

SP07123

Jaypee University of Information Technology

Waknaghat, Distt. Solan (H.P.)

Learning Resource Center

Class Num :

Book Num :

Accession No.: SP07123 / SP0711121

This book was issued is overdue due on the date stamped below. If the book is kept over due, a fine will be charged as per the library rules.

Due Date	Due Date	Due Date

Invisible Digital Image Watermarking Using DCT &
DWT

Abhijit Singh Wander 071030

Amit Bhardwaj 071034

Kartikeya Khanna 071085

Name of Supervisor Mr. Pardeep Garg



Submitted in partial fulfillment of the degree of
Bachelor of Technology

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING

Jaypee University of Information Technology
Waknaghat, Solan - 173234, Himachal Pradesh

TABLE OF CONTENTS


Certificate.....	III
Acknowledgement.....	IV
Summary.....	V
List of Figures.....	VI
List of Tables.....	VIII
1. Introduction.....	1
2. Review of Watermarking.	
2.1 History.....	4
2.2 General Model of Watermarking.....	6
2.3 Types Of Watermarking.....	7
2.4 General Framework for Watermarking.....	9
2.5 Watermarking vs. Steganography.....	10
2.6 Watermarking vs. Cryptography.....	10
2.7 Watermarking vs. Digital Signatures.....	11
2.8 Properties of Watermarking.....	11
2.9 Watermarking Techniques.....	12
2.10 Applications.....	13
2.11 Limitations.....	15
3. Invisible Watermarking using Discrete Cosine Transform	
3.1 Overview.....	16
3.2 Technique Implemented.....	18
3.3 Algorithm	
3.3.1 Flowchart.....	20
3.3.2 Stepwise Implementation.....	21

4. Invisible Watermarking using Discrete Wavelet Transform	
4.1 Overview	
4.1.1 What are Wavelets?	24
4.1.2 Continuous Wavelet Transform.....	27
4.1.3 Discrete Wavelet Transform.....	30
4.2 Technique Implemented.....	35
4.3 Algorithm	
4.3.1 Stepwise Implementation.....	36
5. Results and Observations	
5.1 Results using DCT.....	39
5.2 Results using DWT.....	41
6. Source Code	
6.1 Watermark Embedding Using DCT.....	43
6.2 Watermark Recovery Using DCT.....	46
6.3 Watermark Embedding Using DWT.....	47
6.4 Watermark Recovery Using DCT.....	49
7. Conclusion and Future Work.....	51
References.....	52
Brief Bio Data.....	54

CERTIFICATE

This is to certify that project report entitled “**Invisible Digital Image Watermarking Using DCT & DWT**”, submitted by **Abhijit Singh Wander, Amit Bhardwaj and Kartikeya Khanna** in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Wagnaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor.....

Name of Supervisor ..PARDEEP SINGH

Designation ..LECTURER

Date ..23-05-2011

ACKNOWLEDGEMENT

As we conclude our project with the grace of God, we look back to thank; for all the help, guidance and support they lent us, throughout the course of our endeavor. First and foremost, we thank laudable Mr. Pardeep Garg, our Project Guide, who has always encouraged us to put in our best efforts and deliver a quality and professional output. His methodology of working over the basics and laying a strong foundation has taught us that output is not the END of project. We really thank him for his time and efforts. Secondly, we thank Prof. S. V. Bhooshan, Dr. Vivek Sehgal and Mr. Salman Raju for their patient hearing of our ideas and opening up our minds to newer horizons by pointing out our flaws, providing critical comments and suggestions to improve the quality of our work and appreciating our efforts. We also thank Mr. Mohan Sharma for guiding us throughout our lab work and providing us an immense support. Apart from these, countless events, countless people and several incidents have made a contribution to this project that is indescribable.

Name of the students:

Abhijit Singh Wander

Amit Bhardwaj

Kartikeya Khanna

Signature of the students:

Abhijit

Amit Bhardwaj

Khanna


Date: 23-05-2011

SUMMARY

Digital image watermarking has gained a great interest in the last decade among researchers. Having such a great community which provides a continuously growing list of proposed algorithms, it is rapidly finding solutions to its problems. However, still we are far away from being successful. Therefore, more and more people are entering the field to make the watermarking idea useful and reliable for digital world. Of these various watermarking algorithms, some outperform others in terms of basic watermarking requirements like robustness, invisibility, processing cost, etc. The recent progress in the digital multimedia technologies has offered many facilities in the transmission, reproduction and manipulation of data. However, this advancement has also brought the challenge such as copyright protection for content providers. Digital watermarking is one of the proposed solutions for copyright protection of multimedia data. This technique is better than Digital Signatures and other methods because it does not increase overhead.

In this project, we have tried to create different watermarking algorithms for generating invisible watermarks. We have also implemented those algorithms and tried to find out their effectiveness. Algorithms are chosen to be representatives of different categories such as spatial and transform domain. We have made use of DCT and DWT domains for embedding and extracting a watermark into an image. We try to figure out the properties of the methods that make them vulnerable or invulnerable against these attacks.

Image content authentication is to verify the integrity of the images, i.e. to check if the image has undergone any tampering since it was created. This would sustain the integrity and help us to forgo the forgery or tampering of images so that they can be shared safely over the World Wide Web.

Signature of Supervisor.....

Name of Supervisor ..PARDEEP GARG

Date23-05-2011

LIST OF FIGURES USED

1. Introduction	
1.1 Motivations behind Watermarking.....	2
2. Background of Watermarking	
2.1 Watermarks in Currency Notes.....	5
2.2 Watermark Embedding Model.....	6
2.3 Watermark Recovery Model.....	6
2.4 Types of Watermarking Techniques.....	7
2.5 Visible and Invisible Watermark.....	8
2.6 Digital Watermarking System.....	9
3. Invisible Watermarking Using Discrete Cosine Transform	
3.1 Components of a typical image/video transmission system.....	17
3.2 Definition of DCT regions.....	19
3.3 DCT Embedding Algorithm.....	20
3.4 DCT Extraction Algorithm.....	20
3.5 Lena & Watermark.....	21
3.6 Watermarked image	22
3.7 Extracted Watermark.....	23
4. Invisible Watermarking Using Discrete Wavelet Transform	
4.1 Some common wavelet types.....	24
4.2 Haar Wavelet.....	25
4.3 Signal having low frequencies throughout and high frequencies for a very short duration.....	26
4.4 Signal and wavelet function for different values of τ (tau) for $s=1$	28
4.5 Signal and wavelet function for different values of τ (tau) for $s=5$	29
4.6 Signal and wavelet function for different values of τ (tau) for $s=20$	29
4.7 Subband Coding.....	33
4.8 Signal and Image decomposition structure by dwt.....	36
4.9 Cover and Watermark image used.....	36
4.10 Decomposed Original Image into its various frequency components.....	37

4.11 Watermarked image.....	37
4.12 Extracted Watermark.....	38
5. Results and Observations	
5.1 DCT Watermarking.....	39
5.2 Extracted Watermark from Flipped Image.....	40
5.3 Extracted Watermark from Tampered Image.....	40
5.4 Extracted Watermark from Inverted Image.....	40
5.5 Watermarked Images obtained by varying values of 'g' as 2, 5 and 20.....	41
5.6 DWT Watermarking.....	41
5.7 Extracted Watermark from Flipped Image.....	42
5.8 Extracted Watermark from Inverted Image.....	42
5.9 Watermarked Images obtained by varying values of 'k' as 2, 5, 20,200.....	42

LIST OF TABLES USED

4. Invisible Watermarking Using Discrete Wavelet Transform	
4.1 Difference between a Wave and Wavelet.....	25

CHAPTER 1

INTRODUCTION

Information hiding (or data hiding) is a general term encircling a wide range of problems beyond the embedding messages in content. The term hiding can refer to either for information imperceptibility (watermarking) or information secrecy (Steganography). Watermarking and Steganography are two important sub disciplines of information hiding that are closely related to each other and may coincide but with different underlying properties, requirements and designs, thus result in different technical solutions [1].

Steganography is a term derived from the Greek words steganos, which means "covered," and graphia, which means "writing." It is the art of concealed communication. The existence of a message is secret. Examples include invisible ink which would glow over a flame used by both the British and Americans to communicate secretly during the American Revolution and hidden text using invisible ink to print small dots above or below letters and by changing the heights of letter-strokes in texts used by German spies in World Wars [2].

Watermarking which a term used back from paper watermarking, on the other hand has the additional concept of resilience against attempts to remove the hidden data. This is because the information hidden by watermarking systems is always associated to the digital object to be protected or its owner while steganographic systems just hide any information.

Robustness criteria are also different since Steganography mainly concerns with detection of hidden message while watermarking concerns potential removal by a pirate. Besides, Steganography typically relates to covert point-to-point communication while watermarking is usually one-to-many [3].

Digital watermarking is a technique that hides some additional information, which is called a 'watermark', into the 'cover data' by slightly modifying the data content. The term 'cover data', also known as host data, is used to describe the original media data, such as audio, image and video. After a watermark is embedded, the cover data becomes the 'watermarked data'. The process of inserting a watermark into the cover data is known as embedding, while the process of extracting or verifying the presence of a watermark is known as watermark detection or extraction.

Digital watermarking technology is becoming increasingly important due to the proliferation of digital images on the World Wide Web and in electronic commerce. There have been different types of watermarks proposed in the literature, designed for different applications.



Figure 1.1 Motivations behind Watermarking

The task for this project has been to investigate the field of image watermarking and develop suitable algorithms that embed a watermark in the image in least computation times and make it quite difficult for a hacker to tamper with it so that the authenticity of image can be maintained.

The major part of this project has been about the new algorithms proposed for digital watermarking that utilize the frequency domain analysis of the image to embed and extract the watermark. This method seems ideal, in that it promises to embed watermarks that cannot be detected by the eye, and being able to extract the watermarks from images exposed to severe alterations.

This project only deals with the digital watermarking of binary and grayscale images, though watermarking are also used for video, audio and other digital media. Digital watermarking seems to be a rising field of research. The research began in the early 1990's and slowly grew until the millennium where interest in this field seems to have exploded. This interest might be reflected in an actual demand for efficient tools needed by companies that publishes digital media on the internet. This is an important issue in, for example, legal applications, news reporting and medical archiving, where we want to be sure that the digital image in question truly reflects what the scene looked like at the time of capture. Another need of image authentication arises in, for example, electronic commerce where the seller transmits a digital image to the buyer over the network. In this case the buyer wants to be sure

that the received image is indeed genuine. Here we not only want to verify the integrity of an image, but we also want to check for original ownership.

Other applications include electronic advertising, real time video and audio delivery, digital repositories and libraries, and Web publishing. An important issue that arises in these applications is the protection of the rights of all participants. It has been recognized for quite some time that current copyright laws are inadequate for dealing with digital data. This has led to an interest towards developing new copy deterrence and protection mechanisms. One such effort that has been attracting increasing interest is based on digital watermarking techniques.

Thus, we realize that digital watermarking has become an active and important area of research, and development and commercialization of watermarking techniques is being deemed essential to help address some of the challenges faced by the rapid proliferation of digital content.

CHAPTER 2

REVIEW OF WATERMARKING

2.1 History of Watermarking

Although paper was invented in China over a thousand years ago, the Europeans only began to manufacture it in the 11th and 12th centuries, after Muslims had established the first paper mills in Spain. Soon after its invention, Chinese merchants and missionaries transmitted paper, and knowledge of papermaking, to neighboring lands such as Japan, Korea, and Central Asia. It was there that Muslims first encountered it in the 8th century. Islamic civilization spread knowledge of paper and papermaking to Iraq, Syria, Egypt, North Africa and finally, Spain. Most accounts of the history of paper focus either on its origins in China or its development in Europe, and simply disregard the centuries when knowledge of paper and papermaking spread throughout the Islamic lands. Some of this neglect is due to the difficulty of studying Islamic paper, since Islamic papers, unlike later European papers, did not have watermarks and were consequently very difficult to localize and date [4]. This explains why the oldest watermarked paper found in archives dates back to 1292, in Fabriano, Italy. The marks were made by adding thin wire patterns to the paper molds. The paper would be slightly thinner where the wire was and hence more transparent. At the end of 13th century about 40 paper mills were sharing the paper market in Fabriano and producing paper with different format, quality and price.

Competition was very high and it was difficult for any party to keep track of paper provenance and thus format and quality identification. The introduction of watermarks was the best method to eliminate any possibility of confusion.

More than 700 years ago, paper watermarks were used in Fabriano, Italy to indicate the paper brand and the mill that produced it. After their invention, watermarks quickly spread over Italy and then over Europe, and although originally used to indicate the paper brand or paper mill, they later served as indication for paper format, quality, and strength and were also used to date and authenticate paper. By the 18th century it began to be used as anti counterfeiting measures on money and other documents [5]. They are still widely used as security features in currency today.

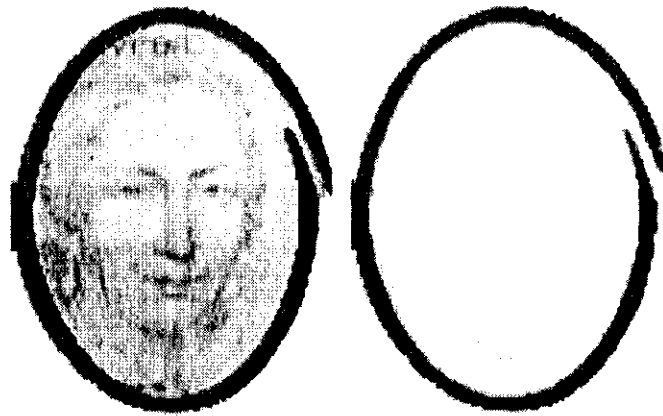


Figure 2.1 Watermarks in Currency Notes

The term watermark was introduced near the end of the 18th century. It was probably given because the marks resemble the effects of water on paper.

The digitization of today's world has expanded the watermarking concept to include digital approaches for use in authenticating ownership claims and protecting proprietary interests [1, 3].

The first example of a technology similar to digital watermarking is a patent filed in 1954 by Emil Hembrooke for identifying music works. In 1988, Komatsu and Tominaga appear to be the first to use the term "digital watermarking" [6].

Digital watermarking allows a person to hide copyrights on audio, video or images. This information usually includes the maker, the copyright itself and any other data the owner wants to include. Watermarking began as a way to keep money from being copyrighted and developed into ways on the World Wide Web to keep documents and other items safe from being reproduced or shared without credit.

The Internet is now rampant with illegal file sharing on all levels. While the original copyright holders are trying to track down and keep their work within the circle they want, it is impossible with the speed of the World Wide Web to keep track of everything on it. Digital watermarking is one such way to keep track of their work and make sure they are getting their dues for creating, distributing and selling it.

2.2 General Model of Watermarking

A generalized watermark model consists of watermark encoding and detection processes.

The inputs to the embedding process are the watermark, the cover object and a secret key. The key is used to enforce security and to protect the watermark. The output of the watermarking scheme is the watermarked data. The channel for the watermarked data could be a lossy, noisy, unreliable channel. Thus the received data may be different from the original watermarked data. The inputs for extraction are the received watermarked data and the key corresponding to the embedding key. The output of the watermark recovery process is the recovered watermark [7].

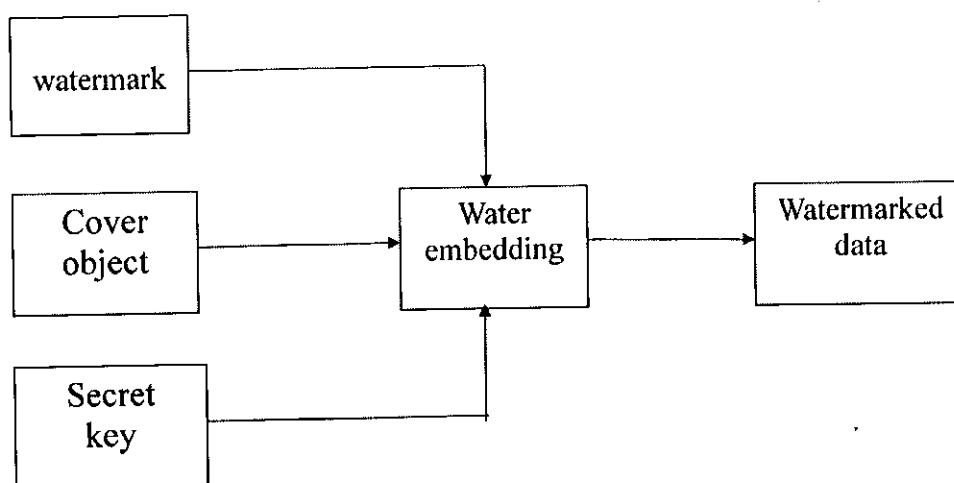


Figure 2.2 Watermark Embedding Model [8]

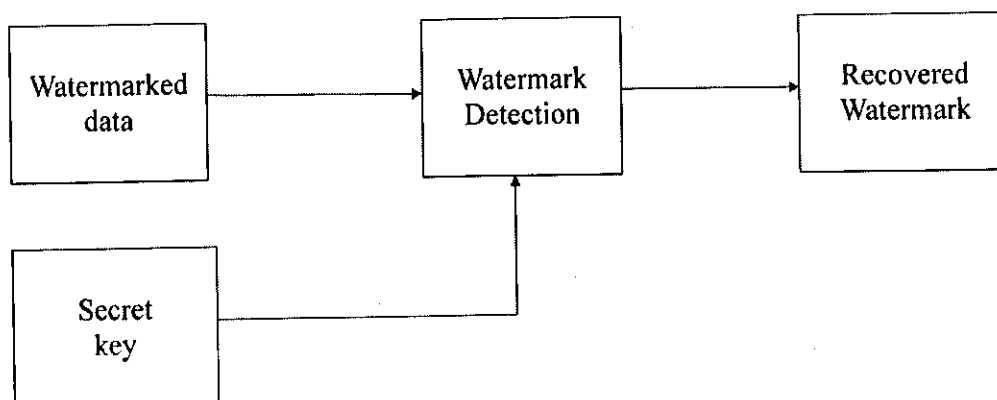


Figure 2.3 Watermark Recovery Model [8]

2.3 Types of Digital Watermarking

Digital Watermarking techniques can be divided into many categories based on certain parameters [9].

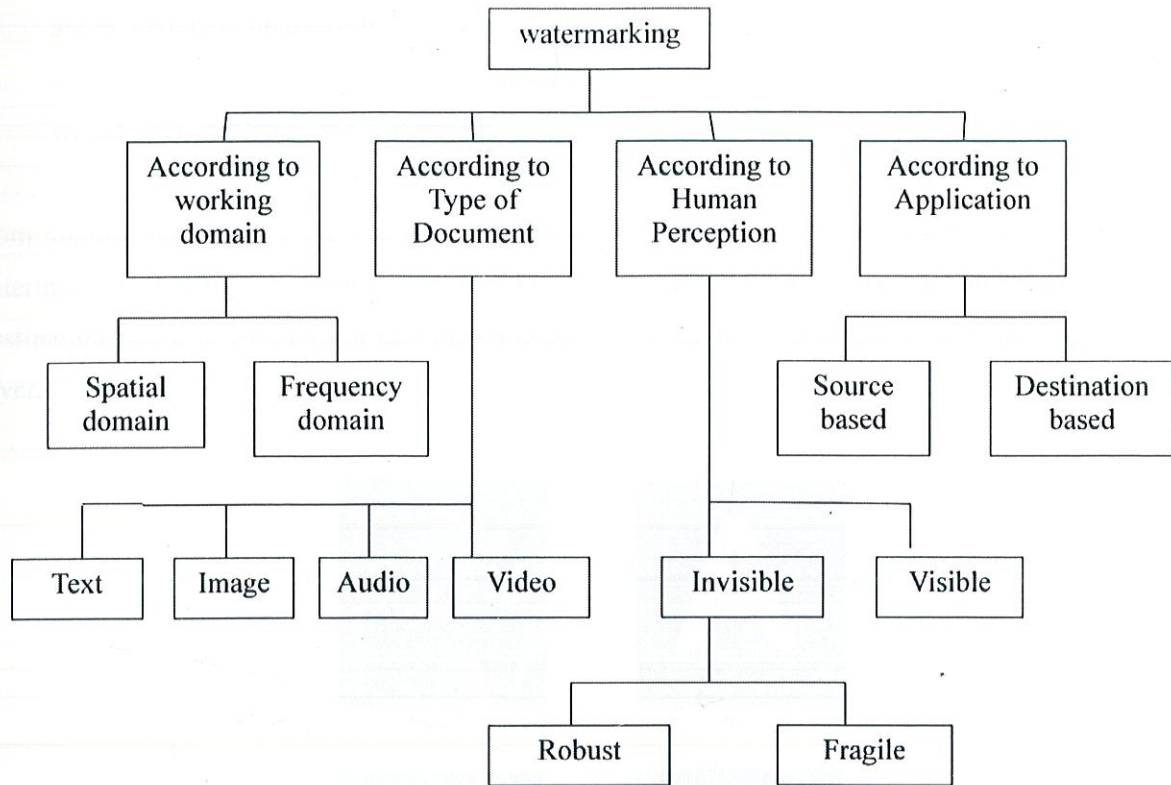


Figure 2.4 Types of Watermarking Techniques

In the case of images, watermarking techniques are commonly distinguished based on two working domains:

Spatial domain – pixels of one or two randomly selected subsets of an image are modified based on perceptual analysis of the original image.

Transform domain – values of certain frequencies are altered from their original.

According to the type of document to be watermarked: Text Watermarking, Image Watermarking, Audio Watermarking and Video Watermarking.

Based on human perception, they can be divided into three categories as follows:

Visible watermark is where the secondary translucent overlaid into the primary content and appears visible on a careful inspection.

Invisible-Robust watermark is embedded in such a way that alterations made to the pixel value are perceptually unnoticed.

Invisible-Fragile watermark is embedded in such a way that any manipulation of the content would alter or destroy the watermark.

From application point of view, digital watermarks could also be: Source based is where a unique watermark identifying the owner is introduced to all the copies of a particular content being distributed. Destination based is where each distributed copy gets a unique watermark identifying the particular buyer.

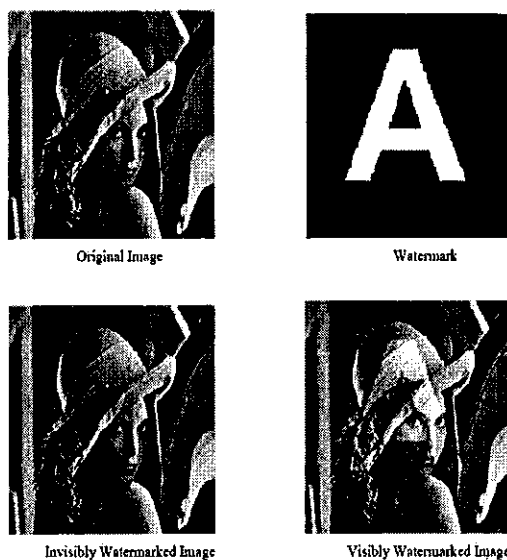


Figure 2.5 Visible and Invisible Watermarks

Classifications can be on countless parameters. Here are some more:

Inserting Watermark Type: watermark can be inserted in the form of noise tagged information, or image.

Public & Private Watermarking: In public watermarking, users of the content are authorized to detect the watermark while in private watermarking the users are not authorized to detect the watermark.

Asymmetric & Symmetric Watermarking: Asymmetric watermarking (also called asymmetric key watermarking) is a technique where different keys are used for embedding and detecting the watermark. In symmetric watermarking (or symmetric key watermarking) the same keys are used for embedding and detecting watermarks.

Steganographic & Non-Steganographic watermarking: Steganographic watermarking is the technique where content users are unaware of the presence of a watermark. In non steganographic watermarking, the users are aware of the presence of a watermark. Steganographic watermarking is used in fingerprinting applications while non steganographic watermarking techniques can be used to deter piracy.

2.4 General Framework for Digital Watermarking

Digital watermarking is similar to watermarking physical objects except that the watermarking technique is used for digital content instead of physical objects. In digital watermarking a low-energy signal is imperceptibly embedded in another signal. The low energy signal is called watermark and it depicts some metadata, like security or rights information about the main signal. The main signal in which the watermark is embedded is referred to as cover signal since it covers the watermark. The cover signal is generally a still image, audio clip, video sequence or a text document in digital format. The digital watermarking system essentially consists of a watermark embedder and a watermark detector as shown in Fig. 1.6.

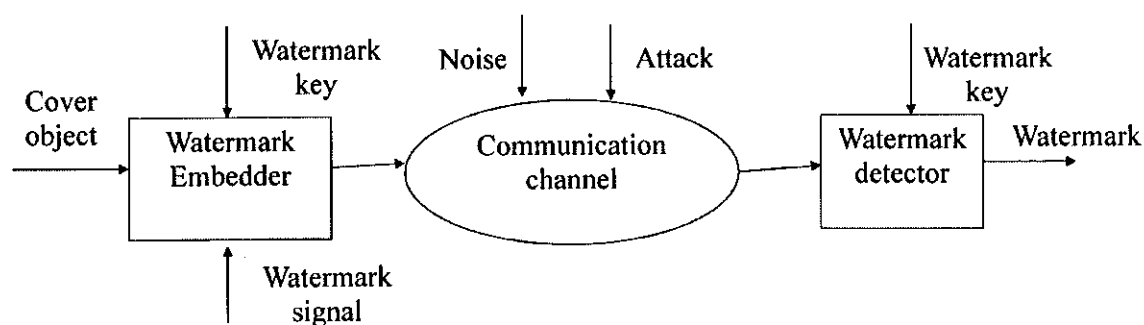


Figure 2.6 Digital Watermarking System

The watermark embedder inserts a watermark onto the cover signal and the watermark detector detects the presence of watermark signal.

Note that an entity called watermark key is used during the process of embedding and detecting watermarks. The watermark key has a one-to-one correspondence with watermark signal (i.e., a unique watermark key exists for every watermark signal). The watermark key is private and known to only

authorized parties and it ensures that only authorized parties can detect the watermark. Further, note that the communication channel can be noisy and hostile (i.e., prone to security attacks) and hence the digital watermarking techniques should be resilient to both noise and security attacks [10].

2.5 Watermarking vs. Steganography

Watermarking is not a new technique. It is descendent of a technique known as Steganography. Steganography is a technique for concealed communication. Here the existence of the message that is communicated is a secret and its presence is known only by parties involved in the communication.

In Steganography a secret message is hidden within another unrelated message and then communicated to the other party. As opposed to this in Watermarking again one message is hidden in another, but two messages are related to each other in some way.

Steganographic methods are in general not robust, i.e., the hidden information cannot be recovered after data manipulation. Watermarking, as opposed to Steganography, has the additional notion of robustness against attacks. Even if the existence of the hidden information is known it is difficult—ideally impossible—for an attacker to destroy the embedded watermark, even if the algorithmic principle of the watermarking method is public [6, 11, 12].

2.6 Watermarking vs. Cryptography

Watermarking is a totally different technique from cryptography. Cryptography only provides security by encryption and decryption. However, encryption cannot help the seller monitor how a legitimate customer handles the content after decryption. So there is no protection after decryption. As shown in the figure 1.4 in this case Customer can make illegal copies of the digital content. Unlike cryptography, watermarks can protect content even after they are decoded [6, 11, 12].

Other difference is cryptography is only about protecting the content of the messages. Because watermarks are inseparable from the cover in which they are embedded so in addition to protecting content they provide many other applications also, like copyright protection, copy protection, ID card security etc. Also the concept of breaking the system is different for cryptosystems and watermarking systems. A cryptographic system is broken when the attacker can read the secrete message. But Breaking of a watermarking system has two stages:

1. The attacker can detect that watermarking has been used.
2. The attacker is able to read, modify or remove the hidden message.

2.7 Watermarking vs. Digital Signatures

In watermarking we embed metadata into the multimedia content directly in such a way that it needs not additional bandwidth. Historically, integrity and authenticity of digital data has been guaranteed through the use of digital signatures. In that we use header part of the document for signature embedding. So additional bandwidth is required, which increases the overhead [12].

2.8 Properties of Watermark

Perpetual Transparency: Use characteristics of HVS to ensure that the watermark is not visible under typical viewing conditions. Basically, it means that a watermarked image should not seem any different from the original one; i.e. one should not notice any degradation in the perceived quality. Other types of watermarks are meant to be visible but in most application they are not and this is why we treat transparency as a basic requirement of digital watermarking.

Robustness: It is a measure of the ability of the embedding algorithm to introduce the watermark in such a way that it is retained in the image despite several stages of image processing. The image may be filtered (high-pass or low-pass or median) rotated, translated, cropped, scaled etc. as part of image processing. A good watermarking algorithm embeds the watermark in the spatial or frequency regions of the image, which would be least affected by such processing. Good correlation is possible between the recovered watermark and the original watermark in spite of noise errors introduced in it by processing. There is a special class of watermarks called "fragile" watermarks, which are intentionally made non-robust. These are intended for authentication of original material rather than tracing it back to a source after being processed. A fragile watermark is lost with the slightest of image processing since such processing alters the image in a manner not intended for by the original owner of the material. "Semi-fragile" watermarks are able to survive standard unintentional image processing such as image compression for storage unlike fragile watermarks; robust watermarks are resilient to intentional or un-intentional attacks or signal processing operations.

Computational Simplicity: Consideration for computational complexity is important while designing robust watermarks. If a watermarking algorithm is robust but computationally very intensive during encoding or decoding then its usefulness in real-life may be limited.

Payload of Watermark: That represents the amount of information that can actually be stored in a particular data stream. This requirement is highly dependent on the host medium, the intended application as well the quality aimed for.

Security: Watermarking security can be interpreted as encryption security leading to the principle that it must lie mainly in the choice of the embedded key. How easy or difficult is it for someone who does not know only the secret key used in the watermarking process to modify an image without modifying the watermark; or by inserting a new but valid watermark.

2.9 Watermarking Techniques

Spatial Domain: Spatial domain watermarking is applied to graphic images and text. Spatial domain watermarking slightly modifies the pixels of one or two randomly selected subsets of an image. Modifications might include flipping the low-order bit of each pixel. However, this technique is not reliable when subjected to normal media operations such as filtering or lossy compression.

LSB Substitution: There are many variants of this technique. It essentially involves embedding the watermark by replacing the least significant bit of the image data with a bit of the watermark data. This technique may involve other approaches such as converting the watermark sequence into a PN sequence which is then embedded into the image or repeated embedding of the watermark when the watermark is much smaller than the host image.

Correlation Based Approach: The watermark is converted into a long PN sequence, which is then weighted & added to the host image with some gain factor k . For detection, the watermarked image is correlated with the watermark image.

Frequency Domain

DCT Based Approach: The Discrete Cosine Transform is a real domain transforms which represents the entire image as coefficients of different frequencies of cosines (which are the basis vectors for this transform). The DCT of the image is calculated by taking 8×8 blocks of the image, which are then transformed individually. The 2D DCT of an image gives the result matrix such that top left corner represents lowest frequency coefficient while the bottom right corner is the highest frequency. DCT also forms the basis of the JPEG image compression algorithm, which is one of the most widely used image data storage formats. The DCT approaches are able to withstand some forms of attack very well such as Low-pass/High-pass filtering/median filtering etc.

Wavelet Based Approach: The techniques involve the embedding of information in the LH blocks of the wavelet transform of the image. Observers due to characteristics of the Human Visual

System do not notice changes to these regions. These are also utilized for fragile watermarking which is a significant tool for content authentication.

2.10 Applications

Content Authentication: Multimedia editing software makes it easy to alter digital content. Digital signature essentially represents some kind of summary of the content. If any part of the content is modified, its summary, the signature, will change making it possible to detect that some kind of tampering has taken place. One example of digital signature technology being used for image authentication is the trustworthy digital camera described in Digital signature information needs to be somehow associated and transmitted with a digital content it was created from. Watermarks can obviously be used to achieve that association by embedding signature directly into the content. Since watermarks used in the content authentication applications have to be designed to become invalid if even slight modifications of digital content take place, they are called fragile watermarks. Fragile watermarks, therefore, can be used to confirm authenticity of a digital content. They can also be used in applications where it is important to figure out how digital content was modified or which portion of it has been tampered with. For digital images, this can be done by dividing an image into a number of blocks and creating and embedding a fragile watermark into each and every block.

Broadcast Monitoring: Many valuable products are regularly broadcast over the television network: news, movies, sports events, advertisements, etc. Broadcast time is very expensive, and advertisers may pay hundreds of thousands of dollars for each run of their short commercial that appears during commercial breaks of important movies, series or sporting events. The ability to bill accurately in this environment is very important. It is important to advertisers who would like to make sure that they will pay only for the commercials which were actually broadcast. And, it is important for the performers in those commercials who would like to collect accurate royalty payments from advertisers. Broadcast monitoring is usually used to collect information about the content being broadcast, and this information is then used as the bases for billing as well as other purposes. A solution for active monitoring is based on watermarking. The watermark containing broadcast identification information gets embedded into the content itself, and the resulting broadcast monitoring solution becomes compatible with broadcast equipment for both digital and analog transmission.

Copyright Communication: Content often circulates anonymously, without identification of the owner, or an easy means to contact the owner/distributor to obtain rights for use. Digital watermarks enable copyright holders to communicate their ownership and offer links to copyright and purchase information. This helps to protect their content from unauthorized use, enabling infringement detection and promoting licensing. The watermark payload carries a content identifier that can be linked to information about the content owner and copyright information. Copyright communication is applicable to images, audio and video.

Authentication and Integrity: Digital watermarks can verify that content is genuine and from an authorized source, as well as verify the content has not been altered or falsified. For example, digital watermarks can verify authenticity and identify counterfeiting as an additional layer of security for encrypted content or for content in the open. The presence of the digital watermark and/or the continuity of the watermark can help ensure that the content has not been altered. No database is required as the processing happens on the local machine. Authentication is applicable to images, print documents, audio, and video.

Fraud and Tamper Detection: When multimedia content is used for legal purposes, medical applications, news reporting, and commercial transactions, it is important to ensure that the content was originated from a specific source and that it had not been changed, manipulated or falsified. This can be achieved by embedding a watermark in the data. When the photo is checked, the watermark is extracted using a unique key associated with the source, and the integrity of the data is verified through the integrity of the extracted watermark. The watermark can also include information from the original image that can aid in undoing any modification and recovering the original. Clearly a watermark used for authentication purposes should not affect the quality of an image and should be resistant to forgeries. Robustness is not critical as removal of the watermark renders the content inauthentic and hence of no value.

ID card Security: Information in a passport or ID (e.g., passport number, person's name, etc.) can also be included in the person's photo that appears on the ID. By extracting the embedded information and comparing it to the written text, the ID card can be verified. The inclusion of the watermark provides an additional level of security in this application. For example, if the ID card is stolen and the picture is replaced by a forged copy, the failure in extracting the watermark will invalidate the ID card.

Data Hiding: The transmission of private data is probably one of the earliest applications of watermarking. As one would probably have already understood, it consists of implanting a strategic message into an innocuous one in a way that would prevent any unauthorized person to detect it.

Medical Safety: Embedding the date and patient's name in medical images could increase the confidentiality of medical information as well as the security.

2.11 Limitations

Much of the work on trying to model and understand some of the fundamental properties and limitations of watermarking algorithms is based on drawing parallels to communications systems. Many popular watermark embedding algorithms are variations of the idea of spread-spectrum techniques for secure communication systems, where information bearing narrow band signal is converted into a wideband signal prior to transmission, by modulating the information waveform with a wideband noise like waveform. As a result of the bandwidth expansion, within any narrow spectral band, the total amount of energy from the information signal is small. By appropriately combining all the weak narrowband signals at the demodulator, the original information signal is recovered.

There has been some interesting work in trying to model and understand some of the fundamental properties and limitations of watermarking algorithms. An information theoretic analysis of watermarking is presented where an elegant framework is proposed for the hiding capacity problem (watermark payload). The framework shows the tradeoff between achievable information hiding rates and allowed distortions for the information hider (watermark embedder) and the attacker (possible distortions to remove or alter the watermark). The attack is a Wiener estimate of the actual watermark signal which leads to an effective watermark design which attempts to match the power spectrum of the watermark as a scaled version of the power spectrum of the original host signal. Intuitively, this says that the watermark should look like the original signal. This also supports the use of visual models for watermark embedding where the watermark signal very closely matches the general characteristics of the host signal. Unlike previous work where quantization is usually modeled as additive noise which is adequate for fine quantization or high data rates, the authors look at modeling the watermarking and quantization effect as dithered quantization where the dither is represented by the watermark.

CHAPTER 3

INVISIBLE WATERMARKING USING DISCRETE COSINE TRANSFORM

3.1 Overview

Transform coding constitutes an integral component of contemporary image/video processing applications. Transform coding relies on the premise that pixels in an image exhibit a certain level of correlation with their neighboring pixels. Similarly in a video transmission system, adjacent pixels in consecutive frames show very high correlation. Consequently, these correlations can be exploited to predict the value of a pixel from its respective neighbors. A transformation is, therefore, defined to map this spatial (correlated) data into transformed (uncorrelated) coefficients. Clearly, the transformation should utilize the fact that the information content of an individual pixel is relatively small i.e., to a large extent visual contribution of a pixel can be predicted using its neighbors.

Quantization is a procedure of constraining something from continuous set of values (such as real numbers) to a discrete set (such as integers). The quantizer sub-block utilizes the fact that the human eye is unable to perceive some visual information in an image. Such information is deemed redundant and can be discarded without introducing noticeable visual artifacts. Such redundancy is referred to as psycho-visual redundancy. This idea can be extended to low bit rate receivers which, due to their stringent bandwidth requirements, might sacrifice visual quality in order to achieve bandwidth efficiency.

If the input sequence has more than N sample points then it can be divided into sub-sequences of length N and DCT can be applied to these chunks independently.

A typical image/video transmission system is outlined in figure below. The objective of the source encoder is to exploit the redundancies in image data to provide compression. In other words, the source encoder reduces the entropy, which in our case means decrease in the average number of bits required to represent the image. On the contrary, the channel encoder adds redundancy to the output of the source encoder in order to enhance the reliability of the transmission.

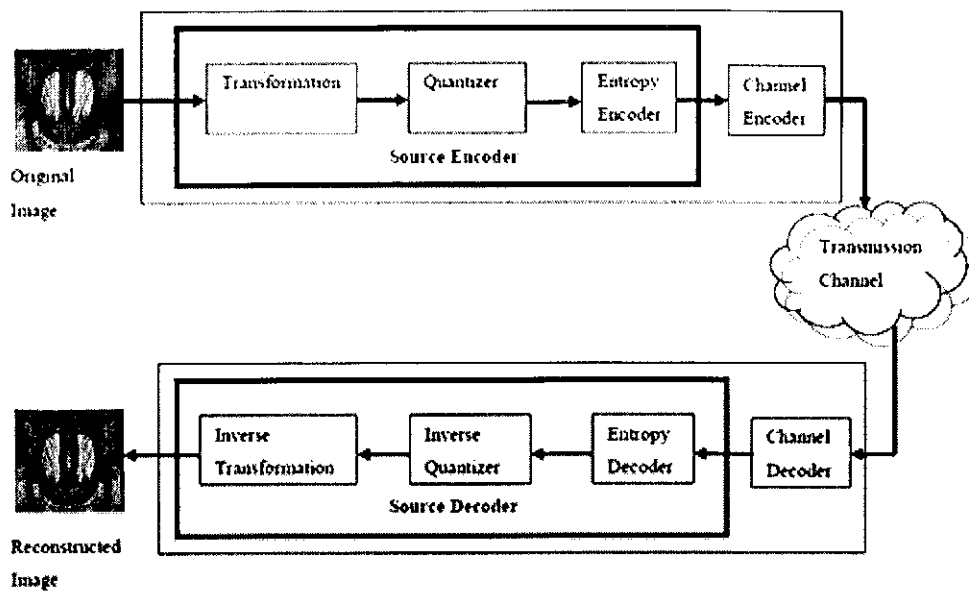


Figure 3.1 Components of Typical Data Transmission System

A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. It is similar to the discrete Fourier transform (DFT), but using only real numbers. These are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample [16].

DCT attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency.

There are eight standard DCT variants, of which four are common. The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT"; its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT".

Formally, the discrete cosine transform is a linear, invertible function $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ (where \mathbb{R} denotes the set of real numbers), or equivalently an invertible $N \times N$ square matrix. The N real numbers x_0, \dots, x_{N-1} are transformed into the N real numbers X_0, \dots, X_{N-1} according to the following formula:

DCT-II

$$X_k = \sum_{n=0}^{N-1} x_n \cos[\pi/N(n+1/2)k] \quad k=0,1,\dots,N-1$$

The DCT-II is probably the most commonly used form, and is often simply referred to as "the DCT". A two-dimensional DCT-II of an image or a matrix is simply the one-dimensional DCT-II, from above, performed along the rows and then along the columns (or vice versa). That is, the 2d DCT-II is given by the formula:

$$X_{k_1, k_2} = \sum_{n_1=0}^{N_1-1} \left(\sum_{n_2=0}^{N_2-1} X_{n_1, n_2} \cos\left[\frac{\pi}{N_2}(n_2+1/2)k_2\right] \right) \cos\left[\frac{\pi}{N_1}(n_1+1/2)k_1\right]$$

Advantages of DCT

1. Primitive and easy to implement.

Disadvantages of DCT

1. Only spatial correlation of the pixels inside the single 2-D block is considered and correlation from the pixels of the neighboring blocks is neglected.
2. Impossible to completely decorrelate the blocks at their boundaries using DCT.
3. Undesirable blocking artifacts affect the reconstructed images or video frames (high compression ratios or very low bit rates).
4. Scaling as add-on additional effort.
5. DCT function is fixed and cannot be adapted to source data.

3.2 Technique Implemented

Discrete-Cosine-Transform or DCT is a popular transform domain watermarking technique. The DCT allows an image to be broken up into different frequency bands namely the high, middle and low frequency bands thus making it easier to choose the band in which the watermark is to be inserted. The literature survey reveals that mostly the middle frequency bands are chosen because embedding the watermark in a middle frequency band does not scatter the watermark information to most visual important parts of the image i.e. the low frequencies and also it do not overexpose them to removal through compression and noise attacks where high frequency components are targeted. Numerous watermarking techniques based on DCT are proposed. Although some of the watermarking techniques embed the watermark in the DC component i.e. the low frequency component most technique utilizes the comparison of middle band DCT coefficients to embed a single bit of watermark information into a DCT block. The middle-band frequencies (FM) of an 8*8 DCT block can be shown below in figure:

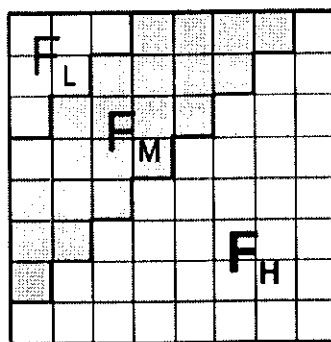


Figure 3.2 Definition of DCT Regions

DCT of the image is taken in a block dimension of 8×8 resulting in DCT blocks of dimension 8×8 . A DCT block consists of three frequency bands. FL is used to denote the lowest frequency components of the block, while FH is used to denote the higher frequency components. FM is the middle frequency band and is chosen for embedding copyright information. This provides additional resistance to lossy compression techniques which targets the high frequency components, while avoiding significant modification of the cover image. From the frequency band FM two locations $M_i(u_1, v_1)$ and $M_i(u_2, v_2)$ are chosen as the region for comparison.

The two locations which have identical quantization values are chosen for embedding one watermark bit of information. From the table the coefficients at (5,2) and (4,3) or (1,2) and (3,0) would make suitable candidates for comparison, as their quantization values are equal. The DCT block will swap for "0" if $M_i(u_1, v_1) < M_i(u_2, v_2)$, otherwise it will swap for "1", where 0 and 1 are the elements in the watermark array. The coefficients are then swapped if the relative size of each coefficient does not agree with the bit that is to be encoded. This shows that the number of watermark bits that can be embedded is directly dependent on the number of pairs of locations in quantization table for which the value in the table is similar. The swapping of such coefficients should not alter the watermarked image significantly, as it is generally believed that DCT coefficients of middle frequencies have similar magnitudes. The robustness of the watermark can be improved by introducing a watermark "strength" constant k , such that $M_i(u_1, v_1) - M_i(u_2, v_2) > k$. Increasing k thus reduces the chance of detection errors at the expense of additional image degradation [17].

3.3 Algorithm

3.3.1 Algorithm Flowchart

Embedding:

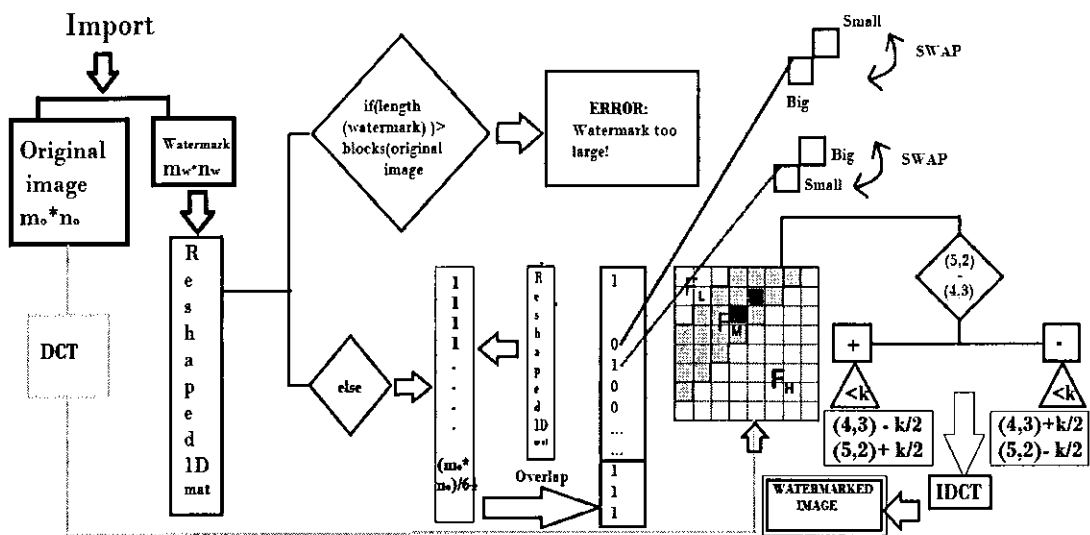


Figure 3.3 Embedding Algorithm

Extraction:

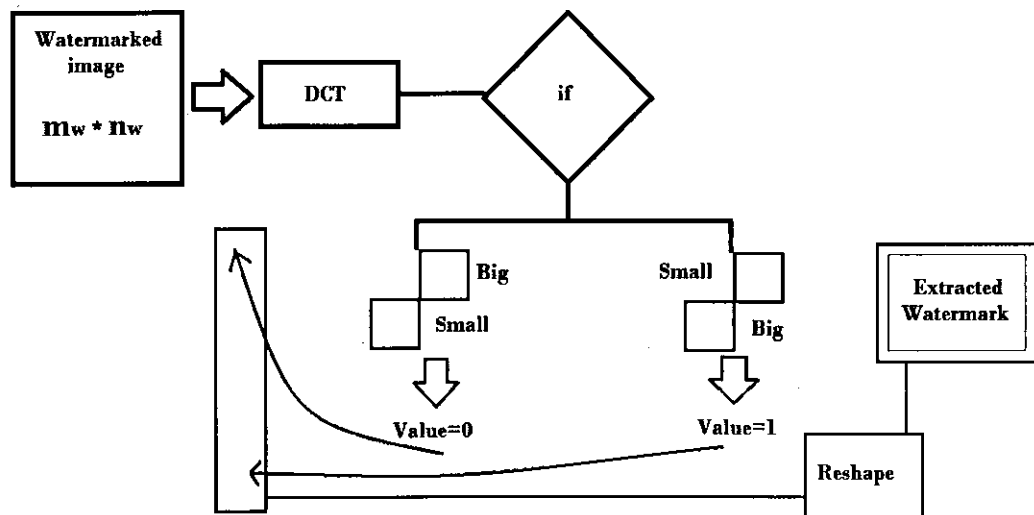


Figure 3.4 Extraction Algorithm

3.3.2 Stepwise Implementation

Invisible watermarking is a process of applying a watermark or embedding/modifying original pixel values in such a way that changes in the resultant are not perceptible.

The explanation of Invisible Watermarking Embedding process using DCT we have used has been shown below.

(Note: We are working on gray scale and binary images)

1. We set value for a coefficient difference depending on how strongly we need our watermark to be embedded, but at the cost of greater image quality degradation with higher values of it.
2. We set the size for each block in the original image.
3. We have taken 'lena.bmp' as our image to be watermarked and 'bb.bmp' as our image to be used as a watermark.



Figure 3.5 Lena & Watermark

4. After storing the sizes of both these images into some variables in MATLAB, we reshape the watermark image into a single row or column matrix.
5. We calculate the total number of blocks in the original image by dividing its size by size of each block. This gives us the maximum length of watermark we can insert.
6. Now, we check whether our watermark can be inserted into the original image or not.
7. If the condition holds true i.e. watermark can be inserted, we pad ones into a matrix having size equal to the maximum length of the watermark.
8. Now we pad the watermark matrix into the matrix having all ones.
9. Running a loop till the length of the newly padded matrix, we take the blockwise DCT of the original image.

10. As soon as we encounter a black pixel('0') in our padded matrix, we swap the (5,2) & (4,3) coefficients of the first DCT block depending on which one is greater and vice-versa we do when we encounter a white pixel('1'). We have used these blocks as they lie in the mid-frequency band of the image, and tampering within this band would not degrade the image quality as it only gives the overview of the image.

11. Now we need to adjust the values of (5,2) & (4,3) such that their coefficient difference is \geq the minimum coefficient difference.

12. Similarly we traverse all the blocks and perform the same operation as above.

13. We now take the Inverse DCT of the DCT blocks we get and store it in a new matrix having similar dimensions as the original image. This gives our watermarked image.



Figure 3.6 Watermarked Image

Invisible watermarks are hence more valuable in applications where the visible watermarks are deemed to be inappropriate, such as sale of high quality images.

Now if a dispute arises over the ownership of the image, or we wish to see if an image has been tampered with, we can perform the Recovery of Watermark from the Watermarked Image.

RECOVERY:

1. We set the size of each block to be used for DCT calculations.
2. We store the size of the watermarked image in some variable. This would help in reshaping the watermark image matrix.
3. Also we get the maximum size of the watermark that we could embed in the original image.
4. Running a loop till this maximum length, we calculate the blockwise DCT of the watermarked image.
5. Now we compare the (5,2) & (4,3) coefficients of the first DCT block. If $(5,2) > (4,3)$ we pad zero in the corresponding pixel value of the watermark image matrix else we pad a one.

6. Similarly we traverse all the blocks and perform the same operation.
7. We now reshape this 1D watermark image matrix into a 2D matrix.
8. This obtained 2D matrix is our recovered watermark image.



Figure 3.7 Extracted Watermark

CHAPTER 4

INVISIBLE WATERMARKING USING DISCRETE WAVELET TRANSFORM

4.1 Overview

4.1.1 Wavelets, Wavelet Transform and Multiresolution Analysis

A wavelet is a kind of mathematical function used to divide a given function or continuous-time signal into different components and study each component with a resolution matched to its scale. The term wavelet means a small wave. The smallness refers to the condition that this (window) function is of finite length (compactly supported). The wave refers to the condition that this function is oscillatory. The term mother implies that the functions with different region of support that are used in the transformation process are derived from one main function, or the mother wavelet. In other words, the mother wavelet is a prototype for generating the other window functions. The following is the formula for the mother wavelet [18].

$$\Psi^{\tau,s}(t) = 1/\sqrt{s} \psi((t-\tau)/s)$$

Here τ & s are the translation and scaling factors.

Some common wavelets used are: Haar, Meyer and Morlet.

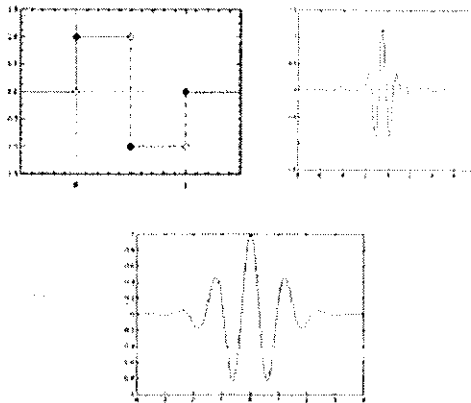
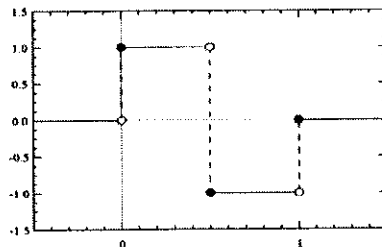


Figure 4.1 Some Common Wavelet Types

	Wave	Wavelet
Definition	A never ending repetitive signal	A small confined signal confined within a finite region
Energy	Infinite because Signal never ends	Finite and concentrated around a point
Statistical properties	Time invariant i.e Stationary signal	Time variant i.e. non-stationary signal
Associated analytical properties	Fourier transform	wavelet transform
Examples	Cosine wave	Haar, debauchies, Mexican hat, etc.

Table 4.1 Difference between a Wave and Wavelet

In our project, we have used Haar Wavelets. The Haar wavelet is also the simplest possible wavelet. The technical disadvantage of the Haar wavelet is that it is not continuous and therefore not differentiable [18].



$$\psi(t) = \begin{cases} 1 & 0 \leq t < 1/2, \\ -1 & 1/2 \leq t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Figure 4.2 Haar Wavelet [19]

Wavelet Transform

The wavelet transform is the representation of a function by wavelets. The wavelets are scaled and translated copies of a finite-length (known as "daughter wavelets") or fast-decaying oscillating waveform (known as "mother wavelet").

Wavelet transforms are classified into discrete wavelet transforms (DWTs) and continuous wavelet transforms (CWTs). Note that both DWT and CWT are of continuous-time (analog) transforms. They can be used to represent continuous-time (analog) signals. CWTs operate over every possible scale and translation whereas DWTs use a specific subset of scale and translation values or representation grid [18].

Multiresolution Analysis

MRA is designed to give good time resolution and poor frequency resolution at high frequencies and good frequency resolution and poor time resolution at low frequencies. This approach makes sense especially when the signal at hand has high frequency components for short durations and low frequency components for long durations. Fortunately, the signals that are encountered in practical applications are often of this type.

For example, the following shows a signal of this type. It has a relatively low frequency component throughout the entire signal and relatively high frequency components for a short duration somewhere around the middle [18].

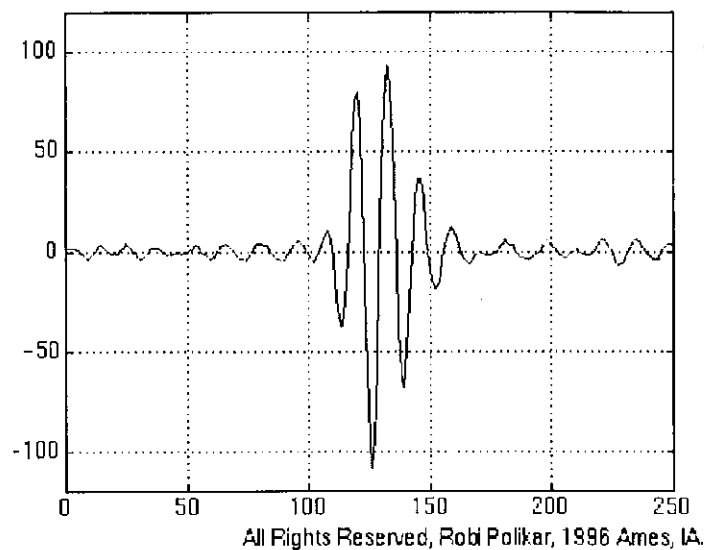


Figure 4.3 Signal having low frequencies throughout and high frequencies for a very short duration

4.1.2 Continuous Wavelet Transform

The continuous wavelet transform is defined as follows:

$$\text{CWT}_x^\psi(\tau, s) = \psi_x^\psi(\tau, s) = 1/\sqrt{s} \int x(t) \psi^*((t-\tau)/s) dt$$

As seen above, the transformed signal is a function of two variables, tau and s, the translation and scale parameters, respectively. Psi (t) is the transforming function, and it is called the mother wavelet.

The term translation is related to the location of the window, as the window is shifted through the signal. This term corresponds to time information in the transform domain.

The parameter scale in the wavelet analysis is similar to the scale used in maps. As in the case of maps, high scales correspond to a non-detailed global view (of the signal), and low scales correspond to a detailed view. Similarly, in terms of frequency, low frequencies (high scales) correspond to a global information of a signal (that usually spans the entire signal), whereas high frequencies (low scales) correspond to a detailed information of a hidden pattern in the signal (that usually lasts a relatively short time).

Fortunately in practical applications, low scales (high frequencies) do not last for the entire duration of the signal, unlike those shown in the figure, but they usually appear from time to time as short bursts, or spikes. High scales (low frequencies) usually last for the entire duration of the signal.

Scaling, as a mathematical operation, either dilates or compresses a signal. Larger scales correspond to dilated (or stretched out) signals and small scales correspond to compressed signals. In terms of mathematical functions, if f(t) is a given function f(st) corresponds to a contracted (compressed) version of f(t) if s > 1 and to an expanded (dilated) version of f(t) if s < 1. However, in the definition of the wavelet transform, the scaling term is used in the denominator, and therefore, the opposite of the above statements holds, i.e., scales s > 1 dilates the signals whereas scales s < 1, compresses the signal.

CWT Computation

Let x(t) is the signal to be analyzed. The mother wavelet is chosen to serve as a prototype for all windows in the process. All the windows that are used are the dilated (or compressed) and shifted versions of the mother wavelet. There are a number of functions that are used for this purpose.

The procedure will be started from scale s=1 and will continue for the increasing values of s, i.e., the analysis will start from high frequencies and proceed towards low frequencies. This first value

of s will correspond to the most compressed wavelet. As the value of s is increased, the wavelet will dilate.

The wavelet is placed at the beginning of the signal at the point which corresponds to time $=0$. The wavelet function at scale "1" is multiplied by the signal and then integrated over all times. The result of the integration is then multiplied by the constant number $1/\sqrt{s}$. This multiplication is for energy normalization purposes so that the transformed signal will have the same energy at every scale. The final result is the value of the transformation, i.e., the value of the continuous wavelet transform at time zero and scale $s=1$. In other words, it is the value that corresponds to the point $\tau=0, s=1$ in the time-scale plane.

The wavelet at scale $s=1$ is then shifted towards the right by τ amount to the location $t=\tau$, and the above equation is computed to get the transform value at $t=\tau, s=1$ in the time-frequency plane.

This procedure is repeated until the wavelet reaches the end of the signal. One row of points on the time-scale plane for the scale $s=1$ is now completed. Then, s is increased by a small value. Note that, this is a continuous transform, and therefore, both τ and s must be incremented continuously. However, if this transform needs to be computed by a computer, then both parameters are increased by a sufficiently small step size. This corresponds to sampling the time-scale plane.

The above procedure is repeated for every value of s . Every computation for a given value of s fills the corresponding single row of the time-scale plane. When the process is completed for all desired values of s , the CWT of the signal has been calculated.

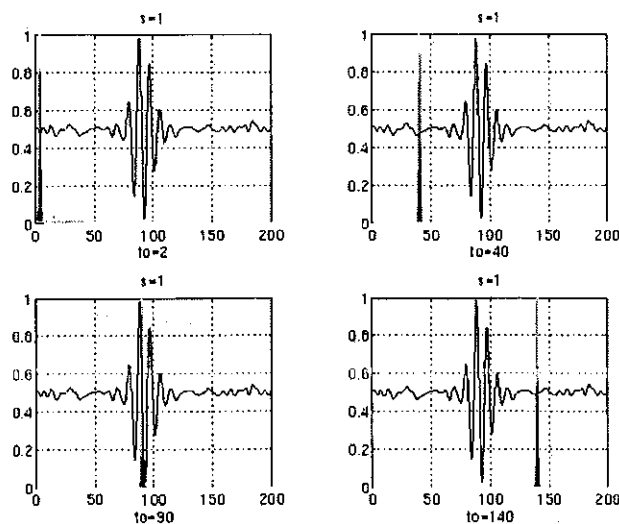


Figure 4.4 Signal and wavelet function for different values of τ (tau) for $s=1$

If the signal has a spectral component that corresponds to the current value of s (which is 1 in this case), the product of the wavelet with the signal at the location where this spectral component exists gives a relatively large value. If the spectral component that corresponds to the current value of s is not present in the signal, the product value will be relatively small, or zero.

The signal in above figure has spectral components comparable to the window's width at $s=1$ around $t=100$ ms. The continuous wavelet transform of the signal in above figure will yield large values for low scales around time 100 ms, and small values elsewhere. For high scales, on the other hand, the continuous wavelet transform will give large values for almost the entire duration of the signal, since low frequencies exist at all times [18].

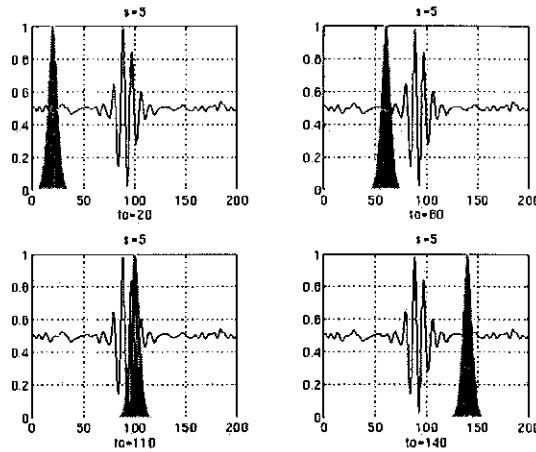


Figure 4.5 Signal and wavelet function for different values of τ (tau) for $s=5$

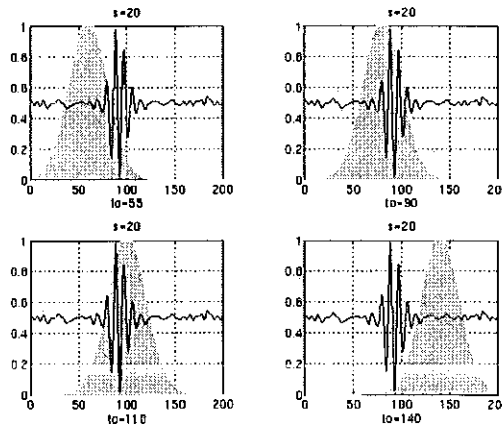


Figure 4.6 Signal and wavelet function for different values of τ (tau) for $s=20$

As the window width increases, the transform starts picking up the lower frequency components. As a result, for every scale and for every time (interval), one point of the time-scale plane is computed. The computations at one scale construct the rows of the time-scale plane, and the computations at different scales construct the columns of the time-scale plane.

Note that the axes are translation and scale, not time and frequency. However, translation is strictly related to time, since it indicates where the mother wavelet is located. The translation of the mother wavelet can be thought of as the time elapsed since $t=0$.

Lower scales (higher frequencies) have better scale resolution (narrower in scale, which means that it is less ambiguous what the exact value of the scale) which correspond to poorer frequency resolution. Similarly, higher scales have scale frequency resolution (wider support in scale, which means it is more ambiguous what the exact value of the scale is), which correspond to better frequency resolution of lower frequencies.

4.1.3 Discrete Wavelet Transform

In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. The Wavelet Series is just a sampled version of CWT and its computation may consume significant amount of time and resources, depending on the resolution required. The DWT, which is based on sub-band coding is found to yield a fast computation of Wavelet Transform. It is easy to implement and reduces the computation time and resources required. In DWT, a time-scale representation of the digital signal is obtained using digital filtering techniques. The signal to be analyzed is passed through filters with different cut-off frequencies at different scales.

Although the discretized continuous wavelet transform enables the computation of the continuous wavelet transform by computers, it is not a true discrete transform. As a matter of fact, the wavelet series is simply a sampled version of the CWT, and the information it provides is highly redundant as far as the reconstruction of the signal is concerned. This redundancy, on the other hand, requires a significant amount of computation time and resources. The discrete wavelet transform (DWT), on the other hand, provides sufficient information both for analysis and synthesis of the original signal, with a significant reduction in the computation time.

The main idea is the same as it is in the CWT. A time-scale representation of a digital signal is obtained using digital filtering techniques. Recall that the CWT is a correlation between a wavelet at different scales and the signal with the scale (or the frequency) being used as a measure of similarity. The continuous wavelet transform was computed by changing the scale of the analysis

window, shifting the window in time, multiplying by the signal, and integrating over all times. In the discrete case, filters of different cutoff frequencies are used to analyze the signal at different scales. The signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies.

The resolution of the signal, which is a measure of the amount of detail information in the signal, is changed by the filtering operations, and the scale is changed by upsampling and downsampling (subsampling) operations. Subsampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. For example, subsampling by two refers to dropping every other sample of the signal. Subsampling by a factor n reduces the number of samples in the signal n times.

Upsampling a signal corresponds to increasing the sampling rate of a signal by adding new samples to the signal. For example, upsampling by two refers to adding a new sample, usually a zero or an interpolated value, between every two samples of the signal. Upsampling a signal by a factor of n increases the number of samples in the signal by a factor of n .

The procedure starts with passing this signal (sequence) through a half band digital lowpass filter with impulse response $h[n]$. Filtering a signal corresponds to the mathematical operation of convolution of the signal with the impulse response of the filter. The convolution operation in discrete time is defined as follows:

$$X[n]*h[n] = \sum_{K=-\infty} x[k].h[n-k]$$

A half band lowpass filter removes all frequencies that are above half of the highest frequency in the signal. For example, if a signal has a maximum of 1000 Hz component, then half band lowpass filtering removes all the frequencies above 500 Hz [18].

The unit of frequency is of particular importance at this time. In discrete signals, frequency is expressed in terms of radians. Accordingly, the sampling frequency of the signal is equal to 2π radians in terms of radial frequency. Therefore, the highest frequency component that exists in a signal will be π radians, if the signal is sampled at Nyquist's rate (which is twice the maximum frequency that exists in the signal); that is, the Nyquist's rate corresponds to π rad/s in the discrete frequency domain. Therefore using Hz is not appropriate for discrete signals. However, Hz is used whenever it is needed to clarify a discussion, since it is very common to think of frequency in terms

of Hz. It should always be remembered that the unit of frequency for discrete time signals is radians.

After passing the signal through a half band lowpass filter, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of $\pi/2$ radians instead of π radians. Simply discarding every other sample will subsample the signal by two, and the signal will then have half the number of points. The scale of the signal is now doubled. Note that the lowpass filtering removes the high frequency information, but leaves the scale unchanged. Only the subsampling process changes the scale. Resolution, on the other hand, is related to the amount of information in the signal, and therefore, it is affected by the filtering operations. Half band lowpass filtering removes half of the frequencies, which can be interpreted as losing half of the information. Therefore, the resolution is halved after the filtering operation. Note, however, the subsampling operation after filtering does not affect the resolution, since removing half of the spectral components from the signal makes half the number of samples redundant anyway. Half the samples can be discarded without any loss of information. In summary, the lowpass filtering halves the resolution, but leaves the scale unchanged. The signal is then subsampled by 2 since half of the number of samples is redundant. This doubles the scale.

This procedure can mathematically be expressed as:

$$y[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[2n-k]$$

Having said that, we now look how the DWT is actually computed: The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information. DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with low pass and highpass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive highpass and lowpass filtering of the time domain signal. The original signal $x[n]$ is first passed through a halfband highpass filter $g[n]$ and a lowpass filter $h[n]$. After the filtering, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of $\pi/2$ radians instead of π . The signal can therefore be subsampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_n x[n] \cdot g[2k-n]$$

$$y_{low}[k] = \sum_n x[n] \cdot h[2k-n]$$

Where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the highpass and lowpass filters, respectively, after subsampling by 2.

This decomposition halves the time resolution since only half the number of samples now characterizes the entire signal. However, this operation doubles the frequency resolution, since the frequency band of the signal now spans only half the previous frequency band, effectively reducing the uncertainty in the frequency by half. The above procedure, which is also known as the subband coding, can be repeated for further decomposition. At every level, the filtering and subsampling will result in half the number of samples (and hence half the time resolution) and half the frequency band spanned (and hence doubles the frequency resolution). The following figure illustrates this procedure, where $x[n]$ is the original signal to be decomposed, and $h[n]$ and $g[n]$ are lowpass and highpass filters, respectively. The bandwidth of the signal at every level is marked on the figure as "f".

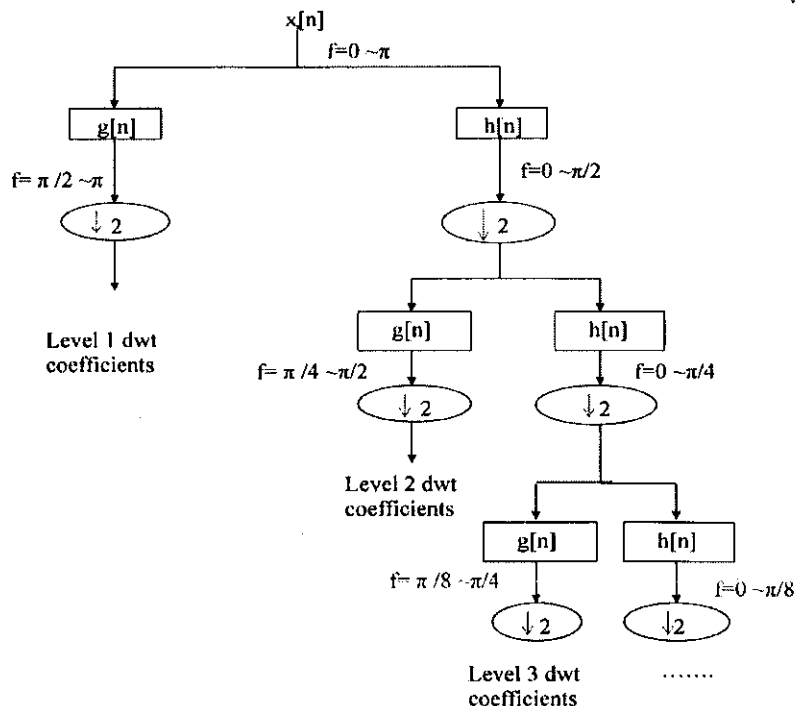


Figure 4.7 Subband Coding [18]

The frequencies that are most prominent in the original signal will appear as high amplitudes in that region of the DWT signal that includes those particular frequencies. The difference of this transform from the Fourier transform is that the time localization of these frequencies will not be lost. However, the time localization will have a resolution that depends on which level they appear. If the main information of the signal lies in the high frequencies, as happens most often, the time localization of these frequencies will be more precise, since they are characterized by more number of samples. If the main information lies only at very low frequencies, the time localization will not be very precise, since few samples are used to express signal at these frequencies. This procedure in effect offers a good time resolution at high frequencies, and good frequency resolution at low frequencies. Most practical signals encountered are of this type. The frequency bands that are not very prominent in the original signal will have very low amplitudes, and that part of the DWT signal can be discarded without any major loss of information, allowing data reduction.

One area that has benefited the most from this particular property of the wavelet transforms is image processing. As you may well know, images, particularly high-resolution images, claim a lot of disk space.

For a given image, you can compute the DWT of, say each row, and discard all values in the DWT that are less than a certain threshold. We then save only those DWT coefficients that are above the threshold for each row, and when we need to reconstruct the original image, we simply pad each row with as many zeros as the number of discarded coefficients, and use the inverse DWT to reconstruct each row of the original image. We can also analyze the image at different frequency bands, and reconstruct the original image by using only the coefficients that are of a particular band [18].

Advantages of DWT

1. Using wavelets, the whole image is seen as one block – the edges are no longer given.
2. Wavelets property of multiresolution analysis reduces the computational time of the detection procedure.
3. Allows good localization both in time and spatial frequency domain.
4. Better identification of which data is relevant to human perception.
5. Higher flexibility: Wavelet function can be freely chosen.

Disadvantages of DWT

1. The cost of computing DWT as compared to DCT may be higher.
2. The use of larger DWT basis functions or wavelet filters produces blurring and ringing noise near edge regions in images or video frames.
3. Longer compression time.
4. Lower quality than JPEG at low compression rates.

4.2 Technique Implemented

The DWT (Discrete Wavelet Transform) separates an image into a lower resolution approximation image (LL) as well as horizontal (HL), vertical (LH) and diagonal (HH) detailed components. The process can then be repeated to compute multiple "levels" wavelet decomposition.

Embedding watermarks in these regions allow us to increase the robustness of our watermark, at little to no additional impact on image quality.

The technique now used is to modify the pixel values of the HL & LH components by adding some random sequence of numbers to it. On performing the Inverse DWT of this image would now give us the watermarked image. Since, we add a specific set of random numbers (by providing a particular 'seed' value in the code that is kept confidential), none of the codes or algorithm hence applied would be able to tamper with the image and extract the watermark. The more we decompose our image to embed our watermark, the more robust watermarking it would be.

Recovery of the watermark from the watermarked image can be simply done by first decomposing the watermarked image into its approximate and detailed coefficients and then correlating their corresponding pixel values with same random sequence generated while on embedding. If the correlation value exceeds the certain threshold, the pixel value in the watermark is modified. We also need to be aware of the precise size of watermark so that we can reshape the matrix generated by these threshold values. If not so, the perfect image or shape in the watermark cannot be detected [20, 21].

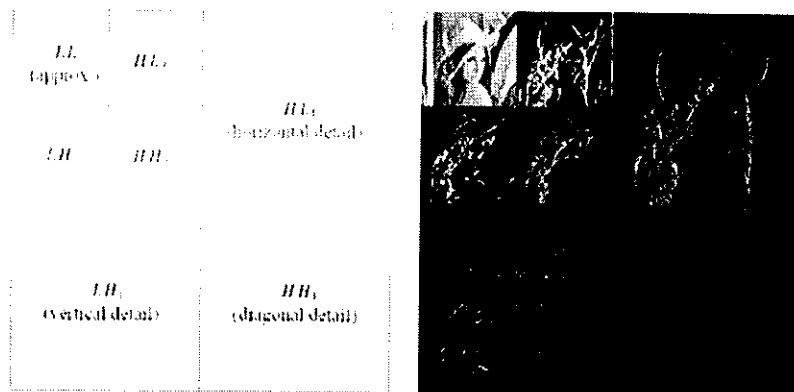


Figure 4.8 Signal and Image decomposition structure by DWT

4.3 Algorithm

4.3.1 Steps Implemented

Invisible watermarking is a process of applying a watermark or embedding/modifying original pixel values in such a way that changes in the resultant are not perceptible.

The explanation of Invisible Watermarking Embedding process we have used has been shown below.

(Note: We are working on gray scale and binary images)

1. We have taken 'lena.bmp' as our image to be watermarked and 'bb.bmp' as our image to be used as a watermark.



Figure 4.9 Cover and Watermark image used

2. After storing the sizes of both these images into some variables in MATLAB, we reshape the watermark image into a single row or column matrix.
3. We set a particular 'seed value' for random number generator. This is the value that would always generate a same sequence of random numbers and a different seed value would give different sequence. This seed value is used in the extraction algorithm as well.

4. The original image is decomposed into its DWT component matrices.

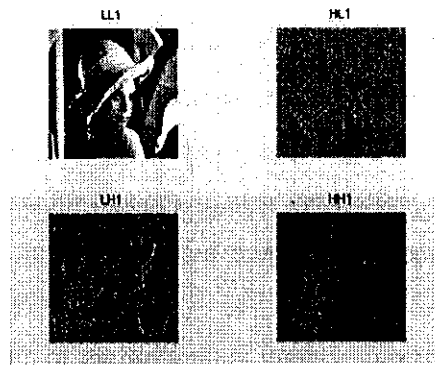


Figure 4.10 Decomposed Original Image into its various frequency components

5. Running a loop till the length of the watermark image, as soon as we encounter a black pixel in the watermark image ('0' value), we add random sequences to the HL & LH components of the decomposed original image. By this we have encrypted our watermark into the original image.
6. Performing the inverse DWT we get our watermarked image.



Figure 4.11 Watermarked Image

Since here we have modified only the HL & LH components (mid-frequency components of the image, that provide us just an overview of the image), not much difference is seen in our watermarked & original image, hence making it less perceptible to another person.

Invisible watermarks are hence more valuable in applications where the visible watermarks are deemed to be inappropriate, such as sale of high quality images.

Now if a dispute arises over the ownership of the image, or we wish to see if an image has been tampered with, we can perform the Recovery of Watermark from the Watermarked Image.

RECOVERY:

1. We store the size of the watermarked image in some variable. This would help in reshaping the watermark image matrix.
2. Also we get the size of the watermark that we had embed and reshape it into a 1-D matrix. Pad ones into this matrix. This hence is a watermark (image) that is completely white.
3. Again we input the seed value for random number generation.
4. Now taking the DWT of our watermarked image, we decompose it into various frequency components.
5. Running a loop till the length of the watermark image, random sequences of same sizes as the HL & LH components are generated.
6. Now we correlate these random sequences with the HL & LH components of our watermarked image and store the results into different correlation vectors.
7. Averaging the 2 correlation vectors into 1 vector, we get the overall result of correlation.
8. Again running the same loop, we compare the final result of correlation with the mean correlation value in this case, and correspondingly pad 0's into the message vector matrix.
9. Reshaping this matrix into 2-D matrix we get our watermark back!!



Figure 4.12 Extracted Watermark

CHAPTER 5

RESULTS AND OBSERVATIONS

5.1 Results using DCT

We have embedded the watermark image i.e. “bb”(30 X 30) into the original image i.e. “lena” (512 X 512) using our DCT embedding algorithm; hence we get our Watermarked Image as shown below.

We calculated the time taken to embed this watermark image into the original image and it came out to be ‘1.0024’ seconds.

Now to get back our Watermark from the watermarked image, we use our DCT extraction algorithm .Time taken to extract the watermark is around ‘1.7701’seconds.

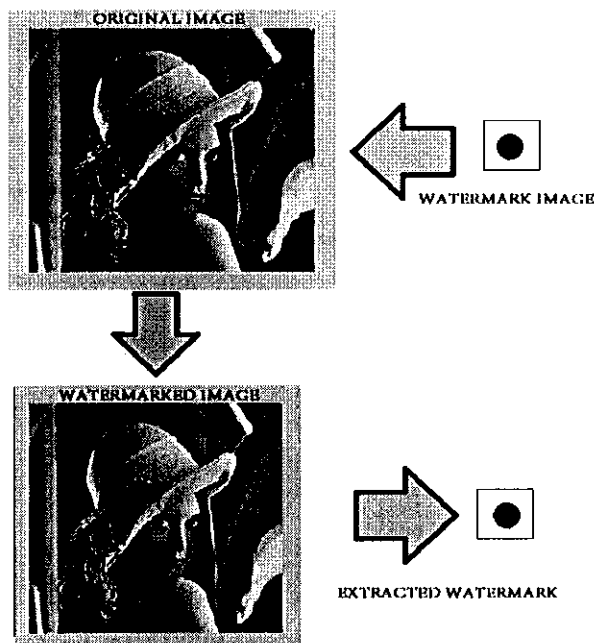


Figure 5.1 DCT Watermarking

The time taken for embedding and extraction depends on

1. Size of the image.
2. Hardware specifications of the computer.

Now we applied various operations on our watermarked image like inversion, flipping, tampering and then extracted our watermark from the image.

The results were as follows:

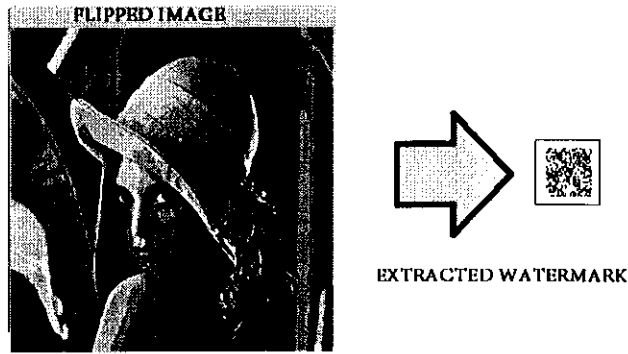


Figure 5.2 Extracted Watermark from Flipped Image

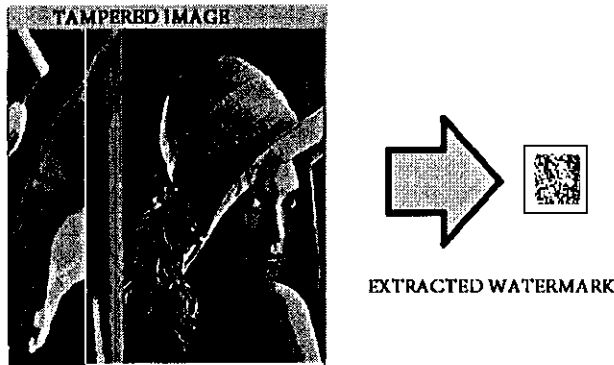


Figure 5.3 Extracted Watermark from Tampered Image

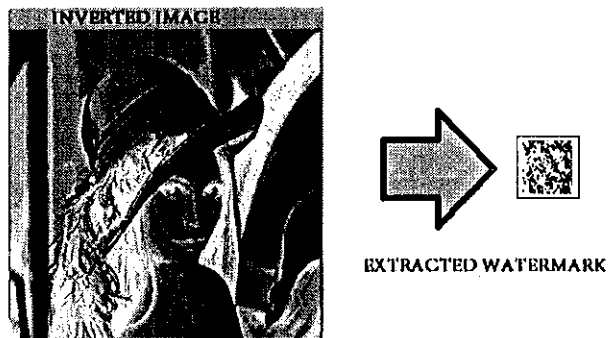


Figure 5.4 Extracted Watermark from Inverted Image

We observe that by performing different operations with the image, we also get distorted watermarks and hence this can be useful when authenticity of an image has to be maintained.



Figure 5.5 Watermarked Images obtained by varying values of 'g' as 2, 5 and 20

Clearly, more is the gain factor, more are the changes in pixel intensity values of watermarked image and hence more it is perceptible.

5.2 Results using DWT

Similarly the results for embedding and extraction of watermarks using DWT algorithms were obtained.

Time taken to embed the same watermark "bb" into "lena" took '14.7986' seconds & '22.3607' seconds to extract it.

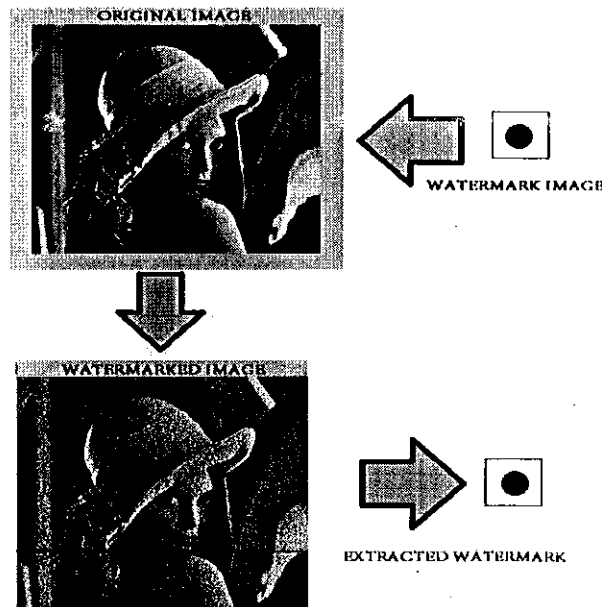


Figure 5.6 DWT Watermarking

Applying some operations like flipping and inversion on the image we get the following results:

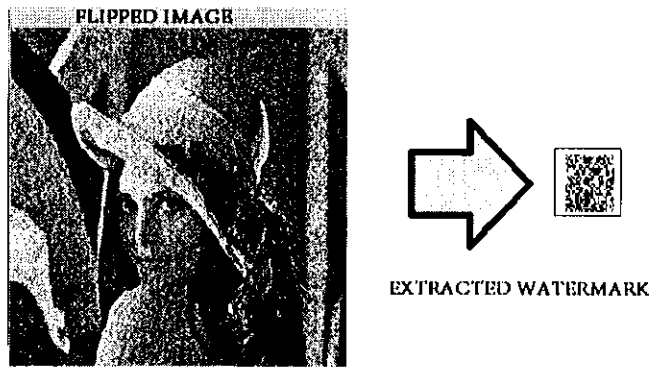


Figure 5.7 Extracted Watermark from Flipped Image

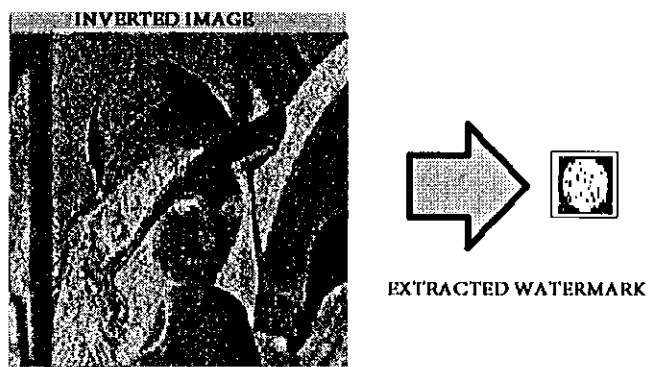


Figure 5.8 Extracted Watermark from Inverted Image



Figure 5.9 Watermarked Images obtained by varying values of 'k' as 2, 5, 20,200(Top Left to Right Bottom)

CHAPTER 6

SOURCE CODE

6.1 Watermark Embedding Using DCT

```
clear all;
k=200;          % set minimum coeff difference
blocksize=8;   % setting the size of the block in original image

file='lena.bmp';      % reading the original image
original=double(imread(file));
mo=size(original,1);  % getting the image height
no=size(original,2);  % getting the image width
maxlength=mo*no/(blocksize^2);

% reading the watermark image
file='bb.bmp';
watermark=double(imread(file));
mw=size(watermark,1);
nw=size(watermark,2);

% reshaping the watermark image to 1D matrix
watermark=fix(reshape(watermark,mw*nw,1));
watermark=watermark(1:end);

if (length(watermark) > maxlength)
    error('Watermark too large to fit in Original Image');
end

% pad the watermark out to the maximum watermark size with ones
watermarkpad=ones(1,maxlength);
watermarkpad(1:length(watermark))=watermark;
```



```

final=original;

x=1;
y=1;
for (w = 1:length(watermarkpad))
% transform block using DCT
dctblock=dct2(original(y:y+blocksize-1,x:x+blocksize-1));
if (watermarkpad(w) == 0)

% swap values such that (5,2) > (4,3) when watermark(w)=0
if (dctblock(5,2) < dctblock(4,3))
temp=dctblock(4,3);
dctblock(4,3)=dctblock(5,2);
dctblock(5,2)=temp;
end
elseif (watermarkpad(w) == 1)

% making (5,2) < (4,3) when watermark(w)=1
if (dctblock(5,2) >= dctblock(4,3))
temp=dctblock(4,3);
dctblock(4,3)=dctblock(5,2);
dctblock(5,2)=temp;
end
end

% now we need to adjust the two values such that their difference >= k
if dctblock(5,2) > dctblock(4,3)
if dctblock(5,2) - dctblock(4,3) < k
dctblock(5,2)=dctblock(5,2)+(k/2);
dctblock(4,3)=dctblock(4,3)-(k/2);
end
else

```

```

    if dctblock(4,3) - dctblock(5,2) < k
        dctblock(4,3)=dctblock(4,3)+(k/2);
        dctblock(5,2)=dctblock(5,2)-(k/2);
    end
end
% transforming the block back into spatial domain
final(y:y+blocksize-1,x:x+blocksize-1)=idct2(dctblock);

% move on to next block. At end of row move to next row
if (x+blocksize) >= no
    x=1;
    y=y+blocksize;
else
    x=x+blocksize;
end
end

% convert to uint8 and write the watermarked image out to a file
finalint=uint8(final);
imwrite(finalint,'final2.bmp');

figure(1);
subplot(1,2,1);
imshow(original,[]);      % display the original image
title('Original Image')
subplot(1,2,2);
imshow(final,[]);        % display the final watermarked image
title('Watermarked Image')

```

6.2 Watermark Recovery Using DCT

```
clear all;
blocksize=8;
file='final2.bmp';      % reading the final watermarked image
final=double(imread(file));
mw=size(final,1);
nw=size(final,2);
maxlength=mw*nw/(blocksize^2);
x=1;
y=1;
for (w = 1:maxlength)
    % transform block using DCT
    dctblock=dct2(final(y:y+blocksize-1,x:x+blocksize-1));
    % if dctblock(5,2) > dctblock(4,3) then water1(w)=0
    if dctblock(5,2) > dctblock(4,3)
        water1(w)=0;
    else
        water1(w)=1; % otherwise water1(w)=1
    end
    % move on to next block. At and of row move on to the next row
    if (x+blocksize) >= mw
        x=1;
        y=y+blocksize;
    else
        x=x+blocksize;
    end
end
% reshaping the extracted 1D watermark in 2D
watermark=reshape(water1(1:900),30,30);
figure(1)
imshow(watermark,[]);
title('Extracted Watermark');
```

6.3 Watermark Embedding Using DWT

```
1 clear all;
% set the gain factor for embedding
g=2;
% read in the original image and determining its size
file_name='lena.bmp';
cover_object=double(imread(file_name));
Mc=size(cover_object,1); %Height
Nc=size(cover_object,2); %Width

% read in the water image and reshaping it into 1D matrix
file_name='water.bmp';
water=double(imread(file_name));
Mm=size(water,1); %Height
Nm=size(water,2); %Width
water_vector=reshape(water,Mm*Nm,1);

% resetting the state of random generator
rand('state',10);

[LL1,HL1,LH1,HH1] = dwt2(cover_object,'haar');

% add random sequences to HL1 and LH1 components when water = 0
for (k=1:length(water_vector))
    seq_hl=round(2*(rand(Mc/2,Nc/2)-0.5));
    seq_lh=round(2*(rand(Mc/2,Nc/2)-0.5));

    if (water(k) == 0)
        HL1=HL1+g*seq_hl;
        LH1=LH1+g*seq_lh;
    end
end
end
```

```
% perform IDWT
watermarked = idwt2(LL1,HL1,LH1,HH1,'haar',[Mc,Nc]);

% convert back to uint8
watermarked_uint8=uint8(watermarked);

% write watermarked Image to a file
imwrite(watermarked_uint8,'watermarked.bmp','bmp');

% display watermarked image
figure(1)
imshow(watermarked_uint8,[])
title('Watermarked Image')

figure(2)
subplot(2,2,1);imshow(LL1,[]);title('LL1');
subplot(2,2,2);imshow(HL1,[]);title('HL1');
subplot(2,2,3);imshow(LH1,[]);title('LH1');
subplot(2,2,4);imshow(HH1,[]);title('HH1');
```

6.4 Watermark Recovery Using DWT

```
clear all;
% read in the watermarked object
file_name='watermarked.bmp';
watermarked=double(imread(file_name));
Mw=size(watermarked,1);    %Height
Nw=size(watermarked,2);    %Width

% read in original watermark
file_name='water.bmp';
water=double(imread(file_name));

% determine size of original watermark
Mo=size(water,1); %Height
No=size(water,2); %Width

% resetting the state of random generator
rand('state',10)

% initialize watermark vector bits to all ones
water_vector=ones(1,Mo*No);
[LL1,HL1,LH1,HH1] = dwt2(watermarked,'haar');

% add random sequences to HL1 and LH1 components when message = 0
for (k=1:length(water_vector))
    seq_hl=round(2*(rand(Mw/2,Nw/2)-0.5));
    seq_lh=round(2*(rand(Mw/2,Nw/2)-0.5));

    correlation_h(k)=corr2(HL1,seq_hl);
    correlation_v(k)=corr2(LH1,seq_lh);
    correlation(k)=(correlation_h(k)+correlation_v(k))/2;
end
```

```
for (k=1:length(water_vector))
    if (correlation(k) > mean(correlation))
        water_vector(k)=0;
    end
end

% reshape the watermark vector and display extracted watermark.
figure(2)
message=reshape(water_vector,Mo,No);
imshow(message,[])
title('Extracted Watermark')
```

CHAPTER 7

CONCLUSION & FUTURE WORK

We have studied and implemented the Discrete Cosine Transform and the Discrete Wavelet transform techniques in our project. We made a literature survey from various internet sites, books and papers as mentioned in our bibliography to study the various techniques for embedding and recovering an invisible watermark from the image. After understanding some of the techniques we implemented DWT and DCT for watermarking the images. We made use of the MATLAB for simulation of our proposed algorithm. We observed that working in transform domains was much efficient than working in spatial, both in terms of robustness and visual impact. We had made use of minimum coefficient difference, 'k' & gain factor, 'g' in our DCT and DWT algorithms respectively. By anticipating their values we were able to make our watermarked images even more robust but with some image quality degradation. Overall we observed that DWT was a more efficient technique to watermark images as it led to less image quality degradation at the cost of higher processing time. More sophisticated wavelet domain techniques would improve upon both the robustness and visual impact, also reducing the computational requirements. The computation time depends upon the parameters such as size of the images used and the specifications of the computer (specially RAM and processor used).

We have applied the Discrete Cosine Transform and the Discrete Wavelet Transform technique in our project to watermark the images. We could henceforth use these techniques to watermark audio, speech and video signals too, but first step would be to implement this algorithm for colored images. Moreover, in the proposed algorithm, we have added a sequence of random numbers according to the intensity value of the watermark; however, embedding the exact watermark as such in the image would be the goal we will be looking forward to. Also we will try to build algorithms for faster computation and decrease the processing time.

REFERENCES

- [1] I. J. Cox, M. L. Miller and J. A. Bloom, Digital Watermarking, Morgan Kaufmann Publishers, 2002.
- [2] J. Cummins, P. Diskin, S. Lau and R. Parlett, "Steganography and Digital Watermarking", Student Seminar Report, School of Computer Science, University of Birmingham, 2004.
- [3] S. Katzenbeisser and F. A. P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, 2000.
- [4] Jonathan M. Bloom, "Revolution by the Ream: A History of Paper", Saudi Aramco World May/June 1999 print edition, vol. 50, pp. 26-39, May/June 1999.
- [5] "Digital Watermark" available at <http://www.ncd.matf.bg.ac.yu/casopis/05/Vuckovic>
- [6] Frank Hartung, Martin Kutter, "Multimedia Watermarking Techniques", Proceedings of The IEEE, Vol. 87, No. 7, pp. 1085 – 1103, July 1999.
- [7] Y. Zhao, "Dual Domain Semi-fragile Watermarking for Image Authentication", M.S. Thesis, University of Toronto, Canada, 2003.
- [8] S. A. Naveed, "Improved Watermarking Scheme Using Decimal Sequences", M.S. Thesis, Louisiana State University, United States, 2005.
- [9] S. P. Mohanty, "Watermarking of Digital Images", M.S. Thesis, Indian Institute of Science, India, 1999.
- [10] Saraju Prasad Mohanty, "Watermarking of Digital Images", Submitted at Indian Institute of Science Bangalore, pp. 1.3 – 1.6, January 1999.
- [11] A White paper on "Digital Watermarking: A Technology Overview", Wipro Technologies, pp. 2 – 8. Aug. 2003.
- [12] CHAN Pik-Wah, "Digital Video Watermarking Techniques for Secure Multimedia Creation and Delivery", submitted at The Chinese University of Hong Kong, pp. 7 – 15, July 2004
- [13] Alper Koz, "Digital Watermarking Based on Human Visual System", The Graduate School of Natural and Applied Sciences, The Middle East Technical University, pp 2 – 8, Sep 2002.
- [14] J.J.K.O. Ruanaidh, W.J.Dowling, F.M. Boland, "Watermarking Digital Images for Copyright Protection", in IEE ProcVis. Image Signal Process., Vol. 143, No. 4, pp 250 - 254. August 1996.
- [15] Brigitte Jellinek, "Invisible Watermarking of Digital Images for Copyright Protection" submitted at University Salzburg, pp. 9 – 17, Jan 2000.
- [16] http://en.wikipedia.org/wiki/Discrete_cosine_transform

- [17] Tribhuvan Kumar Tiwari and Vikas Saxena ,“ An Improved and Robust DCT Based Digital Image Watermarking Scheme”, International Journal of Computer Applications (0975 – 8887) Volume 3 – No.1, June 2010
- [18] Wavelet Tutorials by Robi Polikar
- [19] http://en.wikipedia.org/wiki/Haar_wavelet
- [20] Kamran Hameed, Adeel Mumtaz, and S.A.M. Gilani, “Digital Image Watermarking in the Wavelet Transform Domain”. World Academy of Science, Engineering and Technology 13 2006
- [21] Ming-Shing Hsieh, Din-Chang Tseng, Member, IEEE, and Yong-Huai Huang, “Hiding Digital Watermarks Using Multiresolution Wavelet Transform”. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 48, NO. 5, OCTOBER 2001

BRIEF BIO DATA

ABHIJIT SINGH WANDER

Department of Electronics and Communication Engineering,
Jaypee University of Information Technology,
Waknaghat, Solan (H.P)

Pursuing	Name of School/University	Year	CGPA
B.Tech (ECE)	Jaypee University of Information Technology, Solan (H.P.)	2011	8.3 (85%) (Up Till 7 th sem)

Currently working on the **Invisible Digital Image Watermarking Using DCT & DWT.**

AMIT BHARDWAJ

Department of Electronics and Communication Engineering,
Jaypee University of Information Technology,
Waknaghat, Solan (H.P)

Pursuing	Name of School/University	Year	CGPA
B.Tech (ECE)	Jaypee University of Information Technology, Solan (H.P.)	2011	6.3 (70%) (Up Till 7 th sem)

Currently working on the **Invisible Digital Image Watermarking Using DCT & DWT.**

KARTIKEYA KHANNA

Department of Electronics and Communication Engineering,
Jaypee University of Information Technology,
Waknaghat, Solan (H.P)

Pursuing	Name of School/University	Year	CGPA
B.Tech (ECE)	Jaypee University of Information Technology, Solan (H.P.)	2011	7.5 (79%) (Up Till 7 th sem)

Currently working on the **Invisible Digital Image Watermarking Using DCT & DWT.**