NEW WATERMARKING METHODS FOR DIGITAL IMAGES

By

Md. Wahedul Islam

M.Sc., University of Rajshahi, 1997
B.Sc., University of Rajshahi, 1996

THESIS SUBMITED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER SCIENCE

UNIVERSITY OF NORTHERN BRITISH COLUMBIA

September 2012

Your file Votre référence
ISBN: 978-0-494-94112-6

Our file Notre référence
ISBN: 978-0-494-94112-6

Abstract

The phenomenal spread of the Internet places an enormous demand on content-ownership-validation. In this thesis, four new image-watermarking methods are presented. One method is based on discrete-wavelet-transformation (DWT) only while the rest are based on DWT and singular-value-decomposition (SVD) ensemble. The main target for this thesis is to reach a new blind-watermarking-method. Method IV presents such watermark using QR-codes. The use of QR-codes in watermarking is novel. The choice of such application is based on the fact that QR-Codes have errors self-correction-capability of 5% or higher which satisfies the nature of digital-image-processing. Results show that the proposed-methods introduced minimal distortion to the watermarked images as compared to other methods and are robust against JPEG, resizing and other attacks. Moreover, watermarking-method-II provides a solution to the detection of false watermark in the literature. Finally, method IV presents a new QR-code guided watermarking-approach that can be used as a steganography as well.

Table of Contents

List of Tables

List of Figures

Publication from This Thesis

1. **Md Wahedul Islam**, Saif Zahir "A Novel QR Code Guided Image Steganography", IEEE International Conference on Consumer Electronics (ICCE 2013), Las Vegas, USA, 2013

2. Md Wahedul Islam, Saif Zahir, "A robust color image watermarking scheme" IASTED Int. Conf. on Visualization, Imaging, and Image processing, VIIP 2012, Banff, Canada, July 3-5, 2012, 8 pages

3. Saif AlZahir and **Md. Wahedul Islam**, "A Balanced Semi-blind Digital Watermarking Scheme Using DWT", AHU Journal of Engineering & Applied Sciences ,vol. 4, no. 1, pp. 89 – 101, 2011.

4. Saif AlZahir and **M Wahedul Islam**, " A New Wavelet-Based Image Watermarking Technique", IEEE International Conference on Consumer Electronics (ICCE 2011), Las Vegas, USA, 2011, pp. 723-724

Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Prof. Saif Zahir, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my master's degree to his encouragement and effort and without him this thesis, too, would not have been possible to complete.

Second, I gratefully acknowledge my committee members, Prof. Eli Korkmaz of Department of Physics, and Prof. Han Donker of Department of Accounting, University of Northern British Columbia, for their valuable time, helpful advice and important comments.

Third, I am also grateful to the Computer Science Research Award committee of the University of Northern British Columbia that provided the partial financial support for this research. The library facilities (notably the inter library loan), was indispensible.

The department of Computer Science, University of Northern British Columbia has provided the support and equipment that I have needed to produce and complete my thesis. I appreciate their support.

Last but not the least, this thesis would have been incomplete without unconditional love and sacrifices from my wife Laila Sharmin Afroz and my daughter Yeasha Binte Islam. They witness every ups and down; the contentment and struggling of being a student, a husband and a father. The fruition of this thesis is result of collaboration and supports.

Chapter 1

Introduction

The phenomenal spread of the internet has increased access to digital multimedia content. This increased access allowed for illegal operations such as replication, content-alteration, copyright infrengment, and ownership violation. These unauthorized operations not only raid the property rights of the digital content's owners but also demote motivation for their constructions. Therefore, the fortification of intellectual property rights of any digital content is essential.

Researchers have identified two different technologies for multimedia content protection: encryption and digital watermarking [1, 2]. Despite the difference between these two technologies, some researchers consider digital watermarking as a complementary approach to cryptographic process [2]. Digital watermarking was introduced as a mean to protect digital property and already established itself as the most effective one. Popular medias (web pages, television, magazines, etc.) have frequently tried to present information in the form of images which raises demand to develop efficient image watermarking techniques.

Digital image watermarking is a process that embeds perceptible or imperceptible data (the watermark) in an image in a specific way. Another brach of science that is

concerned with hiding information in information is steganography. The concept of digital watermarking is closely related to steganography, in that they both hide a message or information inside a digital media such as image. However, what separates them is their goal. Watermarking tries to hide information related to the actual content of the digital signal and the final goal is to keep the marked content as near identical as possible to the original content, while steganography main goal is to hide message into the digital content no matter how the stego-object is far from the original object as long as stego-objects are not suspectabl. Analog and paper watermarking has been around for several centuries. However, the field of digital watermarking has been used during the last three decades and already proved itself worthy for many different applications.

According to image watermarking goal, the most important attribute of any watermarking method is the image fidelity or imperceptibility, i.e., the watermarked image has to be visually near identical to the original image. Another significant characteristic of watermarking method is watermark detection or extraction process which can be either blind or non-blind, in blind process, the original image is not needed, whereas, in the non-blind process we need the original image to detect or extract the watermark. However, blind and non-blind extraction/detection proces must leads to the proof of authorship which is the main target of digital watermarking. Alternatively, any image watermarking method can be classified into either of the two categories: robust and fragile. This classification is based on watermarked image robustness against various types of attacks such as image compression, scaling, noise addition and other intentional and unintentional image processing attacks.

1.1 Digital Watermarking Framework

Digital watermarking is a process that embeds data (the watermark) or digital signature into a multimedia object such that the watermark can be detected or extracted later on to make an assertion about that object. The object may be an image or audio or video file. A very simple example of a digital watermark would be a visible *seal* placed over an image to varify ownership. Moreover, a digital watermark might contain additional information such as the identity of the owner of a certain copy of the image. In general, digital watermarking system consists of three parts:

- The watermark
- The encoder (Mark embedding procedure)
- The decoder (Mark extraction or detection procedure)

In any watermarking method the owner may have a unique watermark or any other mark(s) in different objects. The decoding method authenticates both the owner and the integrity of the objects.

1.1.1 The Watermark

A digital watermark is a pattern of bits or arbitrary real or integer number inserted into a digital media. One of the most known digital watermarks is the pseudo-random-numbers (PRN) sequence of uniue random numbers generated with a seed or a secret key value. The use of logo or small images as a watermark is gaining more popularity day by day as they are easy to identify by bare eyes. However, the following are commonly used watermarks:

- PRN sequence
- Sequence of 0s and 1s (Binary bits)
- Digital signature
- Encoded text
- Small binary Image
- Small Grey scale/colour image or logo.

1.1.2 The Encoder

Let us denote an image by I , a watermark by W and the watermarked image by I_w. An encoder function E takes an image I and a watermark W, and generates a new image which is called watermarked image I_w, mathematically, as follows:

$$E(I, W) = I_W \tag{1}$$

It should be noted that the watermark, W, may be independent or dependent of host image I. In such a case, the encoding process described by equation (1) still holds. Figure 1 illustrates the encoding process.

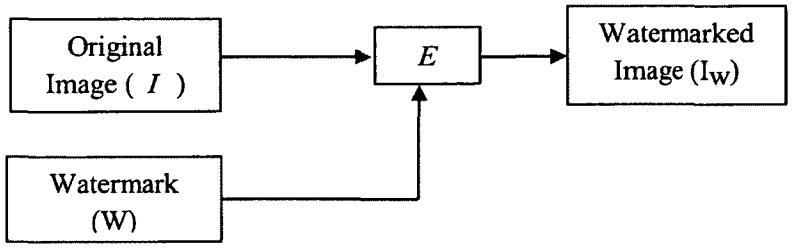


Figure 1.1: Encoder

1.1.3 The Decoder

A decoder function D takes an image I_W (I_W can be a watermarked or possibly corrupted watermarked image) whose ownership is to be determined. For recovering the watermark, W′, from the image, an additional image I_O (which is the original version of I_W) can also be included. This is due to the fact that some decoding methods may make use of the original images in the watermarking process to provide extra robustness against intentional and unintentional corruption of pixels. This process can be represented as:

$$D(I_w, I_o) = W' \tag{2}$$

The extracted watermark W' will then be compared with the original watermark sequence by a comparator function C_δ and a binary output decision generated. It is 1 if there is a match and 0 otherwise, which can be represented as follows.

$$C_\delta(W', W) = \begin{cases} 1, & c \le \delta \\ 0, otherwise \end{cases} \tag{3}$$

where c is the extracted value of the watermark, $x = C_\delta(W', W) * c$ is the correlation of two signatures and δ is a threshold. Without loss of generality, watermarking methods can be treated as a three-tuple (E, D, C_δ). Figures 1.2 and 1.3 demonstrate the decoder and the comparator, respectively.

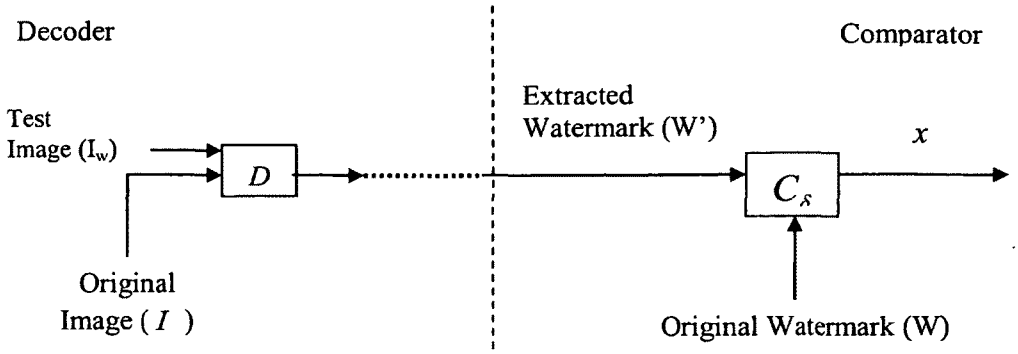


Figure 1.2: Decoder

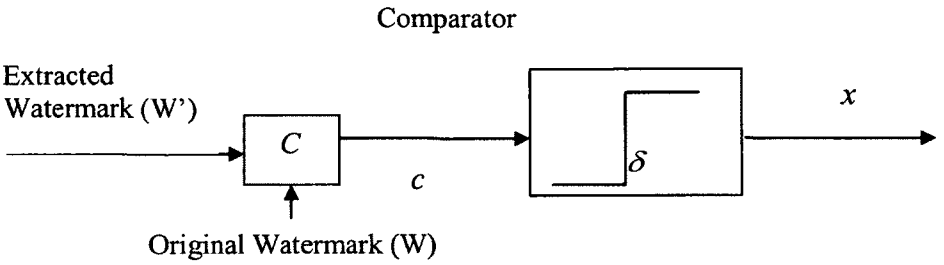


Figure 1.3: Comparator

A watermark must be easily detectable or extractable to be functional. Depending on the way the watermark is embedded and depending on the nature of the watermarking method. In a watermarking method, a watermark can be extracted in its exact or degraded form. In other methodses, watermark can be detected only to determine whether a embedded watermark is present in an image or not. This procedure is called watermark detection.

1.2 Types of Digital Watermark

Watermarks can be divided into various categories. Watermarks can be applied in spatial domain i.e. pixelwise. An alternative to spatial domain watermarking is transformed domain watermarking. Different types of watermarks are shown in the figure 1.4. Watermarking

methods can be divided into two, four, two, and two different categories according to working domain, type of documents, human perception and application, respectively. The visible-watermark is easily visible to a casual viewer on a careful inspection. The invisible-robust watermark is embedded in such a way that Visible watermark is a semi-transparent object overlaid into the primary image alternation to the pixel value is perceptually unnoticed; it can be recovered only with appropriate decoding mechanism and can withstand image processing. The fragile watermark is embedded in such a way that any manipulation or modification of the image would alter or destroy the watermark. Dual watermark is a combination of a visible and an invisible watermark [30].

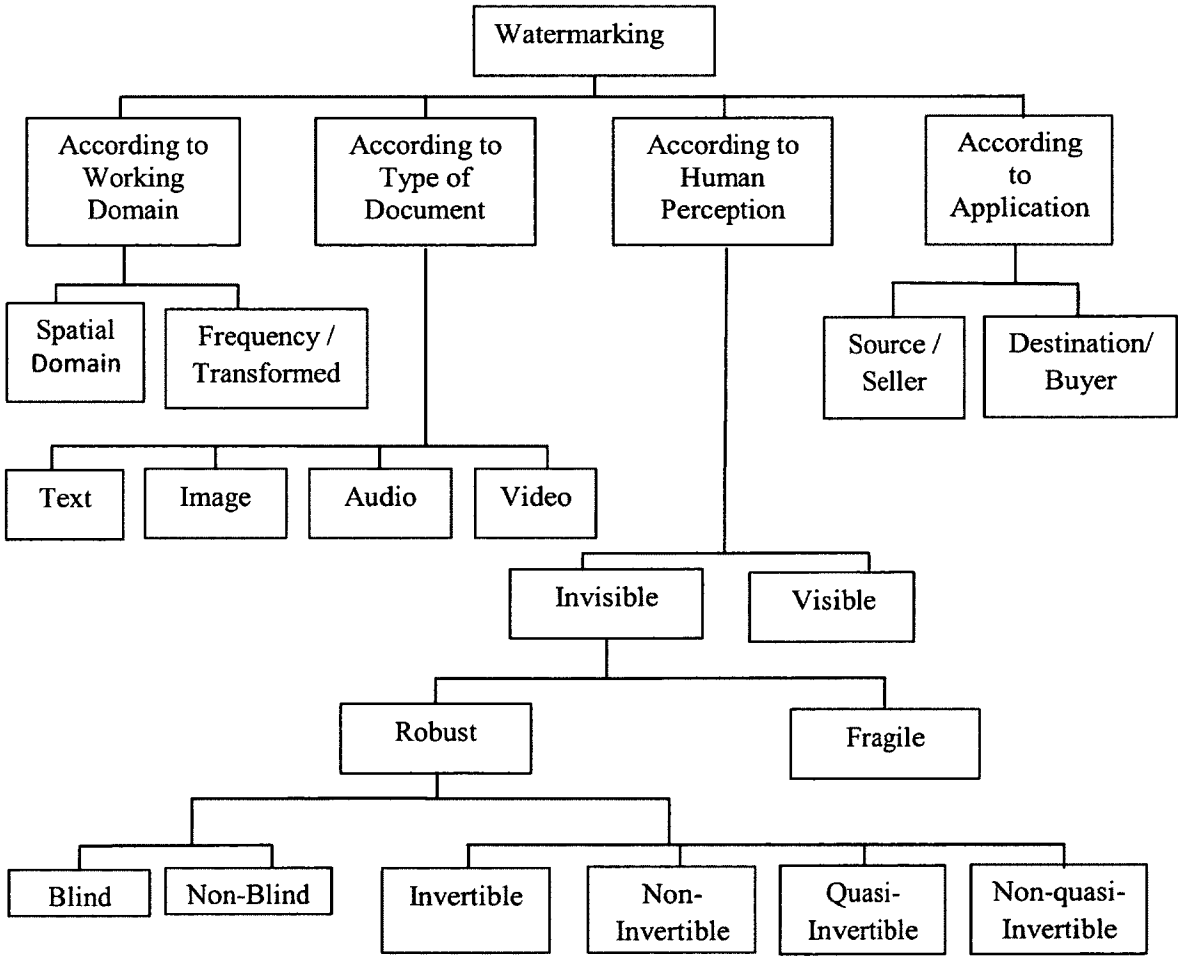


Figure 1.4: Types of watermarking techniques.

By definition, watermark extraction/detection process of an imperceptible robust blind watermarking method does not need to be aided by the original for watermark extraction/detection. However, non-blind watermarking methods do require the original image to help extracting them. The class of invisible invertible robust watermarking method is one where the original image can be obtained by reversing the watermarking procedure. An invisible robust watermarking scheme (E, D, C_δ) is called invertible, if for any watermarked image I_w, there exists a function E^{-1} such that:

$$E^{-1}(I_w) = (I_o, W) \tag{4}$$

$$E(I_o, W) = (I_w) \tag{5}$$

$$C_\delta(D(I_w, I_o), W) = 1 \tag{6}$$

where, E^{-1} is a computationally feasible function, W belongs to the set of allowable watermarks and the images I_o and I_w are perceptually similar. Otherwise, the watermarking scheme is non-invertible. A watermarking scheme (E, D, C_δ) is called quasi-invertible, if for any watermarked image I_w, there exists a function E^{-1} such that,

$$E^{-1}(I_W) = (I'_o, W) \tag{7}$$

$$C_\delta(D(I_w, I_o), W) \approx 1 \tag{8}$$

where, E^{-1} is a computationally feasible function, W belongs to the set of allowable watermarks and the images I_o and I'_o are perceptually similar. Otherwise, the watermarking scheme is non-quasi-invertible.

Digital watermark can be also classified in terms of application as follows:

- Source based
- Destination based

Source-based watermarks are desirable for ownership identification or authentication where a unique watermark identifying the owner (seller) is introduced to all the copies of a particular image being distributed. A source-based watermark could be used for content authentication and to determine whether a received image or other electronic data has been tampered with or not. The watermark could also be destination based where each distributed copy gets a distinctive watermark identifying the particular buyer. In legal selling cases, the destination based watermark can be used to track the buyer.

1.3 Applications of Digital Watermarking

To ensure digital copyright management and protection, digital watermarking has found many different types of applications.

One of the first watermarking applications was the broadcast monitoring. With watermarked media, advertiser can monitor whether or not the commercials they have paid for are aired by the broadcaster according to the contracts. Interested reader, can find details about this procedure [3].

Another significant application of watermarking is owner identification. In such a scheme a robust watermark containing the identification information of the content owner is embedded into the watermarked media as proof of copyright or ownership.

A third application is the transaction tracking. For example watermarking could be used to record the recipient of every legal copy of digital media by embedding different watermarks in each recipient-copy. Then, if a leak or illegal distribution occurred, the distributor can identify which recipient-copy is the source of leak.

A fourth application is copy control which can be achieved with this watermarking technology. For example, watermarking can be used to prevent illegal copying of any digital media, by embedding a watermark in them that would instruct a watermark compatible CD or DVD writer not to write any watermarked media since it is protected with copyright.

Finally, watermark integrity check can be verified if the watermarked media has been altered or not as well as determine the location of alteration.

1.4 Image Steganography

To fight against data security issue, cryptography technology comes with many different solutions, such as applied encryption standard (AES), RSA and secure sockets layer (SSL). However, it has been proven that the best encryption technique may be broken in a manner of time by the steady progress of the art [4]. Therefore, digital steganography contribution has already gained more importance in information security field. Encryption can be compared to keeping of information in a safe vault whereas steganography can be compared to keeping of information in a jungle that people do not know. The word steganography, comes from Greek origin means 'Covered writing'. It also refer to hiding information in digital media in order to hide the existence of the information [5]. In a simple way steganography is an art of hiding secret data in an innocent looking container called cover data. This cover data may be any digital media such as digital image, audio, movie file etc. Usually the embedded secret data is called payload. Once the payload has been embedded into a cover media it may be transmitted to the receiver or posted in public place from where intended receiver can download it such as a webpage. Since the transmission or posting is open, cover media should focus on introducing as little distortion as possible to make it innocent looking or

undetectable under most examinations. Steganography can be use to verify legal and illegal casess; however, every invention contain both of these cases. Watermarking is a protecting technique which protects the author's property right for any digital media by some hidden watermarks within the digital media. On the contrary, steganography protects the confidential data from unintended users; however, the techniques steganography, watermarking and cryptography are interlinked [6]. Moreover, it is very hard to draw a distinction between steganography and watermarking. Table 1 partially taken from reference [6] can help explain the differences.

Table 1 Comparison of steganography, watermarking, and cryptography.

Terminology	**Steganography**	**Watermarking**	**Cryptography**
Container/Carrier	Any Digital media (Cover)	Any Digital media (Cover)	Usually Text or Binary data
Secret Data	Payload	Watermark	Palin Text
Result	Stego-file	Watermarked-file	Cipher-Text
Key	Optional	Optional	Necessary
Input-files	Two	At least two unless in self-embedding	One
Detection/ Extraction	Blind or Non-blind	Blind, Semi-blind, Non-blind	Blind
Objective	Secret Communication	Copyright preservation, Data-Integrity Authentication	Data Protection
Failure	When detected	When Removed	When De-ciphered
Relation to Carrier	No, payload is more important than carrier	Some time yes, Carrier is more important than Watermark	N/A
Concern	Payload Capacity, Sustainability against Steganalysis	Robustness for Non-Fragile Watermarking, Temper detection for Fragile Watermarking	Robustness
Types of Attack	Steganalysis	Image Processing	Cryptanalysis

In this study we have proposed four methods for image watermarking. Two are transformed-domain-invisible robust (non-blind) image watermarking, one is transformed-domain-invisible semi-fragile (non-blind), and one is transformed-domain-invisible-fragile (blind) watermarking methods. The proposed method III and method IV can also serve as image steganography techniques. Our literature review includes more on those articles that are related to our works rather than all types of watermarking and steganography.

Chapter 2

Related Works

Image watermarking methods are generally classified into two broad categories: spatial and transformed approaches. Spatial approach is found to be dissatisfactory since it cannot defy lossy data compression and other image processing. Transform-based technique can embed more bits of watermarks and resist more attacks. First few transformed domain watermarking techniques are based on the discrete cosine transformation (DCT) that have been proposed by Cox *et al.* [2], Koch and Zhao [7], and Hsu and Wu [8]. In this study, four different methods for image watermarking have been developed. Traditional discrete wavelet transformation (DWT) based robust-non-blind has been desived in method-I for grayscale images. In method-II we use DWT and singular value decomposition (SVD) which is also a robust-non-blind watermarking approach. In the literature we also have briefly descrided the strength and weakness of existing DWT-SVD based watermarking methods. Method-II has been further extended for color images; thus, significant color image watermarking methods are also included in this literature review. Method-III use block-DWT-SVD based approach which achieve image fidelity criteria for medical image watermarking. Section 2.4 includes the state-of-art medical image watermarking methods. Table 2 shows categorizations of different watermarking methods in the literature review.

Section 2.5 includes study on image steganography as method IV also works for blind image steganography. However, any steganography system must have to maintain the following characteristics [6]. First, Image fidelity: for non-blind steganography human eyes should not distinguish the difference between the original image and the stego image, and for blind steganography stego image should not introduce any abrupt change in the image texture. Second, the capacity: the more payloads a cover media can carry the better it is.

Table 2 Different methods for different types of images watermaking covered in the literature

Image Type	Methods		References
Gray Scale Image	Robust	Discrete Cosine Transformation (DCT)	[2][7][8]
		Discrete Wavelet Transformation (DWT)	[9][10][11][12][13][14]
		Singular Value Decomposition(SVD)	[15][17]
		DWT & SVD	[16][18][19][20][21][22]
Color Image	Robust	DWT	[24][30]
		DWT-SVD	[25][28]
		DCT	[26]
		DWT-DCT	[27][29]
Medical Image	Robust	DWT	[33]
		Contourlet Transformation	[34]
	Fragile	Module 256/ LSB/ IWT	[35][36][37]
	Semi-Fragile	Integer Wavelet Transformation (IWT) and matrix norm quantization	[38]

2.1 Discrete Wavelet Transformation Based Image Watermarking

One very effective and popular way to transfer spatial domain information into transform domain based information is the DWT. In this transform, the wavelet multi-resolution and direction (horizontal, vertical, diagonal etc) selective decomposition of image exploits the characteristics of a human visual system (HVS) [9]. Moreover, the DWT has been used to embed data in the transformed domain to minimize noise, and to offer extra robustness

against asymmetrical attacks. The fundamental idea for the DWT of a 2-D image is that, it decomposes an image into four parts of high, middle, and low frequency (i.e., *LL* (Low-Low), *HL* (High-Low), *LH* (Low-High), *HH* (High-High)) sub-bands by cascading the image horizontally and vertically with critically sub-sampled filter banks. The sub-bands labeled LL_1, HL_1, LH_1, and HH_1 are attained by first decomposition that represents the finest-scaled wavelet coefficients. The sub-band LL_1 can be further decomposed if it is necessary to obtain the coarser-scaled wavelet coefficients. This decomposition process is repeated a number of arbitrary times which is determined by the application at hand. Dugad et al. [10] used a wavelet transform domain spread spectrum watermarking method, which embed random number sequence in selected coefficients with predetermined thresholds (T_1 = 40) in the transformed domain in order to guarantee non-eras ability of the watermark. We implemented Dugad et al [10] method and noticed that watermarked image has low pick signal to noise ratio (PSNR) when compared with original image (< 40 dB). Kundur and Hatzinakos [11] address non-blind watermarking with multi resolution wavelet decompositions. They embed a binary watermark (which is a sequence of 1 and -1) into the detail wavelet coefficients with the use of a key that determines certain wavelet coefficients addressee of the mark. However, their method found not robust when attacked-watermarked-image's PSNR drop to 30 dB from watermarked-image. Elbasi et al [8] extends the idea of Dugad et al [10], using two bands (*LL* and *HH*). Their experiment shows that watermark embedding in LL sub-band is more robust against JPEG compression, resizing, and low pass filtering. Also, watermark embedding in *HH* sub-band is more robust to resist against histogram equalization and contrast adjustment but their study did not describe how using of both *LL* and *HH* band can overcome the threshold calculation issue. Hajjara et al [13] suggest biorthogonal wavelets and insertion of watermark into HL subbabnds and tested their

method's robustness against only non-geometrical attack. Zahir and Islam [14] have used *LH* & *HL* subbabnds of DWT image to embed watermark but their method lacks of sufficient number of coefficients for some types of images.

2.2 Singular Value Decomposition and DWT-SVD Based Image Watermarking

The singular value decomposition (SVD) based watermarking algorithm was first introduced by Liu et al [15]. The SVD of a square matrix was discovered by Beltrami in 1873 and Jordan in 1874 independently, and extended to rectangular matrices by Eckart and Young in the 1930s [16]. Singular values are the luminance values of any SVD transformed image, changing this values slightly, does not affect the image quality significantly. On the other side, the orthogonal matrices in the SVD of the image are related to the image details [17]. However, SVD based image watermarking is susceptible to quality degradation of the host image and also it is weak against general attacks. Dual transform domain has also been used for embedding the watermark in many recent articles. Li et al [18] proposed a hybrid DWT-SVD domain watermarking scheme with human visual system (HVS) properties. After decomposing the host image into four sub-bands, they applied SVD to each sub-band and embedded singular values of the watermark into the sub-bands. Since they only embed SVs of the watermarks into SVs of wavelet coefficients, this method would have ambiguous watermark problem which has been discussed in details in section 2.3.1. Bhatnagar and Raman [19] used a visually meaningful gray logo instead of a noise type Gaussian noise. They embed watermarks into a reference image by modifying the SVs of reference image using the SVs of the watermark. The reference image has been formed using threshold based directive contrast calculation. As their method use U and V matrix of SVD decomposed

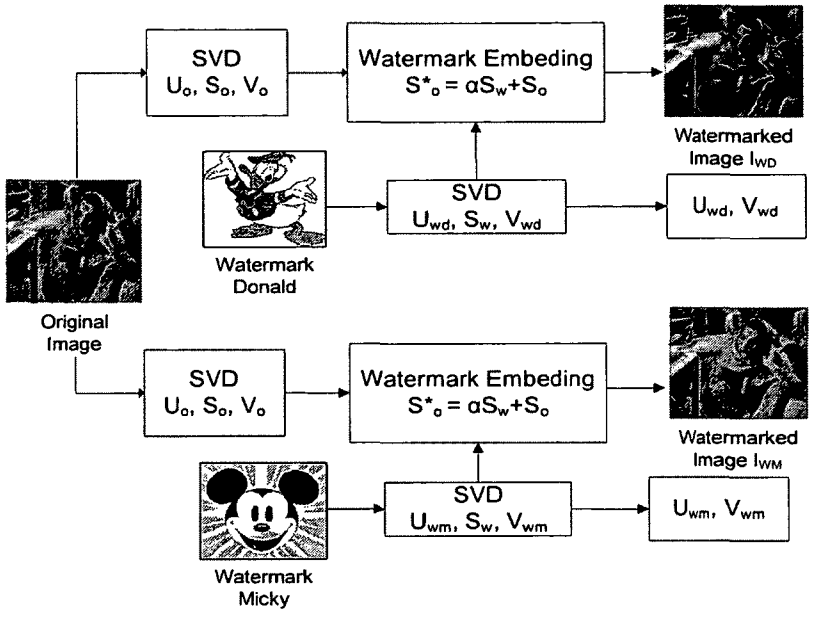
original watermark or logo in their watermark extraction process there may exist false positive ambiguity; moreover, the PSNR between the original and watermarked image was found less than 44 dB. In recent years, Yavuz et al [20] proposed SVD-DWT watermarking scheme which embeds the watermark bit into the left singular vectors (U), however, any image fidility measurement of watermarked image has not been found in this study. With traditional SVD-DWT approach watermark has been embedded in high frequency component in [21]. Emir et al [22] presents another hybrid approach where images are first decomposed into four bands using DWT, then singular value decomposition (SVD) was applied to each band and embedding was performed by modifying the singular values and is susceptible to false positive ambiguity problem. Hence, many DWT-SVD based visual-watermark extraction algorithms suffer false positive detection problem [23] no matter how robust they are.

2.2.1 Problems with Existing DWT-SVD based Method

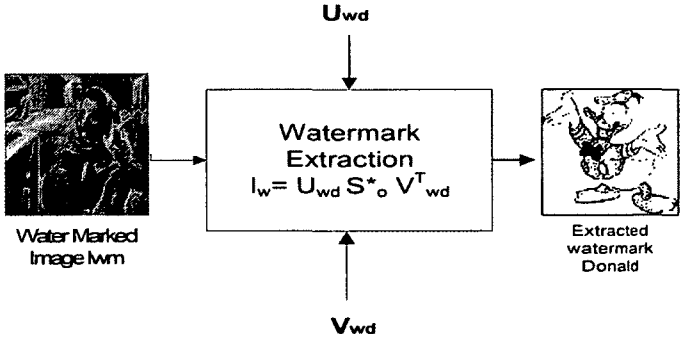
In the watermark detection stage for DWT –SVD based methods, most extracted visual-watermarks are reconstructed using the difference between the singular values of the original image and modified SVD of the watermarked image with the orthogonal matrices (U & V) of original watermarked image. Hence, these methods have the ambiguity problem, i.e. embedded watermark 'Micky' can be detected as a 'Donald'. Figure 1 shows the result of the experiment that we have done, which supports existence of this false positive detection problem.

In the Watermark insertion phase for these types of method, singular value decomposition is applied on the original image or DWT-image. Again, same type of

decomposition is applied on the watermark which is also an image. Thereafter, SVs of the original image are modified by adding the SVs of watermark with some multiplication factor α. Finally, in watermark extraction, S^*_o is just a diagonal matrix of the original watermark and the orthogonal matrices U_{wd} and V_{wd} represent the geometry of a different watermark.



(a)



(b)

Figure 2.1: Watermark embedding with (a) Donald and Micky in and (b) extraction of false watermark

Therefore, in extraction, no matter what value of the diagonal S^*_o takes (diagonally modified watermark) , the resulting watermark geometry will be defined by the U_{wd} and V_{wd}.

2.3 Color Image Watermarking

In [24] Elbasi et al extended their DWT-based gray image watermarking method proposed in [8] for color images. They embed their watermark in the luminance layer of YUV color model. 42.26 dB PSNR has been reported between original Lena color image (512x512) and watermarked image. Yin et al [25] proposed a DWT-SVD watermarking scheme using a scrambled watermark into the green component of the color image. The green component has been decomposed into LL_n, HL_n, LH_n, and HH_n sub-band up to n^{th} level. For each sub-band different embedding method are applied. They use encrypted image as their watermark. 42.82 dB PSNR between original and watermarked image has been reported for 200 by 200 color Baboon image.

Lee et al [26] proposed a YCbCr color model based watermarking method. They opted Cb component to embed their watermark. A logo or small image is combined with seed key using modified Hadamard kernel to generate their watermark and has been embedded into DCT coefficient along a pre-selected zigzag scan line path. With weak watermark strength they achieve a average PSNR of 40 dB between an original and watermarked image. Gui et al [27] proposed a DWT-DCT based watermarking scheme using HSI color model. DCT coefficient of intensity component of the host image has been used for watermark embedding. Before embedding, watermark (a small image or logo) is DWT decomposed and then scrambled to generate a chaos sequence. Intensity component of the host image is decomposed in 8x8 block of DCT. Then the binary watermark sequence is embedded into DCT coefficients. However, no watermark invisibility measurement has been reported in this work. Dharwadkar et al. [28] proposed a DWT-SVD based color image watermarking method using the blue channel of RGB color space. SVs of the watermarks are embedded

into the SVs of all sub-band coefficients. Watermark extraction process may have false positive or ambiguity problem as the extraction process uses the U and V matrix of the original watermark. No watermark image fidility measurement has been reported in this article also. Kapoor et al. [29] used hue (H) channel of HSV color model using DWT and DCT, however, their study did not reported any image fidility measure of the watermarked image. Vahedi et al. [30] used a 64x64 binary logo as watermark and opted Intensity channel 'I' of HSI color model to embed their watermark. In their scheme embedding procedure take place in the 3rd decomposition level of DWT image and 16 non-overlapping block obtained from binary logo (watermark) are inserted into the most preserving wavelet coefficients (optimized by genetic algorithm) such that most important blocks of the watermarks are inserted into most preserving wavelet coefficients.

2.4 Medical Image Watermarking

Nowadays, medical images are produced by a wide variety of digital imaging equipment such as magnetic resonance imaging (MRI), Computed Tomography (CT), computed radiography (CR) and so forth. Copying, editing, and transporting these digital images are much easier than analog images. Usually medical images are stored in electronic patient record (EPR) system. However, these systems collect information over years by numbers of health professionals and are used for different purposes such as patient care, clinical research and Insurance company [31]. So this information needs to be confidential and authenticate. Moreover, volume of medical image is large hence require huge storage capacity. Continuously updated digital imaging and communication in medicine (DICOM) standard provides guidelines to ensure authentication, integrity and confidentiality of medical images

[32]. DICOM has already recognized and included pixel data compression using Joint Photographic Experts Groups (JPEG), JPEG2000 and other compression standards. Insurance companies, patients and health professionals may want to alter the content of these medical images for various reasons. Therefore, protecting medical images against such type threat is necessary. Watermarking can be used as a solution. All medical images watermarking scheme can be distinguished into three broad categories: robust, fragile, and semi-fragile. This section defines each categories and brief review of existing methodology in each sector.

Robust medical image watermarks are designed to resist different image processing attack or manipulation. The fundamental issues with robust watermarking are for copyright protection and content authentication. Many traditional robust methods are transformed domain scheme, where the watermark is spread over a wide range of image frequencies. Giakoumaki et al. [33] proposed a wavelet-transformation based medical image watermarking method. In their method, the host image is decomposed into three levels using Harr DWT. BCH encoded watermarks are embedded into horizontal details coefficient of the second and third level, respectively, which consist of physician's digital signature and patient's personal data. However, for robustness verification they only showed results for JPEG compression. The average PSNR of Watermarked image has been found 46 dB. Rahimi et al. [34] presented a dual and oblivious (blind) watermarking scheme in the contourlet domain. This adaptive method embeds their watermarks with different embedding strength in region-of-interest (ROI) and region-of-non-interest (RONI) of singular value vectors of the lowpass sub-band in contourlet domain. The average PSNR value of whole watermarked image is reported greater than 45 dB as well as SSIM value has been reported

more than 0.96. This method is robust against various image processing attacks. However, this method did not report any tampering detection capability.

Fragile watermark has tampering detection capability. The watermark should be destroyed even if the image is manipulated in the slightest manner, especially if some block of pixels or portion of image is forged. Fragile watermark comes with localization capability of the forged area. Traditional methods embed checksums or watermarks in least significant bit (LSB) plane. Wu et al. [35] proposed a block based medical image watermarking methods where module 256 is opted to achieve information hiding and image authentication. The image has been sub divided into 256x256 pixels, and any forgery is identified in the altered block not the exact location of the forgery. The obtain PSNR of 48.2 dB for their watermarked image. Memom et al. [36] proposed a LSB based fragile watermarking algorithm. Their method separates a medical image (CT scan image of chest area) into ROI and RONI region. Watermark has been generated by converting hospital logo, patient's information and computed hash function of the host image into BCH code. After setting all LSBs of the host image to zero, generated BCH code is embedded into the scrambled pixels in LSBs of RONI region. This method reported a high PSNR value (>55 dB) for watermarked image. Piao et al. [37] proposed fragile watermarking method using integer wavelet transformation (IWT). Most significant bits (MSB) and image information are used as watermark which are converted into a hash value and inserted into the LSB of the *MxM* blocks of *LL* sub-band of IWT image. This method also achieved high PSNR value for watermarked medical images (i.e. more than 50 dB).

Semi-fragile watermarks combine the properties of both robust and fragile watermarks. Like fragile methods, they are capable of localizing regions of an image those

have been altered. Semi fragile medical image watermarking algorithm has been found very scarce. Though there are many robust DWT-SVD based image watermarking algorithms they are not suitable for image integrity verification, i.e. they do not have tamper-localizing capability. Liu et al. [38] proposed a semi-fragile medical image watermarking method using IWT and matrix norm quantization. Their method has been found robust against many image processing attack. However, for tampering detection they have developed a function called Tamper assessment function (T_{AF}) which only can identify whether a tampering has been attempted or not, but does not have tamper-localizing capability.

2.5 Digital Image Steganography

For centuries people were belligerent to develop innovative methods for secret communications. In the 5th Century BC Histaiacus shaved one of his slave's head, tattooed a message on his skull and then the slave was sent to its receiver after his hair grew back [39-40]. However, with the computational power of modern computer digital era of steganography begins. Figure 2.2 shows different types of modern Information security systems.

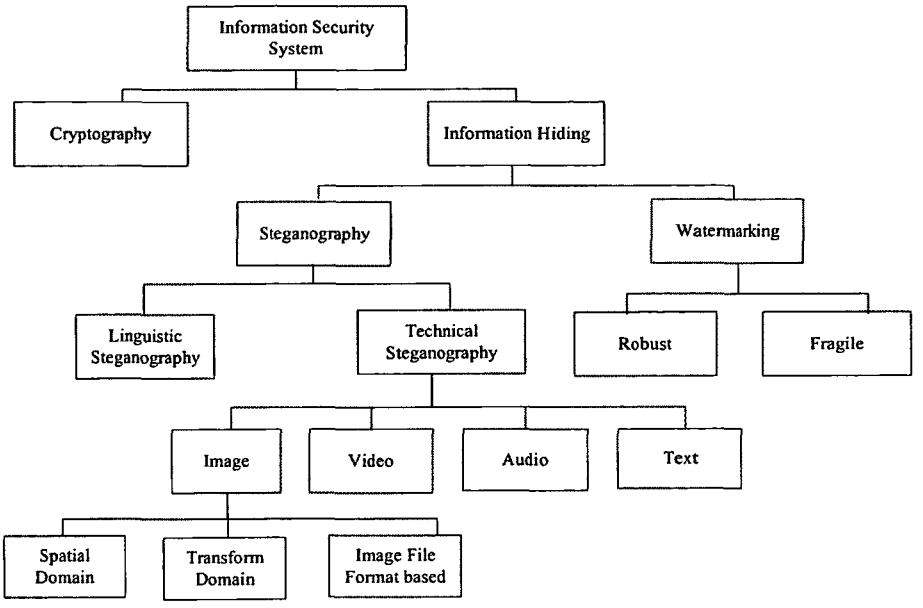


Figure 2.2: The outlet of information security system

Many steganographic schemes have been proposed with various types of digital media, however, this study focus on steganography in digital images. For a detailed survey on steganography in other media the readers are referred to [41]. We can classify the existing digital image steganography into three broad categories: i) Steganography on image file format; ii) spetial-domain based approach; iii) transformation domain based approach.

Due to the existence of different digital image file format, steganography methods have been developed exploiting these file format. Software, such as Camouflage, Jpegx, Data-Stash [42] have used this kind of steganography methods. Some color palette modification based image steganography has been proposed in [43, 44 and 45]. Unfortunately, any change in the palette of indexed images leaves a clear signature which is prone to detection and would not resist any kind of little editing to the stego-image.

In the special domain approach, the secret messages are embedded directly into the image pixels. The most popular special domain approach is Least Significant Bit (LSB)

approach which is improved by several algorithms. Moreover, many staganogarphic tools utilize this technique, such as Steghide, S-Tools, steganos, etc. are available online [43]. Most of this type mechanism replaces the LSBs of the randomly selected pixels of the cover image with the secret message bits [47, 48, 49, and 50] where pixel selections are based on some secret key and achieve high payload, for example more than 50000 bits in 512x512 gray scale images. Chung et al. proposed singular value decomposition and vector quantization based image steganography in [51] and achieved a PSNR 37.002 dB between stego and cover image. However, spatial domain steganography is venerable to blind steganalysis, meaning easily detected by statistical analysis (for example chi-square test). Moreover, this group is also prone to file format changing which annihilates the stego message.

In the transformation domain approach, the secret message is embedded into the coefficients of discrete cosine transformation (DCT), Discrete Fourier Transformation (DFT), discrete wavelet transforms (DWT) and similar transformation. Despite medium or low payload capacity of this type scheme they are gaining popularity due to their sustainability from most blind steganalysis which is currently most important factor for any steganography method. Most of the DCT transformed domain steganography approach exploits the redundancies in the DCT domain which is also the base of JPEG compression. Li and Wang et al. presented a stego method by modifying the quantization table (QT) and adding the hidden bits in the middle frequency components [52]. Iwata et al.[53] and Chang et al. [54] and also done similar work by modifying the QT and adding hidden data in the successive zero sequence of middle and high frequency components respectively. Chang et al. achieved PSNR between 26.46 dB and 40.55 dB with payload capacity between 4096 to 36710 bits.

They also show that their stego method was undetected by chi-square steganography test program provided by Guillermito [55]. However, in DCT domain researcher are limited to exploit those coefficients, as many of the 64 coefficients are equal to zero and changing this zero to non-zero values will have an effect on the compression rate. Two most DCT coefficient modification based data hiding are JSteg and F5 which take advantage of JPEG compression algorithm.

Raja et al.[56] claims, due to round off errors presence, fast Fourier transformation is not a suitable candidate for image data hiding; however, Johnson et al.[57] and Mckeon et al. [58] disagreed with them and employed the 2D DFT to generate Fourier based steganography for image and movies, respectively. Though DWT based methods have been exploited for digital watermarking, they are still in its infancy for steganography. However, few of the proposed DWT based steganography already have shown its strengths over other transformations since they resist JPEG compression.

Chen et al. [59] proposed a DWT based image steganography scheme where they embed their secret bit in the high frequency components of the DWT decomposed image using 2 LSB substitutions with wavelet coefficients. They apply some basic mathematical operation on the secret bits and according to those secret bit information they embed those bits in *HH* sub-bands. In their scheme, *LH* and *HL* sub-bands are used for extended embedding depending on secret bits (with high payload). They report stego-image with PSNR difference in the range 39.0033 dB to 54.94 dB from cover image, however, they did not report any staganalysis on their method. Driskell et al. [60] also achieve high image fidelity using D4 Daubechie wavelet filter by substituting wavelet coefficients that fall below a threshold with coded letters. However, they do not report any steganalysis on their method.

Sajedi et al. [61] proposed an adaptive contourlet based image steganography. The key idea is based on changing contourlet coefficients values with proportional regional values and their algorithm was undetected with most steganalysis methods with a very high PSNR in the range of 31.08 to 41.58 dB. However, we have not found any DWT and SVD based Image steganography that have used QR code either as watermark or as payload.

Chapter 3

DWT, SVD and Image Quality Measures

This chapter presents an overview of the mathematical and statistical techniques used in this research.

3.1 Discrete Wavelet Transformation (DWT)

Wavelet is mathematical function that divides data into different frequency components and then study each component with a resolution matched to its scale. Each wavelet family begins with a father wavelet φ(t), or scaling function, that satisfies a dilation equation of the form [62]:

$$\varphi(t) = \sum C_k \, \varphi(2t - k) \tag{9}$$

The constant C_K are known as the refinement coefficients [63]. For the Haar wavelet transformation (which is same as db1 of Daubechies) family $C_0=C_1=1$ with other refinement coefficients being zero. The mother wavelet function $\psi(t)$ is then derived from the father wavelet and wavelet offspring functions are then defined based on parent [63]. So wavelet function is a band-pass filter (high and low) and conventional scaling function where for each level halves its bandwidth. From filtering perspective, the filtering process incorporates two filter, *H*, the low-pass filter, and *G*, the high –pass filter [64]. However, both of these filters

are themselves signals, which in case of most wavelets are zero everywhere in a short finite stretch. These low and high pass filters can be defined as:

$$(H_s)_k = \sum h_k - 2_j S_j \tag{10}$$

$$(G_s)_k = \sum g_k - 2_j S_j \tag{11}$$

Here h_k and g_k are filter coefficients. In the wavelet decomposition process both filters are applied to the original signal. The low-pass operator H has filter coefficients based on the coefficients C_k, obtained simply by dividing the C_k by $\sqrt{2}$. The high pass filter G has coefficients g_k which are calculated from the low-pass filter coefficients h_{1-k} by :

$$g_k = (-1)^k h_{1-k} \tag{12}$$

The filtering process also requires additional values before and after signal, for example zeros padding. Once a signal is padded and filtered it is said that the signal has been decomposed into its wavelet coefficients.

Any one dimensional wavelet transform can be extended into its 2D form. The simplest 2D- DWT is Haar which is same as db1 of Daubechies wavelet transformation which consists of two operations: one is the horizontal operation and the other is the vertical operation. A simple conceptual algorithmic procedure of a 2D signal for db1 of Daubechies can be illustrated in figure 3.1 [65].

At first this process scan the pixels from left to right in the horizontal direction and perform addition and subtraction operation on each pair of neighboring value left to right. All summation goes to the left and difference goes to the right as illustrated in figure 3.1(a).

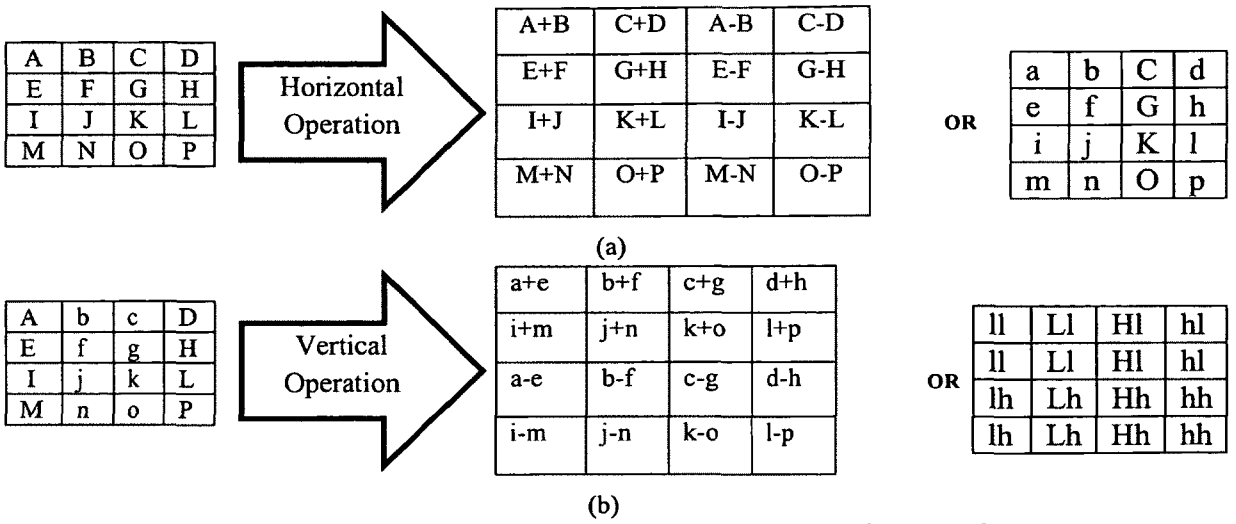


Figure 3.1: The conceptual horizontal and vertical operation of 2D DWT [65]

In the second stage, this process scan the pixels from top to bottom in the vertical direction and perform addition and subtraction operation on each pair of neighboring value top to bottom. All summation goes to the top and difference goes to the bottom as illustrated in figure 3.1(b). After two of these sequential phases we finally obtain 4 sub-bands defined as *LL*, *HL*, *LH* and *HH*, among which *LL* sub-band is the low frequency components and contain most energy of the original image. Figure 3.2 shows a real one level DWT sub-band decomposition of a portion of image.

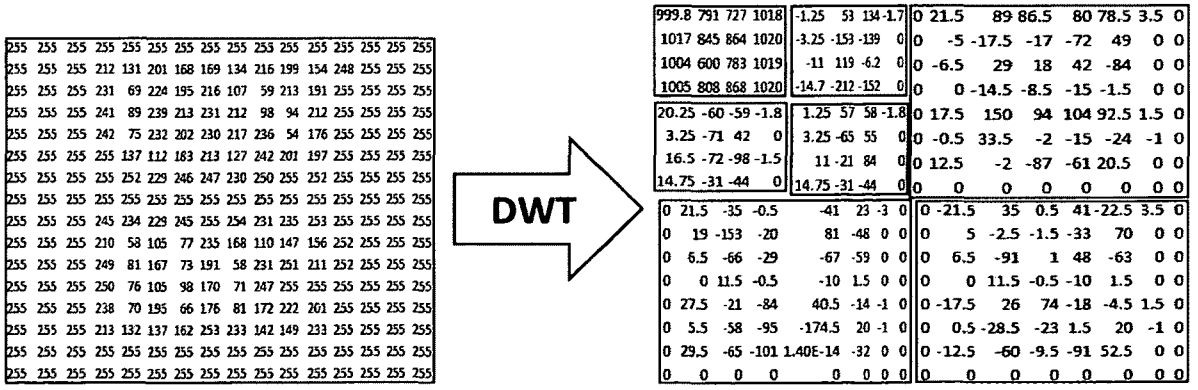


Figure 3.2: 1st and 2nd level DWT decomposition of a portion of image using db1filter.

The power of wavelet comes from the use of multiresolutions, rather than examining entire signals through the same window, different parts of the wave are viewed through different size windows [66]. To use wavelet transformation as watermarking tool we first need to understand the image compression mechanism using wavelet transformation. We shall outline the basic steps that are common to all wavelet-based image compression algorithms. These steps are reversible and therefore we can assume that it is lossless. The only lossy part is the quantization part. Quantization in wavelet transforms will produce approximations to the images when an inverse transform is performed. Thus, creating the error is inherent in lossy compression. The final step is encoding the quantized information. Usually encoding is not done with only one method, rather it combins different encoding methods such as Huffman, arithmetic, run length encoding, etc.

It is well known that DWT has strong relations with multiresolution analysis and filter banks [67]. The output of each low pass filter feeds the new pair of filters in the next level of decomposition. Sub-band *LL* corresponds to the lower frequencies and sub-band *HH* corresponds to higher frequencies and other two sub-bands are the intermediate frequency sub-bands.

The information in natural images is predominant at low frequency component. Moreover, the significance of the coefficients decreases as the level decreases inside the transformed image [67]. The EZW coding scheme is strongly based in the high probability that the coefficients of the 2D structure of the decomposed image, at the higher levels of decomposition are more important than those of lower levels. Many wavelet coefficients are close or equal to zero and hence their upper level decompositions are not that significant. By

exploiting these redundant number of zero coefficients or near to zero coefficients EZW work. The above observance was first reported by Shapiro [68].

In a simple word if a wavelet coefficient at a course scale is insignificant (less in mode value), then all wavelet coefficients of the same orientation in the same spatial location at finer scale are likely to be insignificant and are replaced with zero in JPEG2000 image format [80]. Figure 3.3 shows some significant (in red color) and insignificant (in green color) coefficient values.

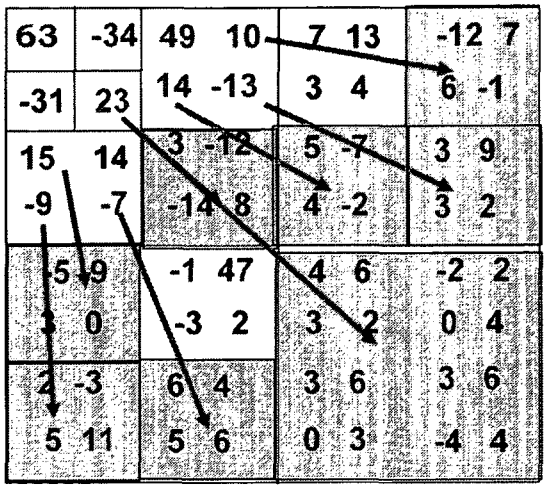


Figure 3.3: Significant and insignificant coefficients

3.2 Singular Value Decomposition (SVD)

Matrix decomposition is one of the most functional initiatives in the theory of matrices. Singular Value decomposition has already established itself as a powerful matrix decomposition technique and has many significant properties that are useful in many image-processing applications. In simple terms, singular value decomposition of any rectangular matrix is a factorization of the matrix into product of three matrices.

The SVD theorem states:

Consider a unitary matrix I (such as an image matrix) with m rows and n column, with rank r, and $r \leq \min(n, m)$. Then I can be factored into:

$$I = USV^T \tag{13}$$

where I is any matrix of real number of size $m \times n$ of an image. U and V are left and right singular matrices (UU^T = Identity, VV^T = Identity) of size $m \times m$ and $n \times n$ respectively and S = diag ($\sigma 1, \sigma 2, \ldots, \sigma r$) with size $m \times n$, is the diagonal matrix with r (rank of I matrix) nonzero elements called singular values of I.

The eigenvectors of $I.I^T$ form the columns of the matrix U and the eigenvectors of $I^T.I$ form the columns of V. The singular values (SVs) are the square roots of the eigenvalues of $I^T.I$ or $I.I^T$. Also, the singular values in S are arranged in descending order. The decomposition can be best illustrated with an example shown in figure 3.4. Let consider a 3x3 matrix I_{3x3} that can be decomposed into USV^T matrices with U_{3x3}, S_{3x3} and $V_{3x3}{}^T$.

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} -0.32 & -0.31 & -0.89 \\ -0.64 & -0.62 & 0.44 \\ -0.69 & 0.72 & 0.00 \end{bmatrix} \times \begin{bmatrix} 5.28 & 0 & 0 \\ 0 & 1.03 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} -0.43 & -0.56 & -0.69 \\ -0.80 & -0.10 & 0.58 \\ 0.40 & -0.81 & 0.40 \end{bmatrix}$$

$$\mathbf{A}_{3\times3} \qquad \mathbf{U}_{3\times3} \qquad \mathbf{S}_{3\times3} \qquad \boldsymbol{V}^T_{3\times3}$$

Figure 3.4: Example of decomposing A to USV^T

From linear algebra view point, the digital image is an array of nonnegative scalar entries that is regarded as a matrix. There are three properties of SVD that have significance in image processing application.

i) A small distortion in the image quality does not change SVs of image matrix significantly [69].

ii) Each SV stipulate the luminance of an image layer, whereas the other pair of singular vectors indicates the geometry [70].

iii) SV also has intrinsic algebraic image properties [70].

More analysis of geometric distortion on the singular values of an image is provided in [16]. They can be summarized as follows:

Transpose Analysis: Any matrix I and its transpose I^T produce the same SVs.

Rotation Analysis: Any matrix I and I_r (I rotated by an arbitrary degree) yield the same non-zero SVs.

Row and Column flip: Any matrix I, row flipped I_{rf}, as well column flipped I_{cf}, yields the same non-zero SVs.

Translation: Adding rows or columns of zeros to any matrices I, also yields the same non-zero SVs.

The above properties of SVD method make it a significant tool for image processing.

3.3 Types of Watermark in this Thesis

In this research, we used two forms of watermark. The first is PRN and the second is the QR code.

3.3.1 Pseudo Random Number (PRN)

If the seed to the PRN is known, the same sequence of numbers can be regenerated with PRN generator. Another advantage of PRN sequence is that it allows the detector to statistically check the presence or absence of a watermark. However, embedding a meaningful watermark is essential is many applications such as a binary image, logo or text. Binary image or logo

are considered as visual watermark and in most cases they are extracted from watermarked image rather their existence is checked statistically.

3.3.2 Quick Response Code (QR Code)

QR code was invented by Denso Wave (a division of Denso Corporation) and Toyota in 1994[71]. It is one kind of 2- dimensional code with control points which makes it easier to be interpreted by scanning equipment such as iPhone, Digital Camera and hand held scanner. Moreover, the error correction capability of QR code makes it ideal one for watermark or steganography. QR code ranges from version 1 to version 40 where each version has a maximum data capacity according to the amount of data, symbol type and error correction level. For different version of QR code there are different module configurations where modules refer to the black and white dots which construct the QR Code [71]. Version 1 QR code is designed to be a 21x21 array of data elements [72]. The largest standard QR Code is a version 40 symbol that has 177x177 modules and can hold up to 4296 characters of alphanumeric data (theoretically) compared to 25 characters of a version 1 QR Code [72]. Table 3 shows a version 2 (25x25 modules) QR code with the left side text (hidden message) encoded into it.

Table 3 Sample QR code version 2 and hidden message

QR CODE	Message
	UNBC Computer Science

QR code module 2 – is an extended form of module 1 which is an open system application of QR code. QR code can operate in six different modes: numeric mode, alphanumeric mode, 8 bit byte mode, Kanjii mode, mixing mode, and structured append mode. Alphanumeric numeric mode is convenient to represent user or organization information so in our proposed watermarking methods we have selected only this mode with version 2 symbol which is sufficient enough to store any organization's information (47 characters). Figure 3.5 illustrates the structure of a version 2 QR code [72].

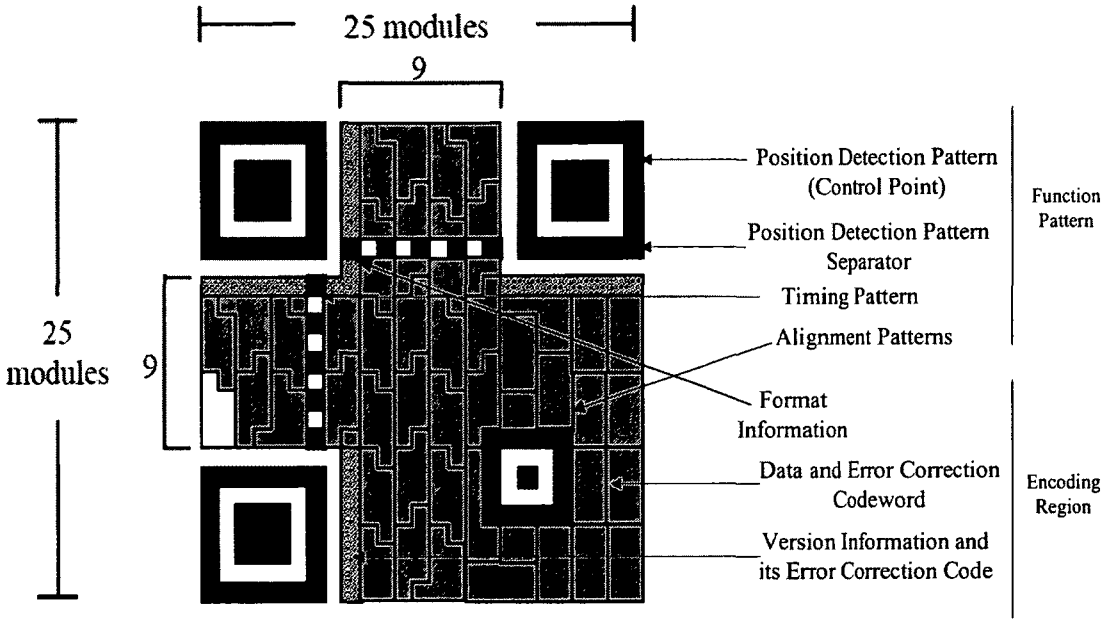


Figure 3.5: Structure of a version 2 QR code [72]

Alphanumeric mode can encodes a data from a set of 45 characters, i.e.10 numeric digits(0-9, ASCII values 30_{HEX} to 39_{HEX}), 26 alphabetic characters (A-Z, ASCII values 41_{HEX} to $5A_{HEX}$), and 9 symbols(SP, $, %, *, +, -, ., /, :). In QR code each codeword is 8 bit long and use the Reed–Solomon error correction algorithm for four error correction levels. Data capacities with error correction level for version 1, 2, 3 symbols are tabulated in table 4 [72].

The error correction capacity of QR code can correct two types of erroneous codeword: erasure and errors. In other word, erasure is a damaged or undecodable symbol character whereas error is a misdecoded symbol character. The number of erasures and errors correctable is given by the following formula [72]:

$$e + 2t \leq d - p \tag{14}$$

where e is the number of erasures, t is the number of errors, d is the number of error correction codeword and p is the number of misdecoded protection codeword. For example in a version 2-L symbol there is a total of 44 codewords, of which 10 are error correction codewords (leaving 34 data codewords). The 10 error correction codewords can correct 5 misdecoded words or substitution errors, i.e. 5/44 or 11.3% of the symbol capacity. However, for QR code error correction levels ISO standard define 4 designated classes which are L(7%), M(15%), Q(25%), and H(30%). For detailed information on QR code interested reader can look on technical specification for QR Code in the ISO-18004 [72].

In our proposed watermarking method III and method IV we opt QR code as our watermark due to the following reasons:

i) 25 x 25 symbol QR code can definitely contain more information than a 25x25 pixel binary image.

ii) The use of QR code is free of any license.

iii) QR code can be restored even if the code is partially damaged. A maximum of 30% codeword can be restored.

iv) QR code can be easily encrypted using DES algorithm.

Table 4 Data Capacity of version 1, 2, and 3 QR code with error correction levels [72]

Version	No of Modules	Data Capacity		Level of Error Correction
		Numeric	Alphanumeric	
1	21x21	41	25	L (7%)
		34	20	M (15%)
		27	16	Q (25%)
		17	10	H (30%)
2	25x25	77	47	L (7%)
		63	38	M (15%)
		48	29	Q (25%)
		34	20	H (30%)
3	29x29	127	77	L (7%)
		101	61	M (15%)
		77	47	Q (25%)
		58	35	H (30%)

3.4 Image Fidelity

Any watermark or stego image is assumed to be identical to the original image as long as the PSNR value between original and stego image is kept higher. The required high fidelity render watermarking in images makes it more critical task than steganography methods where much higher perceptual distortion is tolerated. Chen et al. [73], reported from studies of lossy image compression, a minimum PSNR of 40-50 dB is required for medical images, while 20-30 dB PSNR is acceptable for other multimedia images. The PSNR in dB can be defined as:

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}} \tag{15}$$

where mean-square error (MSE) is defined as :

$$\mathrm{MSE} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(X_{i,j} - \overline{X}_{i,j})^2 \tag{16}$$

where $X_{i,j}$ and $\overline{X}_{i,j}$ are the gray level values of pixels in the host image and watermarked image respectively. m and n are the dimension of the original image. Although PSNR's have

a highanalytical tractability but it does not consider human visual sensitivities. To better evaluate image perceptual quality with another image we also apply structural similarity (SSIM) index which is based on windows rather than single pixels. This metric is idle for testing the similarities of images because it focuses on local rather than global image similarity [74]. The SSIM index is defined as [75]:

$$SSIM = \frac{4Cov(I,C)\mu_I\mu_C}{(SD_I^2+SD_C^2)(\mu_I^2+\mu_C^2)} \tag{17}$$

Where $Cov(I,C)$ indicates the covariance between images I and C inside the local window, μ_I and μ_C the mean values of the windows in images I and C, SD_I and SD_C indicates the standard deviations of the values in the windows of I and C. Hence, SSIM is a product of correlation, luminance similarity, and contrast and ensures human observer inspection very well, within 0 to 1 range, where 1 means perfect quality.

3.5 Robustness and Tampering Localization in watermarked Image

We already know that there are different types of watermarking scheme, such as robust, semi-fragile, and fragile. Robustness refers to the fact that embedded information should be reliably detectable or extractable even after alterations of the watermarked data (i.e., attacked watermarked image). For robust watermarking methods robustness test is an important criteria for the acceptability of such type of methods whereas for fragile watermarking methods it is necessary that there should be a tampering-detection or tempering-localization mechanism. Since our method I and method II are in robust group, method III algorithm in semi-fragile group, and method IV in fragile group we tested our first two method's performance against different standard attacks commonly known as StirMarks [76]. For the semi-fragile and fragile method (method III & method IV) we have devised a tampering-

detection and localization mechanism which is described in the next chapter (Proposed method).

Following are the attacks those are included in StirMarks attacks. However, a difference between intentional and unintentional attacks not has been shown.

JPEG Compression – JPEG is the most widely used image compression standard and any robust watermarking scheme should be resilient to some degree of compression.

Scaling – Scaling can be divided into two groups: uniform and non-uniform scaling. In uniform scaling alteration of horizontal and vertical direction maintain the same scaling whereas in non-uniform scaling uses different scaling factors in both directions (change of aspect ratio). Watermarking method resilient to non-uniform scaling is very scarce.

Rotation – Rotate an image with small angle with or without cropping combination. However, it does not change the commercial value of the image but can destroy the watermark.

Low pass filtering - These are linear and non-linear filters. Commonly used filters are median, Gaussian, and standard average filters.

Noise addition - In the communication and signal processing field noise addition and uncorrelated multiplicative noise have been mostly addressed. Authors often claim that their watermarking techniques survive this kind of noise but many forget to mention the maximum level of noise that their method can sustain.

3.6 Steganalysis

The main idea of steganalysis is to conclude whether or not a suspected cover is embedded with secret message, however, Katzenbeisser et al. [77] also urge robustness for steganography which he categorized within steganalysis. Nevertheless, scholars differ about the importance of robustness in steganography system. Cox et al. [2] suggested not considering robustness in the steganography and not categorized it as steganalysis. We also support Cox's view in this regard. Before any steganalysis process start it is imperative that stego image is near identical to cover image which is known as image fidelity. Stego-image must not contain any visual artifacts of embedding especially in homogeneous region of image. Many LSB and DCT based steganography methods leave noticeable distortions in smooth or homogeneous areas of an image. For image fidelity measurement peak signal to noise ratio (PSNR) has already established as an accepted standard. Similarity of both stego-image and cover image histogram also counted as steganalysis tool [78]. However, in blind-steganography, the receiver does not have the original cover image. Hence measuring PSNR between stego and cover image would not be possible. Last but not the least, another most popular blind steganalysis tool is chi-square test which can detect the presence of steganography for JSteg and F5 methods [6]. Westfield and Pfitzmann [79] claim that the pixels in images are not completely random. Rather, they report, the frequencies of each of the two pixel values in each pairs of value (POV) tend to lie far from the mean of the POV. In simple words, it is scarce for the frequency of the pixels value $2k$ to be nearly equal to the frequency of pixel value $2k+1$ in a typical image with no embedded information. The chi-Square attack has been designed to detect these near-equal POVs in images and produce the probability of embedding of even pixel values and their corresponding odd pixel values in the

stego-image. The probability of embedding is near 1 if the image is expected to have embedded information and is near 0 if is unexpected to have embedded information. For steganalysis Chi-Square Statistic, with *n-1* degree (one less than the number of different categories in the histogram for that image) of freedom can be defined as [80]:

$$\chi^2_{n-1} = \sum_{i=0}^{254} \frac{(x_i - z_i)^2}{z_i}, \qquad \text{Where } z_i = \frac{x_i + y_i}{2} \tag{18}$$

Where x and y are two vectors such that x_i= frequency (i) and y_i= frequency $(i+1)$. The final step of this steganalysis is calculating p, the probability of embedding and can be calculated as the integration of the density function [79]:

$$p = 1 - \frac{1}{2^{\frac{n-1}{2}} \Gamma^{\frac{n-1}{2}}} \int_0^{\chi^2_{n-1}} e^{-\frac{u}{2}} u^{\frac{n-1}{2}} du \tag{19}$$

where Γ is the Euler Gamma function. However, it is observed that chi-square test will fail if the image is noisy or each pixel value (0-255) is uniquely distributed in an image. With such type of image even without any message embedded in it chi-square test will fail and yield probability near equal to 1.

Chapter 4

Proposed Methods

This chapter describes our 4 proposed digital image watermarking methods of which method III and method IV can also serve as image steganography techniques. We use Daubechies wavelet as our embedding tool and pseudo number (PN) sequence from Gaussian distribution as our watermark for method I and method II which allows detection process to statistically check the presence of watermark. QR code are introduced as watermark for method III and method IV, a new avenue for information hiding.

4.1 Watermark Embedding Method I

In our proposed method-1, we have used DWT with Biorthogonal Daubechies filter as our embedding tool since the leading edge JPEG2000 image file use Biorthogonal Daubechies filter for coding image file [80]. Besides, protecting digital copyright in digital contents is not enough; a secure watermarking is desirable to protect the participants (Buyers and Sellers) in a digital content transaction. Therefore, watermark is generated using a pseudorandom sequence (PRS) generator, which is initialized with a seed that depends on a secret key (different for sellers and buyers). This study does not express protocol details rather the frame work for the watermark within watermarked image. The two different marks is inserted into the horizontal (*HL*) and vertical (*LH*) sub-bands (i.e. *HL* for buyer and *LH* for sellers) of wavelet coefficient, since in the frequency domain, the contrast-sensitivity-function of HVS depend on frequency and orientation details (horizontal and vertical details)

[2]. There is also evidence that HVS is less sensitive to removal of smaller details (*LL* sub-band) and embedding watermarks in these sub-bands may degrade the image significantly. Embedding in the low frequency sub-bands, however, could increase robustness significantly. Moreover, high pass bands (*HH*) typically contain edge related information of the image; each coefficient in the high frequency bands affects only a spatially limited portion of the image. Therefore, adding the watermark to significant coefficients in the high frequency bands is equivalent to embedding the watermark to only the edge areas of the image which also contains the majority of the noise in the image. Furthermore, when a watermarked image has to be compressed with higher compression ratio, the embedded watermarks may be seriously destroyed. Despite *LL* contains more significant coefficients than any other sub-bands embedding watermark in *LL* for more than one level is self-destructive and will destroy the image quality, and *HH* typically contains more edge related information of the image, then the *HL* and *LH* pair is a better sub-band selection. It is observed that vertical and horizontal channels have noticeable energy (large values) in its first 3 level decomposition. Exploiting this factor, 3 level decomposed *HL* and *LH* sub-band/channels are chosen. Figure 4.1 shows the sub-bands where watermarks are embedded for method I. We embed our watermark into two sub-bands namely *HL* & *LH* of the Daubechies wavelet of the grayscale image.

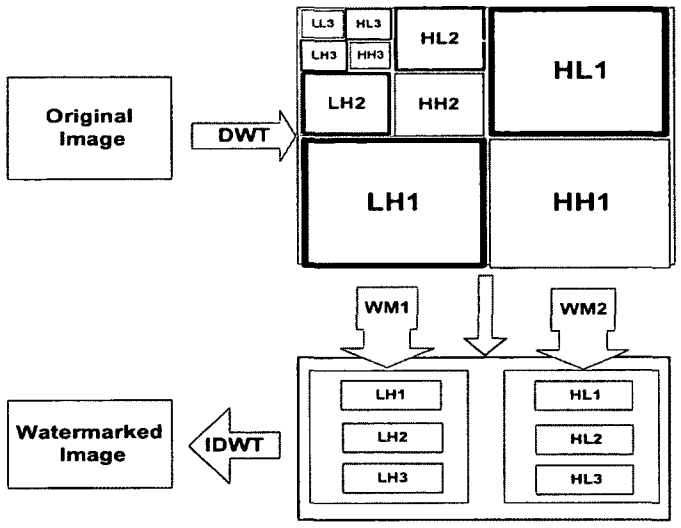


Figure 4.1: *HL* and *LH* sub-bands for watermark embedding

In Discrete Wavelet Transformed (DWT) image watermark (unique sequence of real numbers) has been embedded using coefficient that are near the largest coefficient (*T1*) value of that sub-bands of a given image to increase attack tolerance. Since JPEG2000 image format use Zero Wavelet Tree (ZWT), Opting threshold value near to highest coefficient ensures watermark existence even after high compression ratio.

The following equation is used for watermark embedding.

$$X_i = X_i + \alpha |X_i| W_i \ , \tag{20}$$

where X_i are all significant DWT coefficients, such that i runs over all DWT coefficients greater than T_1, where T_1 can be calculated as :

$$T_1 = 2^{log_2(Higest\ Coefficients-1)} \ , \tag{21}$$

and W_i is defined as:

$$W_i = W(i \mod lp) \ , \tag{22}$$

where lp is a long prime number.

W_i is generated from a uniform distribution of zero mean and unit variance. α is taken as 0.4, 0.3, and 0.2, respectively, for each level of decomposition from 1st to 3rd since there is gradual increase in energy in higher level of decomposition. Furthermore, more robustness can be achieved through two or three times embedding of the same watermarks in the *LH* and *HL* sub-bands; however, in this proposed study we apply one time embedding. Figure 4.2 shows the block diagram of the watermark embedding process.

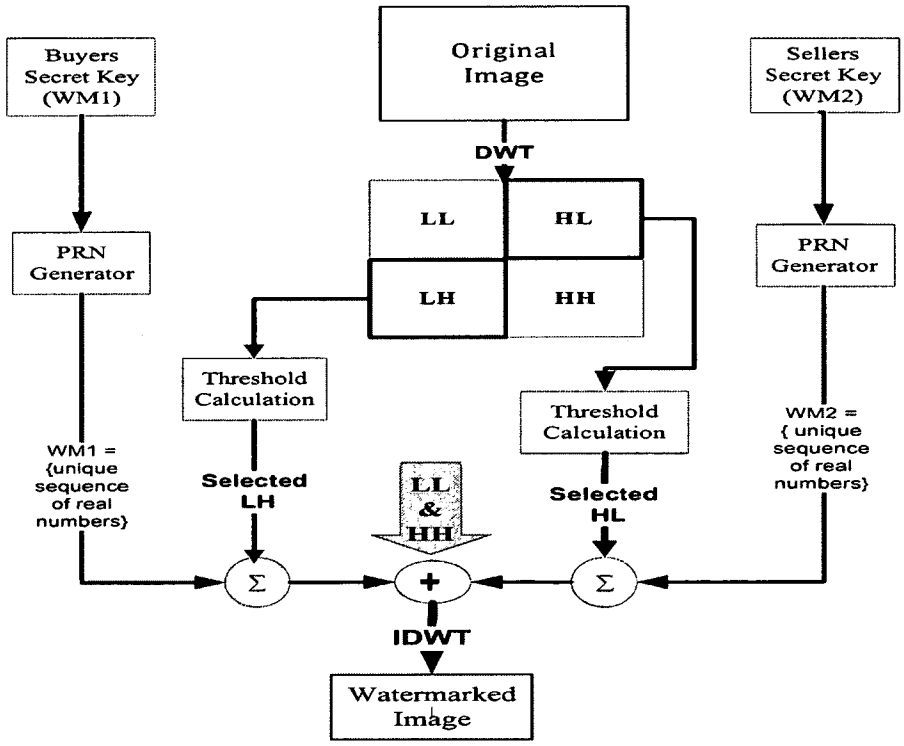


Figure 4.2: Block diagram of proposed watermarking method I

4.1.1 Watermark Detection for Method I

For watermark detection for method I we have implemented our detection function as a correlation detector and a similarity vector measurement. The correlation C between the

difference of watermarked and original images DWT coefficients, d, of the corrupted watermarked image and the original watermark W is computed as:

$$C = \frac{1}{R}\sum d\, W_i, \tag{23}$$

where i run over all significant DWT coefficients and R is the rank of HL or LH matrices.

Besides, to calculate the similarity, the proposed method also utilizes the vector projection defined by Cox et al [2] described in the following equation:

$$Z(w[i], w^*[i]) = \sum w[i].w^*[i] / \sqrt{\sum w^*[i] w^*[i]}, \tag{24}$$

where $w[i]$ is the original watermark and $w^*[i]$ is the extracted watermark. When the similarity measurement value between original watermark and extracted watermark is a way greater than other similarity measurement value, then, we can conclude that the watermark is present.

4.2 Watermark Embedding Method II

For some images such as Barbara, LH or HL subbabnds of DWT images suffer of less number of significant coefficient that are above the threshold (T_I) to embed watermark for method I. To overcome the lack of sufficient number of significant coefficient in method II we introduce singular value decomposition with DWT for watermarking. The image energy spreads over all the singular values in the presence of random texture and first singular value dominates all the others in the presence of smooth regions [16]. Moreover, The use of SVD with DWT has several advantages: (i) watermarks are hidden in two transformed domain; and (ii) watermarks in the watermarked image are less affected even if many general image-

processing. However, the proposed method does not defy rightful ownership problem, i.e., if an owner has more than two watermarks in the image both watermarks will be detected. Figure 4.3 shows the process of watermark embedding.

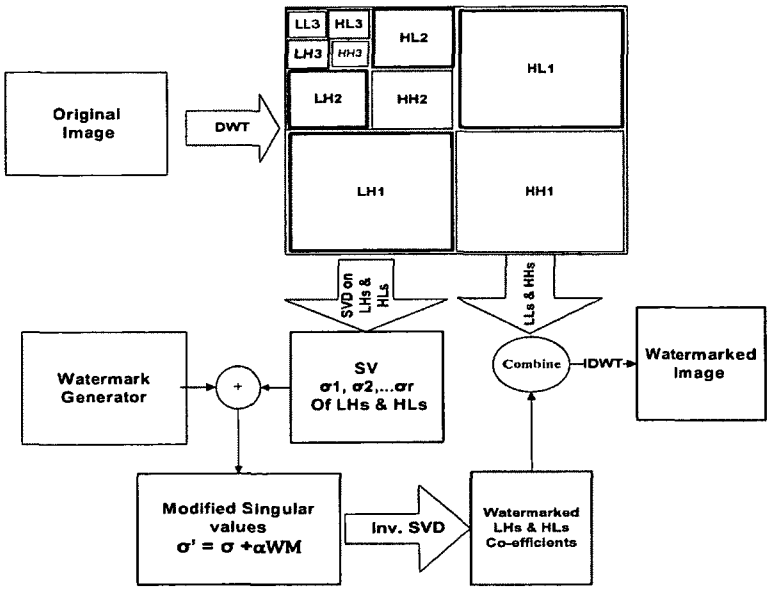


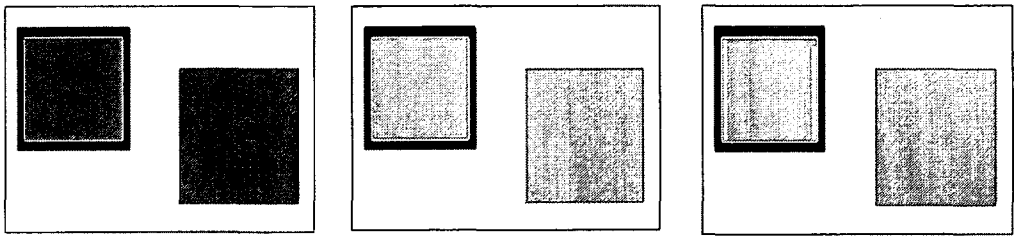
Figure 4.3: Block diagram of the watermark embedding process method II

In this method, 3 level DWT is applied to the original image and HL_1, LH_1, HL_2, LH_2, HL_3 and LH_3 sub-bands are obtained. The watermark (unique sequence of real numbers) has been embedded in the SVs of all *HL* and *LH* sub-bands. Since our watermark is a sequence of real numbers and they are embedded only in the diagonal values of singular values arranged in descending order, this algorithm works well for every types of images. The following equation is used for watermark embedding.

$$\sigma_i^{'} = \sigma_i + \alpha\sigma_i W_i \tag{25}$$

where W_i are random number sequence or watermarks, σ_i are all nonnegative singular values and α is a multiplication factor (0.05 , 0.025, and 0.0125) which differs in values for

different levels of decomposition to hide our watermark in different level. We have not used a single value of α for each level to avoid our watermark to be visible in the watermarked image especially when there is large enough color invariant region, for example, flat wall or something like that. Watermark with different and fixed value of α for 3 level of decomposition is shown in Figure 4.4.



(a) (b) (c)

Figure 4.4: Original Image (a) watermarked image with different value of α (b) watermarked image with fixed value of α (c).

In our method W is an array of unique random numbers and the size of that array is defined by a long prime number lp. Each Wi is defined as:

$$W_i = W(i \mod lp) \tag{26}$$

Where lp, a long prime number, and $i= 1,2, \ldots lp$. Once the value of i is equal to lp it is set back to 1to ensure that water mark index do not exceeds the watermark array index.

The embedding process can be formulated as follows:

1. DWT is Applied to the original Image I and decomposed up to 3 level ; get LH_1, HL_1, LH_2, HL_2, LH_3, and HL_3.

2. Apply SVD to each $LH1$, $HL1$, ..., $LH3$, $HL3$.

$SVD\ (I^k) \longrightarrow U_k, S_k, V_k\ ;\ for\ k= \{LH_1, \ldots\ldots HL_3\}$,

where I_k is LH or HL sub-bands coefficient matrix.

3. Modify the singular values of LH and HL sub-bands by adding watermark sequence using equation (25).

4. Obtain modified LH and HL sub-bands using Inverse SVD.

   $I_k^* = U_k S_k^* V_k T$; *for* $k = \{LH_1, \ldots \ldots HL_3\}$,

   where I_k^* is the modified LH or HL sub-bands coefficient matrices

5. Apply inverse DWT on all sub-bands to produce watermarked Image.

This method was also extended for color images using the Y (luminance) channel of YCbCr color model. YCbCr is scaled and offset version of the YUV color space and has been developed as part of ITU-R BT.601 [81]. In this color format Y (luminance value) is defined to have a 8-bit range of 16-235; Cb and Cr are defined to have a nominal range of 16-240. However, in conventional RGB color space data (color) has a range of 0-255; The transformation of RGB color model to YCbCr color model can be derived as:

$$Y = 0.251\,R' + 0.504G' + 0.098B' + 16 \quad (27)$$

$$Cb = -0.148\,R' - 0.291G' + 0.439B' + 128 \quad (28)$$

$$Cr = 0.439\,R' - 0.368\,G' - 0.071B' + 128 \quad (29)$$

where R',G', and B' are gamma corrected R, G, and B. Chou et al [82] reported that the YCbCr and XYZ color spaces are more suitable for watermark embedding than other color spaces. They attain bit error rate of extracted watermark less than 16% with JPEG compression ratio of 38. Their experimental results are shown in Table 5. Moreover, from table 5 it is observed that YCbCr model's average bit error rate is less than XYZ color model.

So it can be concluded that selection of YCbCr color model for color image watermarking among all color model would be the best.

Table 5 Argument to Support YCbCr and bit error rates of extracted watermark [82].

Attacks	Image	Color Space		
		YCbCr	XYZ	CIELAB
JPEG Compression	Lena	8.75%	9.63%	23.88%
	Pepper	15.63%	15.75%	25.00%
Low-pass Filtering	Lena	0.00%	0.00%	1.00%
	Pepper	0.00%	0.13%	0.50%
Noise adding	Lena	1.25%	3.88%	11.75%
	Pepper	2.38%	2.13%	15.25%

Vahedi et al.[83] reported that the 'I' and 'Y' channels of HSI and YCbCr color model respectively are appropriate candidate to hide watermarks. we chose YCbCr color model over RGB because we wanted to embed our watermark into the luminance values of the color image. In RGB color space there is no separate luminance channel. So we choose YCbCr color space for our method. It has some advantages over embedding the watermark in R, G, or B channel of RGB color space. For example if we embed our watermark in the blue channel to which human eye is not sensitive and some attacker converts this color image into a gray scale image, then the watermark will immediately be destroyed.

4.2.1 Watermark Detection for Method II

For watermark detection for method II we have implemented our detection function also as a correlation detector and a similarity measurement. The correlation C between the extracted watermarks which are the diference of singular values $\bar{\sigma}$ of DWT coefficients of the corrupted watermarked image and original image; and original watermark W with rewriting equation (23) as:

$$C = \frac{1}{R}\sum \bar{\sigma} W_i \tag{30}$$

where i run over all singular values and R is the rank of HL or LH matrices. Besides, to calculate the similarity, the proposed method also utilizes the vector projection defined by Cox et al [2], When the similarity measurement value between original watermark and extracted watermark is a way greater than other similarity measurement value, then, we can conclude that the watermark is present.

4.3 Watermark Embedding Method III

For method III, we introduce QR code as our watermark instead of PRN. Moreover, instead of applying SVD on whole LH and HL coefficients we divide each level LH and HL sub-bands into 16x16 non-overlapping sub-blocks. Thereafter, SVs of these sub-blocks of sub-bands becomes the recipients of our watermarks. There is evidence that any modification on Image has less effect on 1st SVs of the SVD decomposed image [16] so we chose only the first SVs of each block of the sub bands as the watermark/payload recipients. Furthermore, it is observed that vertical and horizontal channels have noticeable energy (large values) in first three decomposition levels. Since LH and HL sub-bands represent mid-tone of the image, slight variation of SV values of these sub-bands do not alter the visual resemblances of the watermark or stego image from the cover image. Table 6 also justify this argument where we include only the 1st block of each band for 1st , 3rd, and 5th SVs before and after JPEG compression attack for Baboon image. The alteration ration of SVs is defined as in the following equation:

$$Alteration\_ratio = \frac{SVs\ value\ before\ image\ processing}{SVs\ value\ after\ image\ processing} \times 100 \tag{31}$$

Table 6 Alteration ratios of 1st, 3rd, and 5th SV of each block of Baboon image.

Sub-band	Before Compression SVs Value			After Compression SVs Value			Alteration Ratio in %		
	1st SV	3rd SV	5th SV	1st SV	3rd SV	5th SV	1st SV	3rdSV	5thSV
LH1	3.23E+02	1.35E+02	95.549016	3.23E+02	1.35E+02	95.764719	0.01445	0.2098	0.22575
HL1	3.57E+02	1.74E+02	1.10E+02	3.57E+02	1.74E+02	1.11E+02	0.00356	0.2396	0.04328
LH2	6.56E+02	4.26E+02	1.78E+02	6.56E+02	4.26E+02	1.78E+02	0.03436	0.0254	0.15603
HL2	8.23E+02	3.25E+02	2.29E+02	8.23E+02	3.25E+02	2.29E+02	0.01227	0.0260	0.02620
LH3	1.12E+03	6.60E+02	4.48E+02	1.12E+03	6.61E+02	4.48E+02	0.02700	0.0500	0.06335
HL3	1.18E+03	5.40E+02	3.49E+02	1.18E+03	5.40E+02	3.50E+02	0.02508	0.0896	0.10667
LH4	1.70E+03	1.50E+03	1.15E+03	1.70E+03	1.50E+03	1.15E+03	0.01716	0.0409	0.03499
HL4	1.65E+03	1.15E+03	7.22E+02	1.65E+03	1.15E+03	7.22E+02	0.01928	0.0199	0.02671
					Average alteration		0.01914	0.0876	0.08537

For mark embedding, at first, the designated 3 level DWT (HL_1, LH_1, HL_2, LH_2, HL_3, and LH_3) sub-bands are obtained. The watermark or payload, QR code, is first generated with our secret message then the QR Code modules (Black and white marks) are converted into a one dimensional vector as a sequence of -3's and 3's. The range of positive and negative value sequence ((-1,1), (-3,3), (-5,5)) has been chosen by using some sensitivity analysis. Now this random sequence of -3s and 3s are inserted into the largest SV of the SVD decomposed sub-band blocks. The following equation is used for watermark embedding:

$$\sigma_i' = \sigma_i + \alpha\sigma_i W_i \tag{32}$$

where Wi is a random sequence of {-3 and 3} or the secret bits obtained from one dimensional QR code vector, σ_i's are all largest or 1st singular values of each block where i

represent the block number and α is a multiplication factor (0.05, 0.025, and 0.0125) which differs in values for different levels of decomposition to hide our watermark in different level.

The value of α and random sequence of {3,-3} also can be defined as the strength of the watermark. By setting greater value of those we can have more compression tolerance watermarking, however, they reduce the fidelity of the watermarked or stego image. So the selection of α and value of the random sequence i.e. {3,-3} is a compromise between the strength of watermark or payload and image fidelity.

The embedding process can be formulated as follows:

1. DWT is Applied to the cover Image I and decomposed up to 3 level ; get LH_1, HL_1, LH_2, HL_2, LH_3, and HL_3. (In case of color images DWT is applied on the Y channel of YCbCr or on the Blue channel of the RGB color model, respectively)
2. Divide each LH_1, HL_1... LH_3, HL_3 into 16x16 size blocks.
3. Apply SVD to each 16x16 block

   $SVD\ (I_i) \longrightarrow U_i, S_i, V_i;\ for\ i = \{1\ to\ N\},$

   where I_i is i^{th} block of LH or HL sub-bands coefficient matrices. N is a positive integer number and depends on the spatial size of the image.
4. Modify the largest singular values of each block by adding watermark sequence using equation-(32).
5. Obtain modified LH and HL sub-bands using Inverse SVD on each sub-blocks.

   $I_i^* = U_i S_i^* V_i^T, for\ i = \{1, 2, \ldots, N\},$

where I_i^* is i^{th} modified block of LH or HL sub-bands coefficient matrices. N is a positive integer number and depends on spatial size of the image

6. Apply inverse DWT on all subbabnds to produce watermarked/stego image.

Figure 4.5 shows the process of Method III watermark or payload embedding Process

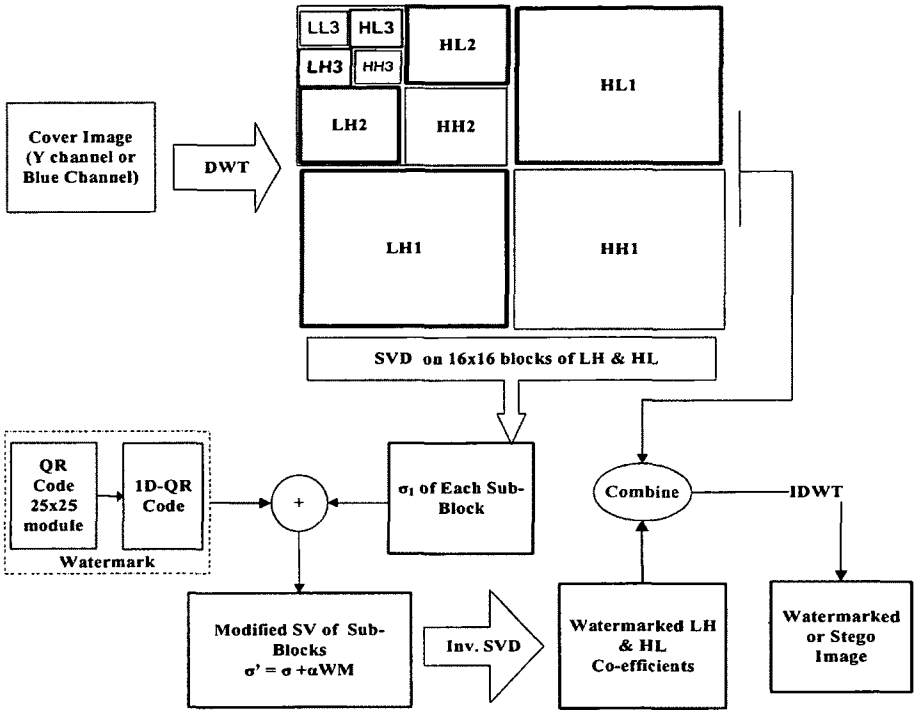
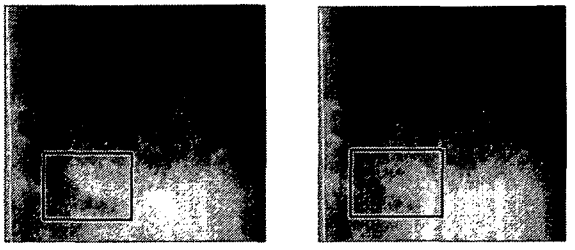


Figure 4.5: Method III watermarks or payload embedding process.

We also have extended this method for color images using 'Blue' channel of RGB color image, since, blue channel is the most insensible for the bare eye [26] and Y channel of YCbCr color model. With method-III we have achieved PSNR more than 40 dB with almost every image so we recommend this method for gray scale medical image watermarking. Moreover, for gray scale image, watermark can be extracted even from watermarked JPEG compressed image and 25% to 200% scaled watermarked image. So this method can be treated as a semi-fragile watermarking method. For medical images, there is a region that is important for diagnosis, called region of interest (ROI) and it should not be altered [35].

However, we suggest to embed watermark in the ROI with as minimum as possible distortion since user (patient, health professional or insurance company) of this Images may change the ROI in an image for various purpose. Specially for archieving purpose (store for longer reriod) tampering detection should be more important than ROI alteration. For example, if a digital mammogram image without any microcalification has been tampered with white spots to make it to look like it has microcalification then it is obviously very necessary to detect tampering within ROI. Figure 4.6 shows some example of tampering in a mammogram image within ROI.



(a) (b)

Figure 4.6: Original mammogram image block with microcalification (a) tampered mammogram image block where microcalification wiped off (b) [35].

This proposed method has tampering detection ability with a tampering location identification capacity.

4.3.1 Watermark or Payload Extraction for Method III

For watermark or payload extraction for method III embedding, proposed method requires only the original image, however, it can be thought of as key, and hence our extraction method can be considered as non-blind extraction process. Figure 4.7 shows the block diagram of the extraction process. The extraction process can be formulated as follows:

1. Apply up to 3rd level DWT to the watermarked or stego image I_w and the original image I_o. (In case of color images DWT is applied on the Y channel of YCbCr or the Blue channel of RGB color model, respectively).
2. Apply SVD on each 16x16 block of $LH_{1\text{-}to\text{-}3}$ and $HL_{1\text{-}to\text{-}3}$ sub-bands of both image I_w and I_o.

   The difference between the largest SV value of every 16x16 blocks of DWT of I_w and I_o are measured. Subtracted values are then map into extracted one dimensional watermark using the following equation:

$$EG_i = \begin{cases} 1 & \text{if } \sigma_w - \sigma_0 > 0 \\ 0 & \text{if } \sigma_w - \sigma_0 \leq 0 \end{cases} \tag{33}$$

3. Finally, one-dimensional watermark or payload is converted into 25x25 matrixes to construct the version 2 QR code.

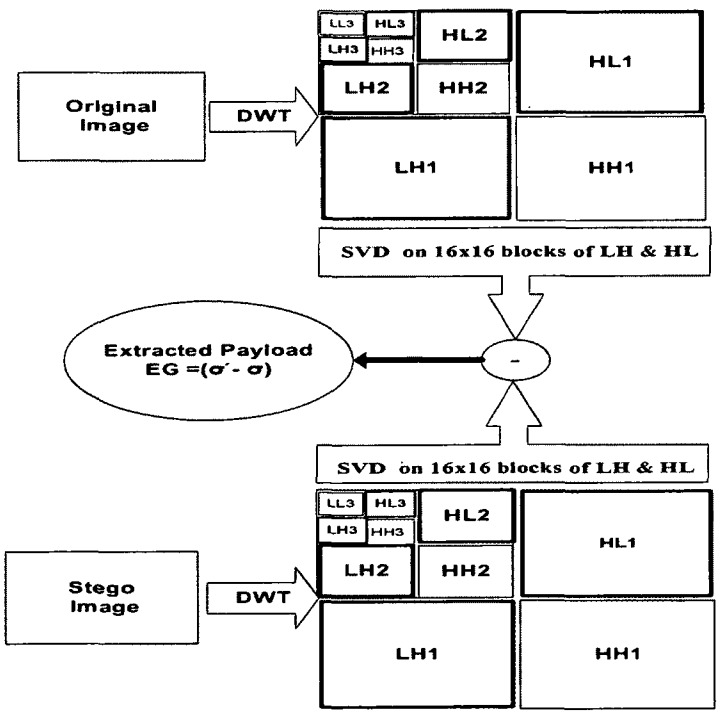


Figure 4.7 Block diagram of the watermark extraction and detection process for method-III

4.4 Watermark Embedding Method IV

Method IV is almost similar to method III. It also use *HL* and *LH* sub-band pair. The watermark or payload, the QR code, is first generated with our secret message, thereafter QR code modules (Black and white marks) are converted into a one dimensional of 1 and 0. The difference in method IV from method III is, it does not embed any marks in singular values rather 2nd singular values of every block is replaced either with 0.99 times of 1st SV or 1.1 times of 3rd SV depending on the value of QR code module (0 or 1). So in this method QR-code vector works as a guide. This method is also iterative, meaning, after iteration one if we cannot extract the exact watermark or payload, we take the watermark or stego image and apply method IV again to ensure 100% extraction of the watermark or payload. The embedding process can be formulated as follows:

1. DWT is Applied to the resized cover Image I and decomposed up to 3 level ; get LH_1, HL_1, LH_2, HL_2, LH_3, and HL_3. (In case of color images DWT is applied on the Y channel of YCbCr or on the Blue channel of the RGB color model, respectively).
2. Divide each LH_1, HL_1... LH_3, and HL_3 into 16x16 size blocks.
3. Apply SVD to each 16x16 block

   $SVD\ (I_k) \longrightarrow U_k, S_k, V_k; for\ k = \{1\ to\ N\},$

   where I_k is k^{th} block of *LH* or *HL* sub-bands coefficient matrices. *N* is a positive integer number and depends on spatial size of the image.
4. Modify the 2nd singular values of each block by the following equation.

$$\sigma_2 = \begin{cases} \sigma_1 * 0.99 & \text{if corresponding QR code module is 1} \\ \sigma_3 * 1.1 & \text{if corresponding QR code module is 0} \end{cases} \tag{34}$$

5. Obtain modified *LH* and *HL* sub-bands using Inverse SVD.

$I_k^* = U_k S_k^* V_k^T, for\ k = \{1, 2, \ldots, N\}$,

where I_k^* is kth modified block of *LH* or *HL* sub-bands coefficient matrices, and *N* is a positive integer number and depends on spatial size of the image.

6. Apply inverse DWT on all sub-bands to produce watermarked image.

Figure 4.8 shows the process of method IV payload embedding Process.

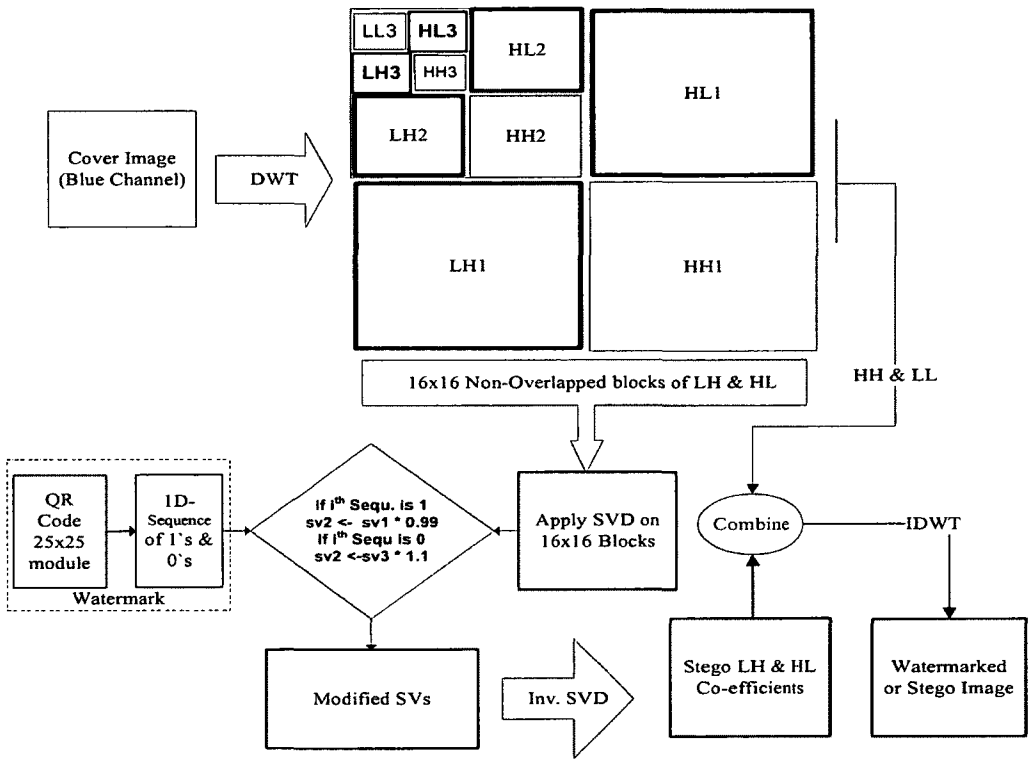


Figure 4.8: Method IV watermark or payload embedding process

To set a stop criteria for the number of iteration we have used PSNR measurement between $(M\text{-}1)^{th}$ and M^{th} iteration resultant image. When PSNR is greater than 100 dB we stop to go for further iteration. In the embedding process with altered value of SVs and we apply inverse DWT transform which generates pixels values in real numbers but when these real numbers are saved in some image file, they are converted into nearest integer values hence some time the extracted value of σ_2 from the watermarked or stego image is not

exactly equal to $\sigma_1 * 0.99$ or $\sigma_3 * 1.1$. We also extend method IV for color image watermarking and steganography. For any RGB color image we convert it to its YCbCr color model and use only Y(Luminance) channel for non-homogeneous image, and for homogeneous image we can chose blue channel to embed our payload since the blue channel is the most insensible for the bar eye[7].

4.4.1 Watermark or Payload Extraction for Method IV

For watermark or payload extraction for method IV embedding, proposed method do not require the cover image, it only requires the threshold information, and can be considered as blind extraction process. The extraction process can be formulated as follows:

1. Apply up to 3rd level DWT to the watermark or stego image I_w. (In case of color images DWT are applied on the Y channel of the YCbCr or on the Blue channel of the RGB color model, respectively).
2. Apply SVD on each 16x16 block of $LH_{1\text{-}to\text{-}3}$ and $HL_{1\text{-}to\text{-}3}$ sub-band of image I_w.
3. Find the ration between the 1st and 2nd Singular value of each block of I_w. Extracted values are then mapped into the extracted one dimensional payload using the following equation:

$$EP_i = \begin{cases} 1 & \text{if} \frac{\sigma_1}{\sigma_2} > T \\ 0 & \text{if } \frac{\sigma_1}{\sigma_2} < T \end{cases}, \tag{35}$$

   where the value of T (threshold) value is in the range 1.1 to 1.4 and depends on individual images.

4. Finally, the one-dimensional watermark is converted into 25x25 matrixes to construct the version 2 QR code.

4.5 Tampering or Forgery Detection in Watermarked Image

Proposed methods III and IV have tampering detection and localization capacity. The tamper detection mechanism of the method III and IV is based on wavelet sub-band decomposition; the altered segment can be identified in several level of decomposition. When there are some altered blocks in the watermarked image, the forgery can be identified in several sub-bands of image when they are analyzed against watermarked Image. If all levels of sub-band or most of the levels of sub-band confirm an alteration then we say that those pixels in the image are forged. The detection process for method III can be formulated as follows:

1. Apply up to 4th level DWT to the watermarked corrupted image I_{wc} and the original watermarked image I_{wo}.
2. Apply SVD on each 16x16 block of $LH_{1\text{-}to\text{-}4}$ and $HL_{1\text{-}to\text{-}4}$ sub-band of both image I_{wc} and I_{wo}.
3. Subtract the largest value of each block of I_{wc} and I_{wo}. Subtracted values are then mapped into extracted one dimensional watermark using the following equation

$$\mathrm{EW_i} = \begin{cases} 1 & \text{if } \sigma_w - \sigma_0 > 0 \\ 0 & \text{if } \sigma_w - \sigma_0 \leq 0 \end{cases} \qquad (36)$$

Finally, one-dimensional extracted watermark is compared with the original one-dimensional watermark to locate the block where tampering or any forgery had been done. If two levels of sub-band or most of the levels of sub-band confirm an alteration at roughly the same spatial location then we say that those pixels in the image are forged. The tampering localization is more acute when it is detected in the very first level of decomposition and becomes less acurate as it is detected in 2nd, 3rd or deeper level of decomposition. In addition, if the

tampered image is similar to the original one and do not have any watermark content in it, then there will be no trace of watermark (QR Code) at all.

Extraction process for method IV is a blind process; only the watermarked image is needed to detect the forged area. The detection process for method IV embedding can be formulated as follows:

1. Apply up to 3 levels DWT to the watermarked corrupted image I_{wc}
2. Apply SVD on each 16x16 block of $LH_{1\text{-}to\text{-}4}$ and $HL_{1\text{-}to\text{-}4}$ sub-band of I_{wc}.
3. Find the ratio between 1st and 2nd SVs of each block of I_{wc}. Extracted values are then map into extracted one dimensional watermark using equation (35).
4. The extracted watermark is compared with the original watermark.
5. If any dissimilarity found, then the corresponding 16x16 block is identified as forged and localized in corresponding 16 X16 blocks of certain DWT decomposition level.

Chapter 5

Result Analysis and Discussion

In order to validate image fidelity, watermark robustness, payload capacity, steganalysis of the proposed methods III and IV, extensive simulations have been performed with MATLAB 7.9 on various types of images including the following:

i) Images from Signal and Image Processing Institute (SIPI) of the University of Southern California (http://sipi.usc.edu/database/database.php) which have 2 different types of color images (28 Arial images, 17 miscellaneous images) and 75 grayscale texture images.

ii) McGill calibrated color image database which is found at: http://pirsquared.org/research/mcgilldb/browsedownload.html. This databse includes 75 flower images, 28 animal images , 28 fruit images, 38 landscape images and 35 manmade object images,

iii) The third database comprises 26 medical images collected from internet which included MRI, CTScan, and mammogram images.

Methods I and II are of robust categories and their experimental results are shown in sections 5.1 and 5.2 with StirMarks attacks on them. To measure the performance of these two methods, we only used few very well known images such as Lena, Baboon, Barbara, F15, etc.

SIPI and McGill image databases have been used to test the performance of method III and IV. Moreover, to test the performance of method III for grayscale image 28 medical images also have been used. Method III has semi-fragile characteristics for only gray scale images. However, for color image method III did not show any robustness so this algorithm can be considered as only non-blind-fragile method for color images. Method IV has been found very sensitive against any image processing or compression attack and extraction of watermark or payload for this method does not require the original image so it is considered as blind-fragile method. Moreover, for method IV embedding we have devised tampering localization mechanism which satisfy the huge demand of any fragile watermarking method and results are shown in section 5.4.

5.1 Results for Method I

Figure 5.1 shows all test images that we used in order to evaluate the performance of method-I.

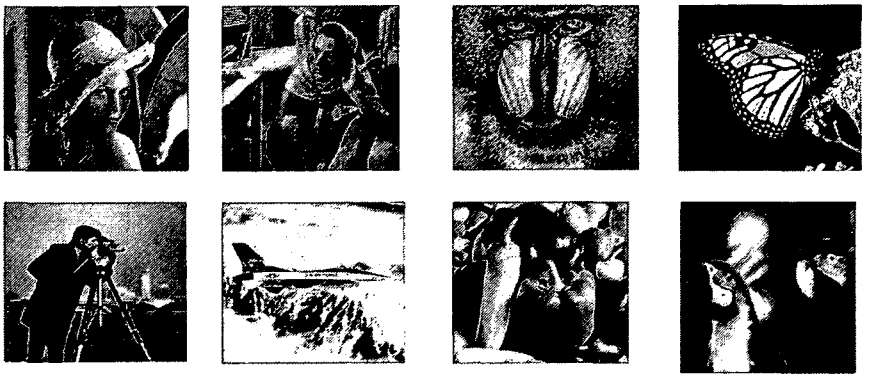


Figure 5.1: Test Images for Method I

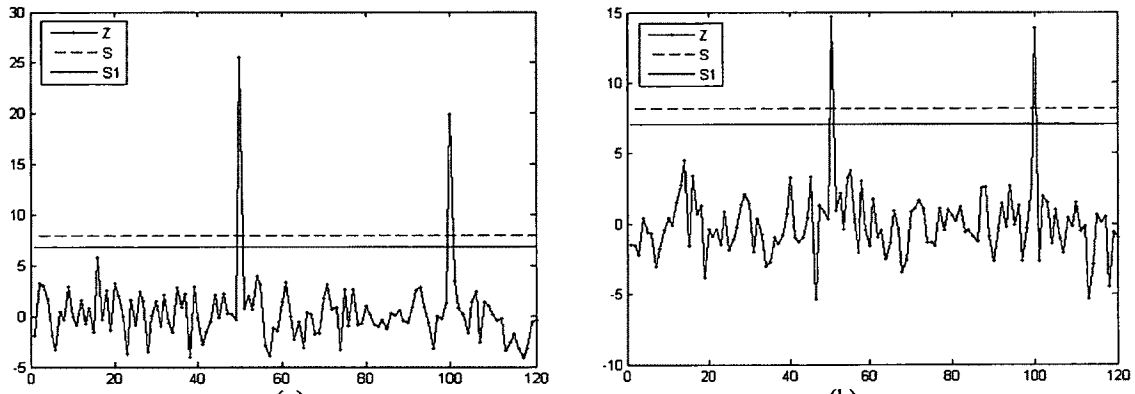
Table 7 shows the PSNR values in dBs of the test images after watermark embedding process is performed using proposed method-I, Dugad *et al.* method [10] and also shows detected correlations for JPEG compressed watermarked images with quality factor q=20.

Table 7 PSNR values of test images after watermark insertion using method I

Image	**Proposed Method PSNR**	**Cor. After JPEG Attack q=20**	**Dugad et al[10] Method PSNR**	**Image**	**Proposed Method PSNR**	**Cor. After JPEG Attack q=20**	**Dugad et al [10] Method PSNR**
Lena	41.352	0.67	38.20	Butterfly	40.037	0.61	31.02
Barbara	35.953	0.66	34.15	Cameraman	41.910	0.63	32.84
Baboon	34.247	0.65	32.21	F16	41.460	0.66	38.53
Peppers	35.321	0.66	33.82	Zurest	39.320	0.62	36.53

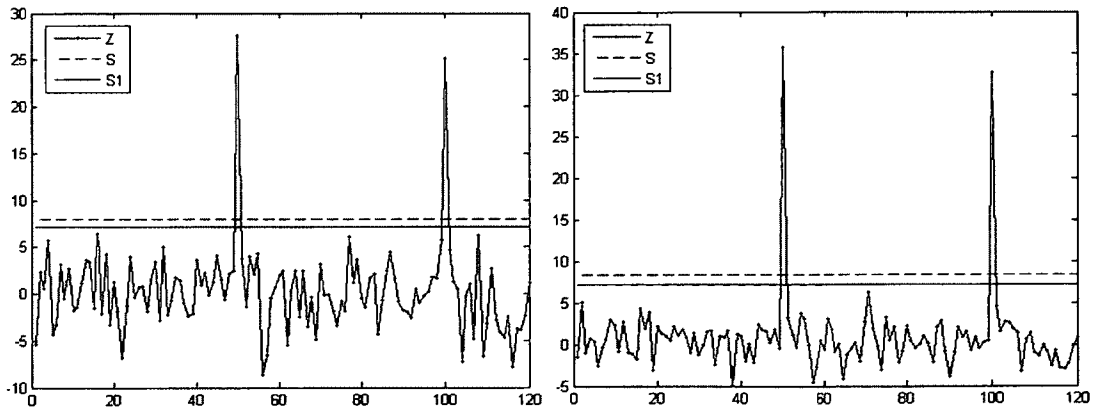
These results show that the watermarked document is visually near identical to the original image as the PSNR value is always greater than 34 dB and there is high correlation after JPEG compression attack even with quality factor 20. The table-7 also shows our method outperforms Dugad et al.[10] in respect of PSNR value. Figure 5.2, 5.3, and 5.4 display the detector responses on the *y*-axis and randomly generated watermark on the *x*-axis. In this method we used a seed of 50 for the byer watermark and a seed of 100 for the seller watermark. Hence, the results as shon in the figures show the location of these watermarks on the x axis at both 50 and 100 seeds. Solid line shows the *LH* threshold (which is close to the next 2nd higher value) value. The circle/plus markers indicate the similarity index values of the corresponding watermarks. However, since two different watermarks are embedded into two different bands and not every picture have enough number of significant coefficients in *LH* and *HL* sub-bands, so sometime one watermark may be detected weakly due to less number of significant coefficients above *T1*. Moreover, the size of the image is also a crucial factor for the proposed method. To test the performance of the proposed method, several image processing techniques were employed including compression (up to 20% JPEG),

resizing, and filtering. These results are clearly depicted in figure 5.2, 5.3, and 5.4. No false positive and false negative have been detected. All experimental results are of 512x512 size image. However, for larger size (more than 512x512) image we have not noticed further significant improvement.



(a) (b)

Figure 5.2 Detector response for JPEG compressed-watermarked F16 image with (a) quality factor q=20 (b) JPEG2000.



(a) (b)

Figure 5.3: Detector response for F16 (a) 256x256 resized image (b) 600x600 resized image.

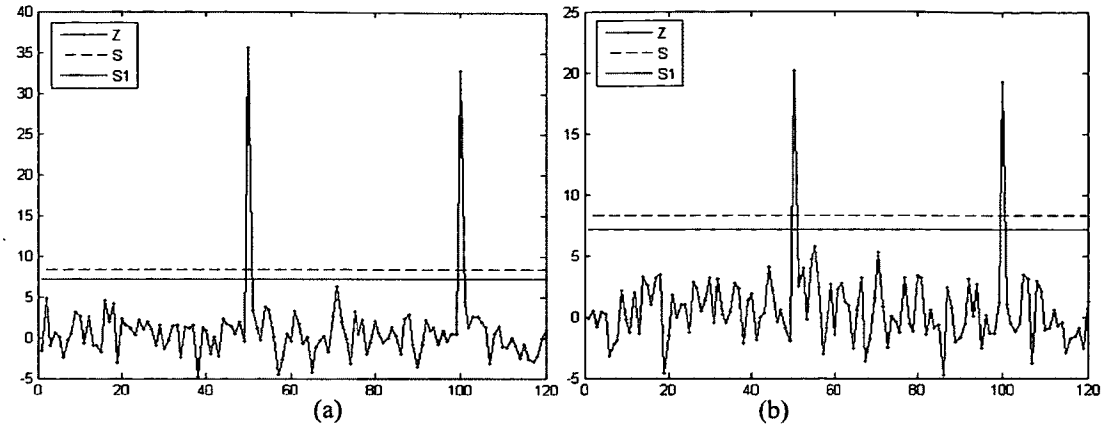


Figure 5.4: Detector response for F16 (a) 5x5 medial filtering (b) Gaussian noise with mean=0 varience=0.01 .

The simulation results show that the proposed watermark is robust against signal processing attacks such as high ratio JPEG compression, resizing, and adding Gaussian noise.

5.2 Results for Method II

In order to validate the effictiveness of method-II for gray scale image, we have performed extensive simulations using MATLAB 7.9 on a set of well-known images from our image database which is shown in Figure 5.5.

Figure 5.5: Test images for method II for grayscale images.

We embedded our watermark in these images and measured the PSNR for each watermarked image. Table 8 shows the PSNR values in dB of the watermarked images for 2, 3 and 4 level decomposition.

Table 8 PSNR after watermark insertion using method II for gray scale images

Image		PSNR(dB)			Image		PSNR(dB)		
#		2 Level	3 Level	4 level	#		2 Level	3 Level	4 level
1	Lena	56.03	55.00	55.27	12	Butterfly	50.60	50.08	50.13
2	F16	52.97	52.47	52.46	13	Food	56.22	55.08	55.18
3	Barbara	50.99	50.84	50.33	14	Airplane	60.06	59.60	59.16
4	Baboon	51.62	51.66	51.75	15	Galaxy	56.05	55.67	56.33
5	Peppers	53.9	52.90	53.44	16	House	58.37	57.36	57.22
6	Cameraman	49.41	50.40	48.71	17	Lake	51.67	51.37	51.35
7	Girl	59.93	58.003	58.53	18	Pirate	52.44	52.21	52.25
8	Paris	50.00	49.96	49.87	19	Room	51.84	51.64	51.56
9	Niagrafall	54.91	54.64	54.74	20	Bridge	50.82	50.70	50.61
10	Texture	51.33	51.11	51.09	21	Bus	49.19	49.05	48.70
11	Flower	52.01	51.8	51.85	22				
	Average (21 Images)	**53.54**	**53.23**	**53.18**					

These results show that the watermarked image is visually near identical to the original image with a PSNR value always greater than 50 dBs with 2, 3 and 4 level decomposition. However, for robustness test we included only 3 level of decomposition, since they are resistant enough to image processing. Figures 5.6-5.10 display the detector responses on the y-axis and randomly generated watermark on the x-axis. The original images have been watermarked with a seed of 50. Therefore, in each figure, the similarity index with the real watermark is located at 50 on the x-axis; the solid line shows threshold value. The circle/plus markers indicate the similarity index values of the corresponding watermarks.

We have not compared our results with other DWT-SVD based watermarking methods as researchers in this area opted to use images as their watermark rather than PRN sequence as we did. One paper, Calagna et al. [17] had used a pseudo-random number

sequence drawn from a Gaussian distribution N(0,1) as their watermark. But, they embed their watermark into the SVs of certain blocks of the original image, not in the SVs of the DWT image as in our case.

In addition to our tests on the *LH* and *HL* sub-bands reported earlier, we have calculated the PSNR of our method on three sub-bands namely *LH, HL*, and *HH* to test for possible higher attack resistance of the watermarked image. In this case, no significant improvement was obtained. Table 9 shows the PSNR values in dBs of the test images after embedding the watermark using *LH, HL,* and *HH* bands.

Table 9 PSNR values of test images after watermark insertion using 3 sub-bands

Image	**PSNR(dB)**	**Image**	**PSNR(dB)**	**Image**	**PSNR(dB)**
Lena	51.31	Paris	49.42	Galaxy	55.91
F16	49.97	Niagrafall	53.1	House	52.60
Barbra	49.89	Texture	50.73	Lake	49.06
Baboon	49.86	Flower	50.38	Pirate	50.94
Peppers	49.58	Butterfly	47.38	Room	50.83
Cameraman	48.42	Food	51.11	Bridge	49.76
Girl	52.71	Airplane	58.38	Bus	47.3
Average	**50.88**				

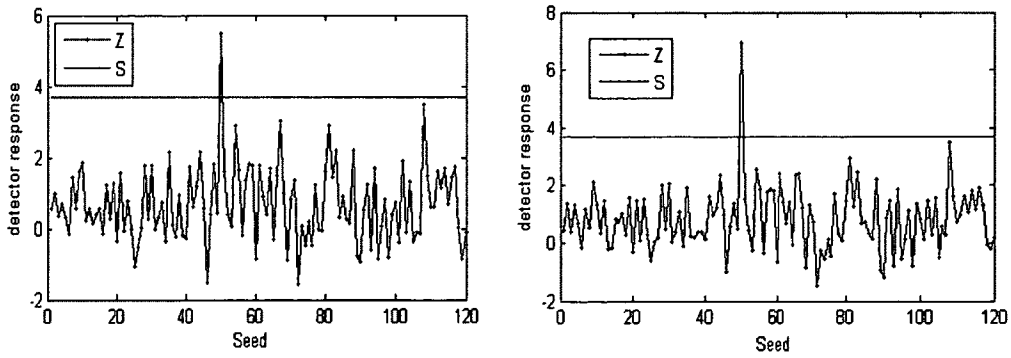
We have used various image processing attacks to test the performance of the proposed method II and its robustness. Some of these attacks are: JPEG compressions (up to 10%), Gaussian Noise, low-pass filtering, resizing, and median filtering. Moreover, since SVD is related to the luminance value of images, we have tested our algorithm robustness against varying the contrast of the watermarked image. Figure 5.11 shows Lena image with 10% luminance variation.

5.2.1 Robustness Against Lossy JPEG Compression

Any watermarking method has to be robust against JPEG compression. Table 10 shows the performance of the proposed method-II against different ratios of compression applied on watermarked images. The watermark survived even after PSNR dropped to 25.32 dBs. Figure 5.6 shows that the watermarks were detected after 10% JPEG compression and JPEG2000 compression.

Table 10 PSNR of watermarked JPEG images of test images

Image	PSNR in dB and Correlation of Attacked and Watermarked Image								Watermark Detected
	Q factor 10	Cor.	Q factor 20	Cor.	Q factor 30	Cor.	Q factor 40	Cor.	
Lena	30.56	.810	33.25	.831	34.68	.876	35.57	.901	√
Baboon	26.32	.832	29.69	.839	31.40	.841	32.47	.891	√
Barbara	25.32	.811	27.93	.817	29.88	.816	31.21	.867	√
Peppers	29.09	.791	31.37	.791	32.47	.795	33.13	.813	√
F16	28.89	.782	31.69	.783	33.30	.783	34.32	.809	√
Camera man	26.41	.803	28.52	.809	29.85	.811	30.80	.821	√



(a) (b)

Figure 5.6: Detector response for JPEG compressed-watermarked Lena image with (a) quality factor q=10 (b) JPEG2000

5.2.2 Robustness Against Gaussian Noise

Method-II is also robust against Gaussian noise for mean zero and up to variance 0.01. The PSNR was dropped to 19.14 dB and watermark detection was found perfect. This is shown in figure 5.7. Table 11 shows the performance of the proposed method against different varying variances of Gaussian noise.

Table 11 PSNR of Gaussian noise added watermarked images of test images

Image	PSNR in dB and Correlation of Attacked and Watermarked Image						Watermark Detected
	Mean = 0 var = 0.005	Corre.	Mean = 0 var = 0.009	Corre.	Mean = 0 var = 0.01	Corre.	
Lena	22.74	.981	20.18	.968	19.73	.951	√
Baboon	22.20	.992	19.66	.991	19.20	.988	√
Barbra	22.54	.991	20.02	.986	19.57	.979	√
Peppers	22.10	.992	19.59	.989	19.14	.986	√
F16	22.20	.975	19.76	.972	19.33	.962	√
Cameraman	23.28	.994	20.86	.991	20.42	.987	√

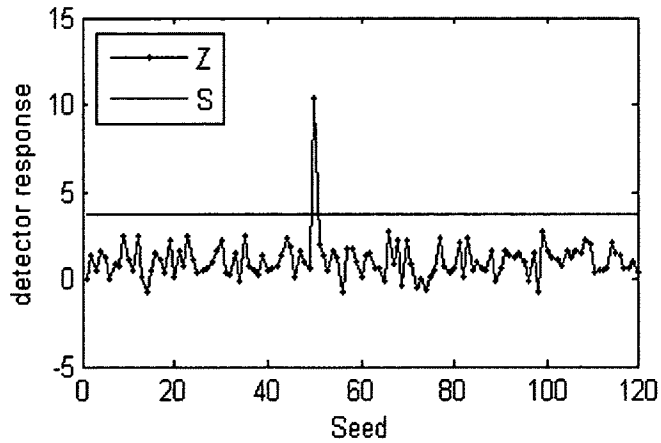


Figure 5.7: Detector response for watermarked Lena image with Gaussian noise mean = 0, variance = .01.

5.2.3 Robustness Against Median Filtering

Median filtering is a widely used image processing method which provides smoothing of finer details with most edges preserved. Table 12 shows the performance of the proposed method against varying window sizes (3 by 3 and 5 by 5) of median filtering.

Table 12 PSNR of median filtered watermarked images of test images.

Image	PSNR in dB and Correlation of Attacked and Watermarked Image				Watermark Detected
	[3x3]	Corre.	[5x5]	Corre.	
Lena	55.01	0.998	54.91	0.992	√
Baboon	51.6	0.996	51.55	0.991	√
Barbra	50.84	0.981	50.47	0.978	√
Peppers	52.98	0.987	52.83	0.988	√
F16	52.47	0.969	52.40	0.975	√
Cameraman	50.40	0.985	50.25	0.987	√

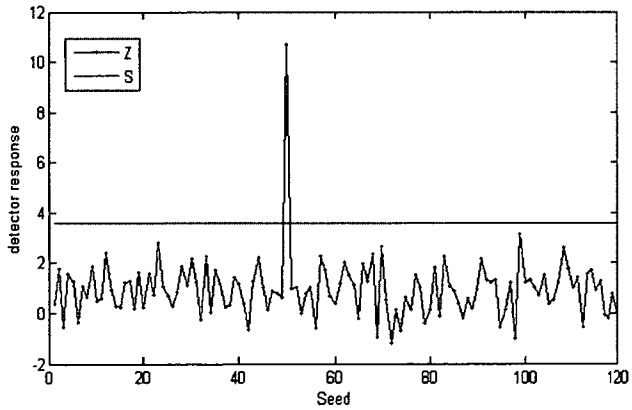


Figure 5.8: Detector response for watermarked Lena image median filetering with window size 5x5.

Besides, the above attack we also tested the performance of the proposed method against image resizing and found that our watermark can be detected with 25% spacial size reduction. Our method is also robust against histogram equalization.

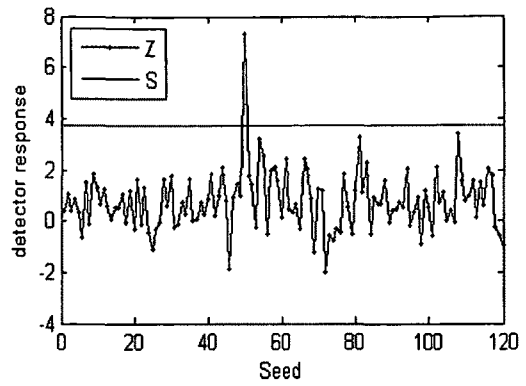


Figure 5.9: Detector response for watermarked Lena image resizing 256x256

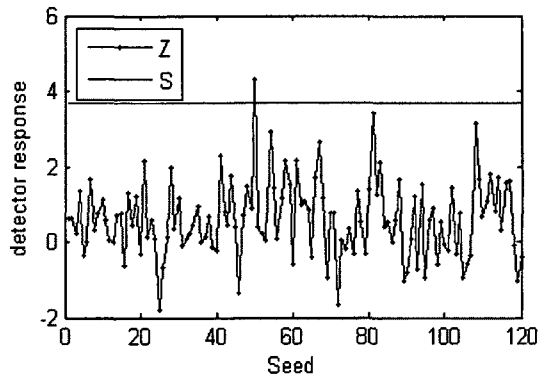


Figure 5.10: Detector responses for watermarked Lena image with histogram equalization

We varied the luminance values and observed that our watermark can withstand a 12% luminance variation on average. No false positive and false negative has been detected. All experimental results are of 512x512 size images. However, for larger size (more than 512x512) images we have not noticed further significant improvement.

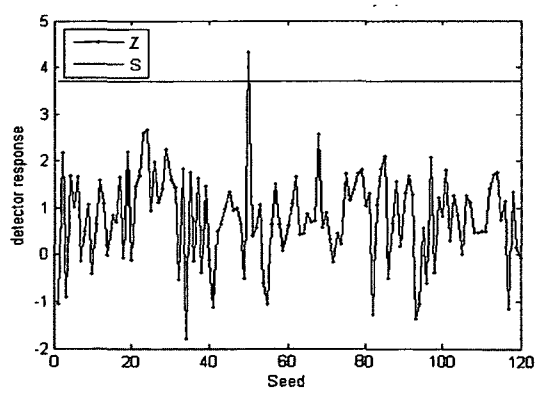


Figure 5.11: Detector response for watermarked Lena image with 10% luminance variation.

Method II embedding was also extended for color images using the Y (luminance) channel of the YCbCr color model. Table 13 shows the PSNR values in dB of the test images after embedding watermark using proposed method II with Y channel and other color image watermarking methods. However, most other researcher reported their results with 1 or 2 tests images.

Table 13 Comparision of PSNR and SSIM of watermarked 4 color images using method II and other authors

	SSIM index	PSNR (dB)				
Image		**Proposed Method-II (With Y Channel)**	**Elbasi et al [12]**	**Yin et al [25]**	**Lee et al [26]**	**Vahedi et al.[83]**
Lena	0.922	49.78	-	-	40.88	41.74
Peppers	0.847	49.89	-	-	40.88	39.27
F16	0.981	49.56	-	-	-	
Baboon	0.898	48.46	41.74	42.82	40.88	39.31
Average	**0.912**	**49.42**	**41.74**	**42.82**	**40.88**	**40.10**

Method II for color image performance has also been tested by converting the color image to it's gray scale form. Figure 5.12 shows the detector response for color to gray scale converted lena image.

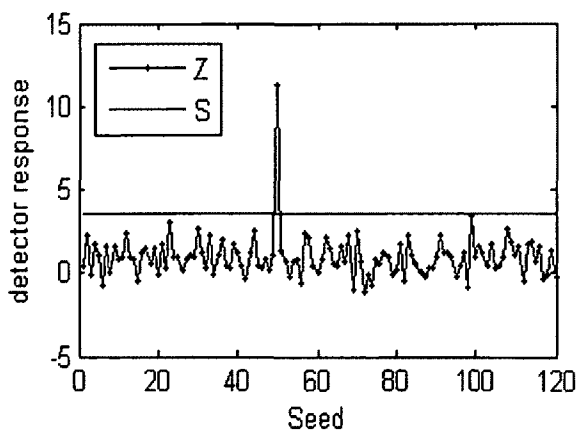


Figure 5.12: Detector response for Lena color image converted to gray-level image.

5.3 Results for Method III

In method III, we have introduced for the first time in image processing research QR code as our watermark instead of PRN. There is no detection rather we have developed an extraction method for watermark from the watermarked image or altered watermarked image. Since this method achieved the requirement of medical image watermarking which is PSNR greater than 40 dBs, we included only grayscale medical image to test the performance of these method for gray scale image. However, the performance of this method for color images is included in section 5.4. In order to validate the robustness and tampering detection capability of this watermarking method, a set of 26 different medical images (Mammogram, MRI, CT Scan) were collected over internet of which 4 images are shown in figure 5.13.

Image-1 Image-2 Image-3 Image-4

Figure 5.13: Sample medical images

Table 14 shows the PSNR in dB and SSIM values of the test images after embedding watermark using method III.

Table 14 PSNR and SSIM values of medical images after watermarked using method III

Test Image	Image Type	Image Fidelity Measure	
		PSNR in dB	SSIM value
Image-1	Mammogram	48.17	0.974
Image-2	MRI Brain	41.64	0.983
Image-3	CT Scan Brain with Glioma	40.86	0.963
Image-4	CT Scan Brain with Parkinson	40.31	0.979
Image-5	Mammogram	48.61	0.985
Image-6	MRI Knee	43.34	0.983
Image-7	MRI Brain	40.26	0.985
Image-8	MRI Abdomen	44.44	0.989
Average		**43.45**	**0.980**

These results show that the watermarked document is visually near identical to the original image with a PSNR value always greater than 40 dB, which is already accepted by the medical image community. Moreover, we also achieved more than 0.96 SSIM values which is more accurate measurement than PSNR. For image 4 we obtained 0.985 SSIM though the PSNR was low for that watermarked image than image 1. Figure 5.14 shows sample results of watermarked image and extracted watermark from them.

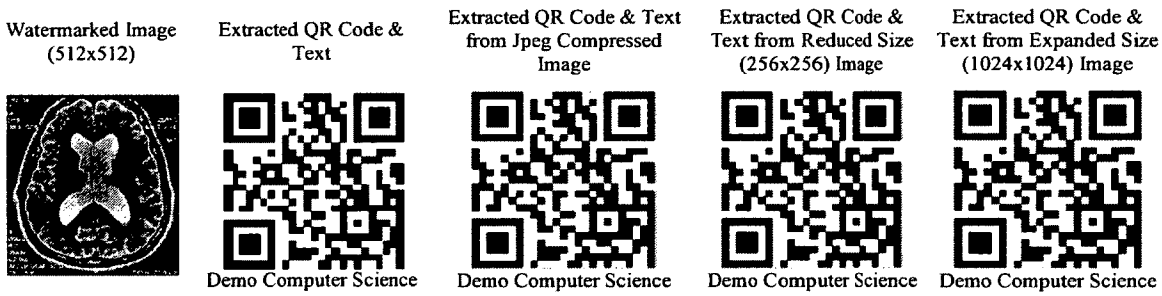


Figure 5.14: Results of watermark extraction from watermarked, compressed, and resized images

The tampering detection approach used in method III is based on DWT sub-band decomposition. Any altered block can be identified in the various levels of decomposition. When there are some altered blocks in the watermarked image, the forgery can be identified in several sub-bands of DWT image when they are analyzed against original watermarked image. The tampering localization is more acute when it is detected in the very first level of decomposition and becomes course as it is detected in 2nd, 3rd level of decomposition. In addition, if the tampered image is similar to the original one and do not have any watermark content in it, then there will be no trace of watermark (QR Code) at all. Figure 5.15 shows detected tampered location of watermarked image 2 and image 4.

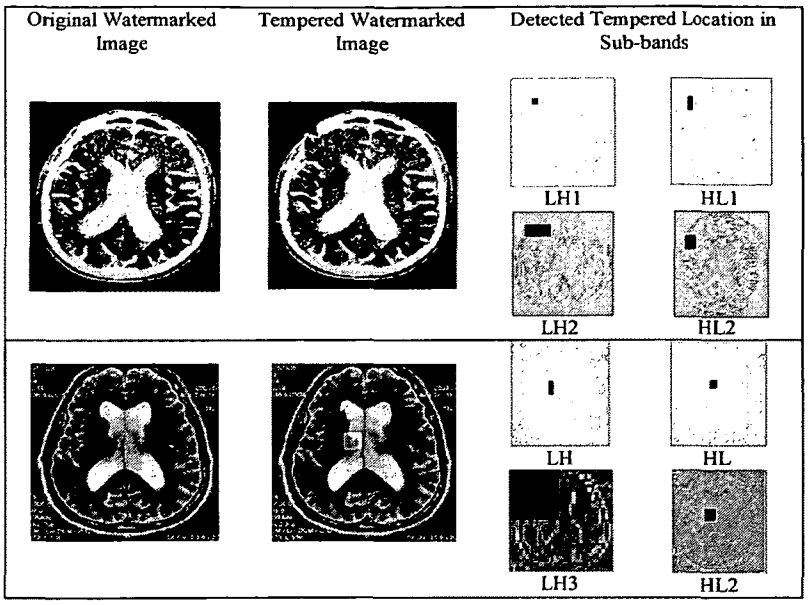
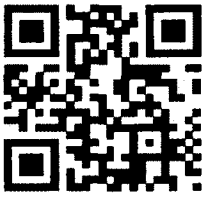


Figure 5.15: Tampering detection and localization in sub-bands of medical images

Our experimental simulation results on medical images show that the proposed watermark is robust against JPEG compression, and resizing, but very sensitive against any forgery of the watermarked image. The proposed method has very good tampering detection capability with localizing tampered area within the image, hence it can be used for forgery detection on medical images. We emphasize this algorithm for medical image since being semi-fragile, the proposed method achieved satisfactory level of SSIM and PSNR for a medical image. Moreover, this method was also extended for color image watermarking with results are given in section 5.5. Moreover, from the experimental results we observed that method III for color image with Blue channel for RGB color model cannot withstand any type of image processing or compression attack, so we recommend method III with blue channel in image steganography, not for watermarking.

5.4 Results for Methods III and IV

The proposed method IV is most significant method among all of these methods since watermark extraction process for this method is blind and it maintains acceptable PSNR and SSIM for watermarked image. We have tested method III and method IV with large number of images from SIPI and McGill image database. Since it consist of PSNR results for more than 250 images, only average results of each category are included in this section, however, detailed results are attached in annex-1. Figure 5.16 shows the QR code that has been used for mark embedding for method IV and it was generated using online software (e.g. qure.com, www.racoindustries.com) which are free. Moreover, we vary the size of message for version 2 QR code from 2 to 40 Characters and found similar ratio in the number of black and white modules.



UNBC Computer Science

Figure 5.16: Watermark used for the experiment

Table 15 shows the average PSNR in dB and SSIM values for 75 texture image of SIPI gray scale image database, and 17 miscellaneous grayscale images.

Table 15 Average PSNR and SSIM values of watermarked images for grayscale images using method IV

Image Database	Category Name	Method III (Non - Blind)		Method IV (Blind)		
SIPI		PSNR	SSIM	PSNR	SSIM	No. of failed images for extraction
	Texture Images (75)	37.54	0.957	34.47	0.967	9
	Miscellaneous images (17)	39.98	0.967	36.68	0.962	3

Table 16 shows the average PSNR in dB and SSIM values of each category of the test images after embedding the watermark or payload using methods III and IV with blue channel of RGB color model. Table 16 also shows the number of images that gives us erroneous message (Watermark or payload) extraction. However, figure 5.17 shows eight test images from each category of SIPI and McGill color image database.

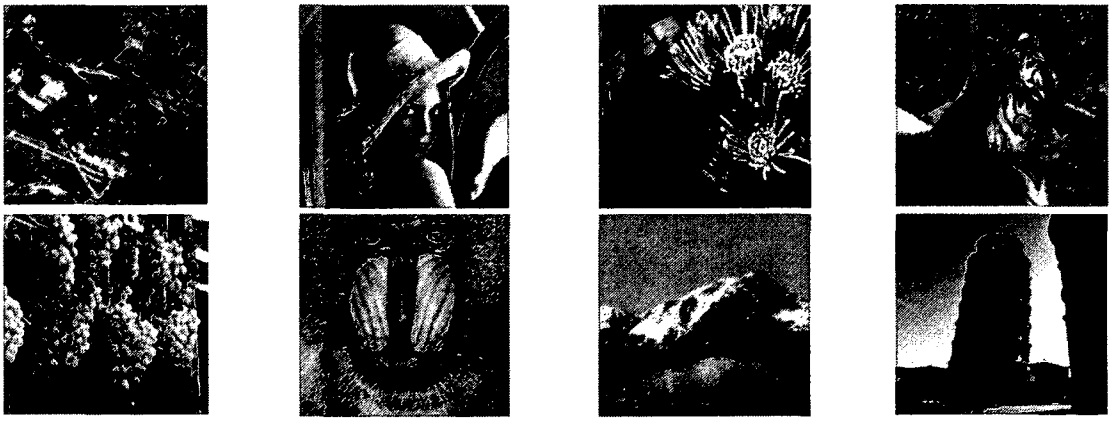


Figure 5.17: Eight test images out of 256 test images

Table 16 Average PSNR and SSIM of watermarked color images using method III & IV (Blue Channel).

Image Database	Category Name	Method – III (Non - Blind)		Method – IV (Blind)		
SIPI		PSNR	SSIM	PSNR	SSIM	No. of failed images for extraction
	Arial Images (28 images)	40.04	0.961	34.06	.956	6
	Miscellaneous images (17 images)	39.44	0.960	36.48	0.955	2
McGill calibrated color image	Flower images (75 images)	44.62	0.991	39.04	0.961	5
	Animal images (28 images)	46.52	0.992	40.36	0.972	1
	Fruit images (28 images)	45.32	0.991	39.23	0.958	4
	Landscape images (38 images)	48.17	0.995	44.16	0.982	7
	Manmade Object images (35 images)	44.28	0.991	38.31	0.959	5

These results show that the watermarked document is visually near identical to the original image with a PSNR value always greater than 30 dB and average greater than 34 dB for each category. Moreover we also achieved more than 0.9 SSIM values. Method III achieved more PSNR in dB values than Method IV. Clearly, method IV has a blind payload extraction process. This means that the receiver does not need the original image to extract the watermark. Nevertheless, no visual artifacts or abnormality in homogeneous image region has been observed in the watermarked image with blue channel.

Since with Y channel, we are embedding our watermark in the luminance value this method can sustain JPEG attack, meaning method IV with YCbCr color model can be considered as blind semi-fragile watermarking method. However, we observed some visual artifact in the watermarked image if there is homogeneous regions, i.e., same or close pixel values in images. We did not observed any such thing in noisy or non-homogeneous images. Figure 18 shows two watermarked images, one with homogeneous region other with non-homogeneous region in the image for Y channel embedding.

(a) (b)

Figure 5.18: Watermarked Image (a) with Homogeneous regions (b) with non-homogeneous regions

Therefore, we do not recommend method-IV using the YCbCr model for color image watermarking and steganography with large portion of homogeneous region within it. However, if the image is non-homogeneous or noisy then it is a ideal candidate to use this method.

Method IV is extremely sensitive to alterations in any part of the watermarked image. Attempt of forgery or alteration in the watermarked image can be localized and identified in one or more sub-bands of DWT. Figure 5.18 shows detected tampered location of forged watermarked image.

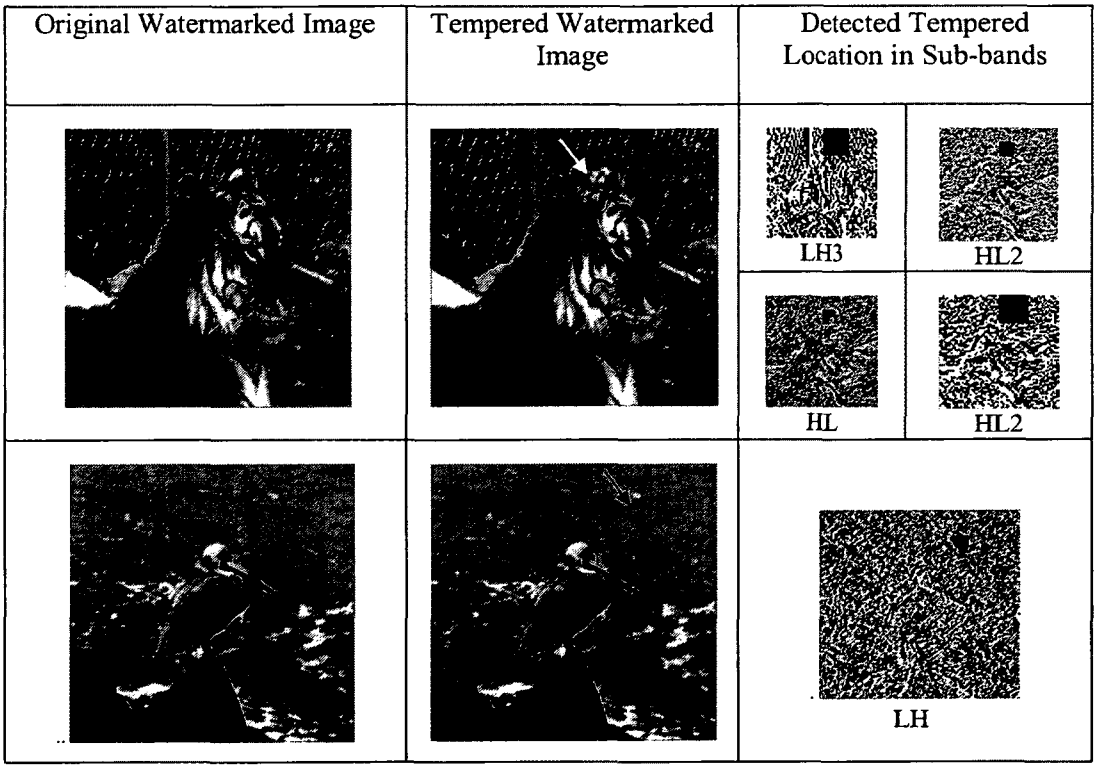


Figure 5.19: Tamperment detection and localization in sub-bands for color images.

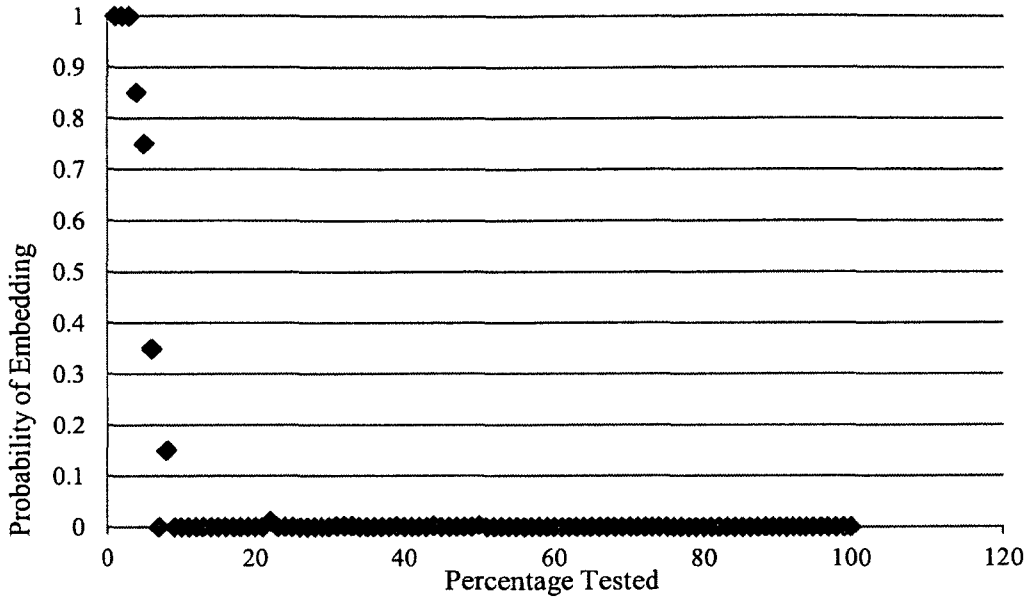
In the tiger image the forgery was flipping all colors in a small region so it has been detected in several sub-bands. In the bird image, the forgery was very minor we just changed few pixels intensity value within the range +10 to -10 of a certain region which is very minor change. Even this minor change has been detected in *LH* sub-band in the first decomposition level. From experimental result it has been observed that minor change are detected from 1st level decomposition and all levels detect forge area when there is any major change in the watermarked image. Moreover, more acurate location of the forged area can be identified from 1st level decomposition than from 2nd and 3rd level decomposition, respectively.

To create our QR code we have used freely available three online QR code generators which are listed in [41]. We have used version 2 QR code to embed the secret message into the 512x512 images with three level of DWT decomposition. Version 2 QR code consist of 25x25 modules with which we can embed up to 47 alphanumeric characters though for the experiment the secret message "UNBC Computer Science" was 19 characters long and shown in figure 5.16 From experiment we found the 16x16 block size is good from image fidelity view point. So to embed message with more than 47 characters we need to select bigger size image. Table 17 shows the relations between message size and cover image size. Table 17 also includes image size of 3264x2448 to show the data embedding capacity within a standard image that can be captured by a standard 14.1 mega pixels camera.

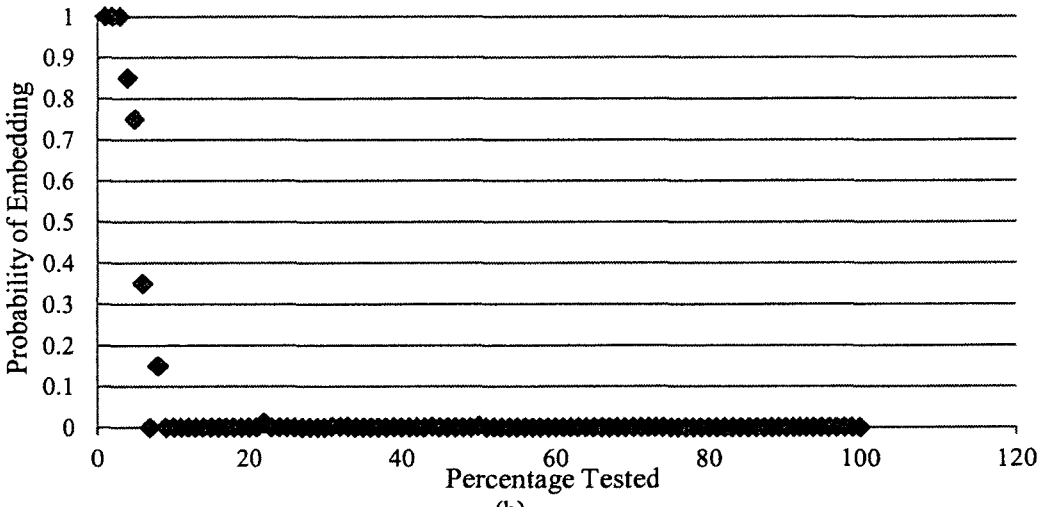
Table 17 Watermark or payload length and image size relations

Image Size	Number of 16x16 blocks	Supported QR-Code Version	# of Modules	Required # of 16x16 Size Block	Highest Character Capacity
512x512	672	1, 2	21x21, 25x25	441, 625	25, 47
768x768	1512	1, 2, 3, 4, 5	21x21, 25x25, 29x29, 33x33, 37x37	441,625, 841, 1089, 1369	25,47, 77, 114, 154
1024x[1024	3200	1, 2, 3, 4, 5, 6, 7, 8, 9	21x21, 25x25, 29x29 33x33, 37x37, 41x41, 45x45, 49x49, 53x53	441, 625, 841 1089, 1369	25,47, 77, 114, 154, 195, 224, 279, 335
3264x2448	20,330	1, 2, 3, 4, 5, 6, 7, 8, 9....32	21x21, 25x25, 29x29, 33x33, 37x37, 41x41, 45x45, 49x49, 53x53...141x141	441, 625, 841 1089, 1369 19881	25,47, 77, 114, 154, 195, ...1952

To test the statistical un-delectability of the stego image with method IV and further verify that our proposed method can also work as steganographic technique as well as withstand steganalysis such as the chi-square attack, we opted to Chi-square steganography test program by Guillermito [55]. Figure 5.18 shows the test results for baboon image. In figure 5.20 the red curve is the result of the chi-square test. It has been always observed close to 0 for all stego images, so the probability for a random embedded message is low, in other words, nothing is hidden in our stego-image.



(a)



(b)

Figure 5.20: Chi-square analysis of (a) cover Baboon image (b) stego Baboon with version 2 QR code.

We performed Chi-square test by Guillermito program on most of our stego image and always find probability equal to or very near to zero.

Chapter – 6

Conclusions

In the era of digital document, digital image watermarking is a very active research field with a lot of promising applications. Within the course of this thesis, four new schemes for digital image watermarking has been developed which address two broad categories of image watermarking: robust and fragile. From watermark detection and extraction view point blind methods are more acceptable and have diverse application from information security view point. This research was started with simple DWT based non-blind robust watermarking and at this point has reached to DWT-SVD based blind fragile watermarking. We also proposed a DWT-SVD based semi-fragile non-blind watermarking method which succeeded to accomplish the requirement of medical images. However, we have not achieved all requirements for the medical image watermarking needs, i.e., blind semi-fragile watermarking with tampering detection and localization capacity; however, we were able to meet 2 of these 3 requirements. The most significant achievement from this study is the finding of blind DWT-SVD based watermarking method, that is method-IV. Moreover, the tampering detection ability with localizing any tampering in the watermarked image adds extra value to this proposed method. What is more, this method also can be used as image steganography with minimum cover image distortion.

Last but not the least, this research has also introduced QR-code as watermark which will open a new avenue in information hiding. However, in this proposed thesis we have only

tested the alphanumeric mode of the QR code, more mode i.e. 8 bit byte mode with encryption can extend the capacity and security.

The threshold of method-V for watermark extraction has been noticed to vary within 1.1 to 1.4. I want to investigate and fine tuning the proposed method –IV to find a unique threshold that can extract watermark or payload.

Bibliography

[1] A. M. Eskicioglu and E.J. Delp, "Security of Digital Entertainment Content From Creation to Consumption," Signal Processing : Image Communication, Special Issue on Image Security, vol. 118, no. 4, pp.237-262, April 2003.

[2] I. J. Cox, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Transactions on Image Processing, vol. 6, pp. 1673-1687, Dec. 1997.

[3] L. De Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes, and G. Depovere, 'Implementation of a real-time digital watermarking process for broadcast monitoring on a TriMedia VLIW processor', IEE Proceedings - Vision, Image and Signal Processing, vol. 147, no.4, pp. 371-376, 2000.

[4] J. Spaulding, H. Noda, M. N. Shirazi, E. Kawaguchi, 'BPCS Steganography using EZW Lossy Compressed images', Pattern Recognition Letters, vol. 23, no. 13, pp. 1579-15-87, Nov 2002.

[5] B. Li, J. He, J. Huang, Y. Q. Shi, 'A Survey on Image Steganography and Steganalysis', Journal of Information Hiding and Multimedia Signal Processing, vol. 2, no. 2, pp. 142-172, 2011.

[6] A. Cheddad, J. Condell, K. Curran, P. McKevitt, 'Digital Image Steganography: Survey and Analysis of current methods', Signal Processing, Elsevier, vol. 9, pp. 727-752, 2010.

[7] E. Koch, and J. Zhao,'Toward robust and hidden image copyright labeling', IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, Greece, 1995, pp. 1- 4.

[8] C. T. Hsu, and J. L. Wu,'Hidden digital watermarks in images', IEEE Trans. Image Processing, vol. 8, no. 1, pp. 58-68, 1999.

[9] D. Levicky, and P. Foris,'Human visual system models in digital image watermarking', Journal of Radio engineering, vol. 13, no. 4, pp. 38-43, 2004.

[10] R. Dugad, K. Ratakonda and N. Ahuja, 'A new wavelet-based Scheme for watermarking images', Proc. Intl. Conf. on Image Processing, Chicago, IL, USA 1998, pp. 419-423.

[11] Kundur, D. and D. Hatzinakos,'Digital watermarking using multiresolution wavelet decomposition', IEEE Trans. Signal Processing, vol. 49, no. 10, pp. 2383-2396, 2001.

[12] E. Elbasi, and A. M. Eskicioglu,'A DWT –Based Robust Semi-Blind Image Watermarking Algorithm using Two Bands', IS&T/SPIE's, 18th Annual Symp. On Elec. Imaging, Security, Steganography and Watermarking of Multimedia Contents.

[13] S. Hajjara, M. Abdallah, and A. Hudaib,'Digital Image Watermarking Using Localized Biorthogonal Wavelets', European Journal of Scientific Research, vol. 26, no. 4, pp. 594-608, 2009,

[14] S. Zahir, and M. W. Islam,'A New Wavelet –Based Image Watermarking Technique', IEEE Int. Con. Consumer Electronics (ICCE), Las Vegas USA, 2011, pp. 723-724.

[15] R. Liu, and T. Tan,'An SVD-based watermarking scheme for protecting rightful ownership', IEEE Transactions on Multimedia, vol. 4, pp. 121–128, 2002.

[16] E. Ganic, and A. M. Eskicioglu,'Robust Embedding of Visual Watermarks Using DWT-SVD', Journal of Electronic Imaging, vol. 14, no. 4, pp. 121-128, 2005.

[17] M. Calagna, H. Guo, L.V. Mancini, and S. A. Jajodia,'Robust Watermarking System Based on SVD Compression', Proceedings of ACM Symposium on Applied Computing (SAC2006). Dijon, France, 2006, pp. 1341-1347.

[18] Q. Li, C. Yuan, and Y. Z. Zong,'Adaptive DWT-SVD domain image watermarking using human visual model', Int. Conf. on Advanced Communication Technology, Phoenix Park, Korea, 2007, pp. 1947–1951.

[19] G. Bhatnagar, and B. Raman,'A new robust reference watermarking scheme based on DWT-SVD', Computer Standards & Interfaces,vol. 31, pp. 1002–1013, 2009.

[20] E. Yavuz, and Z. Telatar,'Improved SVD-DWT Based Digital Image Watermarking Against Watermark Ambiguity', Journal of Electronic Imaging, pp. 1051-1055, 2005.

[21] L. Liang, and S. Qi,'A New SVD-DWT composite watermarking', ICSP'06, 8th Int. Con. On Signal Processing: Proceedings, Guilin, China, November 16-20, 2006.

[22] E. Ganic, and A. M. Eskicioglu, 'Robust DWT-SVD Domain Image Watermarking: Embedding Data in All Frequencies', MM&SEC'04, September 20-21, Magdeburg, Germany, 2004.

[23] X. P. Zhang, and K. Li,'Comments on - An SVD-Based Watermarking Scheme for Protecting Rightful Ownership', IEEE Transactions On Multimedia: Correspondence, vol. 7, no. 2, pp. 593-594, 2005,

[24] E. Elbasi, A. M. Eskicioglu, PRN Based watermarking Scheme for Color Images, Journal of Istanbul Commercial University, vol. 5, no. 10, pp. 119-131, 2006.

[25] C. Yin, L. Li, A. Lv, and L. Qu, 'Color Image Watermarking Algorithm Based on DWT-SVD', IEEE Proc. Intl. Conf. Automation & Logistic, Jinan, China, 2007, pp. 2607- 2611

[26] M. H. Lee, O. J. Kwon,'Color Image Watermarking based on DS-CDMA using hadamard kernel', Proc. 10th IEEE Intl. Conf. Adv. Commun. Technology, Korea, 2008, pp. 1592-1597.

[27] F. Gui and L. Qiwei, 'Adaptive color image watermarking', Proc. SPIE, 6623, 2008, doi:10.1117/12.791517

[28] N. V. Dharwadkar, B.B. Amberker, and A. Gorai,'Non-blind Watermarking scheme for color images in RGB space using DWT-SVD', Intl. Conf. Communication and Signal Processing, 2011, pp. 489-493.

[29] P. Kapoor, K. K. Sharma, S. S. bedi, A. Kumar,'Color image watermarking technique based on HVs using HIS color Model', Proc. of Int. Conf. on Advances in Computer Engineering, Kerala, India, 2011, pp. 20-24.

[30] E. Vahedi , R. A. Zoroofi, M. Shiva,'Toward a new wavelet-based watermarking approach for color images using bio-inspired optimization principles', Digital Signal Processing, Elsevier, vol. 22, pp. 153-162, 2012.

[31] G. Coatrieux, H. Main, B. Sankur, Y. Rolland, and R. Collorec, 'Relevance of watermarking in medical imaging', In IEEE-embs Information Technology Applications in Biomedicine, Arlington, USA, Nov. 2000, pp. 250–255.

[32] http://medical.nema.org/Dicom/2011/11_11pu.pdf, page 27. Accesed on June 12, 2012.

[33] A. Giakoumaki, S. Pavlopoulos, Koutsouris,'Secure and efficient health data management through multiple watermarking on medical images', Med Bio Eng Comput, vol. 44, pp. 619-631, 2006.

[34] F. Rahimi and H. Rabbani ,'A dual adaptive watermarking scheme in contourlet domain for DICOM images' BioMedical Engineering Online , 10:53, 2011.

[35] J. H. K. Wu, R. F. Chang, C. J. Chen, C. L. Wang, T. H. Kuo, W. K. Moon, D. R. Chen, 'Tamper detection and recovery for medical images using near lossless information hiding technique ', Journal of Digital Imaging , vol 21, no. 1, pp. 59-76, 2008.

[36] N. A. Memom, and S. A. M. Gilani, 'NROI watermarking of medical images for content authentication', Proc. Of the 12th IEEE Intl. Multitopic Conference, December 23-24, 2008, pp. 106-110.

[37] C. R. Piao, D. M. Woo, D. C. Park, and S. S. Han, 'Medical image authentication using hash function and integer wavelet transform ', IEEE 2008 Congress on Image and Signal Processing , Snaya, Hainan, 2008, pp. 7-10.

[38] L. Xin , L Xiaoqi, W. Ying, 'A semi fragile digital watermarking algorithm based on integer wavelet matrix norm quantization for medical images', 2nd Intl. Conf. on Bioinformatics and Biomedical Engineering, Shanghai, China , May 16-18, 2008, pp. 776-779.

[39] N. Provos, P. Honeyman,'Hide and Seek: An Introduction to Steganography', IEEE Security and Privacy, vol. 1, no. 3, pp. 32-44, 2003.

[40] S. B. Sadkhan,'Cryptography: current status and future trends' Proc. Of IEEE Intl. Conf. on Information and Communication Technologies: From Theory to Applications, Damascus, Syria, April 19-23, 2004, pp. 417 - 418.

[41] P. Hayati,V. Potdar,E. Chang,'A survey of steganographic and steganalytic tools for the digital forensic investigator' , http://www.pedramhayati.com/images/docs/survey_of_steganography_and_steganalytic_tools.pdf accesed on June 12, 2012.

[42] Stego-Tools:[Camouflage]:http://camouflage.unfiction.com; accessed on June 12,2012

[JpegX]: /http://www.freewarefiles.com/Jpegx_program_19392.html

[DataStash]: /http://www.skyjuicesoftware.com/software/ds_info. html

[43] C. H. Tzeng and W. H. Tsai,'A combined approach to integrity protection and verification of palette im-ages using fragile watermarks and digital signatures', IEICE Transactions on Fundamentals, vol. E87-A, pp. 1612–1619, June 2004,

[44] C. H. Tzeng, Z. F. Yang, and W. H. Tsai,'Adaptive data hiding in palette images by color ordering and mapping with security protection', IEEE Transactions on Communications, vol. 52, pp 791-800, May 2004.

[45] M. Y. Wu, Y. K. Ho, and J. H. Lee,'An iterative method of palette-based image steganography', Pattern Recognition Letters, vol. 25, pp. 301–309, 2004.

[46] http://www.stegoarchive.com accessed on June 12, 2012

[47] C. K. Chan and L. M. Cheng, 'Hiding data in images by simple LSB substitution', Pattern recognition, vol. 2, pp. 469–474, 2004.

[48] A. D. Ker,'Steganalysis of lsb matching in grayscale images', IEEE Signal Processing Letters, vol. 12, pp. 441–444, June 2005.

[49] R. Z. Wang, C. F. Lin, and J. C. Lin,'Image hiding by optimal lsb substitution and genetic algorithm', Pattern Recognition, vol. 34, no. 3, pp. 671–683, 2001.

[50] H. C. Wu, N. I. Wu, C. S. Tsai, and M. S. Hwang,'Image steganographic scheme based on pixel-value differencing and LSB replacement methods', IEEE Proceedings of Visual Image Signal Process, vol. 152, pp. 611–615, Oct. 2005.

[51] K. L. Chung, C H Shen, L. C. Chang,'A novel SVD and VQ-based image hiding scheme', Pattern Recognition Letters, vol. 22, pp. 1051-1058, 2001.

[52] X. Li, J. Wang,'A steganographic method based upon JPEG and particle swarm optimization algorithm', Information Sciences, vol. 15, pp. 3099–31091, 2007.

[53] M. Iwata, K. Miyake, A. Shiozaki,'Digital steganography utilizing features of JPEG images', IEICE Transactions on Fundamentals, vol. E87-A, no. 4, pp. 929–936, 2004.

[54] C. C. Chang, C. C. Lin, C. S. Tseng, W. L. Tai, 'Reversible Hiding in DCT-based compressed images', Information Sciences , vol. 177, pp. 2768-2786, 2007.

[55] Guillermito, Chi-square Steganography Test Program http://www.guillermito2.net/stegano/tools/index.html accessed on June 12, 2012.

[56] K. B. Raja, C. R. Chowdary, K. R. Venugopal, L. M. Patnaik,'A secure image steganography using LSB, DCT and compression techniques on raw images', Proceedings of IEEE 3rd International Conference on Intelligent Sensing and Information Processing, ICISIP'05, Bangalore, India, December14–17, 2005, pp. 170–176.

[57] N. F. Johnson, S. Jajodia,'Exploring steganography: seeing the unseen', IEEE Computer vol. 31, no. 2, pp. 26–34, 1998.

[58] R. T. McKeon,'Strange Fourier steganography in movies', Proceedings of the IEEE International Conference on Electro/Information Technology (EIT), May 17–20, 2007, pp 178–182.

[59] P. Y. Chen, and H. J. Lin,'A DWT based approach for image steganography', International Journal of Appl. Science and Engineering, vol. 4, no. 3, pp. 275-290, 2006.

[60] L. Driskell , 'Wavelet based steganography', Cryptologia, Taylor & Francis, vol. 28, no. 2, pp. 157-174, 2010.

[61] H. Sajadi, M. Jamzad, 'Using contourlet transform and cover selection for secure steganography', International Journal of Information security, vol. 9, pp. 337-352, 2010.

[62] L. Driskell , 'Wavelet based steganography', Cryptologia, Taylor & Francis, vol. 28, no. 2, pp. 157-174, 2010.

[63] E. Aboufadel and S. Schlicker, 'Discovering wavelets', New York: John Wiley & Son, 1999.

[64] I. Daubechies, 'Ten Lectures on wavelets', Philadelphia: SIAM, 1992.

[65] R. T. McKeon, 'Strange Fourier steganography in movies', Proc. IEEE International Conference on Electro/Information Technology (EIT), May 17–20, 2007, pp 178–182

[66] P. C. Cosman, R. M. Gray, and R. A. Olshen,'Evaluating quality of compressed medical images: SNR, subjective rating and diagnostic accuracy',Proc. IEEE, vol. 82, pp. 920–931, June 1994.

[67] B. Roque, J. Salvado, 'A comparative study on JPEG like and EZW based image coders' http://repositorio.ipcb.pt/bitstream/10400.11/101/1/461-131.pdf accessed on June 12, 2012.

[68] J M Shapiro, 'Embedded Image coding using zerotrees of wavelet coefficients', IEEE transaction of Signal processing, vol. 41, pp. 3445-3465, 1993.

[69] H. C. Andrews, and C. L. Patterson, 'Singular Value decomposition (SVD) Image Coding', IEEE Transactions on Communications, vol. 24, no. 4, pp. 425-432, 1976.

[70] A. Sverdlov, S. Dexter, and A.M. Eskicioglu,'Robust DCT-SVD domain image watermarking for copyright protection: Embedding data in all frequencies', 13th Europian Signal Processing Conference EUSIPCO2005, Antalya, Turkey, 2005.

[71] http://www.denso-wave.com/qrcode/index-e.html accessed on June 12, 2012.

[72] http://www.iso.org/iso/catalogue_detail?csnumber=43655 accessed on June 12, 2012.

[73] K. Chen, and T. V. Ramabadran,'Near-lossless Compression of medical images through entropy –coded DPCM', IEEE Trans. Medical Imaging, vol. 13, no. 3, pp. 538-548, 1994.

[74] A. J. Maeder, and B. M. Planitz, 'Medical image watermarking for multiple modalities', *P*roc. 34th Applied Imagery and Pattern Recognition Workshop, 2005, pp. 158-165

[75] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli,'Image quality assessment: From error measurement to structural similarity', IEEE Transactios on Image Processing, vol. 13, no. 1, pp. 1-14, 2004.

[76] A. P. Fabien Petitcolas, R. J. Anderson, and M. G. Kuhn. Attacks on copyright marking systems. In Aucsmith, pp 218-238, ISBN 3-540-65386-4

[77] S. C. Katzenbeisser, F.A.P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Inc, Norwood, 2000.

[78] R. T. McKeon, Strange Fourier steganography in movies, Proc. IEEE Intl. Conf. on Electro/Information Technology (EIT), May 17–20, 2007, pp 178–182

[79] A Westfield, and A Pfitzmann, 'Attacks on Steganographic Systems: Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools and Some Lessons Learned', 3rd Intl. Workshop on Information Hiding (2000).

[80] B Roque, J Salvado 'A comparative study on JPEG like and EZW based image coders' http://repositorio.ipcb.pt/bitstream/10400.11/101/1/461-131.pdf accessed on June 12, 2012

[81] http://inst.eecs.berkeley.edu/~cs150/Documents/ITU601.pdf. accessed on JUne 12, 2012

[82] C. H. Chou, K. C. Liu, Performance Analysis of Color Image Watermarking Schemes Using Perceptually Redundant Signal Spaces, Proc. Intl. Conf. Intelligent Information Hiding and Multimedia Signal Processing, 2006.

[83] E. Vahedi, R. A. Zoroofi, M. Shiva, 'On optimal color coordinate selection for wavelet-based color image watermarking', Proc. of the ICIAS'07 International Conference, Kuala Lumpur, Malaysia, 2007

Appendix-1

Table A.1 watermarked texture images PSNR and SSIM of SIPI image database using Method -III

Image (Texture)	PSNR	SSIM	Image	PSNR	SSIM
1.1.01	40.93	0.998	1.4.01	33.50	0.975
1.1.02	37.79	0.996	1.4.02	28.02	0.913
1.1.03	38.71	0.996	1.4.03	30.51	0.951
1.1.04	36.02	0.979	1.4.04	33.00	0.936
1.1.05	39.34	0.980	1.4.05	26.35	0.861
1.1.06	39.39	0.996	1.4.06	29.90	0.922
1.1.07	38.34	0.990	1.4.07	32.93	0.962
1.1.08	33.97	0.995	1.4.08	25.42	0.889
1.1.09	33.26	0.996	1.4.09	25.29	0.886
1.1.10	37.55	0.997	1.4.10	37.86	0.993
1.1.11	33.43	0.994	1.4.11	28.99	0.953
1.1.12	31.21	0.993	1.4.12	34.78	0.984
1.1.13	41.13	0.997	1.5.01	34.59	0.963
1.2.01	39.84	0.995	1.5.02	23.48	0.948
1.2.02	38.16	0.995	1.5.03	25.16	0.776
1.2.03	39.50	0.989	1.5.04	25.40	0.812
1.2.04	33.86	0.978	1.5.05	25.15	0.647
1.2.05	37.62	0.976	1.5.06	27.04	0.458
1.2.06	38.72	0.986	1.5.07	48.22	0.982
1.2.07	39.69	0.989	5.1.13	29.17	0.794
1.2.08	35.02	0.976	5.1.14	38.22	0.947
1.2.09	31.08	0.956	5.2.08	36.36	0.941
1.2.10	34.45	0.967	5.2.09	35.83	0.983
1.2.11	38.66	0.988	5.2.10	32.53	0.971
1.2.12	28.63	0.962	5.3.01	37.10	0.979
1.2.13	38.43	0.986	5.3.02	42.10	0.979
1.3.01	44.96	0.991	7.1.01	47.01	0.990
1.3.02	41.54	0.994	7.1.02	42.54	0.990
1.3.03	44.23	0.996	7.1.03	48.84	0.995
1.3.04	42.93	0.997	7.1.04	41.44	0.988
1.3.05	37.11	0.991	7.1.05	43.56	0.986
1.3.06	45.08	0.996	7.1.06	45.26	0.996
1.3.07	48.99	0.996	7.1.07	49.17	0.995
1.3.08	50.40	0.996	7.1.08	46.95	0.991
1.3.09	40.00	0.962	7.1.09	47.01	0.991
1.3.10	42.47	0.980	7.1.10	48.66	0.994
1.3.11	49.33	0.996	7.2.01	47.62	0.991
1.3.12	34.70	0.937	Texture Average	37.54	0.957
1.3.13	41.68	0.995			

Table A.2 watermarked miscellaneous images PSNR and SSIM of SIPI image database using Method -III

Image (Miscellaneous)	PSNR	SSIM	Image	PSNR	SSIM
4.1.01	43.48	0.980	4.2.01	36.86	0.921
4.1.02	38.43	0.977	4.2.02	34.99	0.946
4.1.03	37.28	0.947	4.2.03	40.31	0.994
4.1.04	37.28	0.947	4.2.04	41.69	0.970
4.1.05	37.89	0.947	4.2.05	49.16	0.993
4.1.06	39.64	0.965	4.2.06	38.27	0.957
4.1.07	41.70	0.992	4.2.07	38.47	0.957
4.1.08	42.70	0.992	Boat	39.88	0.967
4.1.09	41.67	0.991	Average Miscellaneous	39.98	0.967

Table A.3 watermarked miscellaneous images PSNR and SSIM of SIPI image database using Method –IV

Image (Miscellaneous)	PSNR	SSIM	Image	PSNR	SSIM
4.1.01	40.48	0.978	4.2.01	33.56	0.943
4.1.02	34.43	0.956	4.2.02	33.79	0.944
4.1.03	36.22	0.931	4.2.03	38.45	0.989
4.1.04	33.27	0.921	4.2.04	36.77	0.968
4.1.05	35.91	0.956	4.2.05	39.99	0.987
4.1.06	36.63	0.956	4.2.06	35.54	0.946
4.1.07	38.74	0.987	4.2.07	35.79	0.966
4.1.08	40.65	0.991	Boat	34.22	0.956
4.1.09	39.22	0.989	Average Miscellaneous	36.68	0.962

Table A.4 watermarked texture images PSNR and SSIM of SIPI image database using Method IV

Image (Texture)	PSNR	SSIM	Image	PSNR	SSIM
1.1.01	38.23	0.981	1.4.01	32.50	0.965
1.1.02	32.87	0.982	1.4.02	30.02	0.910
1.1.03	35.10	0.978	1.4.03	30.00	0.941
1.1.04	33.00	0.965	1.4.04	32.75	0.923
1.1.05	34.51	0.976	1.4.05	29.00	0.851
1.1.06	36.34	0.982	1.4.06	30.30	0.922
1.1.07	35.61	0.988	1.4.07	34.83	0.912
1.1.08	31.86	0.945	1.4.08	24.42	0.869
1.1.09	31.26	0.941	1.4.09	27.29	0.816
1.1.10	35.45	0.956	1.4.10	35.86	0.923
1.1.11	32.44	0.944	1.4.11	26.99	0.853
1.1.12	31.66	0.941	1.4.12	31.67	0.924
1.1.13	37.74	0.957	1.5.01	31.67	0.900
1.2.01	36.85	0.951	1.5.02	27.88	0.792
1.2.02	35.92	0.950	1.5.03	26.17	0.776
1.2.03	37.11	0.967	1.5.04	25.55	0.812
1.2.04	32.65	0.921	1.5.05	25.45	0.647
1.2.05	34.62	0.944	1.5.06	27.67	0.758
1.2.06	32.72	0.939	1.5.07	47.13	0.992
1.2.07	35.79	0.980	5.1.13	28.67	0.794
1.2.08	32.51	0.937	5.1.14	34.29	0.939
1.2.09	30.08	0.911	5.2.08	35.79	0.939
1.2.10	31.22	0.921	5.2.09	35.12	0.982
1.2.11	34.43	0.942	5.2.10	31.89	0.970
1.2.12	30.67	0.912	5.3.01	35.22	0.967
1.2.13	34.78	0.962	5.3.02	39.22	0.971
1.3.01	40.12	0.989	7.1.01	43.55	0.980
1.3.02	37.22	0.984	7.1.02	40.05	0.976
1.3.03	41.11	0.985	7.1.03	45.23	0.955
1.3.04	38.99	0.981	7.1.04	39.48	0.981
1.3.05	33.33	0.923	7.1.05	38.99	0.976
1.3.06	39.22	0.981	7.1.06	42.26	0.987
1.3.07	41.99	0.985	7.1.07	47.04	0.997
1.3.08	44.11	0.984	7.1.08	44.56	0.989
1.3.09	37.77	0.922	7.1.09	43.59	0.992
1.3.10	38.82	0.910	7.1.10	46.11	0.986
1.3.11	40.11	0.913	7.2.01	43.22	0.989
1.3.12	32.39	0.922	Texture Average	34.47	0.967
1.3.13	40.01	0.986			

Table A.5 watermarked Flower images PSNR and SSIM of McGill Image Database using Method III

Image (Flowers)	PSNR	SSIM	Image	PSNR	SSIM
merry_flor0003	41.62	0.991	pippin_park0025	43.85	0.991
merry_flor0008	45.11	0.993	pippin_park0053	45.15	0.991
merry_flor0013	43.93	0.996	pippin_Peel010	43.50	0.992
merry_florida0003	47.73	0.992	pippin_Peel040	43.67	0.991
merry_florida0004	43.32	0.996	pippin_Peel041	41.96	0.989
merry_florida0005	43.82	0.994	pippin_Peel049	44.04	0.993
merry_florida0006	42.01	0.993	pippin_Peel052	43.51	0.992
merry_florida0007	43.71	0.996	pippin_Peel064	46.95	0.994
merry_florida0026	42.23	0.995	pippin0163	43.75	0.991
merry_florida0034	42.23	0.995	pippin0164	41.79	0.994
merry_mexico0083	41.19	0.983	pippin0165	43.61	0.994
merry_mexico0094	45.77	0.994	pippin0172	41.87	0.995
merry_mexico0096	45.30	0.991	pippin0211	50.26	0.996
merry_mexico0097	39.71	0.980	pippin0242	44.54	0.994
merry_mexico0106	47.07	0.993	pippin_Peel068	44.06	0.991
merry_mexico0132	46.78	0.992	pippin_summer0004	43.61	0.992
merry_mexico0133	47.00	0.992	pippin_summer0008	42.62	0.992
merry_mexico0152	45.11	0.993	pippin_UniStrt025	44.39	0.992
merry_mexico0228	44.86	0.991	pippin0112	45.06	0.989
merry_mexico0229	45.09	0.990	pippin0113	52.77	0.996
merry_mexico0231	44.40	0.991	pippin0114	46.86	0.992
merry_mexico0244	42.12	0.993	pippin0115	47.21	0.993
merry_mpark0010	42.57	0.991	pippin0116	45.40	0.990
merry0059	44.80	0.981	pippin0119	46.92	0.992
merry0066	47.99	0.992	pippin0126	47.97	0.989
merry0192	43.92	0.991	pippin0159	40.36	0.991
pippin_jamaica0230	47.12	0.992	pippin0161	43.21	0.990
pippin_jamaica0236	43.47	0.983	pippin0162	42.13	0.995
pippin_jamaica0238	43.88	0.992	pippin0253	45.71	0.995
pippin_jamaica0260	44.55	0.989	pippin0258	43.48	0.989
pippin_jtalon0023	43.88	0.990	pippin0265a	48.29	0.994
pippin_jtalon0024	43.13	0.990	pippin0269	46.00	0.990
pippin_jtalon0025	43.15	0.989	pippin0270	46.20	0.994
pippin_jtalon0026	48.24	0.994	***Flower average***	***44.62***	***0.991***
pippin_jtalon0027	45.55	0.992			
pippin_Mex07_010	44.441	0.990			
pippin_Mex07_013	44.57	0.989			
pippin_Mex07_014	43.93	0.992			
pippin_Mex07_019	41.57	0.986			
pippin_Mex07_034	47.34	0.995			
pippin_Mex07_035	47.07	0.993			

Table A.6 watermarked Fruit's images PSNR and SSIM of McGill Image Database using Method III

Image (Fruits)	PSNR	SSIM	Image	PSNR	SSIM
merry_mtl07_015	51.22	0.995	pippin_jtalon0056	39.97	0.984
merry_mtl07_017	42.09	0.990	pippin_jtalon0057	46.27	0.990
merry_mtl07_018	42.62	0.985	pippin_jtalon0058	44.26	0.989
pippin_jtalon0002	46.19	0.993	pippin_jtalon0062	45.47	0.990
pippin_jtalon0003	51.00	0.996	pippin_jtalon0063	47.24	0.993
pippin_jtalon0005	44.66	0.998	pippin_Mex07_012	44.14	0.993
pippin_jtalon0017	49.37	0.995	pippin_Mex07_024	42.16	0.988
pippin_jtalon0036	44.41	0.994	pippin_Mex07_025	44.34	0.995
pippin_jtalon0037	45.51	0.992	pippin_Mex07_027	44.21	0.995
pippin_jtalon0039	44.33	0.990	pippin_Mex07_033	40.98	0.988
pippin_jtalon0040	44.84	0.989	pippin_Peel064	46.95	0.994
pippin_jtalon0042	46.89	0.992	pippin0163	43.75	0.991
pippin_jtalon0043	47.81	0.995	pippin0164	41.79	0.994
pippin_jtalon0046	41.47	0.986	pippin0165	43.61	0.994
pippin_jtalon0047	48.37	0.995	pippin0172	41.87	0.995
pippin_jtalon0049	43.65	0.989	pippin0211	50.26	0.996
pippin_jtalon0051	50.19	0.995	pippin0242	44.54	0.994
			Fruits Average	***45.32***	***0.991***

Table A.7 watermarked Animal's images PSNR and SSIM of McGill Image Database using Method III

Image (Animals)	PSNR	SSIM	Image	PSNR	SSIM
pippin_ParcSafari004	41.47	0.989	pippin0054	43.88	0.994
pippin_ParcSafari008	45.63	0.994	pippin0056	41.92	0.991
pippin_ParcSafari011	51.64	0.977	pippin0057	46.48	0.994
pippin_ParcSafari015	45.25	0.995	pippin0064	42.63	0.9885
pippin_ParcSafari017	45.24	0.995	pippin0094	44.47	0.992
pippin_ParcSafari029	48.64	0.995	pippin0097	45.25	0.993
pippin_ParcSafari036	45.32	0.994	pippin0101	46.05	0.993
pippin_ParcSafari037	48.90	0.995	pippin0265	44.98	0.995
pippin_ParcSafari043	46.48	0.993	pippin0275	46.58	0.995
pippin_ParcSafari049	49.12	0.995	pippin0281	46.23	0.992
pippin0021	44.63	0.994	***Animal Average***	***46.52***	***0.992***
pippin0053	39.95	0.985			

Table A.8 watermarked Landscape's images PSNR and SSIM of McGill Image Database using Method III

Image	PSNR	SSIM	Image	PSNR	SSIM
Merry_0043_Lasalle	48.51	0.996	merry_mexico0168	50.14	0.994
Merry_0046_Lasalle	46.08	0.995	merry_mexico0171	49.45	0.994
Merry_0052_Lasalle	48.86	0.988	merry_mexico0178	46.84	0.994
Merry_0055_Lasalle	46.87	0.996	merry_mexico0184	47.96	0.995
Merry_0056_Lasalle	46.55	0.995	pippin_adirondacks7	45.04	0.995
Merry_0043_Lasalle	47.15	0.995	pippin0062	48.94	0.996
Merry_0046_Lasalle	48.25	0.995	pippin_ParcSafri049	49.12	0.995
Merry_0052_Lasalle	44.86	0.996	merry_mexico0111	48.72	0.996
Merry_0055_Lasalle	46.87	0.996	merry_mexico0125	58.18	0.998
Merry_0058_Lasalle	46.86	0.995	merry_mexico0152	53.74	0.998
merry_italy0149	54.83	0.998	pippin0140	49.50	0.996
merry_mexico0056	48.80	0.994	pippin0143	47.89	0.996
merry_mexico0058	42.91	0.990	pippin0184	51.70	0.996
merry_mexico0109	49.63	0.997	Merry_0038_Lasalle	42.78	0.988
merry_mexico0119	49.25	0.995	***Landscape Average***	***48.17***	***0.995***

Table A.9 PSNR and SSIM of McGill Image Database using Method III

Image (Manmade)	PSNR	SSIM	Image	PSNR	SSIM
Merry_0002_Lasalle	43.53	0.993	merry0081	42.07	0.988
Merry_0003_Lasalle	49.59	0.997	merry0082	43.76	0.993
Merry_0005_Lasalle	41.39	0.989	merry0086	44.17	0.990
Merry_0006_Lasalle	45.11	0.992	merry0091	42.22	0.990
Merry_0014_Lasalle	41.33	0.991	merry0108	45.10	0.994
Merry_0016_Lasalle	43.47	0.992	merry0199	42.36	0.989
Merry_0017_Lasalle	42.23	0.990	merry0201	41.21	0.986
Merry_0021_Lasalle	41.98	0.988	pippin_Peel023	45.31	0.994
Merry_0022_Lasalle	42.25	0.990	pippin_Peel070	46.57	0.995
Merry_0031_Lasalle	43.95	0.989	pippin_UniStrt001	44.56	0.994
Merry_0032_Lasalle	45.48	0.992	pippin_UniStrt019	42.99	0.984
Merry_0047_Lasalle	46.85	0.995	merry_mexico0185	46.30	0.996
Merry_0061_Lasalle	46.85	0.995	merry_mexico0192	50.84	0.997
Merry_0062_Lasalle	46.41	0.993	merry_mexico0200	46.64	0.996
merry_italy0010	45.22	0.992	merry_mexico0201	49.96	0.997
merry_italy0144	42.03	0.991	merry_mexico0223	46.60	0.994
merry_mexico0061	41.45	0.988	merry0129	48.81	0.995
merry_mexico0116	44.62	0.992	merry0130	47.63	0.996
merry_mtl07_029	42.73	0.988	merry0134	44.86	0.995
merry_toys0005	50.81	0.997	merry0135	47.09	0.995
merry_win07_003	44.13	0.991	pippin_adirondacks2	49.74	0.996
merry0078	51.65	0.995	pippin_adirondacks3	50.42	0.997
merry0080	42.35	0.984	pippin_adirondacks4	52.22	0.997
pippin_adirondacks5	52.30	0.996	***Average***	***44.28***	***0.991***

Table A.10 Watermarked Image PSNR and SSIM of McGill Image Database using Method IV.

Image (Flower)	PSNR	SSIM	Thres.	**Image**	PSNR	SSIM	Thres.
merry0066	46.20	0.977	1.28	pippin_jtalon26	38.58	0.938	1.26
merry_flor0003	34.23	0.934	1.30	pippin_jtalon27	42.03	0.971	1.33
merry_flor0008	40.09	0.973	1.10	pippin_Mex7_1	44.23	0.982	1.15
merry_flor0013	40.96	0.972	1.25	pippin_Mex7_3	40.34	0.967	1.20
merry_florida0003	36.15	0.975	1.10	pippin_Mex7_4	47.04	0.986	1.25
merry_florida0004	36.58	0.973	1.30	pippin_Mex7_9	39.57	0.956	1.20
merry_florida0005	36.85	0.963	1.25	pippin_Mex734	46.12	0.983	1.12
merry_florida0006	35.64	0.963	1.26	pippin_Mex735	42.19	0.980	1.20
merry_florida0007	36.89	0.979	1.10	pippin_park025	37.60	0.960	1.12
merry_florida0026	37.01	0.972	1.15	pippin_park053	44.04	0.957	1.10
merry_florida0034	35.56	0.971	1.15	pippin_Peel10	36.31	0.945	1.19
merry_mexico0083	43.49	0.973	1.25	pippin_Peel040	36.16	0.949	1.22
merry_mexico0094	46.06	0.985	1.30	pippin_Peel041	36.18	0.958	1.15
merry_mexico0096	38.99	0.940	1.25	pippin_Peel068	38.86	0.957	1.30
merry_mexico0097	41.15	0.969	1.18	pippin_summe4	36.34	0.959	1.30
merry_mexico0106	39.85	0.969	1.10	pippin_summe8	36.24	0.968	1.10
merry_mexico0132	40.32	0.963	1.22	pippin_UniSrt25	36.60	0.957	1.10
merry_mexico0133	40.99	0.965	1.25	pippin0112	39.15	0.944	1.21
merry_mexico0152	40.09	0.973	1.10	pippin0113	44.61	0.980	1.15
merry_mexico0228	36.74	0.952	1.12	pippin0114	38.29	0.940	1.25
merry_mexico0229	43.55	0.974	1.30	pippin0115	41.27	0.964	1.30
merry_mexico0231	44.24	0.981	1.27	pippin0116	37.91	0.939	1.20
merry_mexico0244	38.27	0.972	1.10	pippin0119	38.89	0.952	1.20
merry_mpark0010	37.05	0.929	1.30	pippin0126	38.76	0.955	1.10
merry0059	38.85	0.946	1.25	pippin0159	32.68	0.940	1.15
merry0192	37.06	0.959	1.20	pippin0161	37.29	0.953	1.33
pippin_jamaica0230	38.29	0.949	1.10	pippin0162	35.10	0.957	1.30
pippin_jamaica0236	36.42	0.927	1.22	pippin0253	40.61	0.976	1.14
pippin_jamaica0238	37.26	0.962	1.15	pippin0258	35.84	0.937	1.12
pippin_jamaica0260	35.47	0.918	1.21	pippin0265a	42.79	0.961	1.33
pippin_jtalon0023	37.64	0.948	1.25	pippin0269	40.26	0.963	1.12
pippin_jtalon0024	39.47	0.969	1.12	pippin0270	39.58	0.964	1.12
pippin_jtalon0025	37.82	0.960	1.11	***Flower average***	***39.04***	***0.961***	

Table A.11 Watermarked Fruit'sImages PSNR and SSIM of McGill Image Database using Method IV.

Image (Fruits)	PSNR	SSIM	Thre.	**Image (Fruits)**	PSNR	SSIM	Thres.
merry_mtl7_015	39.66	0.952	1.30	pippin_jtalon57	37.57	0.946	1.20
merry_mtl7_017	34.57	0.942	1.30	pippin_jtalon58	36.98	0.945	1.25
merry_mtl7_018	44.37	0.979	1.40	pippin_jtalon62	39.72	0.958	1.17
pippin_jtalon02	40.95	0.976	1.10	pippin_jtalon63	44.03	0.979	1.10
pippin_jtalon03	41.94	0.948	1.40	pippin_Mex712	40.42	0.967	1.20
pippin_jtalon05	38.95	0.980	1.10	pippn_Mex7_24	40.90	0.953	1.40
pippin_jtalon17	39.02	0.966	1.10	pippn_Mex7_25	35.84	0.958	1.10
pippin_jtalon36	37.82	0.971	1.10	pippin_Peel049	35.10	0.957	1.10
pippin_jtalon37	43.67	0.971	1.30	pippin_Peel052	36.10	0.949	1.3
pippin_jtalon39	37.40	0.964	1.20	pippin_Peel064	37.98	0.953	1.25
pippin_jtalon40	40.50	0.936	1.30	pippin0163	37.35	0.947	1.12
pippin_jtalon42	37.75	0.957	1.30	pippin0164	35.07	0.971	1.10
pippin_jtalon43	38.41	0.941	1.40	pippin0165	35.38	0.966	1.15
pippin_jtalon46	34.17	0.939	1.35	pippin0172	35.44	0.974	1.13
pippin_jtalon0047	45.27	0.985	1.10	pippin0211	47.31	0.987	1.10
pippin_jtalon0049	38.10	0.966	1.10	pippin0242	42.59	0.977	1.30
pippin_jtalon0051	43.44	0.954	1.35	pippn_Mex7_27	39.99	0.980	1.18
pippin_jtalon0056	32.87	0.930	1.35	pippn_Mex7_33	35.05	0.943	1.25
				Fruits Average	***39.23***	***0.958***	

Table A.12 Watermarked Animal's Images PSNR and SSIM of McGill Image Database using Method IV.

Image (Animal)	PSNR	SSIM	Thre.	**Image (Animal)**	PSNR	SSIM	Thres.
merry_mexico0111	42.54	0.980	1.10	pippin0021	40.18	0.981	1.13
merry_mexico0125	49.99	0.991	1.15	pippin0053	34.60	0.942	1.20
merry_mexico0152	38.37	0.982	1.20	pippin0054	36.38	0.965	1.10
pippin_Mex07_008	42.35	0.979	1.34	pippin0056	35.63	0.962	1.10
pippin_ParcSafari4	38.65	0.962	1.20	pippin0057	41.18	0.975	1.10
pippin_ParcSafari8	38.63	0.969	1.30	pippin0064	38.28	0.955	1.20
pippin_ParcSafari11	44.13	0.984	1.10	pippin0094	39.68	0.959	1.20
pippin_ParcSafari15	40.68	0.978	1.10	pippin0097	40.25	0.973	1.40
pippin_ParcSafari17	39.97	0.970	1.20	pippin0101	40.62	0.973	1.29
pippin_ParcSafari29	43.34	0.981	1.10	pippin0265	40.72	0.980	1.20
pippin_ParcSafari36	39.20	0.973	1.10	pippin0275	40.48	0.980	1.10
pippin_ParcSafari37	43.41	0.984	1.10	pippin0281	39.85	0.967	1.10
pippin_ParcSafari43	39.78	0.976	1.10	***Animal Average***	***40.36***	***0.972***	
pippin_ParcSafari49	40.56	0.974	1.10				
merry_mexico0111	42.54	0.980	1.10				

Table A.13 Watermarked Landscape's Images PSNR and SSIM of McGill Image Database using Method IV.

Image (Landscape)	PSNR	SSIM	Thre.	**Image (Landscape)**	PSNR	SSIM	Thres.
Merry_0043_LS	41.35	0.972	1.20	merry_italy0149	48.97	0.990	1.25
Merry_0046_LS	42.29	0.983	1.20	merry_mexic56	43.89	0.975	1.30
Merry_0052_LS	39.62	0.980	1.10	merry_mexic58	39.45	0.975	1.35
Merry_0055_LS	41.60	0.981	1.10	merry_mexi109	38.95	0.974	1.30
Merry_0056_LS	40.72	0.967	1.25	merry_mexi119	46.43	0.990	1.22
Merry_0043_LS	41.35	0.972	1.20	pippin_adirond7	37.42	0.973	1.27
Merry_0046_LS	42.29	0.983	1.20	pippin0062	41.96	0.979	1.10
Merry_0052_LS	39.62	0.980	1.10	pippin0140	42.12	0.987	1.19
Merry_0055_LS	41.60	0.981	1.10	pippin0143	45.47	0.986	1.30
Merry_0058_LS	40.14	0.980	1.10	pippin0184	45.40	0.976	1.17
merry_italy0125	49.15	0.991	1.25	***Landscape Avg***	***44.16***	***0.982***	

Table A.13 Watermarked Manmade Object's Images PSNR and SSIM of McGill Image Database using Method IV.

Image	PSNR	SSIM	Thre.	Image	PSNR	SSIM	Thres.
Merry_02_Lasal	39.73	0.973	1.21	merry0082	37.36	0.963	1.18
Merry_03_Lasal	40.31	0.974	1.18	merry0086	39.85	0.968	1.30
Merry_05_Lasal	33.04	0.934	1.40	merry0091	35.60	0.925	1.35
Merry_06_Lasal	38.14	0.955	1.35	merry0108	40.19	0.970	1.30
Merry_014_Las	36.64	0.969	1.15	merry0199	37.81	0.963	1.27
Merry_16_Lasa	38.74	0.969	1.30	merry0201	37.44	0.946	1.26
Merry_17_Lasa	35.95	0.951	1.35	pippin_Peel023	40.99	0.973	1.19
Merry_21_Lasal	38.24	0.965	1.29	pippin_Peel070	36.18	0.962	1.24
Merry_22_Lasal	35.93	0.960	1.40	pippin_UnStrt01	38.91	0.974	1.15
Merry_31_Lasal	37.26	0.949	1.40	pippin_UnStrt19	37.57	0.940	1.28
Merry_0032_La	38.28	0.936	1.35	mery_mexco168	43.23	0.977	1.16
Merry_0047_Lasalle	39.01	0.970	1.25	mery_mexco171	50.63	0.991	1.10
Merry_0061_Lasalle	40.54	0.962	1.25	mery_mexco178	47.38	0.987	1.10
Merry_0062_Lasalle	40.68	0.975	1.23	mery_mexco184	48.19	0.988	1.10
merry_italy0010	37.54	0.949	1.22	mery_mexco185	42.19	0.981	1.10
merry_italy0144	36.67	0.961	1.20	mery_mexco192	44.41	0.988	1.20
merry_mexico0061	37.75	0.967	1.10	mery_mexco200	42.55	0.986	1.30
merry_mexico0116	39.99	0.970	1.19	mery_mexco201	46.87	0.987	1.10
merry_mtl07_029	37.65	0.955	1.24	mery_mexco223	42.39	0.981	1.10
merry_toys0005	41.18	0.973	1.14	merry0129	67.20	0.999	1.40
merry_win07_003	37.48	0.960	1.26	merry0130	42.97	0.981	1.40
merry0078	42.44	0.959	1.40	merry0134	40.49	0.978	1.15
merry0080	40.40	0.967	1.35	merry0135	42.38	0.979	1.20
merry0081	37.22	0.945	1.35	***Manmade Obj Image Average***	***38.31***	***0.959***	