771 (Desert), 0.838882 (Pepper) and 0.551286 (Bobcat) respectively at 0.5 bpB and that of the maximum values are obtained as 0.999981 (Airplane), 0.957760 (San Diego) and 0.998400 (San Diego) respectively at 3 bpB.

Table 8.9. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WBT_2x2_GAO scheme

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Lena	0.5	50.170722	0.625184	0.999960	0.997008	0.987649
	1.0	45.083912	2.016921	0.999872	0.993291	0.959293
	1.5	41.591125	4.507832	0.999715	0.981147	0.922039
	2.0	38.369684	9.464810	0.999416	0.970462	0.862324
	2.5	35.521474	18.236139	0.998871	0.931202	0.790104
	3.0	30.812121	53.934688	0.996776	0.883572	0.650087
Baboon	0.5	50.123143	0.632071	0.999966	0.998939	0.997615
	1.0	45.115779	2.002176	0.999894	0.997602	0.991071
	1.5	41.549359	4.551394	0.999759	0.993027	0.982135
	2.0	38.419845	9.356119	0.999507	0.988614	0.961640
	2.5	35.615258	17.846560	0.999059	0.973736	0.941355
	3.0	31.175724	49.602998	0.997409	0.953359	0.889682
Pepper	0.5	46.481090	1.462080	0.999862	0.987135	0.973145
	1.0	40.940125	5.236821	0.999501	0.975797	0.945552
	1.5	39.858673	6.717582	0.999397	0.966531	0.920454
	2.0	34.799753	21.533022	0.997987	0.942456	0.857746
	2.5	33.940204	26.245806	0.997625	0.909712	0.799928
	3.0	28.120981	100.226422	0.990556	0.838882	0.649976

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	50.072236	0.639523	0.999981	0.996426	0.963779
	1.0	45.035336	2.039608	0.999941	0.992074	0.900749
	1.5	41.641056	4.456302	0.999872	0.978052	0.835565
	2.0	38.482172	9.222807	0.999736	0.965752	0.751438
	2.5	35.447509	18.549381	0.999469	0.916269	0.656817
	3.0	31.987770	41.143694	0.998823	0.874394	0.558145
Sailboat	0.5	49.734037	0.691315	0.999965	0.997561	0.989584
	1.0	44.621666	2.243436	0.999887	0.994419	0.967210
	1.5	41.428127	4.680234	0.999763	0.984657	0.940180
	2.0	37.971298	10.374102	0.999478	0.974979	0.895012
	2.5	35.473952	18.436782	0.999070	0.943689	0.845783
	3.0	30.583968	56.843847	0.997154	0.895937	0.723409
Earth	0.5	50.211011	0.619411	0.999963	0.997722	0.994337
	1.0	45.172424	1.976231	0.999882	0.994878	0.980390
	1.5	41.620345	4.477605	0.999734	0.985719	0.959682
	2.0	38.623626	8.927251	0.999469	0.977375	0.920797
	2.5	35.746690	17.314554	0.998974	0.947065	0.869520
	3.0	31.576026	45.235319	0.997183	0.913041	0.752314
San Diego	0.5	50.199054	0.621119	0.999976	0.999143	0.998400
	1.0	45.209008	1.959654	0.999926	0.998088	0.994629
	1.5	41.593146	4.505736	0.999832	0.994483	0.988903
	2.0	38.820831	8.530946	0.999682	0.991701	0.979598
	2.5	35.892176	16.744134	0.999376	0.979506	0.964169
	3.0	32.596018	35.766668	0.998669	0.967229	0.932442
Splash	0.5	47.457569	1.167678	0.999882	0.985263	0.941465
	1.0	45.209008	1.959654	0.999926	0.998088	0.994629
	1.5	41.593146	4.505736	0.999832	0.994483	0.988903
	2.0	38.820831	8.530946	0.999682	0.991701	0.979598
	2.5	35.892176	16.744134	0.999376	0.979506	0.964169
	3.0	32.596018	35.766668	0.998669	0.967229	0.932442
Oakland	0.5	48.711909	0.874760	0.999946	0.997839	0.996125
	1.0	43.422354	2.956956	0.999816	0.995048	0.988226
	1.5	40.966600	5.204995	0.999703	0.987959	0.978031
	2.0	37.132508	12.584313	0.999225	0.980308	0.956873
	2.5	35.161785	19.810789	0.998873	0.956604	0.927573
	3.0	30.645339	56.046235	0.996535	0.926475	0.860268

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	50.223314	0.617659	0.999978	0.995890	0.984041
	1.0	45.116518	2.001836	0.999929	0.990680	0.990680
	1.5	41.629820	4.467847	0.999840	0.974265	0.899483
	2.0	38.567710	9.042933	0.999679	0.959583	0.821400
	2.5	35.670642	17.620412	0.999377	0.905471	0.732473
	3.0	32.003953	40.990670	0.998562	0.849975	0.603185
Anhinga	0.5	48.518520	0.914594	0.999929	0.998785	0.880234
	1.0	42.951837	3.295312	0.999747	0.996217	0.864089
	1.5	40.679109	5.561211	0.999573	0.987486	0.841032
	2.0	36.839708	13.461996	0.998968	0.982169	0.819161
	2.5	33.907746	26.442696	0.997976	0.959799	0.776353
	3.0	30.212855	61.914848	0.993484	0.933199	0.718069
Athens	0.5	48.616577	0.894175	0.999928	0.998795	0.968605
	1.0	43.823776	2.695893	0.999784	0.995426	0.946515
	1.5	41.839157	4.257597	0.999659	0.986936	0.919249
	2.0	39.317588	7.608893	0.999393	0.980645	0.890835
	2.5	35.248960	19.417097	0.998446	0.956621	0.823013
	3.0	33.118049	31.715780	0.997461	0.935279	0.760425
Bardowl	0.5	47.654216	1.115985	0.999886	0.998745	0.996966
	1.0	42.347795	3.787048	0.999613	0.995845	0.986353
	1.5	41.070958	5.081413	0.999487	0.992828	0.982074
	2.0	36.137677	15.823867	0.998369	0.979343	0.942945
	2.5	34.593107	22.582377	0.997715	0.971097	0.926569
	3.0	28.713258	87.448922	0.990947	0.916412	0.827015
Barnfall	0.5	49.699070	0.696904	0.999893	0.998321	0.997215
	1.0	44.604535	2.252302	0.999643	0.995700	0.990077
	1.5	42.040300	4.064904	0.999350	0.989378	0.981332
	2.0	38.545233	9.089855	0.998486	0.982272	0.960951
	2.5	35.180677	19.724798	0.996790	0.953053	0.921111
	3.0	31.430254	46.779430	0.991911	0.914606	0.836815
Butrfly	0.5	49.015265	0.815743	0.999942	0.998879	0.994139
	1.0	43.884581	2.658411	0.999810	0.996064	0.980037
	1.5	41.799181	4.296969	0.999695	0.989412	0.967547
	2.0	38.542639	9.095286	0.999353	0.982823	0.938075
	2.5	35.458750	18.501431	0.998687	0.962524	0.898462
	3.0	32.370627	37.671905	0.997298	0.942210	0.821641

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	49.066413	0.806193	0.999886	0.998370	0.748072
	1.0	44.132313	2.511013	0.999647	0.995202	0.737322
	1.5	41.867859	4.229551	0.999405	0.983840	0.722292
	2.0	38.789515	8.592683	0.998787	0.976651	0.700293
	2.5	35.265755	19.342154	0.997285	0.946083	0.671302
	3.0	26.968917	130.673837	0.981511	0.841805	0.551286
Bodie	0.5	48.151928	0.995147	0.999826	0.998603	0.976576
	1.0	42.784892	3.424452	0.999397	0.996118	0.965995
	1.5	40.697919	5.537176	0.999041	0.991820	0.957279
	2.0	35.783013	17.170345	0.996922	0.976049	0.922955
	2.5	34.235213	24.522177	0.995725	0.963288	0.899700
	3.0	27.711275	110.141960	0.979947	0.875460	0.770080
Bluheron	0.5	49.847640	0.673467	0.999917	0.998128	0.993673
	1.0	44.643374	2.232250	0.999727	0.995287	0.980486
	1.5	42.258275	3.865919	0.999527	0.989276	0.966494
	2.0	39.125987	7.952096	0.999024	0.981801	0.935500
	2.5	35.006495	20.531978	0.997488	0.950407	0.872885
	3.0	32.274658	38.513637	0.995279	0.920110	0.785690
Colomtn	0.5	49.506922	0.728430	0.999943	0.998396	0.983158
	1.0	44.0671121	2.548995	0.999800	0.996136	0.975648
	1.5	41.108296	5.037913	0.999606	0.988939	0.961527
	2.0	37.483829	11.606396	0.999093	0.982674	0.942150
	2.5	34.648314	22.297129	0.998258	0.957951	0.903355
	3.0	30.623054	56.334557	0.995607	0.924233	0.834675
Desert	0.5	44.838152	2.134347	0.999661	0.996140	0.992907
	1.0	39.243565	7.739695	0.998785	0.989701	0.979325
	1.5	38.621977	8.930641	0.998635	0.987934	0.978039
	2.0	32.806069	34.077944	0.994751	0.965713	0.936044
	2.5	32.239637	38.825459	0.994180	0.955545	0.922612
	3.0	25.728587	173.868783	0.973771	0.874366	0.805276
Average case	0.5	48.914940	0.886289	0.999915	0.996804	0.967884
	1.0	43.870500	2.876933	0.999726	0.994083	0.955914
	1.5	41.272720	4.946928	0.999571	0.986409	0.934612
	2.0	37.668980	12.102330	0.998850	0.976654	0.898767
	2.5	35.007330	20.986800	0.998131	0.950956	0.855363
	3.0	30.562470	64.531040	0.993378	0.907389	0.763146

Comparison of average PSNR is made between the WBT_2x2 and WBT_2x2_GAO for the variable payload from 0.5 to 3 bpB. Table 8.10 demonstrated that the maximum average PSNR for both schemes are 48.91 dB at 0.5 bpB. For the payload range (1 – 3 bpB), the following enhancement in terms of average PSNR are observed: – 0.39, 1.34, 1.13, 2.08 and 1.86 dB respectively. Since, the average PSNR of WBT_2x2_GAO is more than 30 dB at 3 bpB of payload, the quality of the watermarked images is well perceptible [148].

Table 8.10. Comparative analysis of obtained average PSNR values between WBT_2x2 and WBT_2x2_GAO with respect to increasing payload

WBT_2x2		WBT_2X2_GAO	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	48.91494	0.5	48.91494
1.0	44.16739	1.0	43.87050
1.5	39.93230	1.5	41.27272
2.0	36.53638	2.0	37.66898
2.5	32.92258	2.5	35.00733
3.0	28.70732	3.0	30.56247

Variable payload based watermarking in Binomial Transform (BT) domain as applicable for 2 x 2 blocks (WBT_2X2_GAO) is compared against Varsaki et al.’s fixed payload based schemes such as, Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] respectively. The comparison is made in terms of PSNR (dB) with respect to the fixed as well as variable payload values (bpB). The fixed payload values of 0.25 bpB and 1 bpB is considered as the significant deficiency for DPTHDI [88] as well as DGTDHS. In order to achieve the variation in payload from 0.5 to 3 bpB, the WBT_2X2_GAO has been used as shown in fig. 8.9. All tested images are of dimensions 512 x 512 and they are labeled as: “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” respectively. The majority of the images are offering more than 30 dB of PSNR against the payload of 3 bpB and hence, the proposed techniques offered good quality watermarked images [148]. In contrast to DPTHDI [88], the WBT_2X2_GAO ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB for “Baboon”, 0.5, 1 and 1.5 bpB of payloads for “Pepper”, 0.5, 1 and 1.5 bpB of payloads for “Airplane” and 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for “Sailboat” respectively. Compared to DGTDHS [129], the WBT_2X2_GAO focused on variable payload values by ignoring the lack of PSNR at 1 bpB.

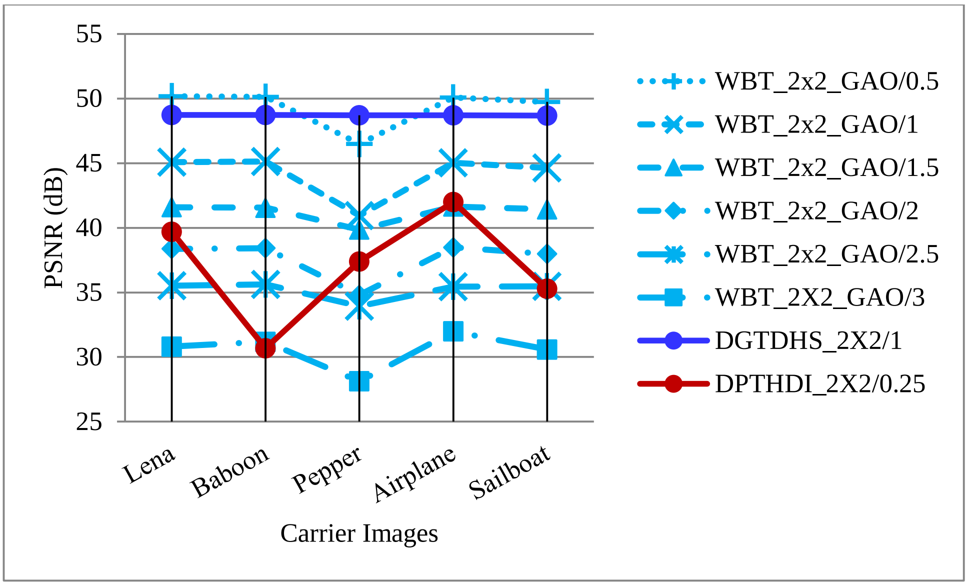


Fig. 8.9. Performance analysis of PSNR (dB) for variable payload based WBT_2x2_GAO and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 8.15 illustrates the analysis of average peak signal to noise ratio (PSNR) values for variable payload (WBT_2x2_GAO, WBT_2x2, WLT_2x2_GAO and WDHT_2x2_GAO) and fixed payload (DPTHDI [88] and DGTDHS [129]) based schemes, respectively. In contrast to WBT_2x2, the improvement of PSNR in WBT_2X2_GAO method can be visually perceived as the payload increases from 0.5 bpB. In comparison with WLT_2X2_GAO and WDHT_2X2_GAO, the superiority of the WBT_2X2_GAO scheme has also been proved as soon as the payload exceeds 1.5 bpB. The average PSNR for DPTHDI [88] is 37.40 dB which is obtained by averaging the PSNR values for "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" at 0.25 bpB of payload. In contrast to DPTHDI [88], the average PSNR of WBT_2x2_GAO ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB which is obtained by taking the averages of PSNR values for "Lighthouse", "Elaine", "Lenna", "Boat", "F16" at 1 bpB of payload. Compared to DGTDHS [129], the WBT_2x2_GAO is lacking in terms of average PSNR at 1 bpB of payload however, the incorporation of variable payload feature (i.e., 0.5 to 3 bpB) made the difference. The majority of the watermarked images obtained for the payload variation of 0.5 to 3 bpB in WBT_2x2_GAO are retaining high level of visual imperceptibility as compared to DPTHDI [88] and DGTDHS [129] since those images are preserved the PSNR values of greater than or equal to 30 dB [148].

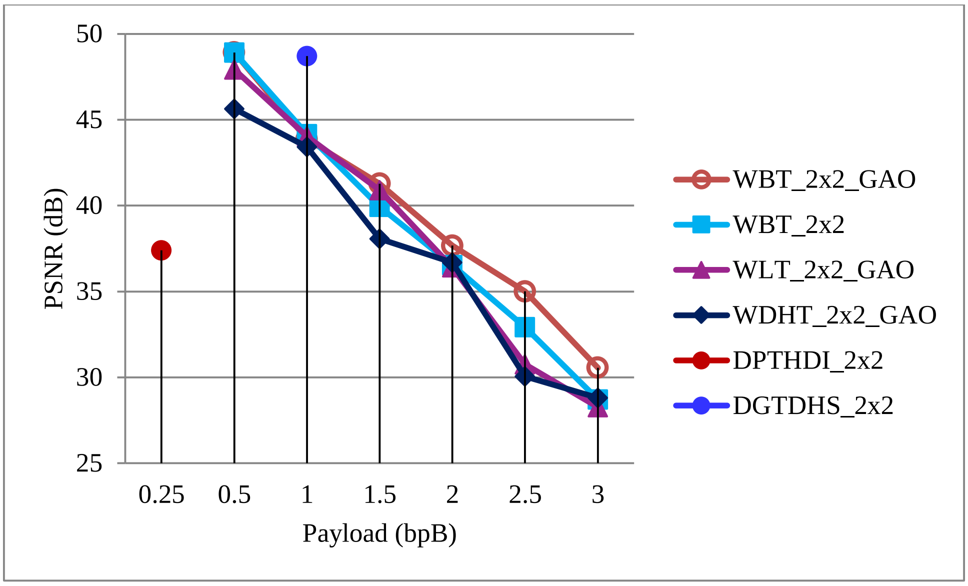


Fig. 8.10. Graphical representation of variation of average PSNR (dB) with respect to payload for WBT_2x2_GAO, WBT_2x2, WLT_2x2_GAO, WDHT_2x2_GAO and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

8.3.3.2. Optimization of 1 x 2 Block based Watermark Fabrication

The host image of size (U x V) is partitioned into 1 x 2 sub-blocks of pixel components which in turn are adjusted according to the principle of adjustment process mentioned in equation (4.16) of section 4.2.2.1. The non-overlapped blocks (1 x 2) are transformed to transform domain by the Binomial Transform (BT). Least significant bits belonging to the range (LSB 0 – LSB 2) is replaced by the secret bits of the message digest, size and content of the watermark. Optimized component corresponding to each embedded component is obtained based on the GA optimization of section 8.2. The optimized component must preserve the fabricated bits of the embedded component. On application of Inverse Binomial Transform (IBT) over 1 x 2 sub-matrices of optimized components, the pixel components are re-computed in spatial domain. Successive repetition of the process conform the fabrication of the entire secret bit-stream and hence, ensured the construction of the watermarked image.

The WBT_1x2_GAO is thoroughly explained with the example given in section 8.3.3.2.1. Computational results, image quality analysis and discussions are given in section 8.3.3.2.2.

8.3.3.2.1. Example

The payload value of 3 bpB yields the modification into the 1 x 2 blocks of red, green and blue pixel components based on the pixel adjustment method which is applied prior to embedding as discussed in equation (4.16) of section 4.2.2.1. The 1 x 2 sub-matrices are as given below:

$$R_1 = [240\ 78] \quad G_1 = [119\ 217] \quad B_1 = [16\ 130]$$

Binomial Transform (BT) converts each 1 x 2 sub-matrix / pair of pixel components into transformed domain. The transformed 1 x 2 sub-matrices of transformed components such as T(R_1), T(G_1) and T(B_1) are obtained as follows:

$$T(R_1) = [240\ 162] \quad T(G_1) = [119\ -98] \quad T(B_1) = [16\ -114]$$

Suppose, the watermark bit stream *"111101110010110000"* is to be fabricated based on the embedding strategy of equation (4.17) of section 4.2.2.1. In this example, three bits are fabricated ($\lambda = \lambda_1 = \lambda_2 = 3$) on each transformed component starting from LSB-0 toward the higher order bit position. Hence, the 1 x 2 sub-matrices of embedded components are as follows:

$$T'(R_1) = [248\ 165] \quad T'(G_1) = [115\ -98] \quad T'(B_1) = [19\ -112]$$

Genetic Algorithm (GA) based optimization ensured the enhancement of quality without losing the embedded bits. Genetic Algorithm (GA) takes each embedded component (t′) as input and returns the optimized component (t″) closest to the pre-embedded component (t) as output. The optimized component (t″) has preserved the embedded bits (LSB-0, LSB-1 and LSB-2) as unaffected. The optimized 1 x 2 sub-matrices are as given below:

$$T''(R_1) = [239\ 165] \quad T'(G_1) = [115\ -98] \quad T'(B_1) = [19\ -112]$$

Applying inverse Binomial Transform (IBT) on 1 x 2 sub-matrices of embedded components conform the generation of 1 x 2 sub-matrices of pixel components which lie in between 0 and 255. The 2 x 2 obtained sub-matrices are as follows:

$$R'_1 = [239\ 74] \quad G'_1 = [115\ 213] \quad B'_1 = [19\ 131]$$

All re-computed pixel components are non-fractional, non-negative and less than 255. Hence, no overflow or underflow situations are arises.

8.3.3.2.2. Results and Discussions

This section represents the results and discussions of the WBT_1X2_GAO scheme which is nothing but the WBT_1x2 method of section 4.2.2 followed by Genetic Algorithm (GA) based optimization of section 8.2. Twenty benchmark (BMP) images [130, 131] of dimension 512 × 512 along with the varying sizes of the "Gold-Coin" image as given in fig. 1.1 are considered to compute results. Perceptual distortion of the WBT_1X2_GAO is evaluated by means of the peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity

(IF), structural similarity index (SSIM) and universal image quality index (UIQ) with respect to 0.5, 1.0, 1.5, 2.0, 2.5 and 3 bits per Byte (bpB) of payloads. In WBT_1x2_GAO, the minimum value of PSNR is 33.21 dB at 3 bpB of payload for the “Desert” image whereas, the maximum value of PSNR is 54.15 dB at 0.5 bpB of payload for the “Foster City” image. The lowest MSE is 0.24 for “FosterCity” at payload of 0.5 bpB and that of the highest value is 31.00 for “Desert” at 3 bpB of payload. Minimum PSNR of “Desert” gives more than 30 dB and hence, the perceived watermarked retains good visual clarity [148]. The minimum values of IF, SSIM and UIQ are obtained at 3 bpB and the values are 0.99536 (Desert), 0.939195 (Splash) and 0.663116 (Splash), respectively. On the contrary, the maximum values of IF, SSIM and UIQ are obtained at 0.5 bpB and the values are 0.999992 (Airplane), 0.999871 (San Diego) and 0.999255 (San Diego) respectively. Usually, the IF, SSIM and UIQ are ranges from [0, 1], the closer the IF, SSIM and UIQ to one, watermarked image is more similar to the original image. The average values are also computed for various metrics of twenty images at variable payload to summarize the experimental results.

Table 8.11. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WBT_1x2_GAO scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.145421	0.250344	0.999984	0.999567	0.994298
	1.0	49.422982	0.742646	0.999953	0.997964	0.984533
	1.5	46.519601	1.449172	0.999910	0.996881	0.967921
	2.0	44.180597	2.483250	0.999844	0.993618	0.949882
	2.5	39.275647	7.682730	0.999552	0.984864	0.880518
	3.0	37.453408	11.687981	0.999300	0.972729	0.841933
Baboon	0.5	54.142422	0.250517	0.999986	0.999850	0.998909
	1.0	49.322226	0.760077	0.999959	0.999290	0.996941
	1.5	46.754014	1.373026	0.999927	0.998866	0.992829
	2.0	44.359107	2.383249	0.999873	0.997706	0.988962
	2.5	39.955176	6.569960	0.999657	0.994000	0.965668
	3.0	37.835126	10.704532	0.999439	0.989369	0.955201
Pepper	0.5	53.876715	0.266323	0.999977	0.999142	0.988760
	1.0	46.422549	1.481922	0.999862	0.988717	0.971649
	1.5	42.162214	3.952382	0.999621	0.981189	0.954462
	2.0	41.521170	4.581031	0.999575	0.979164	0.942817
	2.5	36.271238	15.344633	0.998544	0.962145	0.883838
	3.0	35.503177	18.313129	0.998313	0.951435	0.850751

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	54.138654	0.250734	0.999992	0.999486	0.982639
	1.0	49.423420	0.742571	0.999978	0.997591	0.955813
	1.5	46.151724	1.577276	0.999954	0.995994	0.913311
	2.0	43.981822	2.599549	0.999925	0.992163	0.878373
	2.5	38.879693	8.416102	0.999759	0.978710	0.762684
	3.0	37.229074	12.307587	0.999647	0.964185	0.717901
Sailboat	0.5	54.110007	0.252394	0.999987	0.999685	0.995188
	1.0	49.095326	0.800843	0.999959	0.998367	0.987344
	1.5	46.048619	1.615170	0.999918	0.997428	0.973128
	2.0	44.005460	2.585439	0.999869	0.994855	0.960336
	2.5	39.507090	7.284023	0.999633	0.988311	0.913299
	3.0	37.660013	11.144972	0.999438	0.999438	0.885754
Earth	0.5	54.155713	0.249752	0.999985	0.999672	0.997370
	1.0	49.456888	0.736871	0.999956	0.998448	0.992869
	1.5	46.755357	1.372601	0.999918	0.997697	0.985779
	2.0	44.344451	2.391306	0.999858	0.995246	0.976185
	2.5	40.158131	6.269995	0.999630	0.989653	0.937742
	3.0	38.034093	10.225181	0.999395	0.980226	0.910319
San Diego	0.5	54.137201	0.250818	0.999990	0.999871	0.999255
	1.0	49.422528	0.742724	0.999972	0.999407	0.997982
	1.5	46.906883	1.325537	0.999950	0.999157	0.996067
	2.0	44.419124	2.350541	0.999912	0.998223	0.993411
	2.5	40.767428	5.449259	0.999797	0.996484	0.984288
	3.0	38.396289	9.407005	0.999650	0.992790	0.976574
Splash	0.5	54.000320	0.258850	0.999977	0.998787	0.965256
	1.0	47.299330	1.211008	0.999879	0.987145	0.932784
	1.5	43.110554	3.177056	0.999679	0.981474	0.884149
	2.0	42.198245	3.919727	0.999619	0.977357	0.846695
	2.5	36.686256	13.946161	0.998666	0.957384	0.718135
	3.0	35.705379	17.480041	0.998367	0.939195	0.663116
Oakland	0.5	54.064019	0.255081	0.999986	0.999721	0.998422
	1.0	48.289669	0.964080	0.999942	0.998432	0.995446
	1.5	44.806629	2.149895	0.999864	0.997411	0.991156
	2.0	43.236833	3.086007	0.999817	0.995587	0.986620
	2.5	38.749584	8.672053	0.999470	0.990703	0.966623
	3.0	37.208845	12.365048	0.999280	0.983315	0.951637

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Foster City	0.5	54.156509	0.249706	0.999991	0.999422	0.992571
	1.0	49.501043	0.729417	0.999974	0.997207	0.980486
	1.5	46.632323	1.412043	0.999950	0.995723	0.958860
	2.0	44.249877	2.443951	0.999913	0.991194	0.934870
	2.5	39.796064	6.815125	0.999764	0.980013	0.846826
	3.0	37.834282	10.706612	0.999626	0.963037	0.797353
Anhinga	0.5	52.081071	0.402692	0.999969	0.999698	0.882230
	1.0	48.553767	0.907201	0.999930	0.999048	0.878763
	1.5	43.379737	2.986115	0.999771	0.998078	0.854741
	2.0	41.346418	4.769123	0.999634	0.993043	0.836931
	2.5	35.679246	17.585540	0.998655	0.990181	0.797543
	3.0	34.705352	22.006203	0.998316	0.982376	0.787174
Athens	0.5	51.881060	0.421671	0.999966	0.999551	0.970407
	1.0	49.219801	0.778216	0.999937	0.998957	0.966760
	1.5	45.745309	1.732006	0.999861	0.997997	0.937863
	2.0	42.620309	3.556718	0.999714	0.992437	0.911926
	2.5	37.773228	10.858192	0.999132	0.990175	0.852928
	3.0	36.539600	14.425148	0.998847	0.982908	0.838196
Bardowl	0.5	52.530426	0.363109	0.999963	0.999768	0.998743
	1.0	47.790922	1.081404	0.999890	0.998969	0.996695
	1.5	43.940903	2.624158	0.999730	0.997692	0.989078
	2.0	42.401484	3.740519	0.999620	0.996432	0.984796
	2.5	37.073159	12.757465	0.998690	0.987882	0.946570
	3.0	36.111448	15.919722	0.998378	0.984734	0.939775
Barnfall	0.5	53.104854	0.318122	0.999950	0.999696	0.998398
	1.0	49.326944	0.759251	0.999880	0.998758	0.996527
	1.5	45.360881	1.892309	0.999691	0.997571	0.989102
	2.0	43.448602	2.939139	0.999530	0.995439	0.984282
	2.5	38.487027	9.212501	0.998492	0.989366	0.953329
	3.0	36.822136	13.516577	0.997785	0.980904	0.937735
Butrfly	0.5	52.514547	0.364439	0.999974	0.999703	0.995016
	1.0	49.101758	0.799658	0.999943	0.999161	0.992913
	1.5	46.000310	1.633237	0.999884	0.998368	0.983135
	2.0	43.252550	3.074859	0.999781	0.995174	0.971779
	2.5	38.591025	8.994515	0.999360	0.991691	0.936579
	3.0	37.075438	12.750773	0.999092	0.986277	0.922799

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	52.161319	0.395319	0.999944	0.99954	0.749370
	1.0	49.393287	0.747741	0.999894	0.998862	0.747211
	1.5	45.748122	1.730884	0.999756	0.997917	0.733299
	2.0	42.746466	3.454886	0.999512	0.992161	0.721577
	2.5	37.576058	11.362516	0.998398	0.985217	0.691203
	3.0	36.277837	15.321333	0.997840	0.973454	0.683296
Bodie	0.5	53.142233	0.315396	0.999946	0.999759	0.979354
	1.0	47.720609	1.099054	0.999807	0.998815	0.975527
	1.5	43.327158	3.022487	0.999464	0.997023	0.963289
	2.0	42.144334	3.968687	0.999309	0.996014	0.959811
	2.5	36.570104	14.324185	0.997468	0.984540	0.921500
	3.0	35.780058	17.182029	0.996989	0.980956	0.913899
Bluheron	0.5	52.656006	0.352760	0.999956	0.999649	0.995693
	1.0	49.623876	0.709075	0.999913	0.998578	0.992728
	1.5	44.817446	2.144547	0.999737	0.997240	0.979662
	2.0	42.925853	3.315087	0.999595	0.995218	0.971234
	2.5	37.907743	10.527033	0.998711	0.989731	0.922399
	3.0	36.200649	15.596079	0.998088	0.976583	0.891539
Colomtn	0.5	53.079946	0.319952	0.999974	0.999730	0.984703
	1.0	49.026124	0.813706	0.999936	0.998820	0.982228
	1.5	44.157035	2.496760	0.999804	0.997738	0.969607
	2.0	42.335403	3.797869	0.999703	0.994813	0.961231
	2.5	37.074651	12.753083	0.999003	0.989460	0.923624
	3.0	35.961725	16.478125	0.998714	0.982214	0.912320
Desert	0.5	52.888865	0.334344	0.999952	0.999715	0.998172
	1.0	44.970361	2.070351	0.999675	0.996624	0.992990
	1.5	39.953744	6.572125	0.998971	0.992205	0.981001
	2.0	39.514153	7.272186	0.998883	0.991608	0.979086
	2.5	33.661773	27.983562	0.995705	0.976411	0.937623
	3.0	33.216728	31.003273	0.995360	0.972317	0.931528
Average case	0.5	53.348370	0.306116	0.999972	0.999601	0.973238
	1.0	48.619170	0.933891	0.999912	0.997458	0.965909
	1.5	44.913930	2.311939	0.999768	0.995682	0.949922
	2.0	42.961610	3.435656	0.999674	0.992873	0.937040
	2.5	38.022020	11.140430	0.998904	0.984846	0.887146
	3.0	36.577530	14.927070	0.998593	0.976922	0.865440

Table 8.12 revealed the comparative analysis of average PSNR values between WBT_1x2 and WBT_1x2_GAO against the payload variation 0.5 to 3 bpB. The average PSNR for schemes are identical at 0.5 and 1 bpB respectively. Since, GA optimization has been incorporated into WBT_1x2 for ensuring the enhancement of quality, the average PSNR variation of 1.08, 1.96, 1.36 and 2.69 dB are observed for WBT_1x2_GAO as compared to WBT_1x2 at 1.5, 2, 2.5 and 3 bpB respectively. As the average PSNR values are greater than or equal to 30 dB, the good fidelity watermarked images are to be constructed [148].

Table 8.12. Comparative analysis of obtained average PSNR values between WBT_1x2 and WBT_1x2_GAO with respect to increasing payload

WBT_1x2		**WBT_1X2_GAO**	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	53.34837	0.5	53.34837
1.0	48.61917	1.0	48.61917
1.5	43.83774	1.5	44.91393
2.0	41.00855	2.0	42.96161
2.5	36.66372	2.5	38.02202
3.0	33.88552	3.0	36.57753

The quantitative analysis of perceptual difference among the 1 x 2 block based watermarking using Binomial Transform (BT) followed by genetic algorithm (GA) based optimization (WBT_1X2_GAO), Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] is made through the widely acceptable quality metric named, peak signal to noise ratio (PSNR). Five color images viz. “Lena”, “Baboon”, “Pepper”, “Airplane” and “Sailboat” has been used to carrying out the experiment as these images are also tested against both DPTHDI [88] and DGTDHS [129]. It is observed from fig. 8.11 that the PSNR of the watermarked images for DPTHDI) [88] and DGTDHS [129] are measured with respect to 0.25 and 1 bpB of payload, respectively. The major limitation of the DPTHDI [88] and DGTDHS [129] are their fixed as well as low payload value. Unlikely, the WBT_1X2_GAO provides acceptable visual imperceptibility (i.e., PSNR ≥ 30 dB) for the payload range [0.5 – 3 bpB]. In contrast to DPTHDI [88], proposed WBT_1X2_GAO scheme ensured equal or higher PSNR (dB) at 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB for “Baboon”, 0.5, 1, 1.5 and 2 bpB of payloads for “Pepper”, 0.5, 1, 1.5 and 2 bpB of payloads for “Airplane” and 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for “Sailboat” respectively. Compared to

DGTDHS [129], the WBT_1X2_GAO ensured higher PSNR at 1 bpB of payload for “Lena”, “Baboon”, “Airplane” and “Sailboat” respectively.

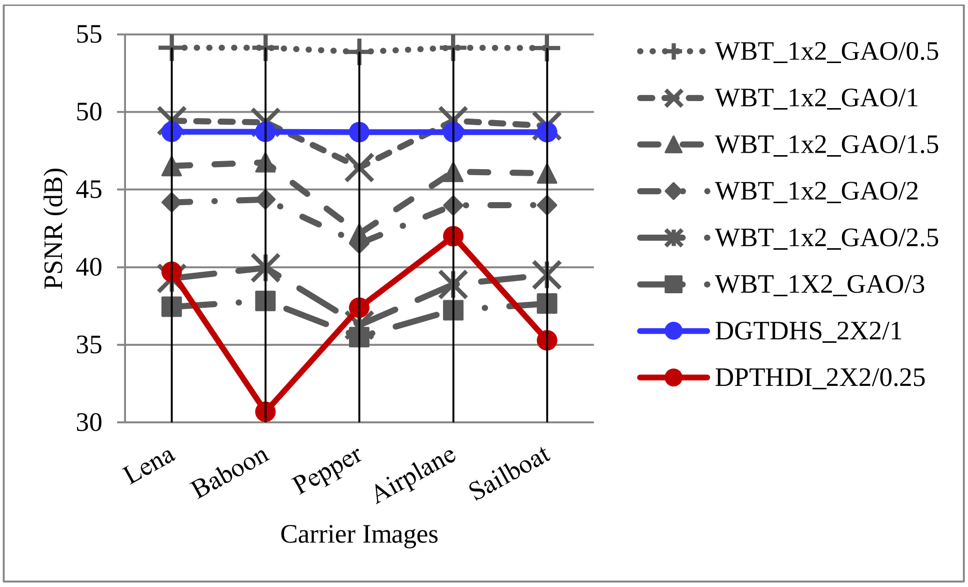


Fig. 8.11. Performance analysis of PSNR (dB) for variable payload based WBT_1x2_GAO and fixed payload based Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 81.2 illustrates a graphical analysis that compares WBT_1x2_GAO against the WBT_1x2, WBT_2x2_GAO, WLT_1x2_GAO, WDHT_1x2_GAO, DPTHDI [88] and DGTDHS [129] respectively by taking the average PSNR (dB) as the unit of performance measurements. Compared to WBT_1x2, the improvement of PSNR in WBT_1X2_GAO can be visually perceived as the payload increases from 1 bpB. In contrast to WDHT_1X2_GAO, WLT_1X2_GAO, WBT_2X2_GAO and WBT_1x2, the WBT_1X2_GAO method ensured less degradation in quality with respect to variable payload (0.5 – 3 bpB). The average PSNR for DPTHDI [88] is 37.40 dB which is obtained by taking the averages of PSNR values for “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images at 0.25 bpB of payload. In contrast to DPTHDI [88], the average PSNR of WBT_1x2_GAO scheme ensured equal or higher PSNR (dB) at 0.5, 1, 1.5, 2 and 2.5 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB which is obtained by taking the averages of PSNR values for “Lighthouse”, “Elaine”, “Lenna”, “Boat”, “F16” images at 1 bpB of payload. In contrast to fixed payload based DGTDHS [129] scheme, WBT_1x2_GAO scheme offers almost identical PSNR (dB) at 1 bpB while the latter scheme also supports fabrication of secret information with variable payload. All watermarked images also preserved a high transparency even at the extreme payload value of 3 bpB [148].

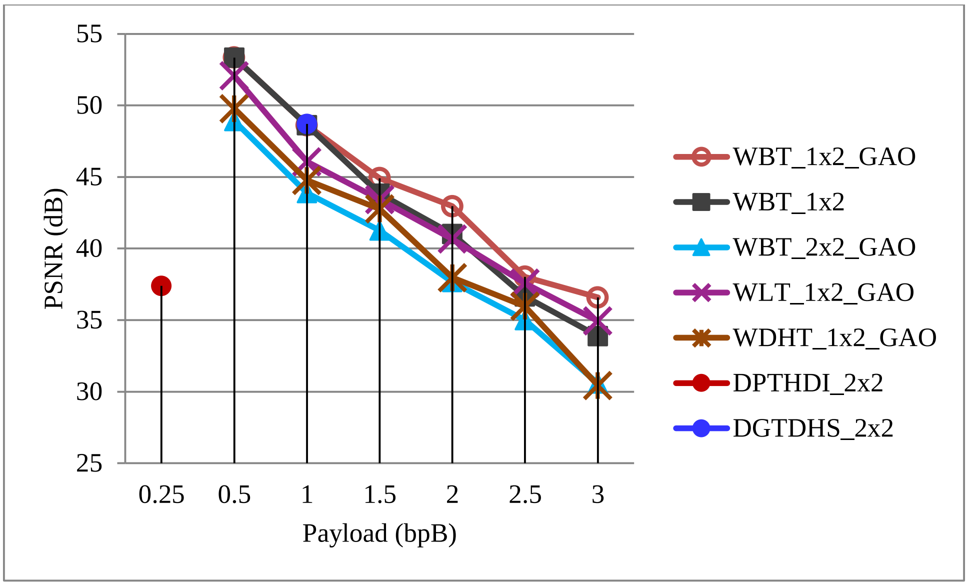


Fig. 8.12. Graphical representation of variation of average PSNR (dB) with respect to payload for WBT_1x2_GAO, WBT_1x2, WBT_2x2_GAO, WLT_1x2_GAO, WDHT_1x2_GAO and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

8.3.4. Optimization for Stirling Transform (ST) based watermarking

As discussed in chapter 5, proposed Stirling Transform (ST) based watermarking has been classified into two categories: 2 x 2 block based watermark fabrication (WST_2x2) and 1 x 2 block based watermark fabrication (WST_1x2).

Section 8.3.4.1 deals with the genetic algorithm (GA) based optimization for 2 x 2 block based watermark fabrication (WST_2X2_GAO). The genetic algorithm (GA) based optimization for 1 x 2 block based watermark fabrication (WST_1X2_GAO) has been discussed in section 8.3.4.2.

8.3.4.1. Optimization for 2 x 2 Block based Watermark Fabrication

The 2 x 2 blocks of the host image is adjusted according to the pixel adjustment rule given in equation (5.7) of section 5.2.1.1. Convert the adjusted blocks of pixel components into transformed components by the Stirling Transform (ST). Message digest, size and content of the watermark are fabricated on first/third/fourth transformed component of each 2 x 2 sub-matrix starting from the least significant bit position (LSB-0) toward higher order bit position. Optimization has been made against each embedded component to search for the optimized component that can preserve the fabricated bits of the embedded component. Inverse Stirling Transform (IST) yields the pixel components corresponding to the optimized components. Successive block embedding operation constructs the watermarked image.

The WST_2x2_GAO is explained with an example as given in section 8.3.4.1.1. Results are computed, analyzed and summarized in section 8.3.4.1.2.

8.3.4.1.1. Example

The pixel components are grouped into three 2 x 2 sub-blocks corresponding to RGB color channels which in turn are adjusted based on the offered payload value which is assumed as 3 bpB in this example. The sub-matrices corresponding to RGB color channels are obtained from the equation (5.7) as given in section 5.2.1.1.

$$R_1 = \begin{bmatrix} 212 & 198 \\ 32 & 45 \end{bmatrix} G_1 = \begin{bmatrix} 97 & 32 \\ 224 & 32 \end{bmatrix} B_1 = \begin{bmatrix} 156 & 166 \\ 118 & 65 \end{bmatrix}$$

Stirling Transform (ST) is applied on each 2 x 2 sub-matrix to convert it from spatial domain into transform domain. The 2 x 2 sub-matrices of transformed components such as T(R$_1$), T(G$_1$) and T(B$_1$) are obtained as given below:

$$T(R_1) = \begin{bmatrix} 212 & 410 \\ 838 & 1835 \end{bmatrix} T(G_1) = \begin{bmatrix} 97 & 129 \\ 417 & 1697 \end{bmatrix} T(B_1) = \begin{bmatrix} 156 & 322 \\ 772 & 2091 \end{bmatrix}$$

Let, secret bit stream *"010000010110010011111001011011010101"* is to be fabricated with 3 bpB of payload which ensured the fabrication of secret bits (λ_1 = 4, λ_2 = 4 and λ_3 = 4) into three least significant bits position of first, third and fourth transformed components as given in equation (5.8) of section 5.2.1.1. In addition, second transformed component is re-adjusted by adding the difference of the pre-embedded and post-embedded values of the first component. Hence, the 2 x 2 embedded sub-matrices are obtained as follows:

$$T'(R_1) = \begin{bmatrix} 210 & 408 \\ 840 & 1830 \end{bmatrix} T'(G_1) = \begin{bmatrix} 98 & 130 \\ 431 & 1705 \end{bmatrix} T'(B_1) = \begin{bmatrix} 150 & 316 \\ 779 & 2090 \end{bmatrix}$$

Optimization of embedded components is carried out by the usage of genetic algorithm (GA). GA is applied on each embedded component to find the optimized component without affecting the embedded bits. The 2 x 2 sub-matrices of optimized components are as follows:

$$T''(R_1) = \begin{bmatrix} 210 & 408 \\ 840 & 1830 \end{bmatrix} T''(G_1) = \begin{bmatrix} 98 & 130 \\ 415 & 1705 \end{bmatrix} T''(B_1) = \begin{bmatrix} 150 & 316 \\ 779 & 2090 \end{bmatrix}$$

On application of inverse Stirling Transform (ST) over RGB color channels, the 2 x 2 sub-matrices of pixel components are generated corresponding to the 2 x 2 sub-matrices of optimized components as follows:

$$R'_1 = \begin{bmatrix} 210 & 198 \\ 36 & 18 \end{bmatrix} G'_1 = \begin{bmatrix} 98 & 32 \\ 221 & 7 \end{bmatrix} B'_1 = \begin{bmatrix} 150 & 166 \\ 131 & 8 \end{bmatrix}$$

8.3.4.1.2. Results and Discussions

The WST_2X2_GAO computes the results against the fabrication of the varying sizes of the Gold-Coin into twenty color images of dimension 512 × 512 as given in table 8.13. Computed results revealed that the peak signal to noise ratio (PSNR) and mean squared error (MSE) values are belonging into the ranges [54.55 dB (Bluheron), 25.49 dB (Desert)] and [0.22 (Bluheron), 183.53 (Desert)] with respect to 0.5 and 3 bpB respectively. The PSNR value is of maximum (by keeping MSE as minimum) at 0.5 bpB and that of the minimum PSNR (or maximum MSE) is obtained at 3 bpB. The image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) values obtained at respective payload values of 0.5 and 3 bpB belongs to the following ranges: [0.999992 (Airplane), 0.972376 (Desert)], [0.999980 (Athens), 0.874742 (Bobcat)] and [0.999331 (San Diego), 0.408747 (Splash)] respectively. It is to be noted that the minimum PSNR obtained in this experiment is falling below 30 dB however, since, the average PSNR crossed the threshold level of PSNR (i.e., 30 dB), the quality of the watermarked images become well perceptible [148].

Table 8.13. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WST_2x2_GAO scheme

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Lena	0.5	54.164743	0.249233	0.999984	0.999684	0.994767
	1.0	46.999298	1.297629	0.999918	0.998892	0.971828
	1.5	43.458391	2.932521	0.999815	0.997047	0.942228
	2.0	38.344103	9.520723	0.999409	0.989707	0.860127
	2.5	34.552063	22.796805	0.998566	0.981986	0.756546
	3.0	30.358763	59.869275	0.996397	0.943168	0.623937
Baboon	0.5	54.143899	0.250432	0.999986	0.999885	0.999015
	1.0	46.962725	1.308602	0.999930	0.999585	0.994255
	1.5	43.421884	2.957276	0.999843	0.998859	0.987094
	2.0	38.260067	9.706743	0.999488	0.995309	0.960692
	2.5	34.489938	23.125253	0.998778	0.992663	0.932187
	3.0	30.316951	60.448455	0.996833	0.974433	0.876867
Pepper	0.5	53.611422	0.283098	0.999975	0.998541	0.987707
	1.0	46.962725	1.308602	0.999930	0.999585	0.994255
	1.5	43.421884	2.957276	0.999843	0.998859	0.987094
	2.0	38.260067	9.706743	0.999488	0.995309	0.96069
	2.5	34.489938	23.125253	0.998778	0.992663	0.932187
	3.0	30.316951	60.448455	0.996833	0.974433	0.876867

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	54.124779	0.251537	0.999992	0.999625	0.983309
	1.0	46.963831	1.308269	0.999962	0.998689	0.925791
	1.5	43.424118	2.955755	0.999915	0.996490	0.868059
	2.0	38.535750	9.109725	0.999739	0.988327	0.747836
	2.5	34.609623	22.496659	0.999355	0.979367	0.631066
	3.0	31.395169	47.158866	0.998649	0.945845	0.532268
Sailboat	0.5	54.096070	0.253205	0.999987	0.999741	0.995591
	1.0	46.812312	1.354718	0.999931	0.999074	0.977181
	1.5	43.169205	3.134438	0.999842	0.997495	0.954202
	2.0	37.974856	10.365606	0.999479	0.990890	0.893236
	2.5	34.388945	23.669324	0.998807	0.984731	0.820938
	3.0	30.205777	62.015841	0.996896	0.948499	0.707935
Earth	0.5	54.167048	0.2491010	0.999985	0.999763	0.997650
	1.0	46.989584	1.300534	0.999922	0.999167	0.986821
	1.5	43.488584	2.912204	0.999827	0.997761	0.970896
	2.0	38.596252	8.983697	0.999464	0.992354	0.919806
	2.5	34.582601	22.637072	0.998654	0.986205	0.839937
	3.0	30.900142	52.852573	0.996701	0.959053	0.731415
San Diego	0.5	54.167580	0.249070	0.999990	0.999911	0.999331
	1.0	47.009242	1.294661	0.999951	0.999687	0.996313
	1.5	43.487280	2.913079	0.999891	0.999154	0.992080
	2.0	38.625172	8.924072	0.999667	0.997177	0.978642
	2.5	34.626879	22.407449	0.999164	0.994826	0.954227
	3.0	31.470237	46.350737	0.998270	0.985879	0.919578
Splash	0.5	53.781897	0.272201	0.999975	0.997960	0.965398
	1.0	45.549754	1.811777	0.999826	0.988325	0.898110
	1.5	41.482417	4.622091	0.999547	0.981752	0.833171
	2.0	36.106360	15.938385	0.998405	0.966984	0.693228
	2.5	33.469567	29.249846	0.997254	0.955555	0.553419
	3.0	29.098915	80.018212	0.992057	0.901762	0.408747
Oakland	0.5	53.978528	0.260152	0.999985	0.999776	0.998472
	1.0	46.305838	1.522286	0.999913	0.998971	0.992453
	1.5	42.400458	3.741402	0.999778	0.997377	0.997377
	2.0	37.065964	12.778619	0.999217	0.991685	0.955214
	2.5	33.929438	26.310951	0.998501	0.986765	0.907619
	3.0	29.949195	65.790142	0.995989	0.963174	0.837429

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	54.174437	0.248677	0.999991	0.999581	0.993350
	1.0	47.021129	1.291122	0.999954	0.998501	0.964335
	1.5	43.505136	2.901126	0.999896	0.995932	0.924045
	2.0	38.653678	8.865689	0.999684	0.986591	0.820071
	2.5	34.788602	21.588382	0.999224	0.976535	0.698535
	3.0	31.613291	44.848841	0.998396	0.937669	0.581562
Anhinga	0.5	53.822131	0.269691	0.999979	0.999952	0.883977
	1.0	45.509950	1.828459	0.999859	0.999069	0.859014
	1.5	41.927247	4.172108	0.999680	0.997926	0.842612
	2.0	36.857854	13.405868	0.998973	0.993313	0.815485
	2.5	33.787402	27.185675	0.997913	0.986531	0.756295
	3.0	29.813832	67.873004	0.994803	0.964388	0.699481
Athens	0.5	54.102421	0.252835	0.999979	0.999980	0.972700
	1.0	45.872174	1.682142	0.999865	0.999101	0.942894
	1.5	43.675100	2.789782	0.999777	0.998182	0.920444
	2.0	39.721621	6.932951	0.999446	0.994554	0.887166
	2.5	34.622894	22.428019	0.998208	0.985834	0.783304
	3.0	32.267589	38.576374	0.996912	0.968414	0.723873
Bardowl	0.5	53.810332	0.270425	0.999972	0.999913	0.999162
	1.0	45.491277	1.836338	0.999814	0.999134	0.993189
	1.5	42.301466	3.827663	0.999610	0.997776	0.984912
	2.0	36.569691	14.325547	0.998521	0.986481	0.944518
	2.5	33.652248	28.045005	0.997158	0.983605	0.911925
	3.0	28.404699	93.888062	0.990301	0.935349	0.819892
Barnfall	0.5	54.282906	0.242543	0.999962	0.999855	0.998867
	1.0	46.841352	1.345690	0.999786	0.999289	0.992385
	1.5	43.576977	2.853530	0.999541	0.998104	0.984590
	2.0	38.257896	9.711598	0.998381	0.992629	0.954161
	2.5	34.480713	23.174429	0.996276	0.987159	0.900463
	3.0	30.761462	54.567509	0.990693	0.955254	0.807997
Butrfly	0.5	54.125218	0.251511	0.999982	0.999940	0.996048
	1.0	46.207664	1.557090	0.999889	0.999265	0.984765
	1.5	43.619316	2.825847	0.999799	0.998545	0.974134
	2.0	39.317643	7.608797	0.999459	0.995449	0.946507
	2.5	34.699212	22.037336	0.998425	0.989990	0.873828
	3.0	32.039491	40.656612	0.997080	0.974895	0.812561

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	54.068113	0.254840	0.999964	0.999913	0.750419
	1.0	45.977910	1.641682	0.999768	0.998702	0.734219
	1.5	43.568527	2.859088	0.999598	0.997605	0.724160
	2.0	39.087537	8.022811	0.998867	0.991231	0.698192
	2.5	35.072939	20.220246	0.997158	0.985790	0.661184
	3.0	26.751700	137.375835	0.980574	0.874742	0.545567
Bodie	0.5	54.100674	0.252937	0.999957	0.999898	0.980116
	1.0	45.770991	1.721794	0.999702	0.999160	0.970550
	1.5	42.016012	4.087701	0.999286	0.997676	0.960292
	2.0	35.708573	17.467189	0.996882	0.983013	0.914773
	2.5	33.405705	29.683135	0.994858	0.979962	0.881866
	3.0	27.435858	117.353087	0.978793	0.896810	0.748096
Bluheron	0.5	54.552339	0.227953	0.999972	0.999860	0.996625
	1.0	47.328067	1.203022	0.999852	0.999245	0.986804
	1.5	44.211753	2.465499	0.999698	0.998170	0.974975
	2.0	38.219167	9.798590	0.998800	0.992690	0.916904
	2.5	34.820904	21.428408	0.997377	0.987645	0.847265
	3.0	31.529044	45.727338	0.994396	0.962338	0.747887
Colomtn	0.5	54.155071	0.249788	0.999980	0.999863	0.985545
	1.0	46.295302	1.525984	0.999880	0.999219	0.975375
	1.5	42.486857	3.667706	0.999713	0.998077	0.963829
	2.0	37.246123	12.259365	0.999042	0.993066	0.932032
	2.5	33.986026	25.970344	0.997971	0.987456	0.879535
	3.0	30.066640	64.034834	0.995003	0.961243	0.814444
Desert	0.5	53.083555	0.319686	0.999953	0.999696	0.998276
	1.0	43.862853	2.671745	0.999590	0.997089	0.991207
	1.5	39.275220	7.683485	0.998809	0.992648	0.979995
	2.0	32.857128	33.679648	0.994837	0.973715	0.931918
	2.5	31.612890	44.852984	0.993261	0.970916	0.913326
	3.0	25.493603	183.535481	0.972376	0.896209	0.794692
Average case	0.5	54.025660	0.257946	0.999978	0.999667	0.973816
	1.0	46.336700	1.540607	0.999862	0.998487	0.956587
	1.5	42.895890	3.462979	0.999685	0.996772	0.938309
	2.0	37.713280	11.855620	0.998862	0.989524	0.886560
	2.5	34.203430	25.121630	0.997784	0.983809	0.821783
	3.0	30.009470	71.169480	0.992898	0.946178	0.730555

Both WST_2X2 and WST_2X2_GAO techniques are operated in Stirling Transform (ST) domain however, the latter one provides better image quality in terms of average PSNR as soon as the payload exceeds 0.5 bpB. Table 8.14 shows the identical values of average PSNR for both schemes at 0.5 bpB. However, for subsequent payload values in the spread from 1 to 3 bpB, the following dispersions of average PSNR values in the latter scheme are observed compared to the former one: 0.07, 0.81, 0.72, 0.90 and 1.43 dB respectively. Hence, the usage GA optimization is proved as successful as the average PSNR values with respect to payload variation (0.5 – 3 bpB) are falling above 30 dB and producing good quality watermarked images [148].

Table 8.14. Comparative analysis of obtained average PSNR values between WST_2x2 and WST_2x2_GAO with respect to increasing payload

WST_2x2		**WST_2X2_GAO**	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	54.02566	0.5	54.02566
1.0	46.24363	1.0	46.33670
1.5	42.08165	1.5	42.89589
2.0	36.99384	2.0	37.71328
2.5	33.30990	2.5	34.20343
3.0	28.57277	3.0	30.00947

Image quality analysis in terms of PSNR has been made among the proposed 2 x 2 block based watermarking using Stirling Transform (ST) followed by Genetic Algorithm (GA) based optimization (WST_2X2_GAO), Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] respectively. The comparison has been done for five different color images such as "Lena", "Baboon", "Pepper", "Airplane" and "Sailboat". It is observed from fig. 8.13 that the PSNR of the watermarked images for DPTHDI [88] and DGTDHS [129] are obtained with respect to 0.25 and 1 bpB of payloads respectively. The major limitation of the DPTHDI [88] and DGTDHS [129] are their fixed as well as low payload. On the contrary, the WST_2X2_GAO is designed based on the principle of variable payload (0.5 – 3 bpB) that offered high-quality watermarked images at respective payload values [148]. Compared to DPTHDI [88], the WST_2X2_GAO scheme ensured equal or higher PSNR (dB) at 0.5, 1 and 1.5 bpB of payloads for "Lena", 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Baboon", 0.5, 1, 1.5 and 2 bpB of payloads for "Pepper", 0.5, 1 and 1.5 bpB of payloads for "Airplane" and 0.5, 1, 1.5 and 2 bpB of payloads for "Sailboat" respectively. In contrast to DGTDHS [129], proposed

WST_2X2_GAO offered less PSNR value at 1 bpB however, the payload variation for a spread from 0.5 to 3 bpB takes the key importance.

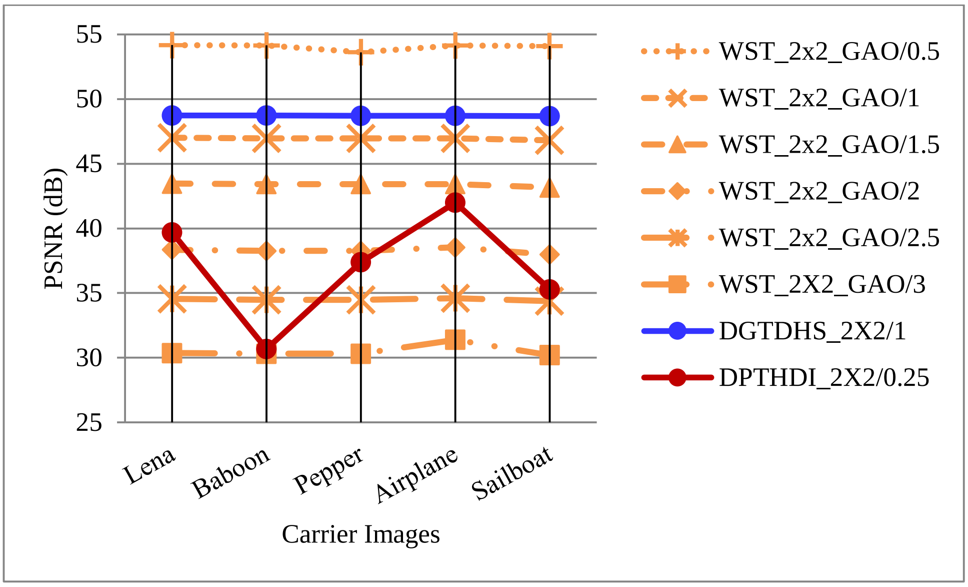


Fig. 8.13. Performance analysis of PSNR (dB) for variable payload based WST_2x2 and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

From table 8.14, the values of average PSNR are observed at varying payload for the spread of 0.5 to 3 bpB. Fig. 8.14 illustrates the trends of average PSNR variation for WST_2x2_GAO, WST_2x2, WBT_2x2_GAO, WLT_2x2_GAO, WDHT_2x2_GAO, DPTHDI [88] and DGTDHS [129] respectively. Compared to WST_2x2 method, the improvement of average PSNR in WST_2X2_GAO method can be visually perceived as the payload increases from 0.5 bpB. In contrast to WBT_2X2_GAO, WLT_2X2_GAO and WDHT_2X2_GAO, the superiority of the WST_2X2_GAO in terms of average PSNR values has also been observed from the chart. The average PSNR of DPTHDI [88] is 37.40 dB as obtained by taking the average of PSNR values for "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" at 0.25 bpB of payload. In comparison with DPTHDI [88], the average PSNR of WST_2x2_GAO ensured equal or higher PSNR (dB) at 0.5, 1, 1.5 and 2 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB as obtained by taking the average of PSNR values for "Lighthouse", "Elaine", "Lenna", "Boat" and "F16" images at 1 bpB of payload. Compared to DGTDHS [129], the WST_2x2_GAO scheme is lacking with the average PSNR at 1 bpB however, the quality distortion of WST_2x2_GAO scheme sacrificed due to the incorporation of variable payload for the spread from 0.5 to 3 bpB for maintaining the perceptible quality level in the watermarked images [148].

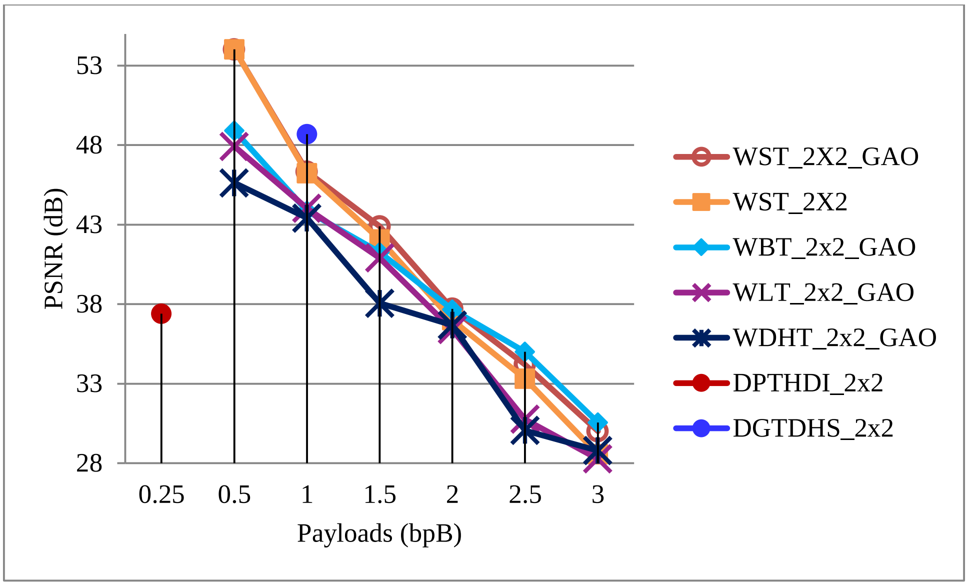


Fig. 8.14. Graphical representation of variation of average PSNR (dB) with respect to payload for WST_2x2_GAO, WST_2x2, WBT_2x2_GAO, WLT_2x2_GAO, WDHT_2x2_GAO and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

8.3.4.2. Optimization for 1 x 2 Block based Watermark Fabrication

Divide the cover image into 1 x 2 non-overlapping blocks in row major order. The pixel components of each block are adjusted to avoid the overflow/underflow situations by applying a pre-embedding adjustment as mentioned in equation (5.22) of section 5.2.2.2. Stirling Transform (ST) converts each 1 x 2 sub-matrix of pixel components into transform domain. The fabrication of the secret bits corresponding to the message digest, size and content of the watermark has been made on each transformed component of a 1 x 2 sub-matrix starting from the least significant bit position (i.e., LSB-0) toward higher order bit positions. Optimized component corresponding to each embedded component is obtained based on the GA optimization of section 8.2. The optimized components reduced the difference with respect to the pixel components without affecting the fabricated bits of the embedded component. Inverse Stirling Transform (IST) is applied over 1 x 2 sub-matrices of optimized components to re-compute the pixel components in spatial domain. The process is repeated until and unless the entire secret information is concealed and the watermarked image is produced.

In section 8.3.4.2.1, all steps including decomposition, adjustment, forward transformation, embedding, optimization and inverse transformation phases of WST_1x2_GAO are clearly described with an example. Experimental results, comparative analysis and discussions have been elaborated in section 8.3.4.2.2.

8.3.4.2.1. Example

Suppose, the cover image is of 512 x 512 in dimension which is decomposed into 1 x 2 non-overlapping blocks. Among the set of blocks, three consecutive blocks corresponding to RGB channels are considered for fabrication with 3 bpB of payload. On adjustment (as discussed in equation (5.22) of section 5.2.2.2), the pixel components are re-computed as follows:

$$R_1 = [164\ 16] \quad G_1 = [240\ 57] \quad B_1 = [71\ 31]$$

Stirling Transform (ST) is applied on each 1 x 2 sub-matrix or pair of pixel components to convert into transform domain. On application of Stirling Transform (ST), following three 1 x 2 sub-matrices viz. T(R_1), T(G_1) and T(B_1) are obtained:

$$T(R_1) = [164\ 180] \quad T(G_1) = [240\ 297] \quad T(B_1) = [71\ 102]$$

Let the secret bit stream *"011101110011100010"* is to be fabricated based on the embedding rule of equation (5.25) of section 5.2.2.1. In this example, three bits are fabricated ($\lambda = \lambda_1 = \lambda_2 = 3$) on each transformed component starting from LSB-0 toward the higher order bit position. Hence, the embedded 1 x 2 sub-matrices becomes:

$$T'(R_1) = [166\ 181] \quad T'(G_1) = [243\ 302] \quad T'(B_1) = [65\ 98]$$

Genetic algorithm (GA) based optimization has been used to ensure the enhancement of quality without hampering the embedded bits. GA is applied on each embedded component by taking the optimized component closest to the pre-embedded component without affecting the embedded bits. The optimized 1 x 2 sub-matrices are as follows:

$$T''(R_1) = [166\ 181] \quad T''(G_1) = [243\ 294] \quad T''(B_1) = [73\ 98]$$

On application of inverse Stirling Transform (IST) over the 1 x 2 sub-matrices of optimized components, the 1 x 2 sub-matrices of pixel components are obtained as follows:

$$R'_1 = [166\ 15]\ G'_1 = [243\ 51]\ B'_1 = [73\ 21]$$

It is observed that the modified pixel components are non-fractional, non-negative, less than or equal to 255 and the degradation of quality has been reduced tremendously.

8.3.4.2.2. Results and Discussions

To analyze the visual clarity of the obtained watermarked images, the peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) are computed against the payload variation of 0.5 to 3 bpB. The experiment takes twenty color images (fig. 1.1) as test images and subsequently fabricates the secret watermark with varying sizes to summarize the performance of the quality metrics. The WST_1x2_GAO offered the minimum value of PSNR as 32.89 dB at 3 bpB of payload for the "Desert" image and that of the maximum value obtained is 54.15 dB at 0.5 bpB of payload for the "Foster City" image. The MSE and PSNR are inversely proportional to each other. The lowest MSE is 0.24 for "FosterCity" at payload of 0.5 bpB whereas the highest MSE is 33.35 for "Desert" at 3 bpB of payload. The minimum obtained values of IF and UIQ are 0.994896 (Desert) and 0.702353 (Bobcat) respectively at 3 bpB whereas, the SSIM is minimum of 0.966280 (Pepper) at 2.5 bpB. Consequently, the maximum obtained values of IF and UIQ are 0.999992 (Airplane) and 0.999333 (San Diego) respectively at 0.5 bpB of payload while the maximum SSIM is 0.999919 (Athens) at 1 bpB. Usually, the IF, SSIM and UIQ are ranges from [0, 1], the closer the IF, SSIM and UIQ to one, watermarked image is more similar to the original image. The average values are computed for these metrics at variable payload to summarize the results.

Table 8.15. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WST_1x2 scheme

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Lena	0.5	54.126228	0.251453	0.999984	0.999646	0.994776
	1.0	50.038663	0.644486	0.999959	0.999683	0.985716
	1.5	47.313069	1.207183	0.999924	0.998752	0.973818
	2.0	44.711515	2.197499	0.999862	0.998804	0.954964
	2.5	41.270474	4.853252	0.999704	0.994929	0.917575
	3.0	38.687600	8.796710	0.999454	0.994901	0.867170
Baboon	0.5	54.041626	0.256399	0.999986	0.999870	0.998967
	1.0	50.027607	0.646129	0.999965	0.999871	0.997203
	1.5	47.252253	1.224207	0.999935	0.999497	0.993926
	2.0	44.680216	2.213394	0.999882	0.999508	0.990003
	2.5	41.081829	5.068709	0.999734	0.997155	0.973964
	3.0	38.584576	9.007882	0.999525	0.997173	0.962705
Pepper	0.5	48.034773	1.022357	0.999900	0.990122	0.977761
	1.0	46.665162	1.401406	0.999868	0.990282	0.972085
	1.5	42.095731	4.013351	0.999611	0.982232	0.956006
	2.0	41.225912	4.903307	0.999536	0.982357	0.943650
	2.5	35.948993	16.526503	0.998389	0.966280	0.902112
	3.0	35.065538	20.254734	0.998081	0.966297	0.862830

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Airplane	0.5	53.971148	0.260594	0.999992	0.999579	0.983422
	1.0	49.968752	0.654945	0.999981	0.999628	0.958382
	1.5	47.254089	1.223690	0.999964	0.998567	0.931284
	2.0	44.709653	2.198441	0.999937	0.998605	0.892525
	2.5	41.700828	4.395390	0.999874	0.994604	0.834021
	3.0	38.861450	8.451529	0.999758	0.994514	0.756468
Sailboat	0.5	53.339840	0.301367	0.999984	0.999694	0.995547
	1.0	49.683212	0.699453	0.999964	0.999722	0.988372
	1.5	46.601886	1.421974	0.999928	0.998909	0.978621
	2.0	44.359979	2.382771	0.999880	0.998948	0.964192
	2.5	40.586356	5.681260	0.999715	0.995128	0.935711
	3.0	38.317187	9.579914	0.999518	0.995132	0.898377
Earth	0.5	54.155713	0.249752	0.999985	0.999732	0.997668
	1.0	50.043001	0.643843	0.999961	0.999767	0.993535
	1.5	47.439663	1.172503	0.999930	0.999082	0.987945
	2.0	44.761540	2.172332	0.999871	0.999111	0.978081
	2.5	41.794594	4.301509	0.999742	0.996313	0.956880
	3.0	38.958986	8.263837	0.999508	0.996289	0.924815
San Diego	0.5	54.121904	0.251703	0.999990	0.999899	0.999333
	1.0	50.037489	0.644660	0.999976	0.999910	0.998168
	1.5	47.425861	1.176235	0.999956	0.999652	0.996659
	2.0	44.760398	2.172903	0.999919	0.999663	0.994002
	2.5	41.856545	4.240585	0.999841	0.998691	0.988918
	3.0	38.979832	8.224266	0.999693	0.998683	0.979961
Splash	0.5	49.270565	0.769172	0.999917	0.989397	0.963107
	1.0	47.535985	1.146784	0.999885	0.989492	0.935045
	1.5	43.452561	2.936461	0.999692	0.984317	0.903258
	2.0	42.269093	3.856301	0.999614	0.984459	0.859429
	2.5	37.586521	11.335174	0.998802	0.974680	0.791730
	3.0	36.252111	15.412363	0.998453	0.974519	0.703385
Oakland	0.5	51.278461	0.484433	0.999968	0.999417	0.997711
	1.0	48.69399	0.878377	0.999947	0.999427	0.995700
	1.5	45.029010	2.042580	0.999868	0.998414	0.991968
	2.0	43.361272	2.998839	0.999818	0.998444	0.987258
	2.5	39.032917	8.124348	0.999464	0.994753	0.974776
	3.0	37.314146	12.068845	0.999255	0.994680	0.957688

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Foster City	0.5	54.156509	0.249706	0.999991	0.999518	0.993383
	1.0	50.058037	0.641618	0.999977	0.999580	0.982171
	1.5	47.433780	1.174092	0.999958	0.998335	0.966862
	2.0	44.766540	2.169832	0.999922	0.998400	0.941266
	2.5	41.885645	4.212266	0.999850	0.993688	0.896054
	3.0	39.013153	8.161406	0.999709	0.993675	0.827783
Anhinga	0.5	50.822348	0.538078	0.999958	0.999807	0.882629
	1.0	49.533637	0.723963	0.999944	0.999856	0.881614
	1.5	43.897760	2.650356	0.999797	0.998758	0.858040
	2.0	42.992535	3.264575	0.999750	0.999029	0.852883
	2.5	38.368759	9.466827	0.999276	0.995962	0.842165
	3.0	37.156935	12.513731	0.999042	0.995478	0.819897
Athens	0.5	51.881060	0.421671	0.999966	0.999864	0.971365
	1.0	51.135386	0.500658	0.999960	0.999919	0.970370
	1.5	46.199546	1.560003	0.999875	0.998980	0.941996
	2.0	45.218493	1.955379	0.999843	0.999326	0.936468
	2.5	42.999221	3.259554	0.999740	0.997225	0.924419
	3.0	40.597041	5.667299	0.999547	0.996773	0.893198
Bardowl	0.5	50.094251	0.636290	0.999934	0.999464	0.997727
	1.0	49.020251	0.814807	0.999916	0.999472	0.997147
	1.5	44.296785	2.417696	0.999751	0.998203	0.989448
	2.0	43.351881	3.005330	0.999692	0.998227	0.987761
	2.5	37.950813	10.423150	0.998909	0.987805	0.954810
	3.0	36.885008	13.322310	0.998619	0.987709	0.947297
Barnfall	0.5	53.104854	0.318122	0.999950	0.999784	0.998572
	1.0	50.239177	0.615407	0.999899	0.999812	0.996867
	1.5	47.007544	1.295167	0.999790	0.999150	0.993415
	2.0	44.657747	2.224875	0.999640	0.999178	0.988455
	2.5	40.990457	5.176480	0.999105	0.995480	0.973622
	3.0	38.379661	9.443091	0.998418	0.995530	0.953810
Butrfly	0.5	52.514547	0.364439	0.999974	0.999853	0.995615
	1.0	50.882130	0.530722	0.999962	0.999869	0.993905
	1.5	46.545614	1.440518	0.999897	0.999267	0.984766
	2.0	45.102054	2.008514	0.999856	0.999378	0.980372
	2.5	42.497849	3.658435	0.999740	0.997735	0.970529
	3.0	40.158629	6.269276	0.999555	0.997544	0.954657

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
Bobcat	0.5	52.161319	0.395319	0.999944	0.999696	0.749616
	1.0	51.007421	0.515630	0.999927	0.999725	0.748645
	1.5	46.296547	1.525547	0.999785	0.998742	0.734346
	2.0	45.147845	1.987448	0.999720	0.998949	0.731919
	2.5	41.873787	4.223782	0.999404	0.995447	0.716921
	3.0	39.846047	6.737140	0.999050	0.994998	0.702353
Bodie	0.5	50.196920	0.621424	0.999888	0.999554	0.978414
	1.0	48.106353	1.005645	0.999822	0.999581	0.975564
	1.5	43.968002	2.607835	0.999532	0.998158	0.967335
	2.0	42.800336	3.412296	0.999398	0.998181	0.963347
	2.5	37.008099	12.950019	0.997627	0.984753	0.931257
	3.0	35.824191	17.008313	0.996963	0.984620	0.914741
Bluheron	0.5	52.656006	0.352760	0.999956	0.999772	0.996039
	1.0	50.119309	0.632629	0.999922	0.999779	0.992999
	1.5	47.141002	1.255972	0.999846	0.999104	0.986159
	2.0	44.993597	2.059304	0.999747	0.999145	0.978268
	2.5	42.145429	3.967686	0.999514	0.996878	0.964195
	3.0	37.754306	10.905604	0.998664	0.996685	0.906332
Colomtn	0.5	52.128962	0.398276	0.999968	0.999769	0.984849
	1.0	49.555255	0.720368	0.999943	0.999805	0.982616
	1.5	45.307793	1.915583	0.999850	0.999044	0.975630
	2.0	43.605367	2.834938	0.999778	0.999122	0.970025
	2.5	39.308274	7.625230	0.999405	0.996325	0.958598
	3.0	37.529969	11.483740	0.999102	0.996223	0.933920
Desert	0.5	45.961851	1.647764	0.999735	0.997282	0.993767
	1.0	45.142962	1.989683	0.999685	0.997345	0.993226
	1.5	40.049655	6.428576	0.998979	0.992915	0.982032
	2.0	39.605029	7.121597	0.998884	0.993036	0.981067
	2.5	33.610840	28.313681	0.995590	0.975230	0.945081
	3.0	32.899735	33.350842	0.994896	0.974986	0.9313737
Average case	0.5	51.900940	0.489554	0.999949	0.998586	0.972513
	1.0	49.374690	0.802561	0.999923	0.998626	0.966967
	1.5	45.600410	2.034476	0.999793	0.997004	0.954676
	2.0	43.854050	2.856994	0.999727	0.997094	0.943797
	2.5	39.974910	7.890192	0.999171	0.991453	0.917667
	3.0	37.853310	11.746140	0.998841	0.991320	0.884938

A comparative study has also been made between the 1 x 2 block based watermark fabrication (WST_1x2) of section 5.2.2 and the 1 x 2 block based watermark fabrication followed by genetic algorithm (GA) based optimization (WST_1X2_GAO) in terms of average PSNR with respect to the variable payload that offers a spread from 0.5 to 3 bpB. Table 8.16 demonstrated that the average PSNR is 51.90 dB at 0.5 bpB and 34.97 dB at 3 bpB for WST_1x2 method. On the contrary, the WST_1X2_GAO offered the values of the average PSNR in the range [51.90 -37.85 dB] with respect to the payload values of 3 and 0.5 bpB respectively. Quality enhancement ensured the improvement of average PSNR for the WST_1X2_GAO over the WST_1x2 and hence, the dispersion of average PSNR values are clearly observed as the payload exceeds 1 bpB. Since, the average PSNR is above 30 dB, the quality of the watermarked images is also highly perceptible [148].

Table 8.16. Comparative analysis of obtained average PSNR values between WST_1x2 and WST_1x2_GAO with respect to increasing payload

WST_1x2		**WST_1X2_GAO**	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
0.5	51.90094	0.5	51.90094
1.0	49.37469	1.0	49.37469
1.5	44.50645	1.5	45.60041
2.0	42.00782	2.0	43.85405
2.5	37.66123	2.5	39.97491
3.0	34.97197	3.0	37.85331

A graphical analysis has been made among the 1 x 2 block based watermarking (WST_1X2_GAO), Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] to analyze the visual imperceptibility as depicted in fig 8.15. The offered payload values for DPTHDI [88] and DGTDHS [129] are 0.25 and 1 bpB respectively. On the contrary, the WST_1X2_GAO provides the payload dispersion in the ranges of 0.5 to 3 bpB. Compared to DPTHDI [88], the WST_1X2_GAO ensured equal or higher PSNR (dB) at 0.5, 1, 1.5, 2 and 2.5 bpB for "Lena", 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Baboon", 0.5, 1, 1.5 and 2 bpB for "Pepper", 0.5, 1, 1.5, 2 and 2.5 bpB for "Airplane" and 0.5, 1, 1.5, 2, 2.5 and 3 bpB for "Sailboat" respectively. Compared to DGTDHS [129], the WST_1X2_GAO ensured higher PSNR for majority of the watermarked images at 1 bpB. The higher PSNR values (i.e., ≥ 30 dB) for payload range (0.5 – 3 bpB) in WST_1x2_GAO ensured good visual imperceptibility [148].

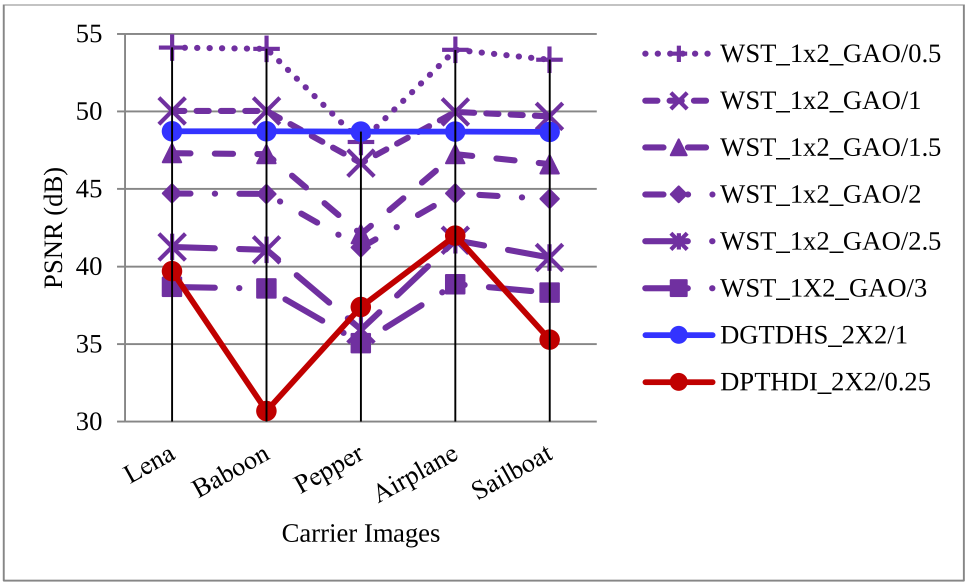


Fig. 8.15. Performance analysis of PSNR (dB) for variable payload based WST_1x2_GAO and fixed payload based Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Fig. 8.16 depicts the comparative analysis in terms of average PSNR values among the variable payload (WST_1x2_GAO, WST_1x2, WST_2x2_GAO, WBT_1x2_GAO, WLT_1x2_GAO and WDHT_1x2_GAO) and fixed payload (DPTHDI [88] and DGTDHS [129]) based schemes, respectively. Apart from the DPTHDI [88] and DGTDHS [129], rest of the schemes mentioned above are supporting varying payload in the range of 0.5 – 3 bpB and computes average PSNR from twenty color images (fig. 1.1). In contrast to WST_1x2, the improvement of average PSNR in WST_1X2_GAO method can be visually perceived as the payload increases from 1 bpB. Compared to WDHT_1X2_GAO, WLT_1X2_GAO, WBT_1X2_GAO, WST_2X2_GAO and WST_1x2, the WST_1X2_GAO method ensured less degradation in fidelity with respect to variable payload. The average PSNR for DPTHDI [88] is obtained as 37.40 dB by averaging the PSNR values of “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” images at 0.25 bpB of payload. In contrast to DPTHDI [88], the average PSNR of WST_1x2_GAO ensured equal or higher PSNR (dB) at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB which is obtained by taking the averages of PSNR values for “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” images at 1 bpB of payload. In comparison with DGTDHS [129], the average PSNR of WST_1x2_GAO offered higher PSNR (dB) at 1 bpB while the latter scheme also supports fabrication of secret information with variable payload. The average PSNR values of WST_1x2_GAO also yields the formation of high-fidelity watermarked images since the obtained values are greater than or equal to 30 dB at respective payload [148].

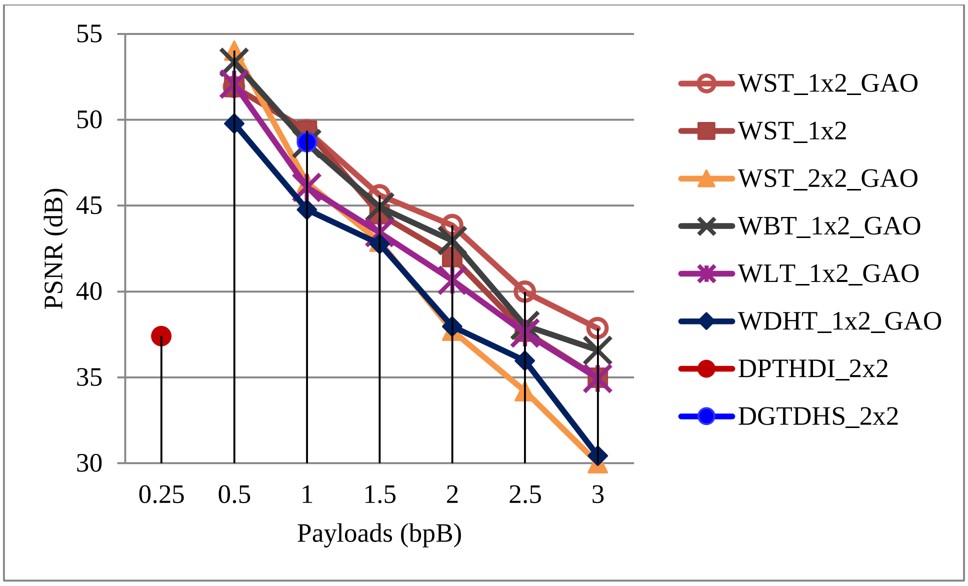


Fig. 8.16. Graphical representation of variation of average PSNR (dB) with respect to payload for WDHT_1X2_GAO, WLT_1X2_GAO, WBT_1X2_GAO, WST_2X2_GAO, WST_1x2 and WST_1X2_GAO and Varsaki et al.'s (DPTHDI [88] and DGTDHS [129]) schemes

8.3.5. Optimization of G-lets D4 domain based watermarking

In chapter 6, two watermarking schemes such as 2 x 2 block based watermark fabrication (WGD4_2x2) and 1 x 2 block based watermark fabrication in G-lets D4 domain (WGD4_1x2) has already been discussed.

In section 8.3.5.1, the 2 x 2 block based watermark fabrication using G-lets D4 domain (WGD4_2x2) scheme is extended as WGD4_2X2_GAO scheme by introducing the genetic algorithm (GA) based optimization of section 8.2 to minimize the quality distortion of the watermarked image. The optimization does not work for 1 x 2 block based watermark fabrication using G-lets D4 domain i.e., WGD4_1x2 since the Vector Co-ordinate Area (VCA) corresponding to each embedded component in transform domain should be unaltered as it plays the important role for embedding as well as extraction of secret information in exploiting modification direction (EMD) based scheme.

8.3.5.1. Optimization of 2 x 2 Block based Watermark Fabrication

The WGD4_2x2_GAO decomposes the cover image into 2 x 2 sub-blocks, adjusts the values of the pixel components (as discussed in equation (2.8) of section 2.2.1.1) and then converts each 2 x 2 sub-block into transform domain based on based on group of linear transformations for dihedral group of order 4 (G-lets D4). It has been discussed in section 6.2.1 that the secret bits corresponding to message digest, size and content of the watermark are fabricated on each transformed component starting from the least significant bit position (i.e., LSB-0) toward higher order bit position. GA optimization is done to search the existence of optimized component which provides reduced quality distortion by preserving the fabricated bits of the embedded component. Inverse transform in G-lets D4 domain is applied over 2 x 2 sub-matrices of optimized components to obtain the pixel components. The process is repeated till the secret information is concealed and the watermarked image is produced.

The detailed discussion regarding the process of embedding is explained with a suitable example as given in section 8.3.5.1.1. Performance of the WGD4_2x2_GAO is computed, analyzed and then compared to prove the effectiveness which in turn is elaborated in section 8.3.5.1.2.

8.3.5.1.1. Example

The block based decomposition of the carrier image yields three 2 x 2 sub-matrices corresponding to red (R_1), green (G_1) and blue (B_1) channels. The pixel adjustment process adjusts the components since the occurrence of overflow/underflow is not allowed. For example, the 3 bpB of payload ensured the following adjustments of the pixel components:

$$R_1 = \begin{bmatrix} 230 & 72 \\ 17 & 155 \end{bmatrix}\ G_1 = \begin{bmatrix} 62 & 215 \\ 56 & 22 \end{bmatrix}\ B_1 = \begin{bmatrix} 111 & 172 \\ 251 & 7 \end{bmatrix}$$

Let, the pseudo random number r = (7809650151167322995 % 8) +1 = 4 is obtained from the secret key (K) = “skghosal”. On application of forward transform in G-lets D4 domain, eight different sets of 2 x 2 sub-matrices of transformed components corresponding to red, green and blue channels are constructed.

Since, r = 4, G_4 i.e., rotation matrix R_3 is chosen. Thus the 2 x 2 transformed sub-matrices corresponding to R/G/B channel is chosen for embedding as given below:

$$T(R_1) = \begin{bmatrix} 17 & 155 \\ -230 & -72 \end{bmatrix}\ T(G_1) = \begin{bmatrix} 56 & 22 \\ -62 & -215 \end{bmatrix}\ T(B_1) = \begin{bmatrix} 251 & 7 \\ -111 & -172 \end{bmatrix}$$

In this example, secret bit stream *“011001100110010011001111011011000110”* is to be embedded into first/second/third/fourth transformed component with an embedding ratio 3: 3: 3: 3 (i.e., $\lambda_1 = 3$, $\lambda_2 = 3$, $\lambda_3 = 3$, $\lambda_4 = 3$) for a payload value of 3 bpB. The order of embedding depends on the sign of the component. If the sign is positive then embed bits in usual order; otherwise, embed bits in reverse order. Hence, the 2 x 2 sub-matrices of embedded components corresponding to red, green and blue channels are obtained as follows:

$$T'(R_1) = \begin{bmatrix} 22 & 156 \\ -228 & -78 \end{bmatrix} T'(G_1) = \begin{bmatrix} 58 & 22 \\ -57 & -215 \end{bmatrix} T'(B_1) = \begin{bmatrix} 254 & 6 \\ -104 & -174 \end{bmatrix}$$

Genetic algorithm (GA) based optimization has been used to ensure the enhancement of quality without hampering the embedded bits. An optimized component is obtained against each embedded component which ensured minimum difference with respect to the pre-embedded component. The optimized 2 x 2 sub-matrices are as follows:

$$T''(R_1) = \begin{bmatrix} 14 & 156 \\ -228 & -70 \end{bmatrix} T''(G_1) = \begin{bmatrix} 58 & 22 \\ -65 & -215 \end{bmatrix} T''(B_1) = \begin{bmatrix} 254 & 6 \\ -112 & -174 \end{bmatrix}$$

Inverse transform in G-lets D4 domain is applied on 2 x 2 sub-matrices of optimized components to re-compute the pixel components. Hence, the 2 x 2 sub-matrices of pixel components are obtained as follows:

$$R'_1 = \begin{bmatrix} 14 & 156 \\ 228 & 70 \end{bmatrix} G'_1 = \begin{bmatrix} 58 & 22 \\ 65 & 215 \end{bmatrix} B'_1 = \begin{bmatrix} 254 & 6 \\ 112 & 174 \end{bmatrix}$$

It has been observed that the modified pixel components for each 2 x 2 matrices are non-fractional, non-negative and less than or equal to 255.

8.3.5.1.2. Results and Discussions

Color images [130, 131] of dimension 512 x 512 are considered for the fabrication of the variable sizes of watermark (Gold-Coin) as given in fig. 1.1. The fabrication is done in such an embedding ratio that the trade-off between the image quality and the payload is preserved. Following image quality metrics has been used to verify the perceptual difference between the original and watermarked images: peak signal to noise ratio (PSNR), mean squared error (MSE), image fidelity (IF), structural similarity index (SSIM) and universal image quality index (UIQ) respectively. The spreading of PSNR and MSE values with respect to the extreme payload values of 0.5 and 3 bpB ensured the following ranges [54.19 dB (Lena) – 39.11 dB (Desert)] and [0.24 (Lena) – 7.97 (Desert)] respectively. The quality of the watermarked images is highly perceptible due to the ability of offering high PSNR even at high payload (3 bpB) [148]. Other metrics such as the IF, SSIM and UIQ are ranges from [0, 1] however, the WGD4_2x2_GAO offered the following ranges [0.998829 (Desert), 0.999992 (Airplane)], [0.974394 (Splash), 0.999911 (San Diego)] and [0.717551 (Bobcat), 0.999334 (San Diego)] with respect to payload values of 3 and 0.5 bpB respectively. These values get closer to one yields higher similarity between watermarked and original images. The average results are also computed from given benchmark images (fig. 1.1) to summarize the experimental results.

Table 8.17. PSNR, MSE, IF, SSIM, UIQ for the carrier/cover images of dimension 512 x 512 with respect to varying payload in WGD4_2x2_GAO scheme

Images	**Payload (bpB)**	**PSNR (dB)**	**MSE**	**IF**	**SSIM**	**UIQ**
Lena	0.5	54.196500	0.247417	0.999984	0.999686	0.994808
	1.0	51.180434	0.495492	0.999968	0.999177	0.990166
	1.5	48.084170	1.010795	0.999936	0.998319	0.978570
	2.0	46.305846	1.522284	0.999903	0.997494	0.968478
	2.5	42.647414	3.534589	0.999776	0.994291	0.933714
	3.0	40.675582	5.565729	0.999648	0.990569	0.903624
Baboon	0.5	54.156000	0.249735	0.999986	0.999886	0.999022
	1.0	51.150645	0.498902	0.999973	0.999666	0.998044
	1.5	48.089079	1.009653	0.999946	0.999367	0.995783
	2.0	46.314292	1.519326	0.999919	0.999075	0.993692
	2.5	42.619487	3.557390	0.999811	0.997952	0.986074
	3.0	40.673029	5.569002	0.999705	0.996703	0.979093
Pepper	0.5	54.034890	0.256797	0.999978	0.999440	0.990092
	1.0	51.056199	0.509871	0.999957	0.998449	0.985853
	1.5	47.781398	1.083778	0.999908	0.996406	0.975437
	2.0	45.949067	1.652622	0.999859	0.994133	0.966926
	2.5	42.021131	4.082885	0.999650	0.987242	0.937369
	3.0	39.916257	6.629100	0.999426	0.979510	0.911108
Airplane	0.5	54.120917	0.251761	0.999992	0.999626	0.983152
	1.0	51.115621	0.502942	0.999985	0.999045	0.969501
	1.5	48.084328	1.010758	0.999971	0.998014	0.941323
	2.0	46.320115	1.517290	0.999956	0.996958	0.918825
	2.5	42.628580	3.549950	0.999898	0.993249	0.852774
	3.0	40.667840	5.575660	0.999840	0.988013	0.805172
Sailboat	0.5	54.114845	0.252113	0.999987	0.999740	0.995608
	1.0	51.125217	0.501832	0.999974	0.999321	0.992255
	1.5	48.068289	1.014498	0.999948	0.998629	0.982660
	2.0	46.297561	1.525190	0.999922	0.997963	0.974539
	2.5	42.569885	3.598253	0.999818	0.995413	0.947939
	3.0	40.626210	5.629364	0.999715	0.992357	0.925721
Earth	0.5	54.194826	0.247512	0.999985	0.999764	0.997682
	1.0	51.165703	0.497175	0.999970	0.999380	0.995612
	1.5	48.094869	1.008308	0.999940	0.998742	0.990129
	2.0	46.314677	1.519191	0.999909	0.998104	0.985030

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	42.621596	3.555664	0.999789	0.995593	0.965876
	3.0	40.680935	5.558873	0.999670	0.992916	0.948186
San Diego	0.5	54.188650	0.247865	0.999990	0.999911	0.999334
	1.0	51.168158	0.496894	0.999981	0.999770	0.998738
	1.5	48.095050	1.008266	0.999962	0.999516	0.997196
	2.0	46.315241	1.518994	0.999943	0.999278	0.995846
	2.5	42.633972	3.545546	0.999867	0.998361	0.990805
	3.0	40.675930	5.565283	0.999792	0.997320	0.986029
Splash	0.5	54.058783	0.255388	0.999977	0.999176	0.966074
	1.0	51.077490	0.507377	0.999955	0.997618	0.949486
	1.5	47.850287	1.066722	0.999905	0.994534	0.916875
	2.0	46.067519	1.608156	0.999856	0.991736	0.891118
	2.5	42.176479	3.939421	0.999645	0.983204	0.817235
	3.0	40.180035	6.238451	0.999437	0.974394	0.762757
Oakland	0.5	54.129326	0.251274	0.999986	0.999804	0.998596
	1.0	51.128024	0.501508	0.999973	0.999479	0.997538
	1.5	47.983759	1.034437	0.999944	0.998900	0.994736
	2.0	46.156534	1.575531	0.999914	0.998349	0.992142
	2.5	42.414450	3.729368	0.999796	0.996349	0.982495
	3.0	40.444980	5.869246	0.999678	0.994024	0.973121
Foster City	0.5	54.186913	0.247964	0.999991	0.999578	0.993367
	1.0	51.160064	0.497821	0.999982	0.998923	0.988152
	1.5	48.093774	1.008562	0.999964	0.997693	0.972649
	2.0	46.305599	1.522370	0.999945	0.996496	0.958687
	2.5	42.658803	3.525332	0.999874	0.992149	0.911265
	3.0	40.714174	5.516490	0.999803	0.987268	0.869567
Anhinga	0.5	52.448813	0.369997	0.999971	0.999895	0.882825
	1.0	49.741529	0.690124	0.999947	0.999654	0.885574
	1.5	47.875708	1.060497	0.999918	0.998962	0.868789
	2.0	46.449414	1.472783	0.999886	0.997968	0.864767
	2.5	43.152331	3.146640	0.999758	0.996879	0.857702
	3.0	41.347886	4.767510	0.999633	0.993769	0.836713
Athens	0.5	51.855675	0.424143	0.999966	0.999907	0.971388
	1.0	49.354203	0.754501	0.999939	0.999683	0.974436
	1.5	47.903793	1.053661	0.999915	0.998943	0.955320
	2.0	46.629618	1.412923	0.999887	0.997869	0.950590

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	43.588696	2.845841	0.999772	0.996749	0.935107
	3.0	41.982122	4.119724	0.999670	0.993871	0.913863
Bardowl	0.5	52.469129	0.368270	0.999963	0.999885	0.999054
	1.0	49.895188	0.666133	0.999933	0.999614	0.998920
	1.5	47.794861	1.080423	0.999892	0.999438	0.996530
	2.0	46.311044	1.520463	0.999847	0.998898	0.994754
	2.5	42.844689	3.377624	0.999661	0.998010	0.989086
	3.0	41.071583	5.080682	0.999490	0.996592	0.982953
Barnfall	0.5	53.432516	0.295004	0.999955	0.999812	0.998833
	1.0	50.605045	0.565686	0.999913	0.999444	0.998217
	1.5	47.842250	1.068698	0.999830	0.998937	0.994826
	2.0	46.141132	1.581128	0.999748	0.998101	0.992990
	2.5	42.546340	3.617814	0.999418	0.996543	0.982041
	3.0	40.601030	5.662096	0.999091	0.993297	0.973719
Butrfly	0.5	52.335775	0.379754	0.999972	0.999887	0.995565
	1.0	49.745587	0.689479	0.999950	0.999630	0.995474
	1.5	47.976468	1.036176	0.999926	0.999193	0.989418
	2.0	46.611337	1.418883	0.999898	0.998370	0.986127
	2.5	43.324613	3.024258	0.999785	0.997295	0.976058
	3.0	41.587503	4.511594	0.999679	0.995466	0.966491
Bobcat	0.5	52.050556	0.405531	0.999942	0.999737	0.749630
	1.0	49.505535	0.728663	0.999897	0.999076	0.750896
	1.5	47.949462	1.042639	0.999853	0.998534	0.740996
	2.0	46.633371	1.411702	0.999799	0.996703	0.737451
	2.5	43.517248	2.893047	0.999593	0.995560	0.739420
	3.0	41.817986	4.278402	0.999398	0.992286	0.717551
Bodie	0.5	54.002689	0.258708	0.999957	0.999882	0.980279
	1.0	51.081650	0.506891	0.999916	0.999619	0.979328
	1.5	47.705236	1.102952	0.999813	0.999253	0.974879
	2.0	45.887266	1.676307	0.999714	0.998554	0.973595
	2.5	42.223046	3.897406	0.999343	0.997598	0.960671
	3.0	40.257182	6.128611	0.998968	0.994617	0.952976
Bluheron	0.5	53.637012	0.281435	0.999965	0.999752	0.996472
	1.0	50.769326	0.544687	0.999933	0.999141	0.995349
	1.5	48.251283	0.972639	0.999880	0.998707	0.989857
	2.0	46.687083	1.394350	0.999829	0.997781	0.986539

Images	Payload (bpB)	PSNR (dB)	MSE	IF	SSIM	UIQ
	2.5	41.895018	4.203184	0.999485	0.995906	0.961584
	3.0	39.579176	7.164118	0.999123	0.987958	0.945879
Colomtn	0.5	53.872445	0.266585	0.999979	0.999826	0.985477
	1.0	50.917212	0.526452	0.999958	0.999502	0.985171
	1.5	48.010593	1.028065	0.999919	0.999011	0.979382
	2.0	46.247529	1.542863	0.999879	0.998332	0.976992
	2.5	42.621546	3.555704	0.999721	0.996780	0.962936
	3.0	40.633909	5.619393	0.999560	0.993451	0.954587
Desert	0.5	53.587786	0.284643	0.999959	0.999841	0.998602
	1.0	50.670424	0.557234	0.999920	0.999565	0.998393
	1.5	47.366850	1.192326	0.999826	0.999113	0.996226
	2.0	45.432909	1.861184	0.999727	0.998399	0.994821
	2.5	41.309350	4.810002	0.999296	0.996383	0.985180
	3.0	39.116142	7.970142	0.998829	0.992510	0.978885
Average case	0.5	53.553700	0.292095	0.999974	0.999752	0.973793
	1.0	50.680660	0.561983	0.999951	0.999288	0.971355
	1.5	47.945080	1.044693	0.999910	0.998511	0.961579
	2.0	46.268860	1.538677	0.999867	0.997528	0.955195
	2.5	42.600730	3.599496	0.999688	0.995075	0.933767
	3.0	40.662470	5.650974	0.999508	0.991345	0.914400

A comparative study of PSNR versus payload has been made between WGD4_2x2 and the WGD4_2X2_GAO in terms of average PSNR at variable payloads. Table 8.18 demonstrated the improvement in average PSNR values for the proposed optimized technique (WGD4_2X2_GAO) over the WGD4_2x2 method is observed as the payload exceeds 1 bpB of payload. For the payload variation (1.5 – 3 bpB), the average PSNR enhancement (in actual sense quality) in WGD4_2X2_GAO over the WGD4_2X2 are summarized as follows: 2.75, 3.39, 4.31 and 4.67 dB respectively.

Table 8.18. Comparative analysis of obtained average PSNR values between WGD4_2x2 and WGD4_2x2_GAO with respect to increasing payload

WGD4_2x2		WGD4_2x2_GAO	
Payload (bpB)	PSNR (dB)	Payload (bpB)	PSNR (dB)
0.5	53.5537	0.5	53.5537
1.0	50.68066	1.0	50.68066
1.5	45.19256	1.5	47.94508

WGD4_2x2		WGD4_2x2_GAO	
Payload (bpB)	**PSNR (dB)**	**Payload (bpB)**	**PSNR (dB)**
2.0	42.87705	2.0	46.26886
2.5	38.29444	2.5	42.60073
3.0	35.99861	3.0	40.66247

Comparison is made among the 2 x 2 block based watermarking (WGD4_2X2_GAO), Discrete Pascal Transform based data hiding scheme (DPTHDI) [88] and Discrete Gould Transform based data hiding scheme (DGTDHS) [129] in terms of PSNR (dB) and payload (bpB). It is observed from fig. 8.17 that the PSNR of the watermarked images for DPTHDI [88] and DGTDHS [129] are obtained at 0.25 and 1 bpB respectively. The major limitation of the DPTHDI [88] and DGTDHS [129] are their fixed as well as low payload. Unlikely, the WGD4_2X2_GAO is focused on the payload variation from 0.5 to 3 bpB and provides acceptable visual imperceptibility by retaining the PSNR values above 30 dB [148]. In contrast to DPTHDI [88], the WGD4_2X2_GAO ensured equal or higher PSNR (dB) at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for “Lena”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB for “Baboon”, 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads for “Pepper”, 0.5, 1, 1.5, 2 and 2.5 bpB of payloads for “Airplane” and 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payload for “Sailboat” respectively. Compared to DGTDHS [129], the WGD4_2X2_GAO ensured higher PSNR at 1 bpB of payload for “Lena”, “Baboon”, “Airplane” and “Sailboat”, respectively.

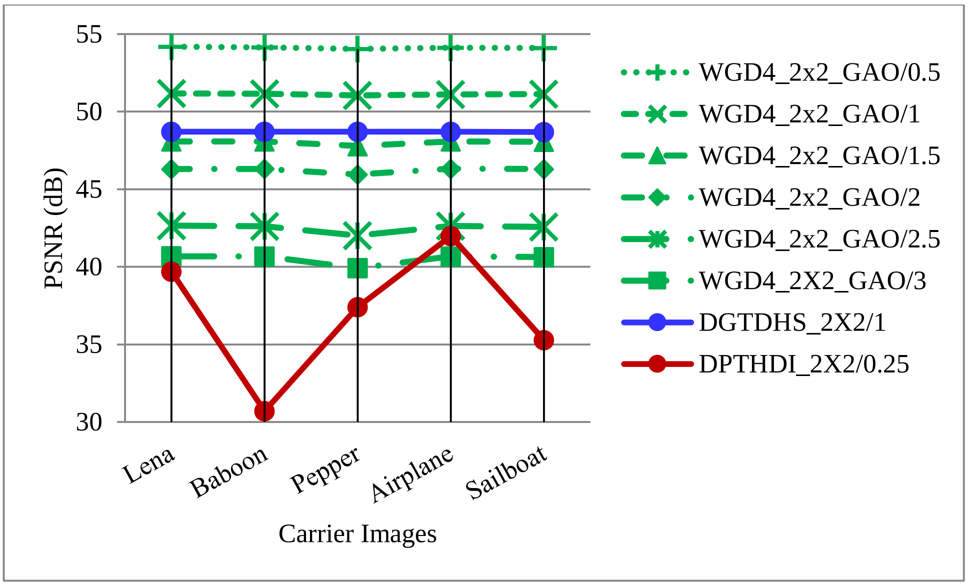


Fig. 8.17. Performance analysis of PSNR (dB) for variable payload based WGD4_2x2_GAO and fixed payload based Varsaki et al.’s (DPTHDI [88] and DGTDHS [129]) schemes with respect to five color images

Generally, the peak signal to noise ratio (PSNR) and payload are two most important parameters to analyze the performance of watermarking techniques. Fig. 8.18 illustrates the average PSNR analysis among WGD4_2x2_GAO, WGD4_2X2, WST_2X2_GAO, WBT_2X2_GAO, WLT_2X2_GAO, WDHT_2X2_GAO, DPTHDI [88] and DGTDHS [129], respectively. In contrast to WGD4_2x2 method, the linear improvement of PSNR in WGD4_2X2_GAO method can be visually perceived as the payload increases from 1 bpB. Compared to other optimized techniques such as the WST_2X2_GAO, WBT_2X2_GAO, WLT_2X2_GAO and WDHT_2X2_GAO, the superiority of the WGD4_2X2_GAO has also been evident from the line chart. The average PSNR for DPTHDI [88] is 37.40 dB by considering the average of PSNR values for "Lighthouse", "Elaine", "Lenna", "Boat" and "F16" images at 0.25 bpB of payload. In contrast to DPTHDI [88], the average PSNR of WGD4_2x2_GAO ensured equal or higher PSNR (dB) at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. The average PSNR for DGTDHS [129] is 48.70 dB by taking the average of PSNR values for "Lenna", "Babboon", "Peppers", "Tiffany", "F16" and "Sailboat" images at 1 bpB of payload. In comparison with DGTDHS [129], the average PSNR of WGD4_2x2_GAO scheme ensured equal or higher PSNR (dB) at 0.5 and 1 bpB of payloads. The WGD4_2x2_GAO provides an average PSNR of greater than 40 dB at 3 bpB which ensured that the quality of the watermarked images is very good and preserves finer details of the images [148].

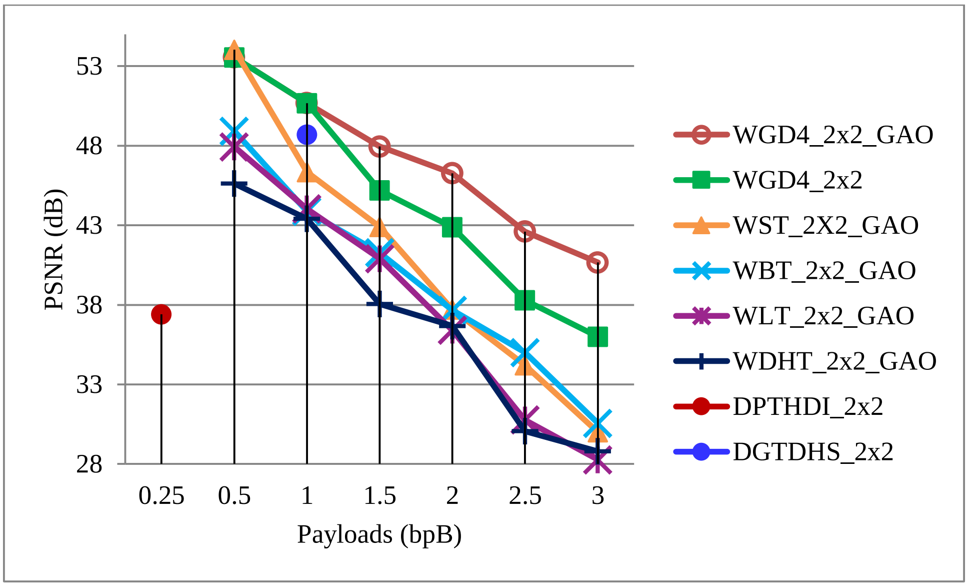


Fig. 8.18. Graphical representation of variation of average PSNR (dB) with respect to payload for WGD4_2x2_GAO, WGD4_2x2, WST_2x2_GAO, WBT_2x2_GAO, WLT_2x2_GAO, WDHT_2x2_GAO and Varsaki et al.'s DPTHDI [88] and DGTDHS [129] schemes

8.4. Salient Features

Genetic Algorithm (GA) based optimization of section 8.2 has been used to enhance the quality of the watermarked images by considering watermarking as an optimization problem. Perceptual distortion is drastically reduced for majority of the proposed watermarking schemes as the payload exceeds 1 bpB. The schemes are applied on each embedded component in such way that the concealed information is kept unaffected. Proposed optimization improves the quality of the watermarked images by keeping the payload unchanged. The optimization may slow down the execution since a lot of intermediate bit-level manipulations are made during population, fitness calculation, cross-over and mutation phases; however, the computational cost has been compromised to achieve high fidelity watermarked images.

Chapter 9

Analysis and Discussions

A comparative analysis has been made among Discrete Hartley Transform (DHT), Legendre Transform (LT), Binomial Transform (BT), Stirling Transform (ST) and group of linear transformations for dihedral group of order 4 (G-lets D4) based fragile watermarking schemes which are described in details at chapter 2, 3, 4, 5 and 6 respectively. In addition, the watermarking techniques followed by quality enhancement of chapter 7 or genetic algorithm (GA) based optimization of chapter 8 has also been analyzed and discussed to investigate the quality improvement. The comparative analysis has also been done to ensure the effectiveness of the proposed techniques over the existing schemes viz. Varsaki et al.’s DPTHDI [88], Lin et al.’s [87] scheme, Yang et al.’s [109] scheme and Varsaki et al.’s DGTDHS [129] in terms of payload and/or quality of the watermarked images. In this pretext, the quality of the watermarked images is primarily analyzed by means of average peak signal to noise ratio (PSNR). The average PSNR for proposed schemes are obtained based on twenty benchmark color images [130, 131] with payload from 0.5 to 3 bpB. In order to achieve variable payload, variable sizes of the watermark are embedded into the benchmark (BMP) images [130, 131] as shown in fig. 1.1. The average PSNR for DPTHDI [88] is 37.40 dB which is obtained by taking the average of PSNR values of six color images such as “Lenna”, “Babboon”, “Peppers”, “Tiffany”, “F16” and “Sailboat” at fixed payload of 0.25 bpB. The average PSNR for Lin et al.’s [87] scheme is 33.23 dB where the average PSNR values are computed for the gray-scale images viz. “Boats”, “F16”, “Mandrill”, “Lena” and “Pepper” at an average payload of 0.34375 bpB. The average PSNR for Yang et al.’s [109] scheme is 49.48 dB as computed by averaging the PSNR values of “Lena”, “Jet” and “Baboon” images at 0.3 bpB of average payload in case of gray-scale images. The average PSNR for DGTDHS [129] is 48.70 dB which is obtained by taking the average of PSNR values at 1 bpB of fixed payload for five images such as “Lighthouse”, “Elaine”, “Lenna”, “Boat” and “F16” respectively. Results obtained from all these schemes in transform domain are compared and validated by extensive analysis.

In fig 9.1, the performance of the proposed schemes based on variable payload (WDHT_2x2, WDHT_1x2, WLT_2x2, WLT_1x2, WBT_2x2, WBT_1x2, WST_2x2, WST_1x2, WGD4_2x2 and WGD4_1x2) are compared against existing schemes (DPTHDI [88], Lin et al.’s method [87], Yang et al.’s method [109] and DGTDHS [129]) to investigate the visual clarity of the watermarked images. In contrast to DPTHDI [88], Lin et al.’s [87], Yang et al.’s [109] and DGTDHS [129] schemes, the proposed schemes have been designed and implemented for a variable payload of 0.5 to 3 bpB. But the available payload of the existing schemes are fixed as well as consisting of low fabrication density. Proposed schemes (WDHT_2x2, WDHT_1x2, WLT_2x2, WLT_1x2, WBT_2x2, WBT_1x2, WST_2x2, WST_1x2, WGD4_2x2 and WGD4_1x2) provide improved results in terms of average PSNR and payload over DPTHDI [88] and Lin et al.'s [87] schemes. Compared to the average PSNR of 37.4 dB of DPTHDI [88], equal or higher average PSNR values are obtained for the following listed schemes at the specified payload ranges: WDHT_2x2, WDHT_1x2, WLT_2x2, WBT_2x2 and WST_2x2 [0.5 – 1.5 bpB], WLT_1x2 and WBT_1x2 [0.5 – 2 bpB] and WST_1X2, WGD4_2X2 and WGD4_1X2 [0.5 – 2.5 bpB] respectively. The minimum payload of the proposed schemes is double (i.e., 0.5 bpB) than that of DPTHDI [88] and offered an average PSNR enhancement of 8 dB at that payload which is considered as the significant level of quality improvement. In contrast to Lin et al.'s [87] scheme, equal or higher perceptual transparency (in terms of average PSNR) is achieved for WDHT_2x2, WLT_2x2 and WBT_2x2 at (0.5 – 2 bpB), for WST_2x2, WDHT_1x2 and WLT_1x2 at (0.5 – 2.5 bpB) and for rest of the proposed schemes at (0.5 – 3 bpB) respectively. Similarly, the proposed schemes revealed an average PSNR improvement of 12.39 dB over Lin et al.'s [87]; at the same time, the average payload increases from 0.3 to 0.5 bpB. In comparison with Yang et al.'s [109] scheme, the WDHT_1x2, WLT_1x2, WBT_1x2, WST_2x2 and WST_1x2 schemes provided an improved average PSNR at 0.5 bpB while the WGD4_2x2 and WGD4_1x2 schemes offered improved average PSNR at 0.5 and 1 bpB respectively. The WBT_1x2, WST_1x2, WGD4_2x2 and WGD4_1x2 schemes offered better average PSNR than DGTDHS [129] at 1 bpB of payload. It is observed that some of the proposed schemes did not obtain good average PSNR over Yang et al.'s [109] and DGTDHS [129] schemes however, those schemes are effective as they are capable of fabrication of secret information in terms of variable payload up to a significant extent. In general, it is observed that the proposed schemes outperform the existing schemes in terms of average PSNR as well as the payload. In addition, proposed schemes (except WDHT_2x2 and WDHT_1x2) have also been compared among themselves in terms of average PSNR to investigate the quality degradation. The average PSNR is improved in a linear fashion with each scheme giving better result than the immediate preceding scheme in the order of sequence: WLT_2x2, WBT_2x2, WST_2x2, WLT_1x2, WBT_1x2, WST_1x2, WGD4_2x2 and WGD4_1x2 respectively. The WDHT_2x2 and WDHT_1x2 have been considered separately from the other proposed schemes as there are large variations of average PSNR with respect to odd/even multiples of 0.5 bpB of payload; as a result, the lines corresponding to these schemes overlap the lines of WLT_2x2, WBT_2x2 and WST_2x2 as evident from fig. 9.1.

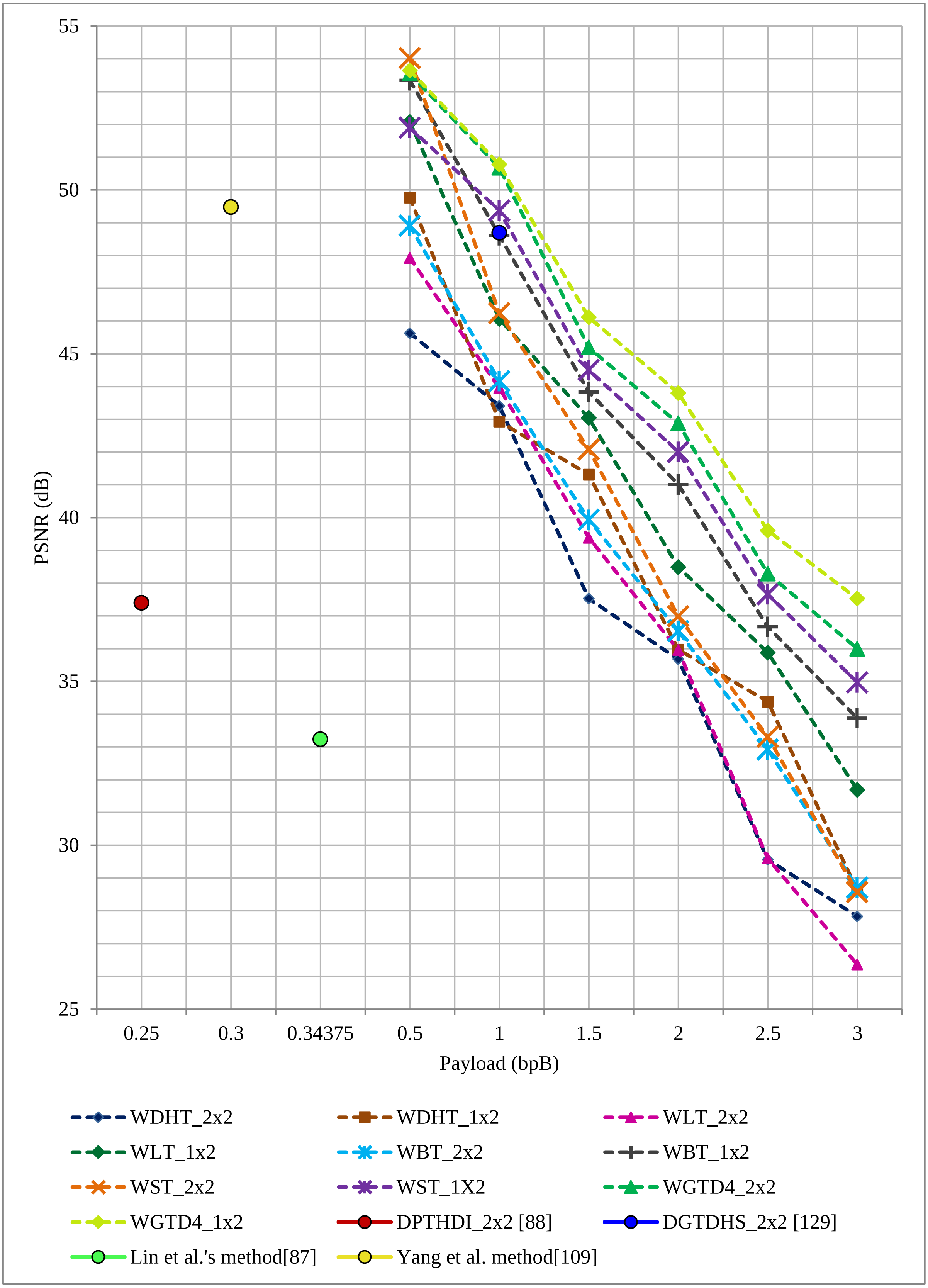


Fig. 9.1. Comparative analysis of PSNR (dB) with respect to payload (bpB) among WDHT_2x2, WDHT_1x2, WLT_2x2, WLT_1x2, WBT_2x2, WBT_1x2, WST_2x2, WST_1x2, WGD4_2x2, WGD4_1x2, Varsaki et al.'s DPTHDI [88], Lin et al.'s [87], Yang et al.'s [109] and Varsaki et al.'s DGTDHS [129] schemes

In fig 9.2., a high-low chart is drawn where, the high and low payload values for the proposed schemes viz. WDHT_2x2, WDHT_1x2, WLT_2x2, WLT_1x2, WBT_2x2, WBT_1x2, WST_2x2, WST_1x2, WGD4_2x2 and WGD4_1x2 are assigned as 3 and 0.5 bpB, respectively. In general, the quality of watermarked images sharply decreases as payload increases and vice-versa. The average PSNR of the proposed schemes lie in the range of 26.35 to 37.52 dB at high payload (3 bpB) whereas, 45.62 to 54.02 dB is achieved at low payload (0.5 bpB). Fig. 9.2 depicts that the WLT_1X2, WBT_1X2, WST_2X2, WST_1x2, WGD4_2x2 and WGD4_1x2 schemes offered an average PSNR of greater than or equal to 50 dB at low payload indicating very high quality watermarked images. However, the WDHT_2x2, WDHT_1x2, WLT_2x2, WBT_2x2 and WST_2x2 offered an average PSNR of less than 30 dB at high payload; but, rest of them offered an average PSNR of 30 dB or more at high payload. The average PSNR values falling below 30 dB indicate a fairly low quality whereas, values above 40 dB ensured almost invisible degradation [148]. Proposed schemes those are suffering from high distortion at high payload can be managed either by turned down the payload as 2.5 bpB or by incorporating a post-embedding quality enhancement/optimization scheme. Fig. 9.2 also depicts that 1 x 2 block based schemes viz. WDHT_1x2, WLT_1x2, WBT_1x2, WST_1x2 and WGD4_1x2 ensured higher values of average PSNR over 2 x 2 block based schemes viz. WDHT_2x2, WLT_2x2, WBT_2x2, WST_2x2 and WGD4_2x2 respectively.

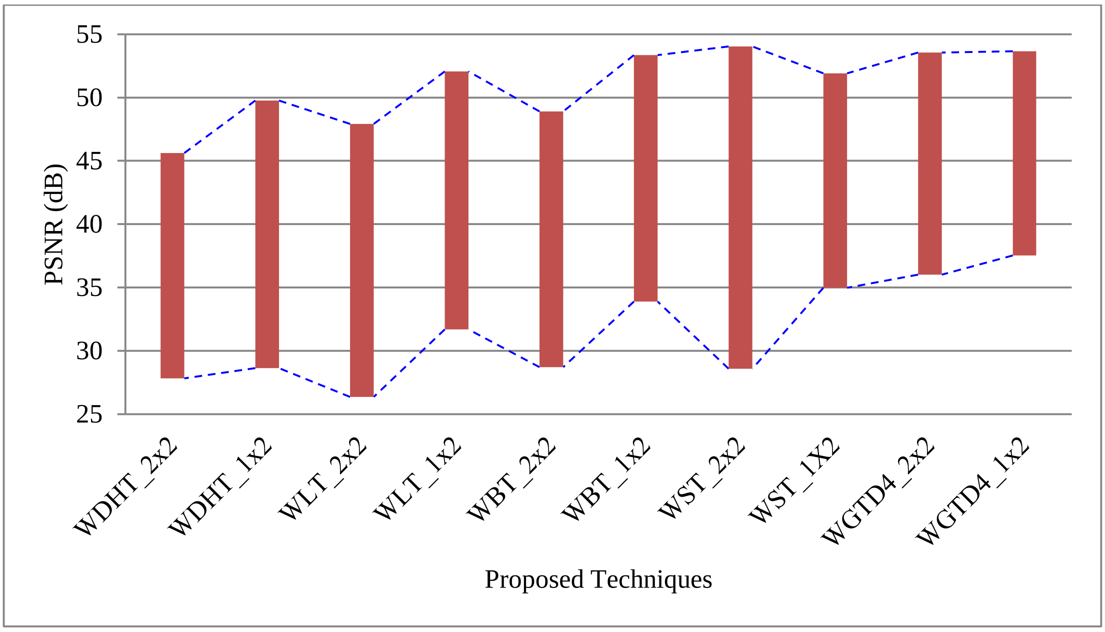


Fig. 9.2. Analysis of PSNR (dB) at high and low payloads (bpB) with respect to WDHT_2x2, WDHT_1x2, WLT_2x2, WLT_1x2, WBT_2x2, WBT_1x2, WST_2x2, WST_1x2, WGD4_2x2 and WGD4_1x2 schemes

In fig. 9.3, a scatter plot has been illustrated in which the independent variable payload is plotted along X-axis and the dependent variable average PSNR is plotted along Y-axis. In order to predict the average PSNR of WDHT_2x2, WDHT_1x2, WLT_2x2, WLT_1x2, WBT_2x2, WBT_1x2, WST_2x2, WST_1x2, WGD4_2x2 and WGD4_1x2 schemes, a linear regression analysis has been done where, the average PSNR of each scheme is plotted with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads. In other words, the data collected from ten different schemes are plotted to perform regression analysis which is used to understand the relationship between average PSNR and the payload. A total of 60 observations have been considered. The "blue" dots in the scatter plot depict the actual values of the average PSNR whereas, the predicted values of average PSNR is plotted through "red" dots. The regression line slopes downward from left to right which ensured a consistent result with the negative relationship between the average PSNR and payload. It is evident from the chart that no line can be found to pass through all points of the plot. Thus no functional relation exists between the two variables viz. average PSNR and payload. The coefficients of the regression line are 54.5728 (intercept) and -7.838 (slope) so that the line has the equation: $y = -7.838x + 54.57$. For instance, one can predict the average PSNR at 2 bpB of payload based on the above mentioned equation of straight line as $y = -7.838 \times 2 + 54.57 = 38.894$, or approximately 39 dB. Similarly, the predicted value of average PSNR at two extreme payload values (3 bpB and 0.5 bpB) has been computed which are around 31 dB and 51 dB, respectively. Since, the average PSNR values falling below 30 dB indicate a fairly low quality watermarked images and values above 40 dB ensured almost invisible degradation [148], the predicted value of average PSNR is highly acceptable. However, with a systematic analysis of the data, three more regression statistics such as the Multiple R, R^2 and adjusted R^2 have also been computed. The value of Multiple R (correlation co-efficient) is 0.908281 which ensured a stronger linear relationship as the value of 1 (one) means perfect positive relationship. Multiple R (correlation coefficient) of 0.908281 also ensured that the prediction of average PSNR is considered quite sure and almost accurate. An R^2 value of 0.825 ensured that 82.5 % of the points fall on the regression line or in other words, 82.5 % values of average PSNR fit the model. Adjusted $R^2 = 0.8219$, which means that the independent variable, payload, explains 82.19 % of the variability of the dependent variable, average PSNR, in the population.

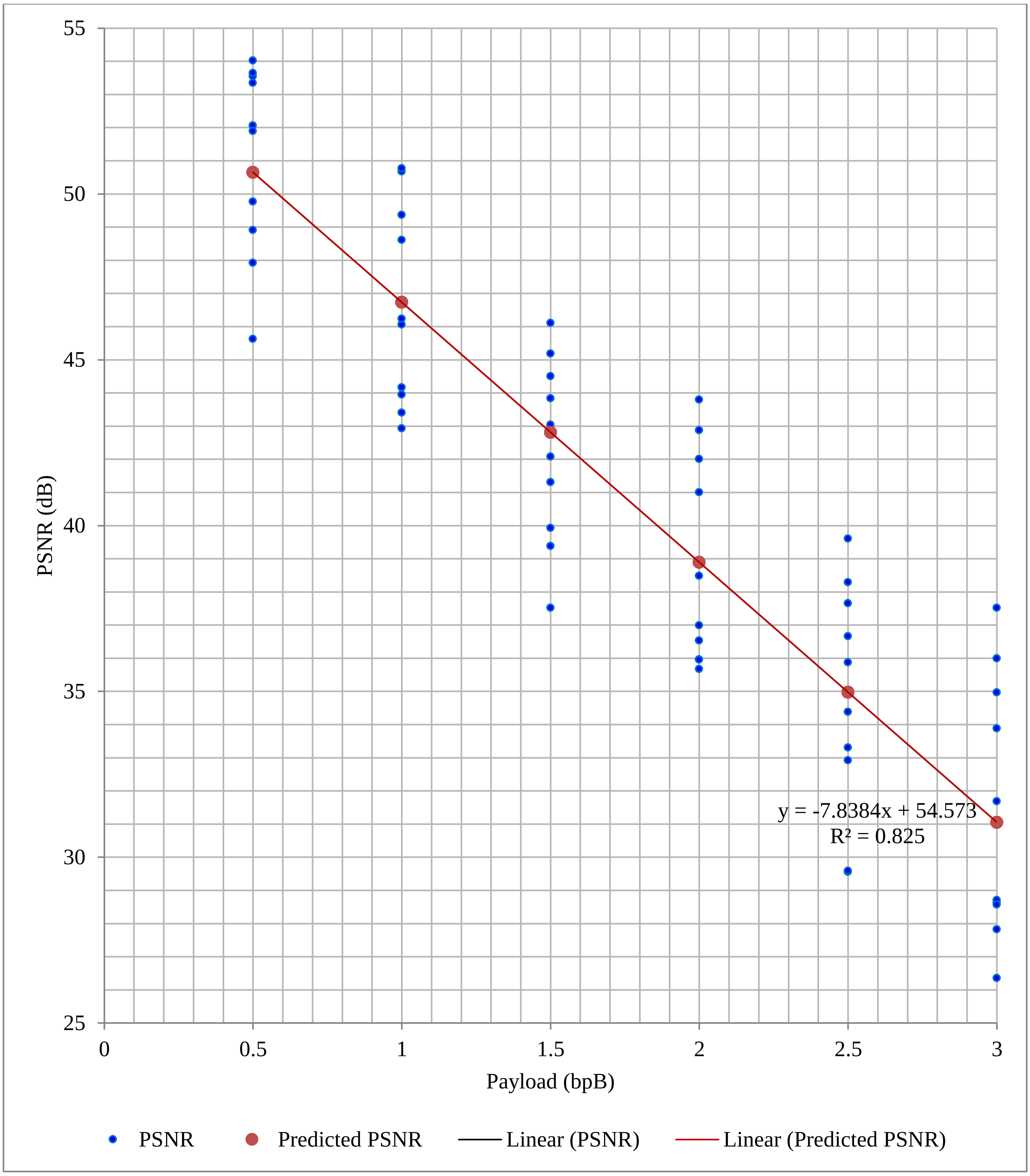


Fig. 9.3. Scatter diagram representing the regression analysis in terms of payload (bpB) and PSNR (dB) for proposed watermarking schemes without quality enhancement\GA optimization

In fig 9.4, the quality enhancement of the WDHT_2x2, WDHT_1x2, WLT_2x2, WLT_1x2, WBT_2x2, WBT_1x2, WST_2x2, WST_1x2, WGD4_2x2 and WGD4_1x2 schemes have been done by introducing the post-embedding quality enhancement which redefines the schemes as: WDHT_2x2_QE, WDHT_1x2_QE, WLT_2x2_QE, WLT_1x2_QE, WBT_2x2_QE, WBT_1x2_QE, WST_2x2_QE, WST_1x2_QE, WGD4_2x2_QE and WGD4_1x2_QE respectively. These schemes are compared against the fixed as well as low payload based existing schemes (Varsaki et al.'s DPTHDI [88], Lin et al.'s method [87], Yang et al.'s method [109] and Varsaki et al.'s DGTDHS [129]) as available in the literature, to investigate the quality of the watermarked images. In comparison with Varsaki et al.'s DPTHDI [88], Lin et al.'s [87] scheme, Yang et al.'s [109] scheme and Varsaki et al.'s DGTDHS [129], proposed schemes deal with improved average PSNR and variable payload (0.5 to 3 bpB). The average PSNR values of the proposed schemes lie above 37.4 dB of average PSNR of DPTHDI [88] for WDHT_2x2_QE and WLT_2x2_QE at 0.5, 1 and 1.5 bpB of payloads, for WST_2X2_QE, WBT_2X2_QE, WDHT_1X2_QE, WLT_1x2_QE and WBT_1x2_QE at 0.5, 1, 1.5 and 2 bpB of payloads and for the rest of the proposed schemes at 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads respectively. In comparison with Lin et al.'s [87] scheme, equal or higher average PSNR is obtained for WDHT_2x2_QE and WLT_2x2_QE with respect to the payload variation [0.5 – 2 bpB]; for WST_2X2_QE, WBT_2X2_QE, WDHT_1X2_QE and WLT_1X2_QE with respect to the payload variation [0.5 – 2.5 bpB]; and for the rest of the proposed schemes with respect to the payload variation [0.5 – 3 bpB] respectively. The WDHT_1x2_QE, WLT_1x2_QE, WBT_1x2_QE, WST_2x2_QE, WST_1x2_QE, WGD4_2x2_QE and WGD4_1x2_QE offered an improved average PSNR over Yang et al.'s [109] by keeping the payload enhancement of 0.2 bpB. Unlike to Varsaki et al.'s DGTDHS [129], the WBT_1x2_QE, WST_1x2_QE, WGD4_2x2_QE and WGD4_1x2_QE achieved higher average PSNR at 1 bpB of payload. It has been observed that some of the proposed schemes are not providing better average PSNR over Yang et al.'s scheme [109] and DGTDHS [129]; regardless of the fact, those schemes are useful as they are providing the variable payload for the range [0.5 – 3 bpB]. The quality enhancement is investigated in terms of average PSNR for the proposed watermarking schemes which is applicable for the payload range [1.5 – 3 bpB] as the PSNR values remains almost constant at 0.5 and 1 bpB for majority of the schemes. Lines representing the schemes overlap each other up to 2 bpB; afterward, the average PSNR is improved in a linear fashion with each scheme giving better result than the immediate preceding scheme in the order: WDHT_2X2_QE, WLT_2X2_QE, WST_2X2_QE, WBT_2X2_QE, WDHT_1X2_QE, WLT_1X2_QE, WBT_1X2_QE, WST_1X2_QE, WGD4_1X2_QE and WGD4_2X2_QE respectively.

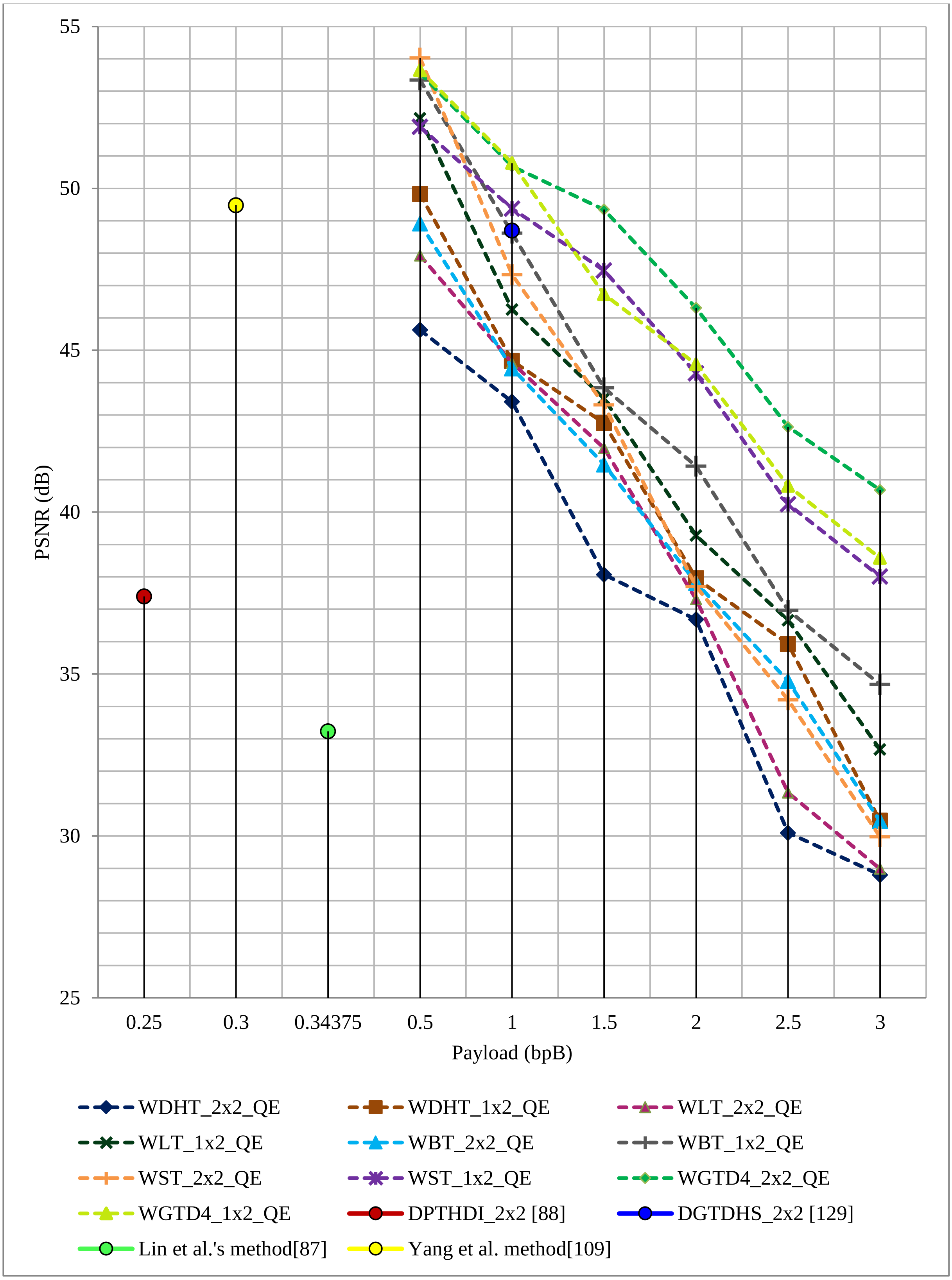


Fig. 9.4. Comparative analysis of PSNR (dB) with respect to payload (bpB) among WDHT_2x2_QE, WDHT_1x2_QE, WLT_2x2_QE, WLT_1x2_QE, WBT_2x2_QE, WBT_1x2_QE, WST_2x2_QE, WST_1x2_QE, WGD4_2x2_QE, WGD4_1x2_QE, Varsaki et al.'s DPTHDI [88], Lin et al.'s [87], Yang et al.'s [109] and Varsaki et al.'s DGTDHS [129] schemes

In fig 9.5., the average PSNR of WDHT_2x2_QE, WDHT_1x2_QE, WLT_2x2_QE, WLT_1x2_QE, WBT_2x2_QE, WBT_1x2_QE, WST_2x2_QE, WST_1x2_QE, WGD4_2x2_QE and WGD4_1x2_QE schemes are evaluated with respect to high (3 bpB) and low (0.5 bpB) payloads respectively. The average PSNR of the proposed schemes lie in the range of 28.79 to 40.68 dB at high payload whereas, 45.62 to 54.02 dB is achieved at low payload. However, it is needless to say that the average PSNR for WLT_1X2_QE, WBT_1X2_QE, WST_2X2_QE, WST_1x2_QE, WGD4_2x2_QE and WGD4_1x2_QE schemes offered an average PSNR of greater than or equal to 50 dB at low payload. It is also observed from the high-low chart of fig. 9.5 that the proposed schemes (except WDHT_2x2_QE and WLT_2x2_QE) offered an average PSNR of more than 30 dB. The WDHT_2x2_QE and WLT_2x2_QE schemes provide the average PSNR of 28.79 dB and 28.98 dB at high payload which are really very close to 30 dB. Generally, the average PSNR value above 30 dB is considered as a well perceptible watermarked image [148]. It is also seen from fig 9.5 that the 1 x 2 block based schemes viz. WDHT_1x2_QE, WLT_1x2_QE, WBT_1x2_QE, WST_1x2_QE and WGD4_1x2_QE are finer than 2 x 2 block based schemes viz. WDHT_2x2_QE, WLT_2x2_QE, WBT_2x2_QE, WST_2x2_QE and WGD4_2x2_QE schemes in terms of average PSNR with respect to variable payload that offers a spread from 0.5 to 3 bpB.

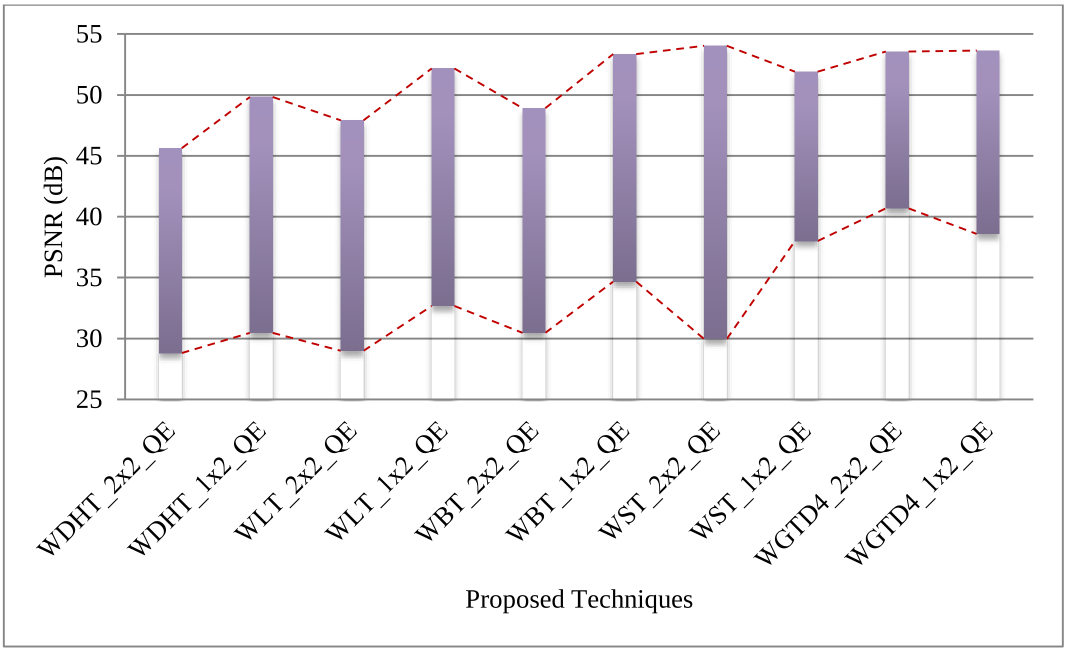


Fig. 9.5. Analysis of PSNR (dB) at high and low payloads (bpB) with respect to WDHT_2x2_QE, WDHT_1x2_QE, WLT_2x2_QE, WLT_1x2_QE, WBT_2x2_QE, WBT_1x2_QE, WST_2x2_QE, WST_1x2_QE, WGD4_2x2_QE and WGD4_1x2_QE schemes

In fig. 9.6, a scatter plot is illustrated to predict the average PSNR with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads through a regression analysis. In this scatter plot, the payload is plotted along X-axis and the average PSNR is plotted along Y-axis. The data collected from ten different schemes such as WDHT_2x2_QE, WDHT_1x2_QE, WLT_2x2_QE, WLT_1x2_QE, WBT_2x2_QE, WBT_1x2_QE, WST_2x2_QE, WST_1x2_QE, WGD4_2x2_QE and WGD4_1x2_QE are plotted to do a regression analysis which can be used to understand the relationship between the parameters: average PSNR and the payload. A total of 60 observations have been considered where, the "blue" dots depict the actual values of the average PSNR and the predicted values of average PSNR is plotted through "red" dots. The regression line slopes downward from left to right which ensured a consistent result with the negative relationship between the average PSNR and payload. Since, none of the line has been passed through all points of the plot, no functional relationship is found between the average PSNR and payload. The coefficients of the regression line are 54.42574 (intercept) and -7.101 (slope) so that the line has the equation: y = -7.838x + 54.57. As a consequence, one can predict the average PSNR at 2 bpB as y = -7.101 × 2 + 54.42 = 40.21, or 40 dB (approximately). Similarly, the predicted value of average PSNR at 3 bpB and 0.5 bpB has been computed which are around 33 dB and 51 dB, respectively. Since, the average PSNR values falling below 30 dB indicate a fairly low quality watermarked images and values above 40 dB ensured almost invisible degradation [148], the predicted value of average PSNR is highly acceptable. However, with a systematic analysis of the data, two more regression statistics such as the Multiple R, R^2 and adjusted R^2 has also been computed. The value of Multiple R (correlation co-efficient) is 0.875421 which ensure a stronger linear relationship as the value of 1 means perfect positive relationship. Since the correlation coefficient (Multiple R) is 0.875421, the prediction of average PSNR is considered reasonably definite and almost correct. An R^2 value of 0.7663 ensure that 76.63 % of the points fall on the regression line or in other words, 76.63 % values of average PSNR fit the model. Adjusted R^2 = 0.7623, which means that the independent variable, payload, explains 76.23 % of the variability of the dependent variable, average PSNR, in the population.

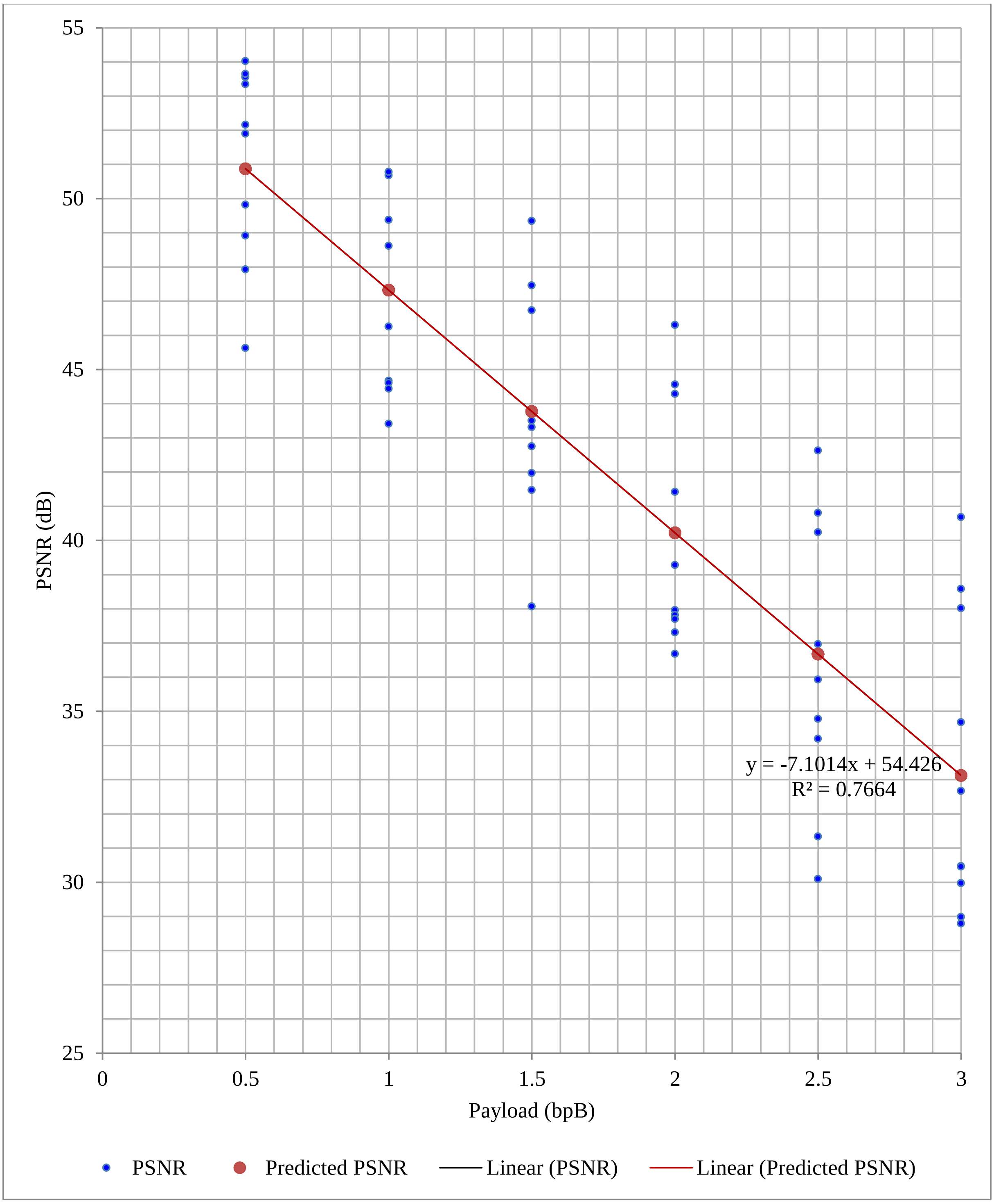


Fig. 9.6. Scatter diagram representing the regression analysis in terms of payload (bpB) and PSNR (dB) for proposed schemes followed by quality enhancement

In fig 9.7, the quality of the WDHT_2x2, WDHT_1x2, WLT_2x2, WLT_1x2, WBT_2x2, WBT_1x2, WST_2x2, WST_1x2, WGD4_2x2 and WGD4_1x2 schemes are improvised by utilizing the genetic algorithm (GA) based optimization as a post-embedding operation. The redefined schemes are: WDHT_2x2_GAO, WDHT_1x2_GAO, WLT_2x2_GAO, WLT_1x2_GAO, WBT_2x2_GAO, WBT_1x2_GAO, WST_2x2_GAO, WST_1x2_GAO and WGD4_2x2_GAO respectively. The quality of these schemes are computed with respect to increasing payload for the range [0.5 – 3 bpB], and are compared against the fixed payload based existing schemes (Varsaki et al.’s DPTHDI [88], Lin et al.’s method [87], Yang et al.’s method [109] and Varsaki et al.’s DGTDHS [129]) respectively. It is seen that the average PSNR of DPTHDI [88] obtained is 37.4 dB at 0.25 bpB of payload. In comparison with DPTHDI [88], equivalent or better results of average PSNR are retained by WDHT_2x2_GAO and WLT_2x2_GAO for payload range [0.5 – 1.5 bpB], WST_2X2_GAO, WBT_2X2_GAO and WDHT_1X2_GAO for payload range [0.5 – 2 bpB], WLT_1x2_GAO and WBT_1x2_GAO for payload range [0.5 – 2.5 bpB] and the rest of the proposed schemes for payload range [0.5 – 3 bpB] respectively. In contrast to Lin et al.’s [87] scheme, the perceived quality level (in actual sense, the average PSNR) ensured equal or higher values at 0.5 – 2 bpB for WDHT_2x2_GAO and WLT_2x2_GAO, at 0.5 – 2.5 bpB for WST_2X2_GAO, WBT_2X2_GAO and WDHT_1X2_GAO and at 0.5 – 3 bpB for the rest of the proposed schemes respectively. The WDHT_1x2_GAO, WLT_1x2_GAO, WBT_1x2_GAO, WST_2x2_GAO, WST_1x2_GAO and WGD4_2x2_GAO schemes provide an improved average PSNR over Yang et al.’s [109] scheme with 0.2 bpB of payload enhancement. As compared to Varsaki et al.’s DGTDHS [129], the WBT_1x2_GAO, WST_1x2_GAO, WGD4_2x2_GAO and WGD4_1x2_GAO schemes achieved higher average PSNR at 1 bpB of payload. It is observed that some of the proposed schemes provide not as good as average PSNR over Yang et al.’s [109] and Varsaki et al.’s DGTDHS [129] schemes, but still those schemes are effective as they are offering variable payload for the spread from 0.5 to 3 bpB. Majority of the proposed schemes does not support optimization for the payload range (0.5 – 1 bpB) and therefore, results are kept unchanged and retained as identical with the average PSNR values prior to optimization. All these schemes are compared among themselves to validate the quality of the watermarked images in terms of average PSNR with respect to 1.5, 2, 2.5 and 3 bpB of payloads. Performance of one scheme overlap another up to 2 bpB however, a new arrangements of the proposed schemes are generated based on the average PSNR for the payload range [2.5 – 3 bpB], where, each scheme giving better result than the immediate preceding in the order: WDHT_2X2_GAO, WLT_2X2_GAO, WST_2X2_GAO, WBT_2X2_GAO, WDHT_1X2_GAO, WLT_1X2_GAO, WST_1X2_GAO, WBT_1X2_GAO and WGD4_2X2_GAO respectively.

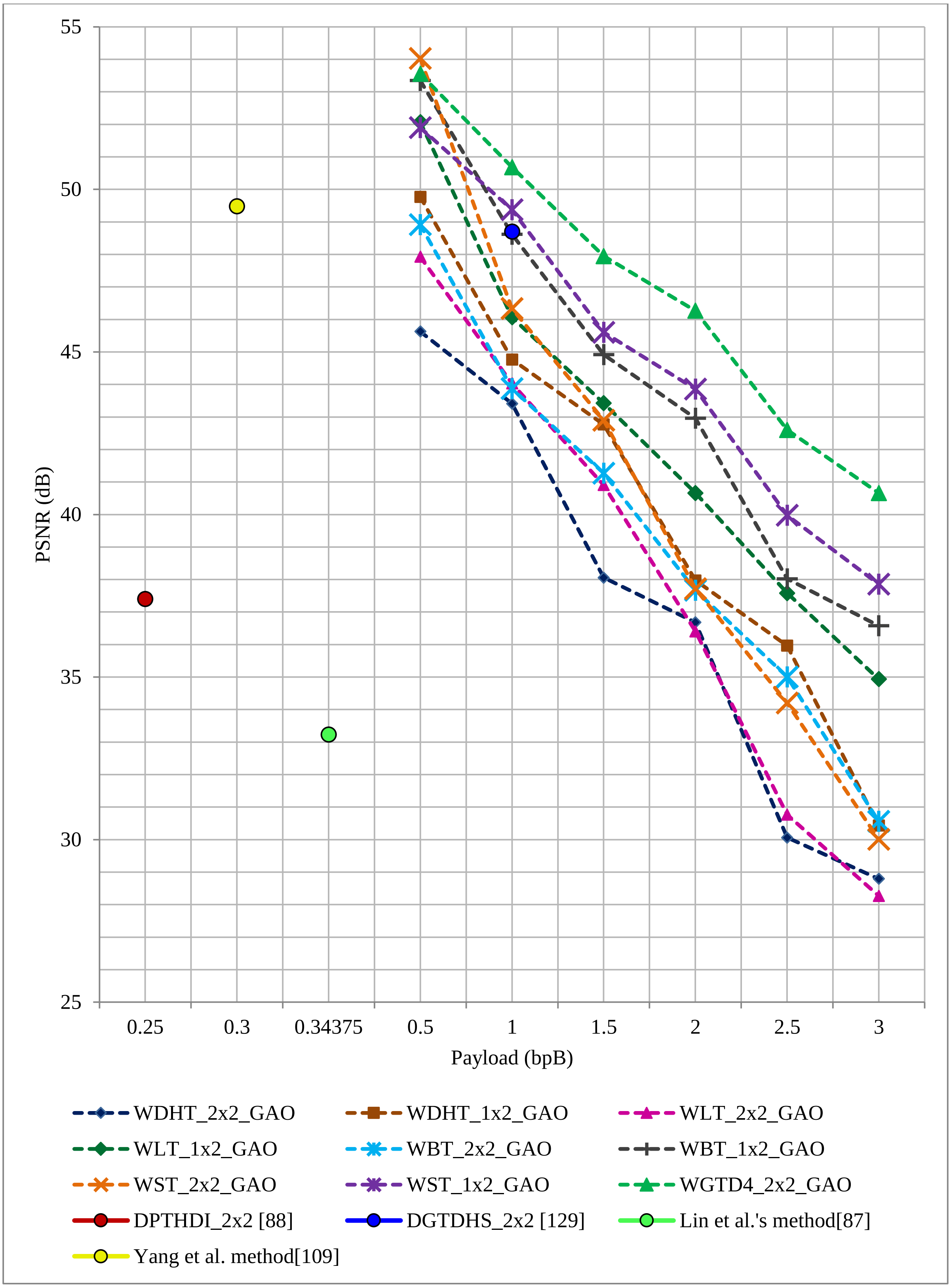


Fig. 9.7. Comparative analysis of PSNR (dB) with respect to payload (bpB) among WDHT_2x2_GAO, WDHT_1x2_GAO, WLT_2x2_GAO, WLT_1x2_GAO, WBT_2x2_GAO, WBT_1x2_GAO, WST_2x2_GAO, WST_1x2_GAO, WGD4_2x2_GAO, Varsaki et al.’s DPTHDI [88], Lin et al.’s [87], Yang et al.’s [109] and Varsaki et al.’s DGTDHS [129] schemes

In fig 9.8., the average PSNR of WDHT_2x2_GAO, WDHT_1x2_GAO, WLT_2x2_GAO, WLT_1x2_GAO, WBT_2x2_GAO, WBT_1x2_GAO, WST_2x2_GAO, WST_1x2_GAO and WGD4_2x2_GAO schemes are plotted against the high and low payload values of 3 and 0.5 bpB respectively. The average PSNR values of the proposed schemes lie into the range of 28.26 to 40.66 dB at 3 bpB and that of average PSNR values of the proposed schemes are falling into the range [45.62 – 54.02 dB] at 0.5 bpB of payload. However, it is needless to say that WLT_1X2_GAO, WBT_1X2_GAO, WST_2X2_GAO, WST_1x2_GAO, WGD4_2x2_GAO and WGD4_1x2_GAO offered an average PSNR of greater than or equal to 50 dB at low payload. It is also seen that the schemes proposed so far (except WDHT_2x2_GAO and WLT_2x2_GAO) offered an average PSNR of more than 30 dB. The WDHT_2x2_GAO and WLT_2x2_GAO schemes provide the average PSNR of 28.79 dB and 28.26 dB at high payload which are really very close to 30 dB. Generally, the average PSNR value above 30 dB is considered as perceptible watermarked image [148]. Therefore, the 1 x 2 block based schemes WDHT_1x2_GAO, WLT_1x2_GAO, WBT_1x2_GAO and WST_1x2_GAO and WGD4_1x2_GAO are finer than 2 x 2 block based schemes viz. WDHT_2x2_GAO, WLT_2x2_GAO, WBT_2x2_GAO and WST_2x2_GAO respectively in terms of average PSNR with respect to variable payload.

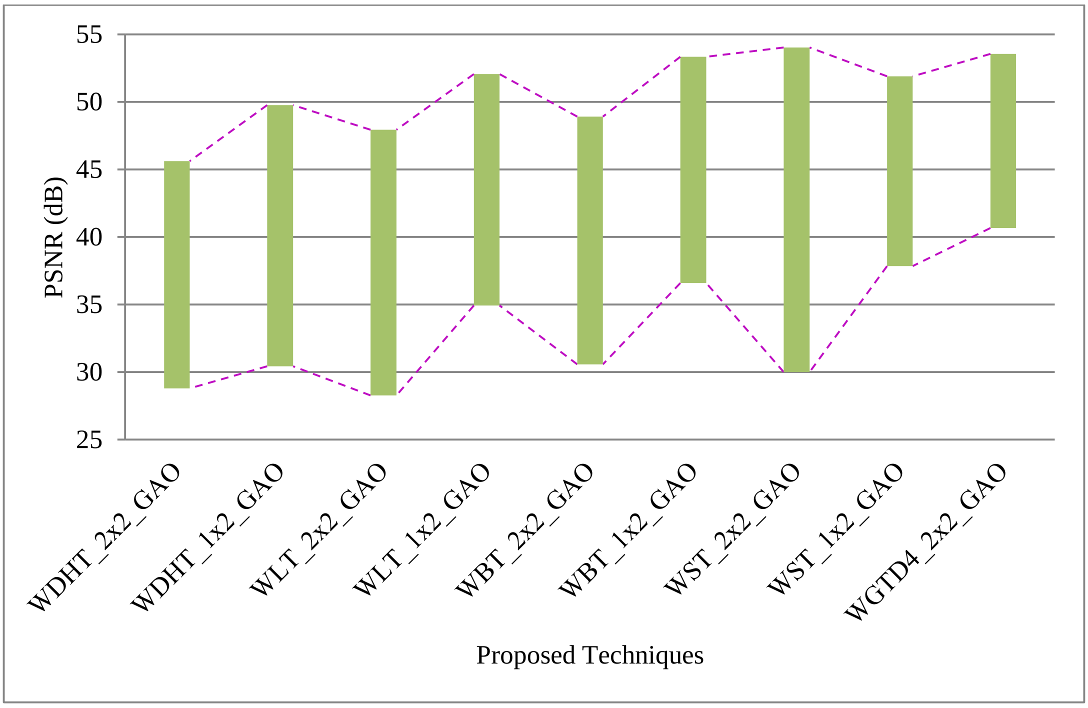


Fig. 9.8. Analysis of PSNR (dB) at high and low payloads (bpB) with respect to WDHT_2x2_GAO, WDHT_1x2_GAO, WLT_2x2_GAO, WLT_1x2_GAO, WBT_2x2_GAO, WBT_1x2_GAO, WST_2x2_GAO, WST_1x2_GAO and WGD4_2x2_GAO schemes

In fig. 9.9, the prediction of average PSNR values with respect to increasing payload for the spread of 0.5 to 3 bpB are made through a regression analysis. The data collected from WDHT_2x2_GAO, WDHT_1x2_GAO, WLT_2x2_GAO, WLT_1x2_GAO, WBT_2x2_GAO, WBT_1x2_GAO, WST_2x2_GAO, WST_1x2_GAO and WGD4_2x2_GAO are plotted along the X-axis and Y-axis of the scatter plot to do a regression analysis which can explain the relationship between the metrics: average PSNR and the payload. A total of 54 observations have been considered where, the "blue" dots depict the actual values of the average PSNR while the predicted values of average PSNR is plotted through "red" dots. The regression line slopes downward from left to right which ensured a consistent result with the negative relationship between the average PSNR and payload. No line can be found to pass through all points of the plot which ensured the absence of functional relationship between the average PSNR and payload. The coefficients of the regression line are 53.806872 (intercept) and -6.994828 (slope) so that the line has the equation: y = -6.994x + 53.80. For example, one can predict the average PSNR at 2 bpB of payload through the above equation of straight line as y = -6.994 × 2 + 53.80 = 39.81, or approximately 40 dB. Similarly, the predicted value of average PSNR at 3 bpB and 0.5 bpB has been computed which are around 33 dB and 51 dB, respectively. Since, the average PSNR values falling below 30 dB indicate a fairly low quality watermarked images and values above 40 dB ensured almost invisible degradation [148], the predicted value of average PSNR is highly acceptable. However, with a systematic analysis of the data, three more regression statistics such as the Multiple R, R^2 and adjusted R^2 has also been computed. The value of Multiple R (correlation co-efficient) is 0.874769 which ensure a stronger linear relationship as the value of 1 means perfect positive relationship. Since the correlation coefficient (Multiple R) is 0.874769, the prediction of average PSNR is considered moderately positive and almost perfect. An R^2 value of 0.7652 ensure that 76.52 % of the points fall on the regression line or in other words, 76.52 % values of average PSNR fit the model. Adjusted R^2 = 0.7607, which means that the independent variable, payload, explains 76.07 % of the variability of the dependent variable, average PSNR, in the population.

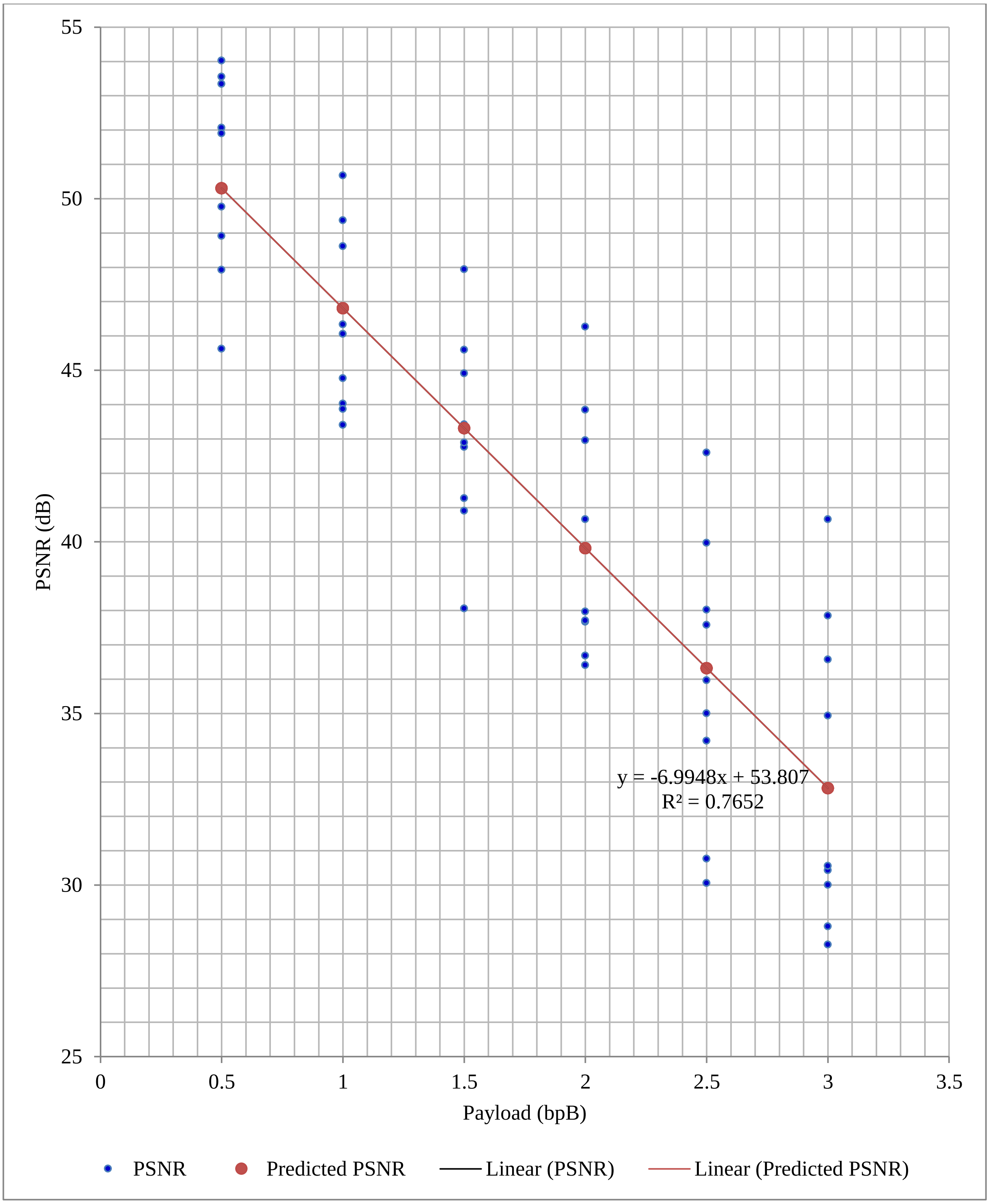


Fig. 9.9. Scatter diagram representing the regression analysis in terms of payload (bpB) and PSNR (dB) for proposed schemes followed by genetic algorithm (GA) based optimization

In fig. 9.10, a comparative analysis has also been made among the proposed watermarking schemes prior to quality enhancement/optimization, the quality enhanced watermarking schemes (QE) and the GA optimization based watermarking (GAO) schemes in terms of predictive PSNR with respect to 0.5, 1, 1.5, 2, 2.5 and 3 bpB of payloads respectively. It is observed from the graph that the QE and GAO based schemes does not affect the fidelity up to 1 bpB of payload. But, as the payload increases, the QE and GAO based schemes giving better results in terms of average PSNR than the proposed watermarking schemes without quality enhancement/optimization. Consequently, the distortions of the watermarked images are minimized tremendously.

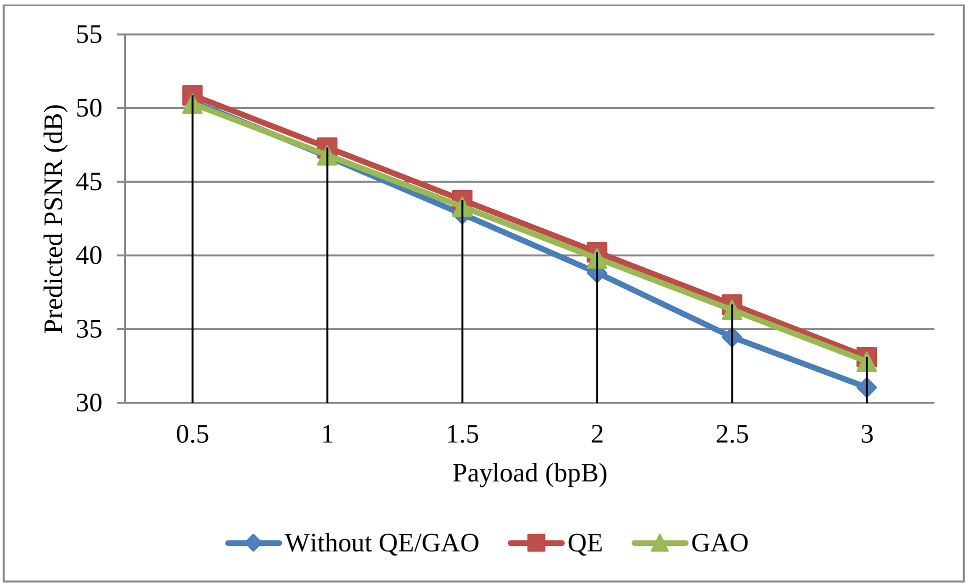


Fig 9.10. Comparative analysis of predictive PSNR among the watermarking based on quality enhancement (QE), GA based optimization (GAO) and the schemes without quality enhancement (QE) / GA based optimization (GAO)

The extensive analysis demonstrated that the watermarking based on Discrete Hartley Transform (DHT), Legendre Transform (LT), Binomial Transform (BT), Stirling Transform (ST) and group of linear transformations based on dihedral group of order 4 (G-lets D4) embeds data in a such a way that the quality of the watermarked images are perceived at variable payloads. By introducing the quality enhancement scheme (QE) of chapter 7 or Genetic Algorithm based optimization (GAO) of chapter 8 has also made the proposed watermarking schemes as an ideal choice for embedding and authentication. The comparative analysis also ensured the effectiveness of these proposed schemes over Varsaki et al.’s DPTHDI [88], Lin et al.’s [87] scheme, Yang et al.’s [109] scheme and Varsaki et al.’s DGTDHS [129] in terms of payload and/or quality.

Chapter 10

Conclusive Discussions

Conclusive Discussions

The major objective of this thesis is to verify the authenticity of color images in terms of embedding the watermark into the transformed coefficients. Popular transformation such as Discrete Hartley Transform (DHT), Binomial Transform (BT), Legendre Transform (LT), Stirling Transform (ST) and group of linear transformations based on dihedral group of order 4 (G-lets D4) have been used for this purpose. It is seen that most of the existing methods available in the literature are suffering from several issues. The first one is selection of proper window size depending upon the transformation chosen for watermark embedding in transform domain. Secondly, the existing methods are focused to achieve high robustness against several visual/geometrical attacks which results a poor payload and high degradation in quality of the watermarked images. Thirdly, most of the existing watermarking techniques are capable of dealing with gray-scale images as the cover image wherein, color image is an effective choice to achieve higher transparency and better payload. The existing methods also suffer from high computational cost of the algorithms in terms of good robustness. Very few methods are available that can deal with the authenticity of color images with acceptable quality degradation when the payload varies from minimum (0.5 bpB) to maximum (3 bpB).

In this thesis, the major achievement is designing and implementation of fragile watermarking schemes for color image authentication. A set of methods have been proposed, implemented and tested in the thesis to deal with watermark embedding in transform domain. It is seen from the literature that existing watermarking schemes used gray-scale images as the cover image in usual cases. But, color images as the cover ensured better transparency and high payload because the watermark information are fabricated separately into the red, green and blue channels. The proposed schemes outperform the existing methods in terms of payload, fidelity and computational overhead.

The general idea for majority of the existing schemes is to achieve higher robustness in lieu of compromising the payload and quality. On the other hand, payload enhancement with permissible degradation in quality is the primary objective of the proposed techniques. Proposed techniques embed watermark bits into 2 x 2 and 1 x 2 transformed matrices as sliding window manner where, variable numbers of bits are embedded into each transformed coefficient to achieve variable payload of 0.5 to 3 bpB. Though the schemes may lack in robustness, watermark bits are fabricated into the low, middle and high frequency sub-bands to achieve variable payload and less distortion in the watermarked images.

In general, either 2 x 2 or 8 x 8 window sizes have been chosen for embedding secret watermarks into the transformed coefficients in case of existing schemes available in the literature. It is seen that the majority of the standard transformations such as Discrete Fourier Transformation (DFT), Discrete Wavelet Transformation (DWT), Discrete Pascal Transform (DPT) and Discrete Gould Transformation (DGT) etc. have supported 2 x 2 block based decomposition of the carrier image and then fabrication of watermark bits into it. Hence, proposed watermarking techniques are basically designed based on the principle of 2 x 2 blocks to compare and validate the experimental results. In addition, 1 x 2 block based schemes is also designed since they offered better visual transparency over the 2 x 2 block based schemes with respect to the following transformations: Discrete Hartley Transform (DHT), Binomial Transform (BT), Legendre Transform (LT), Stirling Transform (ST) and group of linear transformations based on dihedral group of order 4 (G-lets D4) etc.

Unlike Discrete Cosine Transform (DCT) based watermarking scheme [74], Discrete Hartley Transform (DHT), Binomial Transform (BT), Legendre Transform (LT), Stirling Transform (ST) and group of linear transformations based on dihedral group of order 4 (G-lets D4) produced real transformed matrix from real pixel matrix which ensured less computation time and faster execution over floating-point calculations.

It is observed that an intentional/unintentional attack on the watermarked image may alter the content in such a way that visually the image may not considerably degraded. But the frequency distribution is transformed in such a manner that the decoder fails to extract the hidden data. The recipient extracts the fabricated watermark size, content and a message digest (MD) from the watermarked image. The extracted message digest (MD) was actually obtained through MD5 algorithm during embedding. This message digest (MD) is compared with respect to another message digest (MD'), where MD' has been obtained from the extracted watermark at the recipient end. If the retrieved message digest (MD) matches with the re-computed message digest (MD'), then the authentication process is considered to be successful, otherwise, it is treated to be unsuccessful.

Optimizations of the watermarking techniques are another significant achievement. To minimize the distortion of the watermarked images quality, Genetic Algorithm (GA) has been utilized to search the user parameters in a wide range to obtain the optimized results that reflects enhancement in quality. In addition, the quality degradation of the watermarked images has also been reduced by applying a post-embedding quality enhancement scheme.

The optimization/quality enhancement scheme has been applied in such a way that the fabricated watermark information is kept unaffected.

The WST_1x2 of section 5.2.1 dealt with an extra level of security by scrambling the watermark through Arnold’s cat map as operated in the prior phase of embedding. Arnold’s cat map is a chaos based scheme which is utilized for transforming the values of each pair of pixel components. The scrambled bits are embedded into the transformed components in Stirling Transform (ST) domain for improved security.

Proposed watermarking techniques based on the different transformations discussed so far will be widely used to authenticate digital content by preserving the integrity. The major applications of these techniques includes: tamper detection, legal document authentication, ownership verification and source tracking etc.

Limitations and Future Scope

It has already been discussed that watermarking is considered as an optimization problem as the metrics are conflicting with each other. There is a huge scope of research to devise novel watermarking algorithms based on the transformations including Discrete Hartley Transform (DHT), Binomial Transform (BT), Legendre transform (LT), Stirling Transform (ST) and group of linear transformations based on dihedral group of order 4 (G-lets D4) which might exploits other optimization techniques such as Particle swarm optimization, Artificial bee colony optimization, Ant colony optimization and Simulated annealing etc. to improve the watermarked images quality.

In contrast to the proposed schemes, watermarking based on the above mentioned transformations might be used for copy-right protection. The scheme could be investigated with larger block sizes in which minimum numbers of watermark bits are fabricated into the selective frequency bands to achieve high robustness against common visual/geometrical attacks.

Some more transformation techniques such as Exponential Transform (ET) and group of linear transformations based on dihedral group of order 8 (G-lets D8) or group of linear transformations based on dihedral group of order 16 (G-lets D16) etc. could also be investigated for watermarking. Moreover, chaotic sequences such as Logistic map, Tent map and Henon map etc. may also be incorporated to increase the security of the watermark prior to embedding.

It is seen from the literature that the exploiting modification direction based watermarking schemes offers high payload with minimum distortions. In the future course of research, watermarking based on exploiting modification direction in transform domain might open a new direction.

Regardles of these limitations and futures scopes, the thesis will be useful for the personnel those are pursuing research in the field of Computer Science & Engineering, Computer Applications and Information Technology. The schemes are superior over the existing schemes in terms of the following facts: ability to verify the authenticity of color images, less computational overhead, variable payload and acceptable visual clarity. Practicising engineers would also find this thesis to be an excellent reference source.

References

1. Bender W., Gruhl D., Morimoto N., Lu A., "Techniques for data hiding", IBM Systems Journal ,Vol. 35 (3–4), pp. 313–336, 1996.
2. Wong P. W., "A public key watermarking for image verification and authentication", Proc. of IEEE International Conference on Image Processing, Chicago, Vol. 1, pp. 425–429, 1998.
3. Nikolaidis N., Pitas I., "Robust image watermarking in the spatial domain", Signal Processing 66(3), pp. 385–403, 1998.
4. Wang R.-Z., Lin C.-F., Lin J.-C., "Image hiding by optimal LSB substitution and genetic algorithm", Pattern Recognition, 34(3), pp. 671–683, 2001.
5. Chang C.-C., Hsiao J.-Y., Chan C.-S., "Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy," Pattern Recognition,36(7), pp. 1583-1595, 2003.
6. Wu D.-C., Tsai W.-H., "A steganographic method for images by pixel-value differencing", Pattern Recognition Letters, 24(9-10), pp. 1613–1626, 2003.
7. Chang K. C., Huang P. S., Tu T. M., Chang C. P., "Adaptive Image Steganographic Scheme Based on Tri-way Pixel-Value Differencing", Proc. of IEEE International Conference on Systems, Man and Cybernetics, Montreal, pp. 1165 – 1170, 2007.
8. Wu X., Guan Z. -H, Wu Z., "A Chaos Based Robust Spatial Domain Watermarking Algorithm", Lecture Notes in Computer Science, Advances in Neural Networks (ISNN 2007), Nanjing, Vol. 4492, pp. 113-119, 2007.
9. Sathisha N., Madhusudan G.N. , Bharathesh S. , Suresh Babu K., "Chaos based Spatial Domain Steganography using MSB", Proc. of International Conference on Industrial and Information Systems (ICIIS), Mangalore, pp. 177 - 182, 2010.
10. Rawat S., Raman B., "A chaotic system based fragile watermarking scheme for image tamper detection", International Journal of Electronics and Communications (AEÜ), 65(10), pp. 840-847, 2011.
11. Zhang X., Wang S., "Efficient steganographic embedding by exploiting modification direction", IEEE Communications Letters, 10(11), pp. 1-3, 2006.
12. Lee C. F., Wang Y. R., Chang C. C., "A steganographic method with high embedding capacity by improving exploiting modification direction", Proc. of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kaohsiung, Vol. 1, pp. 497-500, 2007.

13. Kuo W. C.,Wang C. C., “Data hiding based on generalized exploiting modification direction method”, Imaging Science, 61(6), pp. 484-490, 2013.

14. Kuok W. C., Chang S.Y., “Hybrid GEMD Data Hiding”, Journal of Information Hiding and Multimedia Signal Processing, 5(3), pp. 420-430, 2014.

15. Bors A.G., Pitas I., “Image watermarking using DCT domain constraints”, Proc. of IEEE International Conference on Image Processing, Lausanne, Vol. 3, pp. 231-234, 1996.

16. Xia X.-G., Boncelet C. G., Arce G. R., “Wavelet transform based watermark for digital images”, Optics Express, 3(12), pp. 497-511, 1998.

17. Lee H.-J., Park J.-H., Zheng Y., “Digital Watermarking Robust against JPEG Compression”, Information Security Lecture Notes in Computer Science, Second International Workshop, ISW’99, Kuala Lumpur, Vol. 1729, pp. 167-177, 1999.

18. Kang S., Aoki Y., “Data hiding system by Fresnel transform”, Proc. of the IEEE Region 10 Conference, TENCON 99 , Cheju Island, Vol.1, pp. 625 - 628, 1999.

19. Huang J., Shi Y.Q. , “Embedding strategy for image watermarking in DCT domain”, Proc. of Fifth Asia-Pacific Conference on Communications and Fourth Asia-Pacific Conference on Optoelectronics and Communications, Bejing, Vol. 2 , pp. 981- 984, 1999.

20. Kim J. R., Moon Y. S., “A robust wavelet-based digital watermarking using level-adaptive thresholding”, Proc. of International Conference on Image Processing, Kobe, Vol.2, pp. 226-230, 1999.

21. Premaratne P. , Ko C.C., “A novel watermark embedding and detection scheme for images in DFT domain”, Proc. of Seventh International Conference on Image Processing and Its Applications, Manchester , Vol.2 , pp. 780-783, 1999.

22. Solachidis V., Pitas I. , “Circularly symmetric watermark embedding in 2-D DFT domain”, Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, Vol.6 , pp. 3469-3472, 1999.

23. Pereira S., Ruanaidh J.J.K.O., Deguillaume F., Csurka G., Pun T., “Template based recovery of Fourier-based watermarks using log-polar and log-log maps”, Proc. of IEEE International Conference on Multimedia Computing and Systems, Florence, Vol.1, pp. 870- 874, 1999.

24. Wang Y. -P., Chen M. -J., Cheng P.-Y., “Robust image watermark with wavelet transform and spread spectrum techniques”, Proc. of Thirty-Fourth Asilomar

Conference on Signals, Systems and Computers, Pacific Grove, Vol. 2, pp. 1846-1850, 2000.

25. Zhicheng N., Sung E., Shi Y.Q., "Enhancing robustness of digital watermarking against geometric attack based on fractal transform", Proc. of IEEE International Conference on Multimedia and Expo, New York, Vol. 2, pp. 1033- 1036, 2000.
26. Suthaharan S., Sathananthan S., "Transform domain technique: robust watermarking for digital images", Proc. of the IEEE Conference Publication, Nasville, pp. 407- 412, Southeastcon, 2000.
27. Zhang X.-D., Lo K.-T., Feng J., Wang D., "A robust image watermarking using spatial-frequency feature of wavelet transform", Proc. of 5th IEEE International Conference on Signal Processing, Bejing, Vol. 2, pp. 1100- 1105, 2000.
28. Fotopoulos V., Krommydas S., Skodras A.N., "Gabor transform domain watermarking", Proc. of International Conference on Image Processing, Thessaloniki, Vol. 2 , pp. 512- 513, 2001.
29. Seto H., Aoki, Y., Seok Kang, "An image data watermarking technique using the average of a Fresnel-transformed pattern", Proc. of International Conference on Image Processing, Thessaloniki, Vol. 2, pp. 534- 537, 2001.
30. Pereira S., Voloshynoskiy S., Pun T., "Optimal transform domain watermark embedding via linear programming", Signal Processing, 81(6), pp. 1251-1260, 2001.
31. Yang J., Lee M. H., Chen X., Park J. Y., "Mixing chaotic watermarks for embedding in wavelet transform domain", Proc. of IEEE International Symposium on Circuits and Systems, Phoenix-Scottsdale, Vol. 2, pp. 668-671, 2002.
32. Ping D., Galatsanos N.P., "Affine transformation resistant watermarking based on image normalization", Proc. of International Conference on Image Processing, Vol. 3, pp. 489- 492, 2002.
33. Ho A.T.S., Jun S., Chow A.K.K., Woon J., "Robust digital image-in-image watermarking algorithm using the fast Hadamard transform", Proc. SPIE 4793, Mathematics of Data/Image Coding, Compression, and Encryption V, with Applications, Vol. 4793, pp. 826-829, 2003.
34. Ashourian M., Yo-Sung Ho, "Analysis of quantization watermarking in the wavelet transform domain", Proc. of Seventh International Symposium on Signal Processing and Its Applications, Vol. 2, pp. 375- 378, 2003.

35. Kang X., Huang J., Shi Y.Q., Lin Y., “A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression”, IEEE Transactions on Circuits and Systems for Video Technology, 13(8), pp. 776- 786, 2003.
36. Kwon K.-R., Kwon S.-G., Nam J.-H., Ahmed T., “Content Adaptive Watermark Embedding in the Multiwavelet Transform Using a Stochastic Image Model”, Lecture Notes in Computer Science, First International Workshop on Digital Watermarking, Seoul, Vol. 2613, pp. 249-263, 2003.
37. Ho A.T.S, Zhu X., Guan Y.L., Marziliano P., “Slant Transform Watermarking for Textured Images”, Proc. of IEEE International Symposium on Circuits and Systems, Vol. 5, Pp. 700-703, 2004.
38. Shiba R., Seok K., Aoki Y., “An image watermarking technique using cellular automata transform”, Proc. of IEEE Region 10 Conference, TENCON 2004, Vol. A, pp. 303- 306, 2004.
39. Lian C., Du S., “Rotation, scale and translation invariant image watermarking using Radon transform and Fourier transform”, Proc. of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, Vol.1, pp. 281- 284, 2004.
40. Liu Z.-B., Fan J.-L., Zhang H.-C., “A blind watermarking algorithm based on wavelet lifting transform”, Proc. of 7th International Conference on Signal Processing, Vol. 1, pp. 843- 847, 2004.
41. Safabakhsh R., Zaboli, S., Tabibiazar A., “Digital watermarking on still images using wavelet transform”, Proc. of International Conference on Information Technology: Coding and Computing, Vol. 1, pp. 671-675, 2004.
42. Jeon J., Lee S.-K., Ho Y.-S., “A Three-Dimensional Watermarking Algorithm Using the DCT Transform of Triangle Strips”, Lecture Notes in Computer Science, Second International Workshop on Digital Watermarking, Seoul, Vol. 2939, pp. 508-517, 2004.
43. Kim B.-S., Choi j.-G., Park K.-H., “RST-Resistant Image Watermarking Using Invariant Centroid and Reordered Fourier-Mellin Transform”, Lecture Notes in Computer Science, Second International Workshop on Digital Watermarking, Seoul, Vol. 2939, pp. 370-381, 2004.
44. Jin J.-Q., Dai M.-Y., Bao H.-J., Peng Q.-S., “Watermarking on 3D mesh based on spherical wavelet transform”, Journal of Zhejiang University Science, 5(3), pp. 251-258, 2004.

45. Shieh C.-S., Huang H.-C., Wang F.-H., Pan J.-S., “Genetic watermarking based on transform-domain techniques”, Pattern Recognition, 37(3), pp. 555-565, 2004.

46. Ahmed A. M., Day D.D., “Applications of the naturalness preserving transform to image watermarking and data hiding”, Digital Signal Processing, 14(6), pp. 531-549, 2004.

47. Areef T. E., Heniedy H. S., Ouda Mansour O.M., Optimal transform domain watermar embedding via genetic algorithms, “Information and Communications Technology”, ITI 3rd International Conference on Enabling Technologies for the New Knowledge Society, Cairo, pp. 607- 617, 2005.

48. Zhang F., Mu X., Yang S., “Multiple-chirp typed blind watermarking algorithm based on fractional Fourier transform”, Proc. of International Symposium on Intelligent Signal Processing and Communication Systems, pp. 141- 144, 2005.

49. Ho A.T.S., Zhu X., Woon W.M., “A semi-fragile pinned sine transform watermarking system for content authentication of satellite images”, Proc. of IEEE International Symposium on Geoscience and Remote Sensing, Vol. 2, 2005.

50. Lee J.-J., Kim W., Lee N.-Y., Kim G.-Y., “A New Incremental Watermarking Based on Dual-Tree Complex Wavelet Transform”, The Journal of Supercomputing, 33(1), pp. 133-140, 2005.

51. Wu X., Liu H., Huang J., “Semi-fragile Watermarking Based on Zernike Moments and Integer Wavelet Transform”, Lecture Notes in Computer Science, 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Vol.3682, pp. 1108-1114, Melbourne, 2005.

52. Yang J., You X., Tang Y. Y., Fang B., “A Watermarking Scheme Based on Discrete Non-separable Wavelet Transform”, Lecture Notes in Computer Science, Second Iberian Conference on Pattern Recognition and Image Analysis, Vol.3522, pp. 427-434, Estoril, 2005.

53. Tsui T. K., Zhang X. –P., Androutsos D., “Color Image Watermarking Using the Spatio-Chromatic Fourier Transform”, Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, Toulouse, Vol. 2, pp. II, 2006.

54. Yu F.Q., Zhang Z.K., Xu M.H., “A Digital Watermarking Algorithm for Image Based on Fractional Fourier Transform”, Proc. of 1^{ST} IEEE Conference on Industrial Electronics and Applications, Singapore, pp. 1- 5, 2006.

55. Wang Z., Zhai G., Wang N., “Digital watermarking algorithm based on wavelet transform and neural network”, Wuhan University Journal of Natural Sciences, 11(6), pp. 1667-1670, 2006.

56. Yin H., Chen Z.-Q., Yuan Z.-Z., “Robust digital watermarking algorithm based on continuous hyperchaotic system and discrete wavelet transform”, Optoelectronics Letters, 2(5), pp. 369-372, 2006.

57. Gui G. -F., Jiang L.-G., He C., “A new asymmetric watermarking scheme based on a real fractional DCT-I transform”, Journal of Zhejiang University SCIENCE, 7(3), pp. 285-288, 2006.

58. Agarwal N., Goyal A.K., “Robust Watermarking in Transform Domain Using Edge Detection Technique”, Proc. of International Conference on Computational Intelligence and Multimedia Applications, Sivakasi, Vol. 4, pp. 59- 63, 2007.

59. Hien T. D., Miyara K.,, Nagata Y., Nakao Z., Chen Y. W., “Curvelet Transform Based Logo Watermarking”, Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications, pp. 305-309, 2007.

60. Lee H.-Y., Lee C.-H., Lee H.-K., “Geometrically invariant watermarking: synchronization through circular Hough transform”, Multimedia Tools and Applications, 34(3), pp. 337-353, 2007.

61. Guo J., Liu Z., Liu S., “Watermarking based on discrete fractional random transform”, Optics Communications, 272(2), pp. 344-348, 2007.

62. Kumaran T., Thangavel P., “Watermarking in Contourlet Transform Domain Using Genetic Algorithm”, Second UKSIM European Symposium on Computer Modeling and Simulation, Liverpool, pp. 257- 262, 2008.

63. Li Y., Wang X., “A watermarking method combined with Radon transform and 2D-wavelet transform”, Proc. of 7th World Congress on Intelligent Control and Automation, Chongqing, pp. 4586- 4590, 2008.

64. Zhu X., “A semi-fragile digital watermarking algorithm in wavelet transform domain based on Arnold transform”, Proc. of 9th International Conference on Signal Processing, Bejing, pp. 2217- 2220, 2008.

65. Korohoda P., Dabrowski A., “The discrete trigonometric transforms for still image watermarking in the transform domain”, Signal Processing Algorithms, Architectures, Arrangements, and Applications, Poznan, pp. 143- 148, 2008.

66. Ozturk M., Cekic Y., Akan A., “Discrete evolutionary transform based robust image watermarking”, Proc. of 23rd International Symposium on Computer and Information Science, Istanbul, pp. 1- 5, 2008.
67. Tsai J.-S., Huang W.-B., Li P.-H., Chen C.-L., Kuo Y.-H., “Robust Digital Image Watermarking Based on Principal Component Analysis and Discrete Wavelet Transform”, Lecture Notes in Computer Science, 9th Pacific Rim Conference on Multimedia, Tainan, Vol. 5353, pp. 506-514, 2008.
68. Falkowski B. J., “Phase Watermarking Algorithm using Hybrid Multi-Polarity Hadamard Transform”, Journal of Mathematical Imaging and Vision, 30(1), pp. 13-21, 2008.
69. Rawat S., Bhatnagar G., Raman B., “ A Robust Watermarking Scheme using Best Tree Wavelet Packet Transform”, Proc. of IEEE International Conference on Advance Computing, Patiala, pp. 883- 887, 2009.
70. Lin Z., “Multipurpose Digital Watermarking Algorithm Based on Morphological Wavelet Transform”, Proc. of WRI International Conference on Communications and Mobile Computing, Yunnan, Vol. 3, pp. 396- 400, 2009.
71. Jianzhong L., Zhang X., Liu S., Ren X., “An Adaptive Secure Watermarking Scheme for Images in Spatial Domain Using Fresnel Transform”, Proc. of 1st International Conference on Information Science and Engineering, Nanjing, pp. 1630- 1633, 2009.
72. Kumar N. N., “Optical image watermarking using fractional Fourier transform”, Journal of Optics, 38(1), pp. 22-28, 2009.
73. Mansouri A., Aznaveh A. Mahmoudi, Azar F. Torkamani, “SVD-based digital image watermarking using complex wavelet transform”, Sadhana, 34(3), pp. 393-406, 2009.
74. Wang F.-H., Pan J.-S., Jain L.C., “Discrete Cosine Transform Based Watermarking Scheme and Band Selection”, Innovations in Digital Watermarking Techniques Studies in Computational Intelligence, Vol. 232, pp. 63-82, 2009.
75. Cintra R.J., Dimitrov V.S., Oliveira H.M. de, de Souza R.M. Campello, “Fragile watermarking using finite field trigonometrical transforms”, Signal Processing: Image Communication, 24(7), pp. 587-597, 2009.
76. Ouhsain M., Hamza, A. B., “Image watermarking scheme using nonnegative matrix factorization and wavelet transform”, Expert Systems with Applications, 36(2, part-1), pp. 2123-2129, 2009.

77. Bohra A., Farooq O., I., “Blind self-authentication of images for robust watermarking using integer wavelet transform”, AEU - International Journal of Electronics and Communications, 63(8), pp. 703-707, 2009.

78. Kang G.S., “Blind digital image watermarking using adpative casting energy in different resolutions of wavelet transform”, Proc. of IEEE International Conference on Computer and Communication Technology (ICCCT), Allahabad, pp. 210- 215, 2010.

79. Peng H., Wang J., Wang W., “Image watermarking method in multiwavelet domain based on support vector machines”, The Journal of Systems and Software, 83(8), pp. 1470–1477, 2010.

80. Zhu X., Wang Y., “Study on algorithm of masking digital watermarking in wavelet transform domain based on chaos encryption and arnold transform”, Proc. of International Conference on Audio Language and Image Processing (ICALIP), Shanghai, pp. 357- 361, 2010.

81. Tian H., Zhao Y., Ni R., Pan J.-S., “Spread Spectrum-Based Image Watermarking Resistant to Rotation and Scaling Using Radon Transform”, Proc. of Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Darmstadt, pp. 442- 445, 2010.

82. Wang C., Ni J., Zhuo J., Huang J., “A geometrically resilient robust image watermarking scheme using deformable multi-scale transform”, Proc. of 17th IEEE International Conference on Image Processing (ICIP), Hong Kong, pp. 3677- 3680, 2010.

83. Sujatha S.S., Sathik M. M., “Feature Based Watermarking Algorithm by Adopting Arnold Transform”, Information and Communication Technologies, Communications in Computer and Information Science, Kerala, Vol. 101, pp. 78-82, 2010.

84. Tian H., Z. Y., Ni R., Pan J.-S., “Geometrically Invariant Image Watermarking Using Scale-Invariant Feature Transform and K-Means Clustering”, Lecture Notes in Computer Science, Computational Collective Intelligence, Technologies and Applications, Kaohsiung, Vol. 6421, pp 128-135, 2010.

85. Liu Z., Lie X., Qing G., Chuang L., Shutian L., “Image watermarking by using phase retrieval algorithm in gyrator transform domain”, Optics Communications, 283(24), pp. 4923-4927, 2010.

86. Singh N., Sinha A., “Digital image watermarking using gyrator transform and chaotic maps”, Optik - International Journal for Light and Electron Optics, 121(15), pp. 1427-1437, 2010.

87. Lin C.C., Shiu P.F., “High capacity data hiding scheme for DCT based images”, Journal of Information Hiding and Multimedia Signal Processing, 1(3), pp. 220-240, 2010.

88. Varsaki E. E., Fotopoulos V.E., Skodras A.N., “On the use of the discrete Pascal transform in hiding data in images”, Proc. of SPIE, Optics, Photonics, and Digital Technologies for Multimedia Applications, Vol. 7723, 77230L, 2010.

89. Yan D., Yang R., Li H., Zheng J., “A digital watermarking scheme based on singular value decomposition and discrete wavelet transform”, Proc. of International Conference on Computer Science and Network Technology (ICCSNT), Harbin, Vol.1, pp. 154- 157, 2011.

90. Li X.-W., Nam T.-H., Lee S.-K., Kim S.-T., “Digital watermarking in transform-domain based on Cellular Automata Transform”, Proc. of 2nd International Conference on Next Generation Information Technology (ICNIT), Gyeongju, pp. 132- 136, 2011.

91. Manoochehri M., Isfahan I., Pourghassem H., Shahgholian G., “A novel synthetic image watermarking algorithm based on Discrete Wavelet Transform and Fourier-Mellin Transform”, Proc. of 3rd IEEE International Conference on Communication Software and Networks (ICCSN), Xi'an, pp. 265 - 269, Xi'an, 2011.

92. Lee Y., Kim J., “Robust Blind Watermarking Scheme for Digital Images Based on Discrete Fractional Random Transform”, Communications in Computer and Information Science, International Conference MulGraB, Held as Part of the Future Generation Information Technology Conference, FGIT 2011, in Conjunction with GDC 2011, Vol. 263, Jeju Island, pp. 139-145, 2011.

93. Kumar D., Kumar V., “Improving the Performance of Color Image Watermarking Using Contourlet Transform”, Communications in Computer and Information Science, First International Conference on Computer Science and Information Technology, CCSIT, springer, Bangalore, Vol. 131, pp. 256-264, 2011.

94. Cancellaro M., Battisti F., Carli M., Boato G., De Natale F.G.B., Neri A., “A commutative digital image watermarking and encryption method in the tree structured Haar transform domain”, Signal Processing: Image Communication, 26(1), pp.1-12, 2011.

95. Saeed R., Namazi F., Yaghmaie K., Aliabadian A., “Hybrid watermarking algorithm based on Singular Value Decomposition and Radon transform”, AEU - International Journal of Electronics and Communications, 65(7), pp. 658-663, 2011.
96. Luo H. Yu F.-X., Huang Z.-L., Lu Z.-M., “Blind image watermarking based on discrete fractional random transform and subsampling”, Optik - International Journal for Light and Electron Optics, 122(4), pp. 311-316, 2011.
97. Maity S.P., Kundu M. K., “Perceptually adaptive spread transform image watermarking scheme using Hadamard transform”, Information Sciences, 181(3), pp. 450-465, 2011.
98. Hao L., Cheng X., Long S., “A kind of image watermarking algorithm based on chaos sequences and fast curvelet transform”, Proc. of 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization (ICSEM), Chengdu, Vol.2, pp. 285- 287, 2012.
99. Elshazly E.H., Ashour M.A., Elrabaie E.M., Abbas A.M., “An efficient Fractional Fourier Transform approach for digital image watermarking”, Proc. of 29th National Radio Science Conference (NRSC), Cairo, pp. 245- 254, 2012.
100. Surekha B., Swamy G., “A semi-blind image watermarking based on Discrete Wavelet Transform and Secret Sharing”, International Conference on Communication, Information & Computing Technology (ICCICT), pp. 1- 5, 2012.
101. Zhao Y., Ni R., Zhu Z. F., “RST transforms resistant image watermarking based on centroid and sector-shaped partition”, Journal-Science China Information Sciences, 55(3), pp. 650-662, 2012.
102. Wu H.-T., Cheung Y.-M., “Secure Watermarking on 3D Geometry via ICA and Orthogonal Transformation”, Lecture Notes in Computer Science, Transactions on Data Hiding and Multimedia Security VII, , Vol. 7110, pp. 52-62, 2012.
103. Rawat S., Raman B., “A blind watermarking algorithm based on fractional Fourier transform and visual cryptography”, Signal Processing, 92(6), pp. 1480-1491, 2012.
104. Huang X., Zhao S., “An Adaptive Digital Image Watermarking Algorithm Based on Morphological Haar Wavelet Transform”, Physics Procedia, Macao, Vol.25, pp. 568-575, 2012.
105. Martino F. D., Sessa S., “Fragile watermarking tamper detection with images compressed by fuzzy transform”, Information Sciences, Vol. 195, pp. 62-90, 2012.
106. Li L., Li S., Abraham A., Pan J.-S., “Geometrically invariant image watermarking using Polar Harmonic Transforms”, Information Sciences, Vol. 199, pp. 1-19, 2012.

107. Hamidreza S., Marzieh A., "A robust spread spectrum based image watermarking in ridgelet domain", AEU - International Journal of Electronics and Communications, 66(5), pp. 364-371, 2012.

108. Betancourth G.P., "Fragile Watermarking Scheme for Image Authentication", Proc. of 5th IEEE International Conference on Human System Interactions (HIS), Merida City, pp. 168 – 174, 2012.

109. Yang C.Y., Lin C.H., Hu W.C., "Reversible data hiding for high-quality Images based on Integer Wavelet Transform, Journal of Information Hiding and Multimedia Signal Processing, 3(2), pp. 142-150, 2012.

110. Tsougenis E.D., Papakostas G.A., Koulouriotis D.E., Tourassis V.D., "Towards adaptivity of image watermarking in polar harmonic transforms domain", Optics & Laser Technology, 54, pp. 84-97, 2013.

111. Makbol N. M., Khoo B. E., "Robust blind image watermarking scheme based on Redundant Discrete Wavelet Transform and Singular Value Decomposition", AEU - International Journal of Electronics and Communications, 67(2), pp. 102-112, 2013.

112. Sang J., Zhang B., Hong D., Xiang H., Xu H., Sang N., "An image watermarking technique based on cascaded iterative Fourier transform", Optik - International Journal for Light and Electron Optics, 124(20), pp. 4522-4525, 2013.

113. Cheng C.-J., Hwang W.-J., Zeng H.-Y., Lin Y.-C., "Fragile Watermarking Algorithm for Hologram Authentication", IEEE Journal of Display Technology, 10(4), pp. 263 – 271, 2014.

114. Han Y., He W., Shang Y., "DWT-Domain Dual Watermarking Algorithm of Color Image Based on Visual Cryptography", Proc. of Ninth IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Bejing, pp. 373 – 378, 2013.

115. Benyoussef M., Mabtoul S., Marraki M. E., Aboutajdine D., "Blind Invisible Watermarking Technique in DT-CWT Domain Using Visual Cryptography", Lecture Notes in Computer Science, Image Analysis and Processing, Vol. 8156, pp. 813-822, 2013.

116. Zheng P., Huang J., "Walsh-Hadamard Transform in the Homomorphic Encrypted Domain and Its Application in Image Watermarking", Lecture Notes in Computer Science, Information Hiding, Vol. 7692, pp. 240-254, 2013.

117. Wang C., Ni J., Zhang D., “Counteracting geometrical attacks on robust image watermarking by constructing a deformable pyramid transform”, EURASIP Journal on Advances in Signal Processing, Vol. 2013, pp. 119, 2013.

118. Tun A., Thein Y., “Digital Image Watermarking Scheme Based on LWT and DCT”, International Journal of Engineering and Technology, 5(2), pp. 272-277, 2013.

119. Kakkirala K.R., Chalamala S.R., “Block based robust blind image watermarking using discrete wavelet transform”, 10th IEEE International Colloquium on Signal Processing & its Applications (CSPA), Kuala Lumpur, pp. 58 – 61, 2014.

120. Minamoto T., Yamaguchi J., “A Blind Digital Image Watermarking Method Based on the Dyadic Wavelet Transform and Chaos Models”, Proc. of 11th IEEE International Conference on Information Technology: New Generations (ITNG), Las Vegas, pp. 459 – 464, 2014.

121. Idrissi N., Roukhe A., “Robust watermarking method based on contourlet transform, maximum entropy, and SVD decomposition”, Proc. of IEEE International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, pp. 261 – 264, 2014.

122. Urvoy M., Goudia D., Autrusseau F., “Perceptual DFT Watermarking With Improved Detection and Robustness to Geometrical Distortions”, IEEE Transactions on Information Forensics and Security, 9(7), pp. 1108 – 1119, 2014.

123. Song X., Wang S., Abd E.-L., Ahmed A., Niu X., “Dynamic watermarking scheme for quantum images based on Hadamard transform”, Multimedia Systems, 20(4), pp. 379-388, 2014.

124. Wang T., Li H., “A novel scrambling digital image watermarking algorithm based on contourlet transform”, Wuhan University Journal of Natural Sciences, 19(4), pp. 315-322, 2014.

125. Forczmański P., “2DKLT-Based Color Image Watermarking”, Image Processing and Communications Challenges 5, Advances in Intelligent Systems and Computing, Springer, Vol. 233, pp. 107-114, 2014.

126. Lei B, Ni D., Chen S., Wang T., Zhou F., “Optimal image watermarking scheme based on chaotic map and quaternion wavelet transform”, Nonlinear Dynamics, 78(4), pp. 2897-2907, 2014.

127. Nezhadarya E., Ward R. K., “Multiscale Derivative Transform and Its Application to Image Watermarking”, Digital Signal Processing, 33, pp. 148–155, 2014.

128. Lang J., Zhang Z.-G., “Blind digital watermarking method in the fractional Fourier transform domain”, Optics and Lasers in Engineering, 53, pp. 112–121, 2014.

129. Varsaki E. E., Fotopoulos V., Skodras A. N., “A discrete Gould transform data hiding scheme”, Mathematical Methods in the Applied Sciences, 37(2), pp. 283-288, 2014.

130. Weber A. G., The USC-SIPI Image Database: Version 5, Original release: October 1997, Signal and Image Processing Institute, University of Southern California, Department of Electrical Engineering. ⟨http://sipi.usc.edu/database/ ⟩ (accessed on 25th January, 2010).

131. University of Granada, Computer Vision Group. CVG-UGR Image Database, ⟨http://decsai.ugr.es/cvg/dbimagenes/c512.php⟩ (accessed on 22nd June, 2014).

132. Watson A.B., Poirson A., “Separable two dimensional discrete Hartley transform”, J.Opt.Soc. Am. A., 3(12), pp. 2001-200, 1986.

133. Schmidt A. L., “Legendre Transforms and Apéry's Sequences”, J. Austral. Math. Soc., Ser. A, 58(3), pp. 358-375, 1995.

134. Jin Y., Dickinson H., “Apéry Sequences and Legendre Transforms.”, J. Austral. Math. Soc., Ser. A, 68, pp. 349-356, 2000.

135. Borisov B., Shkodrov V., “Divergent Series in the Generalized Binomial Transform”, Adv. Stud. Cont. Math., 14 (1), pp. 77-82, 2007.

136. S. Falcon, A. Plaza, “Binomial transforms of the k-Fibonacci sequence”, International Journal of Nonlinear Sciences and Numerical Simulation 10(11-12), pp. 1527-1538, 2009.

137. Bernstein M., Sloane N. J. A., “Some Canonical Sequences of Integers”, Linear Algebra Appl. 226-228, pp. 57-72, 1995.

138. Graham R.L., Knuth D.E., Patashnik O., “Factorial Factors”, §4.4 in Concrete Mathematics: A Foundation for Computer Science, 2nd ed. Reading, MA: Addison-Wesley, pp. 252, 1994.

139. Riordan J., “Combinatorial Identities”, Wiley, New York, R.E. Krieger Pub. Co., pp. 90, 1968 (reprinted with corrections: Riordan, John (1979).

140. Riordan J., “An Introduction to Combinatorial Analysis”, New York: Wiley, pp. 48, 1980.

141. Sloane N. J. A., Plouffe S., “The Encyclopedia of Integer Sequences”, San Diego, CA: Academic Press, 1995.

142. Rajathilagam, B., Rangarajan, M., Soman, K.P., “G-Lets: A New Signal Processing Algorithm”, International Journal of Computer Applications, 37(6), pp. 1-7, 2012.

143. M. Kutter and F. A. P. Petitcolas, “A fair benchmark for image watermarking systems”, Proc. of Electronic Imaging ’99. Security and Watermarking of Multimedia Contents, International Society for Optical Engineering, Vol. 3657, pp. 1-14, 1999.

144. Bandyopadhyay D., Dasgupta K., Mandal J.K. and Dutta P., A novel secure image steganography method based on Chaos Theory in spatial domain, International Journal of Security, Privacy and Trust Management (IJSPTM), 3(1), 2014.

145. Wang Z. and Bovik A.C., A universal image quality index, IEEE Signal Processing Letters, 9(3), pp. 81-84, 2002.

146. Wang Z., Bovik A.C., Sheikh H.R. and Simoncelli E.P., Image quality assessment: From error visibility to structural similarity, IEEE Transactions on Image Processing, 13(4), pp. 600-612, 2004.

147. Mukherjee I., Podder A., DCT Based Robust Multi-bit Steganographic Algorithm, Advanced Computing, Networking and Informatics, Wireless Networks and Security Proceedings of the Second International Conference (ICACNI-2014), Vol. 28, pp. 375-382, 2014.

148. Riad R. (et al.), “Evaluation of a Fourier Watermarking Method Robustness to Cards Durability Attacks”, Lecture Notes in Computer Science, 6th International Conference on Image and Signal Processing (ICISP 2014), Cherbourg, pp. 280-288, 2014.

149. Woolf P., Keating A., Burge C., and Michael Y., “Statistics and Probability Primer for Computational Biologists”, Massachusetts Institute of Technology, BE 490/ Bio7.91, Spring 2004.

150. Steven H. S., “Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering”, Pursues Books, West view, Science Library, Cambridge, 1994.