WAVELET-BASED WATERMARKING METHOD FOR MOVING IMAGES USING HVS

A Dissertation Submitted in partial fulfillment of the requirements
for the award of the degree of

MASTER OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

By

Vikas Deep Raghuwanshi

Delhi University Roll No. 8739

Under the guidance of

Mr. O. P. VERMA

Department Of Electronics and Communication Engineering
Delhi College of Engineering, University of Delhi
DELHI-110042

(Session 2004-06)

CERTIFICATE

This is to certify that the dissertation titled as **"Wavelet-Based Watermarking Method for Moving Images Using HVS"** which is submitted by **Vikas Deep Raghuwanshi**, Class Roll No. 12/E&C/04, University Roll No 8739, in partial fulfillment for the award of "Master of Engineering Degree in Electronics & Communication" in Delhi College of Engineering, Delhi University, Delhi; is the original work carried out by him under my guidance and supervision. The matter contained in this report has not been submitted elsewhere for award of any other degree.

(Project Guide)
Mr. O. P. VERMA
Assistant Professor
Department of Electronics & Communication Engineering
Delhi College of Engineering
Bawana Road, Delhi

ACKNOWLEDGEMENT

It is a great pleasure to have the opportunity to extent my heartiest felt gratitude to everybody who helped me throughout the course of this project.

I would like to express my heartiest felt regards to **Mr. O. P. Verma**, Assistant Professor, Department of Electronics and Communication Engineering for the constant motivation and support during the duration of this project. It is my privilege and honor to have worked under his supervision. Her invaluable guidance and helpful discussions in every stage of this project really helped me in materializing this project. It is indeed difficult to put his contribution in few words.

I would also like to take this opportunity to present my sincere regards to **Professor Asok Bhattacharyya**, Head of the Department, Electronics and Communication Engineering, D.C.E. Delhi, for the support provided by him during the entire duration of degree course and especially in this thesis.

I am also thankful to all teaching and non-teaching staff at DCE, my fellows and classmates for their unconditional support and motivation during this project.

Vikas Deep Raghuwanshi

M.E. (Electronics and Communication)
College Roll No. 12/E&C/04
Delhi University Roll No. 8739

ABSTRACT

Watermarking is finding more and more support as a possible solution for the protection of intellectual property rights. A watermarking algorithm operating in the wavelet domain is presented. Performance improvement with respect to existing algorithms is obtained by means of a new approach to mask the watermark according to the characteristics of the human visual system (HVS). Discrete Wavelet Transform (DWT) exhibiting a strong similarity to the way the HVS processes images.

The video sequence is partitioned into frames and the single shots are projected onto the DWT domain. If a DWT coefficient is modified, only the region of the image where the particular frequency corresponding to that coefficient is present will be modified. This property is utilized while embedding the watermark. The proposed mask takes into account the variance and the luminance of the sub-bands where the watermark is embedded. Finally, the inverse DWT is performed thus obtaining the watermarked video shot. The imperceptibility of the proposed technique along with its robustness has been observed through experimental results.

CONTENTS

List of Tables

List of Figures

List of figures generated in MATLAB

CHAPTER # 1

INTRODUCTION

1.1 Introduction

The pace of the current digital revolution in multimedia distribution is constantly accelerating. Development of the CD, DVD, digital radio and digital television as distribution mechanisms have all been major milestones in this growth. Internet based distribution has also become economically viable, and digital devices have converged into integrated playback and recording systems for many media and content types. The inherent flexibility of Internet facilitates users to transact with one another to create, distribute, store, peruse, subscribe, enhance, modify and trade digital content in various forms like text documents, databases, e-books, still images, audio, video, computer software and games. However, this digital revolution has presented new problems in copyright protection as perfect copies of digital content can easily be made, and the barrier of physicality that has historically inhibited illegal distribution has been removed. This is particularly true of digital images, whose relatively small storage and bandwidth requirements make them easy targets for casual copyright infringement. Additionally, as images have no temporal dimension, they are inherently more pliable than other content types, i.e. copying only a central region of an image is generally much more useful than copying a central region of a song.

Practical solutions for protecting image copyright have been driven mainly by the 'stock-photography' industry, as its revenue model depends on the identification of illegal distribution and also enforcement of legitimate purchasers' 'usage-rights'. The standard cryptographic techniques could provide a solution: An image could be encrypted by the copyright holder prior to transit, and only decrypted by the legitimate purchaser via a secret key. However, it is clear that the copyright holder has no control over how the purchaser then further distributes or uses the decrypted image. A complementary technology is required that can protect the image after it has been decrypted; Digital Watermarking has been proposed as a mechanism for this protection.

1.2 What Is A Digital Watermark?

A digital watermark is a piece of information embedded into a digital image that in some way acts as meta-data for the image. The key idea is that of embedding rather than

appending the information. Embedding in this context means to add the information directly into the image data in such a way that it is not easily removed. If the watermark is removed, then it should render its host image unusable for its original purpose. This is not the case for appended information (for example in the header of a file), as this can generally be removed without destroying the image it describes. Although this report focuses on digital watermarking of images, it should be noted that techniques have been developed for many digital content types, including audio, video, text, and computer software.

The process of embedding information into another object or signal is termed as ***Watermarking.***

1.3 History

First watermark related publications date back to 1979. However, it was only in 1990 that it gained a large international interest. With the growth of the internet & the immediate availability of computing resources to everyone, "Digitized property "can be reproduced & instantaneously distributed without quality loss at basically any cost. Until now, intellectual property (IP) and value has always been bound to some physical container that could not be easily duplicated, thereby guaranteeing that the creator benefits from his work.

As audio, video & other works become available in digital form, it may be that the ease with which perfect copies can be made will lead to large-scale unauthorized copying which will undermine the music, film, book and software publishing industries.

One technical way to make law enforcement & copyright protection for digital media possible and practical is digital watermarking which is aimed to automatically detect & possibly also prosecute copyright infringement. There has therefore been significant recent research into "watermarking" (hiding copyright messages) and "fingerprinting" (hiding serial numbers or a set of characteristics that tend to distinguish an object from other similar objects): the idea is that the latter can be used to detect copyright violators and the former to prosecute them.

1.3.1 Cryptography

Literally, Cryptography is the art of writing in ‘ciphers’; or it is a method of secret communication. In cryptography, the contents of secret message are concealed and only the sender and the receiver of the secret message know the process of extracting the concealed information. Apparently, others can’t easily detect what message is being conveyed. Cryptography is an effective solution to the distribution problem, but in most instances has to be tied to specialized and costly hardware to create tamper-proof devices that avoid direct access to data in digital format. Moreover, most cryptographic protocols are concerned with secured communications instead of ulterior copyright infringements. For instance, access control in set-top-boxes used for digital television demodulation and decoding succeed in avoiding unauthorized access to programs that are being broadcast in scrambled form but fail in precluding further storage and illegal dissemination actions.

1.3.2 Steganography

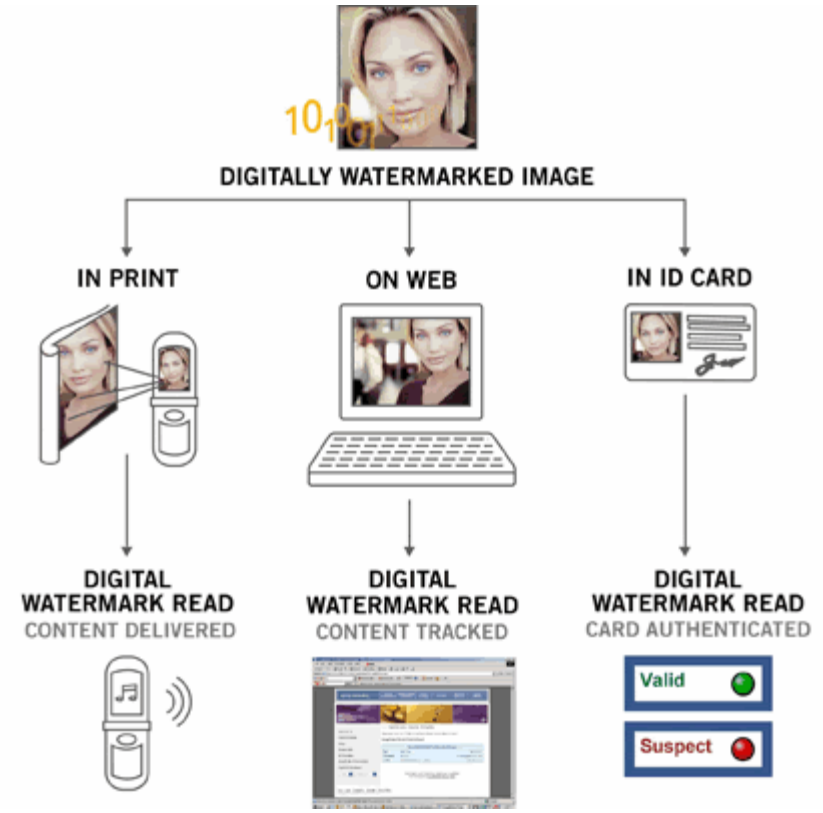
Steganography is a technique for concealed communication. In contrast to cryptography where the content of a communicated message is secret, in steganography the very existence of the message that is communicated is a secret and its presence is known only by parties involved in the communication. Steganography is technique where a secret message is hidden within another unrelated message and then communicated to the other party. Some of the techniques of steganography like use of invisible ink, word spacing patterns in printed documents, coding messages in music compositions, etc., have been used by military intelligence since the times of ancient Greek civilization. In steganography, usually the message itself is of value and must be protected through clever hiding techniques and the “vessel” for hiding the message is not of value. In watermarking, the effective coupling of message to the “vessel”, which is the digital content, is of value and the protection of the content is crucial.

Fragile Invisible Steganography Algorithm "Manipulating LSBs" Goal: To hide image-B in image-A

- ✓ Replace one LSB of image-A with the corresponding one MSB of image-B.
- ✓ Replace two LSBs of image-A with the corresponding two MSBs of image-B.
- ✓ Compare the results of the two manipulations with the original image-A.
- ✓ In general, replace 'k' LSBs of image-A with the corresponding 'k' MSBs of image-B, and observe the results.

1.3.3 Digital Watermark

A Digital Watermark is a special message embedded in an image, whether it is a photo, video or other digital content. Watermarking technologies embeds these "imperceptible" messages by making subtle changes to the data of the original digital content. These watermarks can then be "read" to validate original content and/or deliver an action, such as delivering content to a mobile phone.



The change to the media is so subtle that such digital watermarks are considered imperceptible. Ideal characteristics of a digital watermark have been stated. These characteristics include:

- ✓ Statistical invisibility.
- ✓ Fairly simple extraction should be.
- ✓ Accurate detection.
- ✓ Robustness to filtering, additive noise, compression, and other manipulations.
- ✓ Ability to determine the true owner of the image.

1.3.4 Digital Watermarking

Digital watermarking describes methods and technologies that allow hiding information, for example a number or text, in digital media, such as images, video and audio. The embedding takes place by manipulating the content of the digital data that means the information is not embedded in the frame around the data. The hiding process has to be such that the modifications of the media are *imperceptible*. For images this means that the modifications of the pixel values have to be invisible. Furthermore, the watermark has to be *robust* or *fragile*, depending on the application. Robustness refers to the capability of the watermark to resist to manipulations of the media, such as lossy compression, scaling, and cropping etc. Fragility means that the watermark should not resist tampering, or only up to a certain extent

1.4 Copyright Protection

The goal of watermarking for copyright protection is to embed a “mark” into the image data that can identify the copyright holder of the work. Together with owner identification, one might also want to embed a mark (or fingerprint) identifying the buyer of a work for circulation tracking. The mark can be a registered number (like the UPC found on compact disc media), a text message or graphical logo, or some unique pattern (similar to a DNA fingerprint).The term watermark stems from the ancient art of marking paper with a logo for the same purpose.

Digital watermarks can either be perceptible or imperceptible. Visible image watermarks, often the logo of the copyright holder, can be easily apply to the image but are hard to remove. Many applications require the watermark to be invisible; however this work focuses on invisible watermarks in digital images only.

The embedded , invisible watermark has to be robust against common image processing operations like image compression(e.g. JPEG) , image filtering (Edge enhancement, contract enhancement,…) and geometrical transformations (e.g. cropping ,scaling ,…). Therefore, the watermark can not be stored in the file format, but has to be embedded into the image data itself. In order to establish a proof of ownership in a trial, a watermarking scheme also has to be secure against intentional malicious attacks, here, cryptographic techniques and statistical properties of pseudo–random numbers play an essential role.

1.5 Image & Video Authentication & Data Hiding

Another need for watermarking is for “image authentication” & “temper detection”. Digital photographs are being used more and more often as court evidence nowadays. Here, watermarking is used to detect significant modification of the image. Digital image are susceptible to seamless modifications from sophisticated image processing applications. Watermarks can be used here as a means to verify the genuineness of an image. Verification watermarks are required to be fragile, so that any modification to the image will destroy (or detectable alter) the mark. Unlike cryptographic message digests which can only validate identical copies, watermarking for image authentication should tolerate some well-defined image distortion (e.g. file format conversion, re-sampling, re-compression or progressive transmission).

Data hiding or steganography tries to invisibly embed the maximum amount of data into a host signal (e.g. an image). This allows communication using often enciphered messages without attracting the attention of the third party. Typically, robustness requirements are low for steganographic purposes; instead invisibility and capacity are of prime importance. Image labeling is an application where information about the image content is encoded as a watermark and inserted into the image to assist image retrieval from a database or provide extra information to the viewer.

1.6 Aspects & Requirements Of Watermarking:

- ✓ It should be difficult to insert a false watermark and the watermarking scheme should be able to indicate regions where alterations in the image have taken place.
- ✓ It should be possible to generate large number of the watermarks and the insertion of multiple watermarks should be handled properly.
- ✓ For data hiding and image labeling purposes, the maximum capacity of embedded message is of prime importance. Image labeling techniques require highly localized embedding of watermark information which rules out methods that operate on the entire image.
- ✓ As for cryptography, watermarking methods have to obey the Kerckhoff principle which means that security and robustness claims have to take into consideration that the algorithms for watermarking embedding and extraction are known in detail.
- ✓ The message capacity, that is the number of bits that can be reliably embedded in the image data, is fairly limited. For copyright protection applications that involve identification of the copyright holder as well as the identification of the licensee of the image, different lower capacity bounds have been proposed.

1.7 Report Structure

The dissertation started with a general introduction about background, history and status quo of digital watermarking technique. The second part of the report describes the concepts and basic features of watermarking. In the third part of the report is a literature survey that expounded the essential knowledge about the wavelet transform that will be utilized to define an approach and foundation to implementation. The fourth part gives the detailed description of the possible watermarking attacks and the use of human visual system in watermarking. The fifth part is the main body of the dissertation which contains watermarking technique study implementation and algorithm design. Thereafter, the sixth part described watermark attacks and result analysis, which evaluated and analyzed not only individual aspect but also overall performance after-attacked watermark results. The final part concluded the study and further work.

CHAPTER # 2

WATERMARKING CONCEPTS

2.1 Watermarking Process

In general, any watermarking scheme (algorithm) consists of three parts:

- ✓ The watermark
- ✓ The encoder (marking insertion algorithm)
- ✓ The decoder and comparator (verification or extraction or detection algorithm).

Each owner has a unique watermark or an owner can also put different watermarks in different objects the marking algorithm incorporates the watermark into the object. The verification algorithm authenticates the object determining both the owner and the integrity of the object.

WATERMARK ENCODING PROCESS: The Fig. 2.1 illustrates the encoding process.

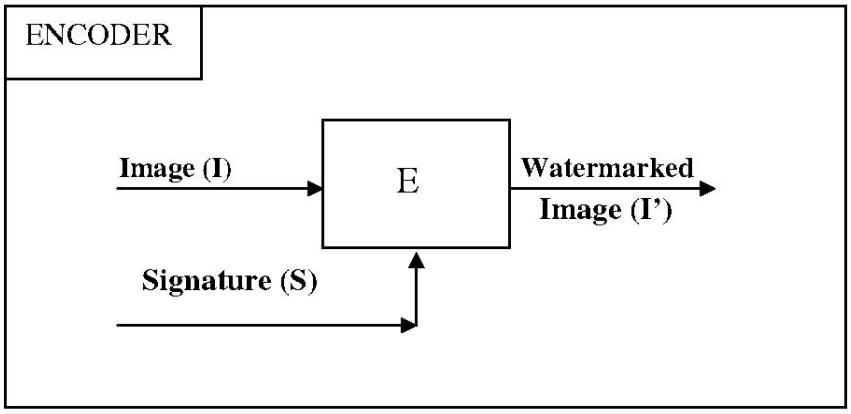


Fig. 1.1 Encoder

Let us denote an image by I, a signature by S = {s1, s2 …} the watermarked image by I'. E is an encoder function, it takes an image I and a signature S, and it generates a new image which is called watermarked image I', i.e.

$$E(I, S) = I' \qquad \ldots\ldots\ldots\ldots (2.1)$$

It should be noted that the signature S may be dependent on image I. In such cases, the encoding process described by (2.1) still holds.

WATERMARK DECODING PROCESS: The fig 2.2 illustrates the decoding process.

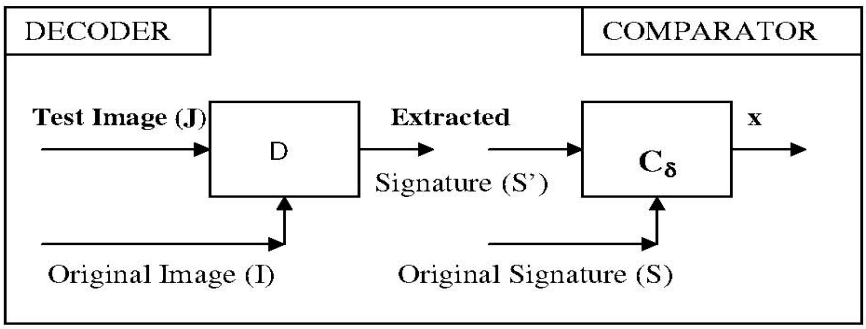


Fig. 1.2 Decoder

A decoder function D takes an image J (J can be a watermarked or un-watermarked image, and possibly corrupted) whose ownership is to be determined and recovers a signature S' from the image. In this process, an additional image I can also be included which is often the original and un-watermarked version of J. This is due to the fact that some encoding schemes may make use of the original images in the watermarking process to provide extra robustness against intentional and unintentional corruption of pixels.

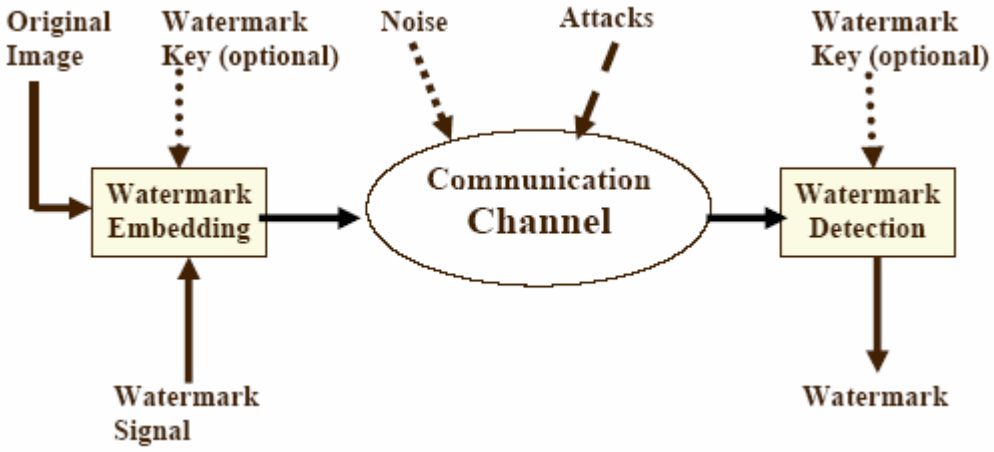


Figure 2.3: A Common Digital Watermarking System

2.2 Types of Watermark

- ✓ **FRAGILE WATERMARK:** Fragile watermarks do not survive lossy transformations to the original host signal and their purpose is tamper detection of the original signal. Placing the watermark information into the perceptually insignificant portions of the data guarantees imperceptibility and provides fragile

marking capabilities. For instance, early watermark techniques for still image data propose inserting watermark information into the least significant bits of the pixel values.

- ✓ **SEMI-FRAGILE WATERMARK:** A semi-fragile watermark is a mark which is (highly) sensitive to a modification of the stego-medium. A fragile watermarking scheme should be able to detect any change in the signal and identify where it has taken place and possibly what the signal was before modification. It serves at proving the authenticity of a document.
- ✓ **ROBUST WATERMARK:** A robust watermark should be stuck to the document it has been embedded in, in such a way that any signal transform of reasonable strength cannot remove the watermark. Hence a pirate willing to remove the watermark will not succeed unless they debase the document too much to be of commercial interest.

Ideally, an effective, robust watermarking scheme provides a mark that can only be removed when the original content is destroyed as well. The degree of robustness and distortion necessary to alter the value of the original content can vary for different applications. Typically, many of the applications for copyright protection involve relatively high quality original content and the imperceptibility criterion is critical for such applications. In order for a watermarking technique to be robust, the watermark should be embedded in the perceptually significant portion of the data.

Some typical distortions or attacks that digital watermarking schemes are expected to survive include re-sampling, rescaling, compression, linear and nonlinear filtering, additive noise, A/D and D/A conversion, and transcoding. Applications for robust watermarking include copyright protection where each copy gets a unique watermark (commonly referred to as a fingerprint) to identify the end-user so that tracing is possible for cases of illegal use; authentication, where the watermark can represent a signature and copy control for digital recording devices. Within the class of robust watermarking techniques there are several different constraints on encoder and decoder design which depends on the particular application.

Semi-fragile watermarking techniques differentiate between lossy transformations that are “information preserving” and lossy transformations which are “information altering.”

Lossy transformations include any signal processing step that alters the original signal values and is not invertible. For example, in authentication applications it may be desirable to have a watermark that can distinguish between a lossy transformation such as compression which does not alter the integrity of the content and an alteration which does alter the integrity, such as manipulating or replacing objects within the scene.

There should be no perceptible difference between the watermarked and original signal, and the watermark should be difficult to remove or alter without damaging the host signal.

2.3 Classes Of Watermarking

- ✓ **PUBLIC OR BLIND WATERMARKING**: In these schemes, the cover that is the original signal is not needed during the detection process to detect the mark.. Solely the key, which is typically used to generate some random sequence used during the embedding process, is required

- ✓ **SEMI-BLIND WATERMARKING**: In some cases you need extra information to help your detector (in particular to synchronize its random sequence on the possibly distorted test signal). In particular some watermarking schemes require access to the 'published' watermarked signal, that is the original signal just after adding the watermark.

- ✓ **PRIVATE OR NON-BLIND WATERMARKING**: The original cover signal is required during the detection process.

- ✓ **ASYMMETRIC OR PUBLIC-KEY WATERMARKING**: In this case, the detection process (and in particular the detection key) is fully known to anyone as opposed to blind watermarking where a secret key is required. So here, only a 'public key' is needed for verification and a 'private key' (secret) is used for the embedding though. Knowledge of the public key does not help to compute the private key (at least in a reasonable time), it does not either allow removal of the mark nor it allows an attacker to forge a mark.

2.4 Applications of Watermarking

- ✓ **Give your images the power of personalization and protection:** Using this you can add a layer of protection to your images by identifying copyright ownership and delivering a tracking capability that monitors and reports where your images are being used. You can protect your images by tracking them beyond your own domain. This will increase control over your assets by tracking and reporting on their authorized and unauthorized use.
- ✓ **Limit unlicensed use:** Using this you can limit unlicensed use and receive information to help recover otherwise lost revenue.
- ✓ **For digital audio and video:** Similar to the process in which artist artistically signed their paintings with a brush to claim their copyrights; artists of today can watermark their work and hide for example their name in the image. Hence, the embedded watermark will allow identifying the owner of the work. It is clear that this concept is also applicable to other media such as digital video and audio. Especially the distribution of digital audio over the Internet in the MP3 format is currently a big problem. In this scenario digital watermarking may be useful to set up a controlled audio distribution and provide efficient means for copyright protection.
- ✓ **Certification:** For example, in the field of data security, watermarks may be used for certification, authentication, and conditional access. Certification is an important issue for official documents, such as identity cards or passports.

- ✓ **To mutually linking information on the documents:** That means that some information is written twice on the document: for instance, the name of a passport

owner is normally printed in clear text and is also hidden as an invisible watermark in the photo of the owner. If anyone would intend to counterfeit the passport by replacing the photo, it would be possible to detect the change by scanning the passport and verifying the name hidden in the photo does not match any more the name printed on the passport.

- ✓ **The authentication of image content:** The goal of this application is to detect alterations and modifications in an image. The three pictures below illustrate an example of this application. The picture on the left shows an original photo of a car that has been protected with a watermarking technology. In the center, the same picture is shown but with a small modification: the numbers on the license plate have been changed. The picture on the right shows the photo after running the watermark detection program on the tampered photo. The tampered areas are indicated in white and we can clearly see that the detected areas correspond to the modifications applied to the original photo.

(a) (b) (c)

- ✓ **Conditional access and copy-control:** For example conditional access to confidential data on CD-ROMs may be provided using digital watermarking technology. The concept consists of inserting a watermark into the CD label. In order to read and decrypt the data stored on the CD, the watermark has to be read since it contains information needed for decryption. If someone copies the CD, he will not be able to read the data in clear-text since he does not have the required watermark. The picture below shows an example of a protected CD. To read the data on the CD, the user starts a program on the CD. This program asks the user to put the CD on the scanner and then reads the watermark. If the watermark is valid the program decrypts the data on the CD and gives the user access the clear-text data. (Patent pending, contact us for license.)

- ✓ **Copy-control:** Several companies work on a watermarking system for copy control in the DVD environment. Fully functioning solutions exist already, however, for the moment they have not been entirely approved by the content producers and providers. Finally, this solution is also an efficient and simple way to prevent the use of illegal copies of software. It has a similar functionally as the anti-piracy device called "dongle", but is more compact and less expensive.
- ✓ **As invisible labels and content links:** For example, photo development laboratories may insert a watermark into the picture to link the print to its negative. This way is very simple to find the negative for a given print. All one has to do is scan the print and extracted the information about the negative. In a completely different scenario digital watermarks may be used as a geometrical reference which may be useful for programs such as optical character recognition (OCR) software. The embedded calibration watermark may improve the detection reliability of the OCR software since it allows the determination of translation, rotation, and scaling.

2.5 Characteristic Features of Watermarks

An invisible watermark for copyright protection should be:

- ✓ **Hidden or Imperceptible:** The insertion of the watermark should not degrade the host signal. Also, the data embedding process should not introduce any perceptible aircrafts into host data. This goal is in conflict with the next two.
- ✓ **Robust:** The watermark should resist manipulations which might occur in legitimate use: filtering, lossy compression, cropping, printing and scanning, conversion to a different data format. Cox and Miller have argued that the watermark must be placed in perceptually significant regions of the host signal to resist lossy compression.
- ✓ **Tamper resistant:** The watermark should resist attempts to remove it. This is not an absolute requirement; rather it is linked to the level of degradation of the host signal. A “brute force “attack which destroys the host signal might well remove the watermark.

- ✓ **Secure:** The watermarked image should not reveal any clues of the presence of the watermark, with respect to un-authorized detection, or (statistical) undetectability or unsuspicious (not the same as imperceptibility).
- ✓ **Public:** The method of watermarking should be known to the general public. Like in cryptography "security through obscurity" is not a valid concept. By keeping a watermarking method secret you remove it from the peer reviewing process and thus make it less secure.
- ✓ **Multiple watermarks:** It should be possible to insert multiple watermarks. The watermarks of all the originals used and the watermark of the creator of the collage should still be detectable.
- ✓ **Scalable:** It should be possible to use better versions of the same technique when more computing power becomes available. This corresponds to the use of bigger keys in cryptographic algorithms. On the other hand the watermark should still be tamper resistant in spite of more computing power.
- ✓ **Self-clocking / arbitrary re-entrant:** If only fragments of the host signal are available - , e.g. after cropping or rotating a picture – the watermark can still be recovered.
- ✓ **Resistance to collusion attack:** If several images marked with different watermarks are "averaged" the result should still be watermarked. This feature is needed in two situations: In fingerprinting, where the same image is watermarked differently for different customers, and in the watermarking of videos, where several similar frames could be averaged.

The watermark could be:

- ✓ A chosen string of bits,
- ✓ An image,
- ✓ A sequence of floating point numbers with certain properties.

2.6 Uses of Digital Watermarking

The list given here is by no means complete and intends to give a perspective of the broad range of possibilities that digital watermarking opens.

✓ *Image Watermarking*

Many techniques have been developed for the watermarking of still image data. For grey-level or color-image watermarking, watermark embedding techniques are designed to insert the watermark directly into the original image data, such as the luminance or color components or into some transformed version of the original data to take advantage of perceptual properties or robustness to particular signal manipulations. Requirements for image watermarking include imperceptibility, robustness to common signal processing operations, and capacity. Common signal processing operations which the watermark should survive include compression (such as JPEG), filtering, rescaling, cropping, A/D and D/A conversion, geometric distortions, and additive noise. Capacity refers to the amount of information (or payload) that can be hidden in the host image and detected reliably under normal operating conditions. The watermark may be scaled appropriately to minimize noticeable distortions to the host. Some examples of watermark information include a binary sequence representing a serial number or credit card number, a logo, a picture, or a signature.

For still image watermarking, watermark embedding is applied directly to the pixel values in the spatial domain or to transform coefficients in a transform domain such as the discrete cosine transform (DCT) or discrete wavelet transform (DWT). Watermark detection usually consists of some preprocessing step (which may include removal of the original host signal if it is available for detection) followed by a correlation operator.

✓ *Video Watermarking*

In this case most considerations made in previous sections hold. However, now the temporal axis can be exploited to increase the redundancy of the watermark. As in the still images case, watermarks can be created either in the spatial or in the DCT domains. In the latter, the results can be directly extrapolated to MPEG-2 sequences, although different actions must be taken for I, P and B frames. Note that perhaps the set of attacks that can be performed intentionally is not smaller but definitely more expensive than for still images.

✓ *Audio Watermarking*

Again, previous considerations are valid. In this case, time and frequency masking properties of the human ear are used to conceal the watermark and make it inaudible. The greatest difficulty lies in synchronizing the watermark and the watermarked audio file, but techniques that overcome this problem have been proposed.

✓ *Hardware/Software Watermarking*

This is a good paradigm that allows us to understand how almost every kind of data can be copyright protected. If one is able to find two different ways of expressing the same information, then one bit of information can be concealed, something that can be easily generalized to any number of bits. This is why it is generally said that a perfect compression scheme does not leave room for watermarking. In the hardware context, Boolean equivalences can be exploited to yield instances that use different types of gates and that can be addressed by the hidden information bits. Software can be also protected not only by finding equivalences between instructions, variable names, or memory addresses, but also by altering the order of non-critical instructions. All this can be accomplished at compiler level.

✓ *Text Watermarking*

This problem, which in fact was one of the first that was studied within the information hiding area, can be solved at two levels. At the printout level, information can be encoded in the way the text lines or words are separated (this facilitates the survival of the watermark even to photocopying). At the semantic level (necessary when raw text files are provided), equivalences between words or expressions can be used, although special care has to be taken not to destruct the possible intention of the author.

✓ *Fingerprinting*

This is similar to the previous application and allows acquisition devices (such as video cameras, audio recorders, etc) to insert information about the specific device (e.g., an ID number) and date of creation. This can also be done with conventional digital signature

techniques but with watermarking it becomes considerably more difficult to excise or alter the signature. Some digital cameras already include this feature.

✓ *Authentication*

This is a variant of the previous application, in an area where cryptographic techniques have already made their way. However, there are two significant benefits that arise from using watermarking: first, as in the previous case, the signature becomes embedded in the message, second, it is possible to create 'soft authentication' algorithms that offer a multi-valued 'perceptual closeness' measure that accounts for different unintentional transformations that the data may have suffered (an example is image compression with different levels), instead of the classical yes/no answer given by cryptography-based authentication. Unfortunately, the major drawback of watermarking-based authentication is the lack of public key algorithms that force either to put secret keys in risk or to resort to trusted parties.

✓ *Copy and Playback Control*

The message carried by the watermark may also contain information regarding copy and display permissions. Then, a secure module can be added in copy or playback equipment to automatically extract this permission information and block further processing if required. In order to be effective, this protection approach requires agreements between content providers and consumer electronics manufacturers to introduce compliant watermark detectors in their video players and recorders. This approach is being taken in Digital Video Disc (DVD).

2.7 Key Points to Remember

- ✓ **Watermark detection** should also be possible in case small modifications have been applied to the marked media. Such modifications can be the result of intentional attacks in order to remove the mark or the result of coding schemes (e.g. lossy compression that is compression where there is some loss of quality) and errors during the transmission. A robust watermarking scheme will be able to retrieve the watermark from this distorted media.

- ✓ **Watermark is embedded in only luminance components rather then in chroma components as well in image/video:** It has more to do with the survivability of the marked areas within an image. Color can easily be changed or converted to grayscale and you still have a "useable" image. In marking an image, one wants to place the mark in the more robust areas of an image. Areas of high luminance are not the correct assessment, because a plain sky may have high luminance but a poor structure for hiding information. What the watermark tools are really interested in are areas with high gradient magnitude. In other words, relatively strong edges with respect to the structure of the image and the luminance variances of the "edges."

  A lot of watermarking schemes hide data in the luminance/intensity due to the fact that the Human Visual System (HVS) use most of its bandwidth on percepting (changes in) brightness. In changing an image, by e.g. JPEG compression, one therefore has to be more gentle to the brightness information than to the color information (hue/saturation) since small changes in lightness might be easily detectable than large changes in color. If the compression changes the brightness in an image, this will give the outcome a poor quality to the HVS, and that is why these changes are avoided. For the watermark to be robust to e.g. compression, the watermark has to be in parts of the image that will not be changed in the compression. That is a reason why hiding data in the Luminance is a good idea.

  In 1997 it was suggested to use the blue channel to embed a spread spectrum based watermark into an image. The blue channel was used because the HVS is less sensitive to blue colors due to the fact that the blue cones (S-cones) are less densely distributed than the green and red cones (M-, L-cones) in the foveal part of the human retina. Since then, we made numerous subjective tests and found that in average the energy of a blue channel watermark is up to 50 times larger than the energy of a luminance watermark, of course both introducing visually equivalent artifacts. This implies that the blue channel watermark is more robust towards attacks such as filtering (averaging, median ...) and additive noise. Furthermore, we found that under lossy JPEG compression both approaches are approximately equivalent. However, one problem that goes with blue channel watermarks is that it is more difficult to control, or predict, the artifacts. That is, the visibility of a luminance watermark is more homogeneous and less dependent on the image colors. Therefore, the design of blue (or any other color) channel watermarks

is more delicate and requires sophisticated models of the HVS to optimally adapt the watermark to the local contrast, intensity, and color. For instance, the attacker takes a coarse estimate of the power density spectrum of an image (very coarse: low pass characteristic), designs the Wiener filter accordingly, and perhaps can remove at least some of the watermark components (e.g., high pass watermark components).Note that the theoretical analysis described above confirms in an analytical fashion the heuristic argument given very early by Cox et al.

- ✓ **The watermark should be embedded into the most significant data components:** Therefore, you should be very careful when designing your watermark based on psycho-acoustic or psycho-visual masking effects. If you put your watermark underneath a masking threshold, an attacker can remove it without any penalty. This approach is not the right one for very robust watermarks. Nevertheless, masking might be appropriate when embedding information just as added value (in this scenario we do not have a malicious attacker). Note that any state- of-the-art compression scheme (for audio and images) will significantly impair the watermark underneath the masking threshold.
- ✓ The theoretical analysis also gives you an idea about the **maximum information that can be embedded per pixel.** Assume that a mean-squared error distortion measurement is used. Further, let the attacker add simple additive white Gaussian noise (AWGN). In this case, Shannon's result for the capacity of an AWGN channel gives the upper limit on the achievable watermark rate, e.g. 0.5 bit/sample if the variance of the AWGN equals the embedding distortion. Everybody can play this attack! Thus, you never can achieve higher rates. Of course, more sophisticated attacks can be invented. Thus, in practice the achievable watermark rate will be much lower. The goal of current research efforts is to tighten this bound. Of course, tights bounds can be obtained only when optimizing the watermarking scheme and the attack for certain signals statistics. An "all-white" image has less (exactly zero) watermark capacity than a Gaussian-noise image.

CHAPTER # 3

LITERATURE REVIEW

3.1 Transforms

First of all, why do we need a transform? Mathematical transformations are applied to signals to obtain further information from that signal that is not readily available in the raw signal. Most of the signals in practice are time-domain signals in their raw format.
Time domain representation is not always the best representation of the signal for most signal processing related applications. In many cases, the most distinguished information is hidden in the frequency content of the signal. The information that cannot be readily seen in the time-domain can be seen in the frequency domain.

Fourier Transform (FT) with its fast algorithms (FFT) is an important tool for analysis and processing of many natural signals. FT has certain limitations to characterize many natural signals, which are non-stationary (e.g. speech). Though a time varying, overlapping window based FT namely STFT (Short Time FT) is well known for speech processing applications, a new time-scale based Wavelet Transform (WT) is a powerful mathematical tool for non-stationary signals.

WT uses a set of damped oscillating functions known as wavelet basis. WT in its continuous (analog) form is represented as CoWT. CoWT with various deterministic or non-deterministic basis is a more effective representation of signals for analysis as well as characterization. Continuous wavelet transform (CoWT) is powerful in singularity detection. A discrete and fast implementation of CoWT (generally with real valued basis) is known as the standard DWT (Discrete Wavelet Transform).

With standard DWT, signal has a same data size in transform domain and therefore it is a non-redundant transform. Standard DWT can be implemented through a simple filter-bank structure of recursive FIR filters. A very important property; Multiresolution Analysis (MRA) allows DWT to view and process different signals at various resolution levels. The advantages such as non-redundancy, fast and simple implementation with digital filters using micro-computers, and MRA capability popularized the DWT in many signal processing applications since last decade. Many researches have successfully applied and proved the advantages of DWT for signal denoising and compression in a number of diverse fields.

3.2 Introduction to Wavelet

3.2.1 *Wavelet Definition*

A ‘wavelet’ is a small wave which has its energy concentrated in time. It has an oscillating wavelike characteristic but also has the ability to allow simultaneous time and frequency analysis and it is a suitable tool for transient, non-stationary or time-varying phenomena [1, 2].

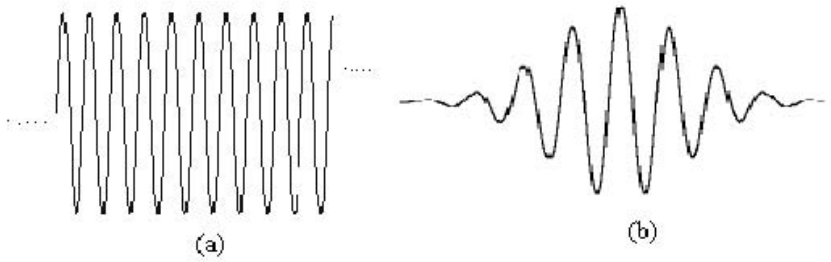


Figure 3.1: Representation of a wave (a), and a wavelet (b)

3.2.2 *Wavelet Characteristics*

The difference between wave (sinusoids) and wavelet is shown in figure (3.1). Waves are smooth, predictable and everlasting, whereas wavelets are of limited duration, irregular and may be asymmetric. Waves are used as deterministic basis functions in Fourier analysis for the expansion of functions (signals), which are time-invariant, or stationary. The important characteristic of wavelets is that they can serve as deterministic or non-deterministic basis for generation and analysis of the most natural signals to provide better time-frequency representation, which is not possible with waves using conventional Fourier analysis.

3.2.3 *Wavelet Analysis*

The wavelet analysis procedure is to adopt a wavelet prototype function, called an ‘analyzing wavelet’ or ‘mother wavelet’. Temporal analysis is performed with a contracted, high frequency version of the prototype wavelet, while frequency analysis is

performed with a dilated, low frequency version of the same wavelet. Mathematical formulation of signal expansion using wavelets gives Wavelet Transform (WT) pair, which is analogous to the Fourier Transform (FT) pair. Discrete-time and discrete-parameter version of WT is termed as Discrete Wavelet Transform (DWT). DWT can be viewed in a similar framework of Discrete Fourier Transform (DFT) with its efficient implementation through fast filterbank algorithms similar to Fast Fourier Transform (FFT) algorithms.

3.3 Evolution of Wavelet Transform

The need of simultaneous representation and localization of both time and frequency for non-stationary signals (e.g. music, speech, images) led toward the evolution of wavelet transform from the popular Fourier transform. Different 'time-frequency representations' (TFR) are very informative in understanding and modeling of WT.

3.4.1 *Fourier Transform*

Fourier transform (FT) is used to find the frequency content of a signal. It allows going back and forwarding between the raw and processed (transformed) signals. However, only either of them is available at any given time. That is, no frequency information is available in the time-domain signal, and no time information is available in the Fourier transformed signal. Fourier transform of a time domain signal x (t) and inverse Fourier transform (IFT) of a frequency domain signal X (f) are given below.

$$X(f) = \int_{-\infty}^{\infty} x(t) \bullet e^{-2j\Pi ft}\, dt \qquad (3.1)$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \bullet e^{2j\Pi ft}\, dt \qquad (3.2)$$

Though FT has a great ability to capture signal's frequency content as long as x(t) is composed of few stationary components (e.g. sine waves). However, any abrupt change in time for non-stationary signal x(t) is spread out over the whole frequency axis in X(f). Hence the time-domain signal sampled with Dirac-delta function is highly localized in

time but spills over entire frequency band and vice versa. The limitation of FT is that it cannot offer both time and frequency localization of a signal at the same time. To overcome the limitations of the standard FT, Gabor introduced the initial concept of Short Time Fourier Transform (STFT).

3.4.2 *Short Term Fourier analysis*

This is the revised version of Fourier transform. There is only a minor difference between Short term Fourier analysis (STFT) and FT. In STFT, the signal is divided into small enough segments, where these segments (portions) of the signal can be assumed to be stationary. For this purpose, a window function "w" is chosen. The width of this window must be equal to the segment of the signal where its stationarity is valid.

This window function is first located to the very beginning of the signal. That is, the window function is located at t=0. Let's suppose that the width of the window is "T" s. At this time instant (t=0), the window function will overlap with the first T/2 seconds. The window function and the signal are then multiplied. By doing this, only the first T/2 seconds of the signal is being chosen, with the appropriate weighting of the window (if the window is a rectangle, with amplitude "1", then the product will be equal to the signal). Assuming the product just as another signal, FT is taken.

The result of this transformation is the FT of the first T/2 seconds of the signal. If this portion of the signal is stationary, as it is assumed, the obtained result will be a true frequency representation of the first T/2 seconds of the signal. The next step would be shifting this window (for some t1 seconds) to a new location, multiplying with the signal, and taking the FT of the product. This procedure is followed; until the end of the signal is reached by shifting the window with "t1" seconds intervals. The following definition of the STFT summarizes all the above explanations in one line:

$$STFT_X^{\omega}(t,f) = \int_t [x(t) \bullet \omega^*(t - t^1)] \bullet e^{-j2\Pi ft} dt \qquad (3.3)$$

In the above equation x (t) is the signal, w (t) is the window function, and * is the complex conjugate. As you can see from the equation, the STFT of the signal is nothing but the FT of the signal multiplied by a window function.

Using STFT one cannot know the exact time-frequency representation of a signal, i.e., one cannot know what spectral components exist at what instances of times. What one can know are the time intervals in which certain band of frequencies exists, which is a resolution problem. This problem occurs because of width of window function used.

Narrow window ===> good time resolution, poor frequency resolution.
Wide window ===> good frequency resolution, poor time resolution and violates the condition of stationarity.

The selection of proper window is application dependent. Once a window has been chosen for STFT, the time-frequency resolution is fixed over the entire time-frequency plane because the same window is used at all frequencies. There is always a trade off between time resolution and frequency resolution in STFT.

3.4.3 *Continuous Wavelet Transform*

The continuous wavelet transform was developed as alternative approach to the short time Fourier transforms to overcome the resolution problem. The wavelet analysis is done in a similar way to the STFT analysis, in the sense that the signal is multiplied with a function, {i.e. the wavelet}, similar to the window function in the STFT, and the transform is computed separately for different segments of the time-domain signal. However, there are two main differences between the STFT and the CWT:

1. The Fourier transforms of the windowed signals are not taken, and therefore single peak will be seen corresponding to a sinusoid, i.e., negative frequencies are not computed.
2. The width of the window is changed as the transform is computed for every single spectral component, which is probably the most significant characteristic of the wavelet transform.

The Wavelet Transform (WT) in its continuous (CWT) form provides a flexible time-frequency window, which narrows when observing high frequency phenomena and widens when analyzing low frequency behavior. Thus time resolution becomes arbitrarily good at high frequencies, while the frequency resolution becomes arbitrarily good at low frequencies. This kind of analysis is suitable for signals composed of high frequency

components with short duration and low frequency components with long duration, which is often the case in practical situations [11].

The continuous wavelet transform is defined as follows

$$CWT_x^{\varphi}(\tau, s) = \Psi_x^{\varphi}(\tau, s) = \frac{1}{\sqrt{s}} \int x(t) \varphi^*(\frac{t-\tau}{s}) dt \qquad (3.4)$$

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

The mother wavelet is a prototype for generating the other window functions. The term translation is related to the location of window, as the window is shifted through the signal. This term corresponds to the time information in transform. The scale parameter is defined as the inverse of frequency. High scales (low frequencies) correspond to global information of a signal (that usually spans the entire signal), whereas low scales (high frequencies) correspond to detailed information of a hidden pattern in the signal (that usually lasts a relatively short time). In practical applications low scales (high frequencies) do not last for entire duration of signal but usually appear from time to time as short bursts and high scales (low frequencies) usually last for the entire duration of the signal.

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.

3.4.4 *Discrete Wavelet Transform*

The CWT has the drawbacks of redundancy and impracticability with digital computers. The discrete wavelet transform (DWT) provides sufficient information both for analysis and synthesis of the original signal, with a significant reduction in the computation time. The DWT is considerably easier to implement when compared to the CWT.

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 lowpass and highpass filters, respectively. The original signal x [n] is first passed through a half-band 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. 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:

$$y_{high}[n] = \Sigma x[k].\, g[2n-k] \qquad (3.5)$$

$$y_{low}[n] = \Sigma x[k].\, h[2n-k] \qquad (3.6)$$

$y_{high}[k]$ 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 sub-band 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). Fig.3.2 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".

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 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. The difference of this transform from the Fourier transform is that the time localization of these frequencies will not be lost.

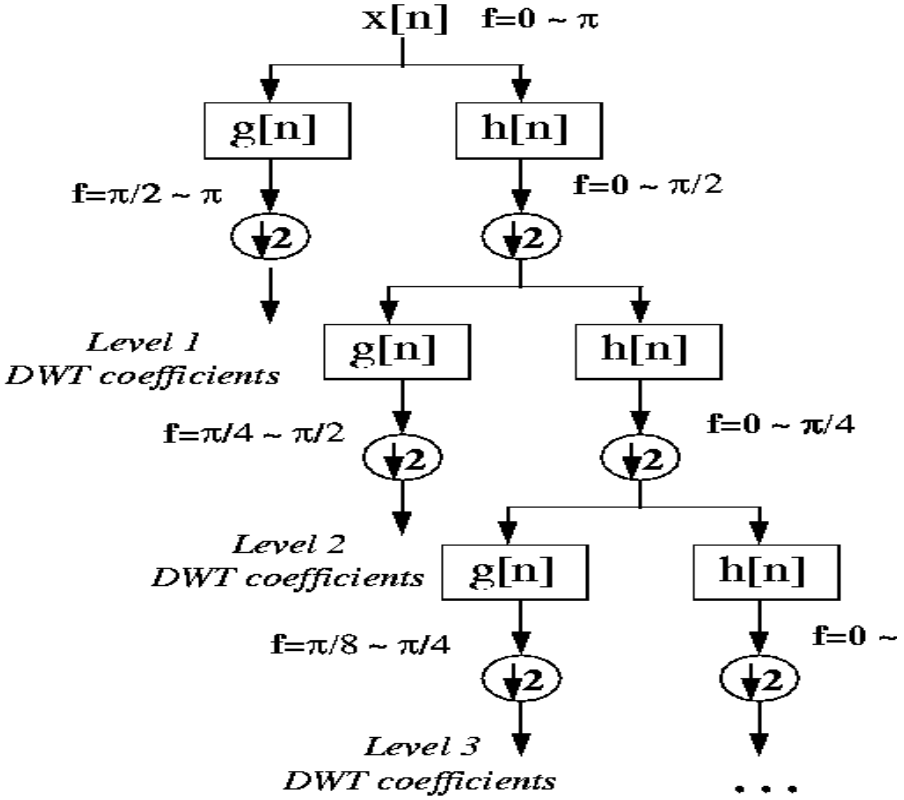
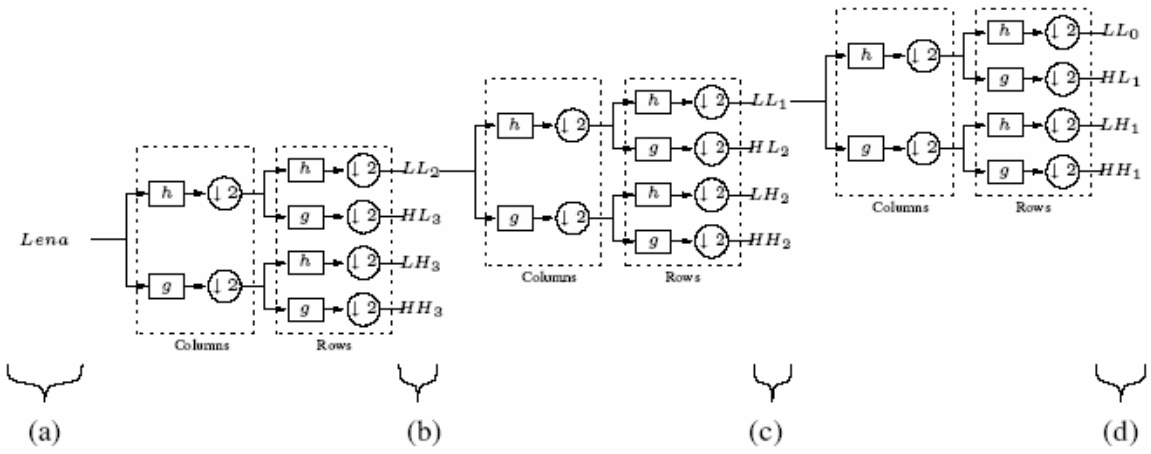


Fig 3.2 DWT Coefficients at different levels

Four resulting sets of wavelet coefficients W_{LL}, W_{HL}, W_{LH} and W_{HH} are conventionally named according to the filtering types along rows and columns respectively (H: high-pass filtering, L: low-pass filtering). These sets are also called wavelet subbands (LL, HL, LH and HH). The perfect reconstruction is also obtained by applying the 1D synthesis scheme on rows and columns successively.



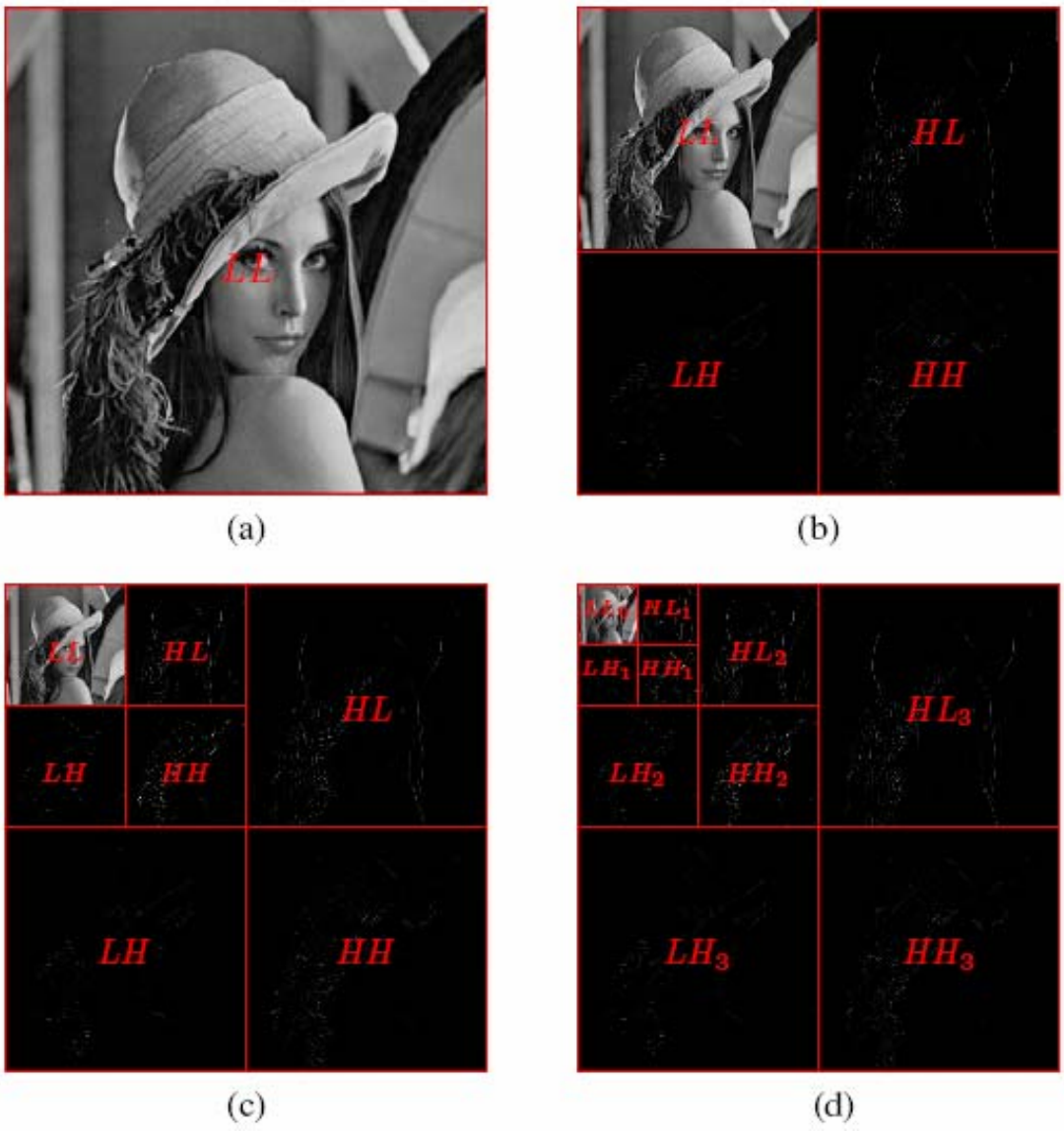


Fig 3.3: Wavelet subbands and resolution levels. (a) Original Lena image (b) First wavelet decomposition levels (c) Second wavelet decomposition levels (d) Third wavelet decomposition levels. Subbands subscripts correspond to resolution level index.

It is worth pointing out that the order in which rows and columns are processed at the analysis and synthesis sides has no importance since the global transformation is linear.

An advantage of wavelet transforms is that the windows vary. In order to isolate signal discontinuities, one would like to have some very short basis functions. At the same time, in order to obtain detailed frequency analysis, one would like to have some very long basis functions. A way to achieve this is to have short high-frequency basis functions and long low-frequency ones. This happy medium is exactly what you get with wavelet transforms.

One thing to remember is that wavelet transforms do not have a single set of basis functions like the Fourier transform, which utilizes just the sine and cosine functions. Instead, wavelet transforms have an infinite set of possible basis functions. Thus wavelet analysis provides immediate access to information that can be obscured by other time-frequency methods such as Fourier analysis.

3.4.5 *Comparative Visualization*

A comprehensive visualization of various time-frequency representations, shown in figure (3.4), demonstrates the time-frequency resolution for a given signal in various transform domains with their corresponding basis functions.

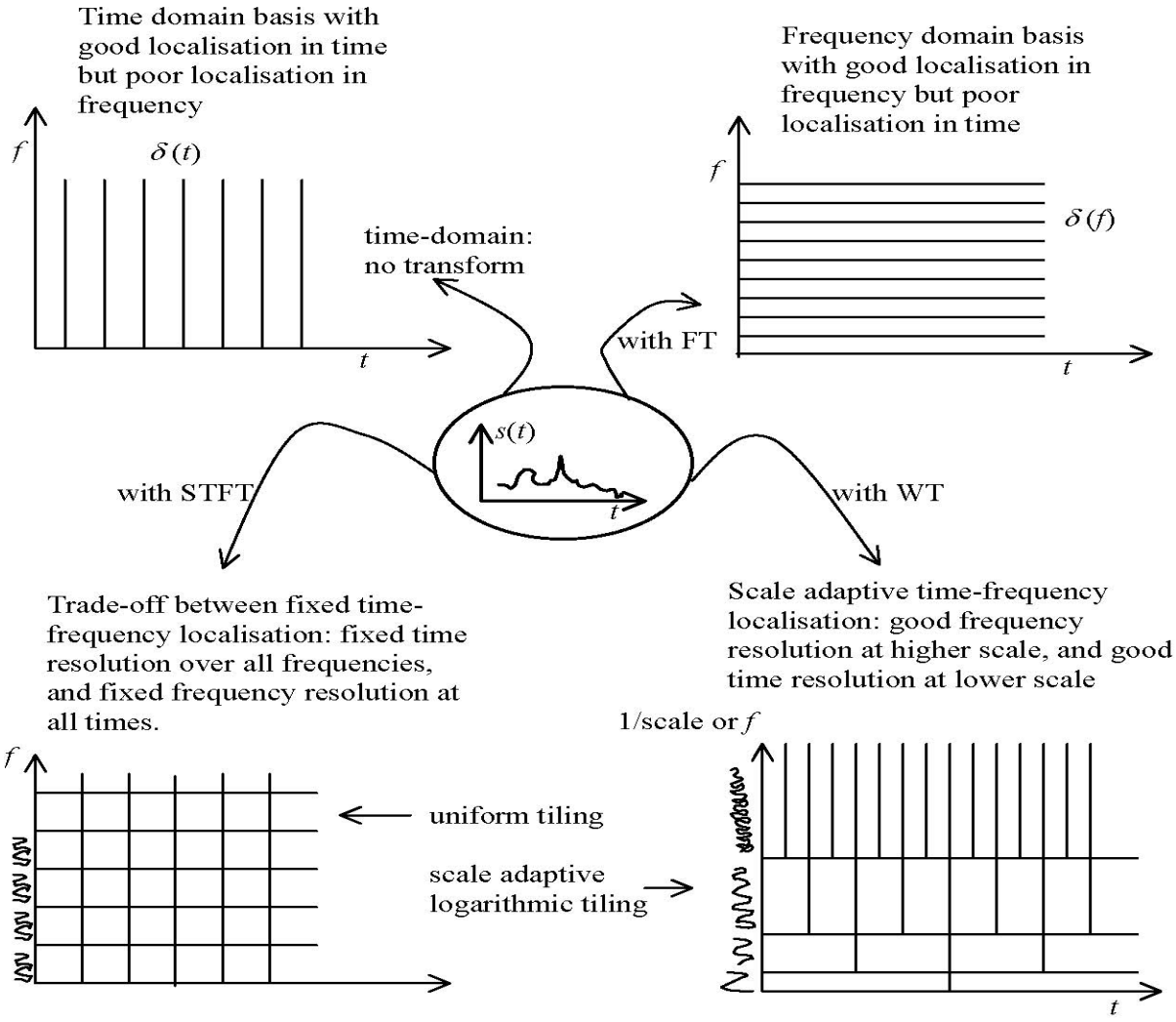


Figure 3.4: Comparative visualization of time-frequency representation of an arbitrary non-stationary signal in various transform domains

3.4 Implementation of DWT

The practical usefulness of DWT comes from its Multi-Resolution Analysis (MRA) ability [48-50], and efficient Perfect Reconstruction (PR) filterbank structures.

3.4.1 *Multiresolution Analysis (MRA)*

Multiresolution analysis (or Multiscale analysis) consists of a sequence of embedded subspaces …$V_2 \subset V1 \subset V_0 \subset V_{-1} \subset V_{-2}$ ….. of $L^2(\Re)$ as shown in figure.

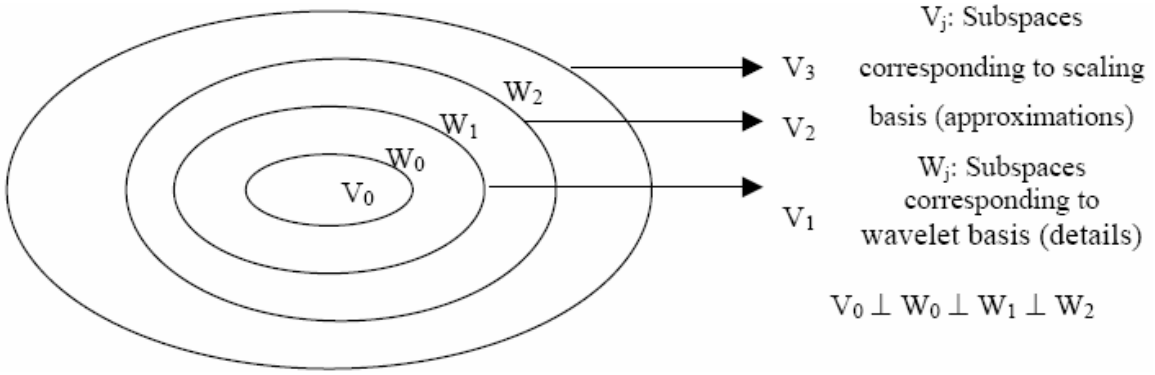


Fig 3.5: Nested vector space spanned by scaling and wavelet basis

The MRA follows the following conditions:

1. $V_j \subset V_{j+1}$ …….$j \in Z$
2. $V_{-\infty} = \{0\}$ and $V_{\infty} = L^2$
3. $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$
4. $V_2 = V_0 + W_0 + W_1$
5. $L^2 = \ldots + W_{-2} + W_{-1} + W_0 + W_1 + W_2 + \ldots = V_0 + W_1 + W_2 + \ldots\ldots$
6. $W_{-\infty} + \ldots + W_{-2} + W_{-1} = V_0$ ..... (3.7)

A scaling function φ(t) (Father Wavelet) is introduced such that for each fixed j, the family

$$\varphi_{j,k} = 2^{-j/2}\,\varphi\,(2^{-j/2}t - k),\ (j, k \in Z) \text{ and } \int \varphi(t)\,dt = 1 \quad \ldots\ldots\ (3.8)$$

is an orthonormal basis of the subspace Vj.

If Wj is orthonormal component of Vj (Wj $\perp$ Vj) in subspace Vj+1, then there exist a function ψ (t) (Mother wavelet) such that for each fixed j the family

$$\psi_{j,k} = 2^{-j/2}\,\psi\,(2^{-j/2}t - k)\,,\ (\ j, k \in \mathrm{Z}\ ) \qquad \ldots\ldots\ldots\ldots (3.9)$$

is an orthonormal basis of the subspace Wj.

Because of the nested subspaces and MRA condition (3), the scaling function satisfies the following 2-scale (dilation or refinement) equation,

$$\varphi(t) = \sqrt{2}\sum_{n=-\infty}^{\infty} h_0[n]\,\varphi(2t-n),\ n \in \mathrm{Z} \qquad \ldots\ldots\ldots\ldots.. (3.10)$$

where it satisfies the admissibility condition $\sum_n h_o[n] = \sqrt{2}$.

The wavelet function satisfies similar equation,

$$\psi(t) = \sqrt{2}\sum_{n=-\infty}^{\infty} h_1[n]\,\varphi(2t-n)\,,\ n \in \mathrm{Z} \qquad \ldots\ldots\ldots\ldots.. (3.11)$$

with the conditions $\sum_n h_1[n] = 0$ and $h_1[n] = (-1)^n\, h_0[-n+1]$.

3.5 Applications of Wavelet Transforms

Finally, applications of widely used standard DWT implementations, utilizing its Multiscale and Multiresolution capabilities with fast filterbank algorithms are numerous to describe. Depending upon the application, extensions of standard DWT namely WP and SWT are also employed for improved performance at the cost of higher redundancy and computational complexity.

A few of such applications in Data compression, Denoising, Source and channel coding, Biomedical, Non-destructive evaluation, Numerical solutions of PDE, Study of distant universe, Zero-crossings, Fractals, Turbulence, and Finance etc. are comprehensively

covered in. Wavelet applications in many diverse fields such as Physics, Medicine and biology, Computer Graphics, Communications and multimedia etc. can be found in various books on wavelets.

3.6 Watermarking in the Wavelet Domain

Presently, the most advanced choice among all the frequency domain methods is probably the DWT. The DWT is a hierarchical transform (unlike the FFT and the DCT), which offers the possibility of analyzing a signal at different resolutions or levels (O). Such multiresolution analysis gives a frequency domain representation as a function of time; i.e., both time/space and frequency localization exists. In order to achieve this, the analyzing functions must be localized in time.

The watermark added at a lower resolution is itself watermarked at a higher resolution. The hierarchical nature of the wavelet representation allows detection of watermarks at all resolutions. Detection of lower resolution watermarks reduces computational complexity, as fewer frequency bands are involved. The multiresolutional property makes the proposed watermarking scheme robust to image/video downsampling operation by a power of two in either space or time.

For watermarking, we need to select an appropriate wavelet or basis. Most of the basis developments have taken place in the context of image compression; and fortunately, watermarking and compression have many things in common. On the other hand, we certainly need to choose a basis that offers compact support. The smaller the support of the wavelet, the more energy the transform compacts in the high frequency sub-bands. Also we are restricted to a class of either orthogonal or bi-orthogonal wavelets. Filter regularity and symmetry and a smooth wavelet function are effective in the reconstructed image quality.

A major advantage of the DWT lies in the fact that it performs an analysis similar to that of the HVS. The HVS splits an image into several frequency bands and processes each band independently.

Finally, more general advantages of the DWT are:

- ✓ It is not a block based transform, and so the annoying blocking artifacts associated with the DCT are absent.
- ✓ Its multiresolution property offers more degrees of freedom compared with the DCT.
- ✓ Lower computational cost than the FFT or DCT: O(n) instead of O(nlog(n)), where n is the order of the transform input vector.
- ✓ Better energy compaction than both the FFT and DCT in the sense that it is closer to the optimal Karhunen-Love transform.

CHAPTER # 4

HUMAN VISUAL SYSTEM

4.1 Digital Media

Digital media take advantage of advances in computer-processing techniques and inherit their strength from digital signals. The following distinguishing features make them superior to the analog media:

- ✓ Robustness—The quality of digital media will not degrade as copies are made. They are most stable and more immune to the noises and errors that occur during processing and transmission. Analog signals suffer from signal-path attenuation and generation loss (as copies are made) and are influenced by the characteristics of the medium itself.

- ✓ Seamless integration—This involves the integration of different media through digital storage and processing and transmission technologies, regardless of the particular media properties. Therefore, digital media eliminate device dependency in an integrated environment and allow easy data composition of nonlinear editing.

- ✓ Reusability and interchangeability—With the development of standards for the common exchange formats, digital media have greater potential to be reused and shared by multiple users.

- ✓ Ease of distributed potential—Thousands of copies may be distributed electronically by a simple command.

4.1.1 Digital Image

Digital images are captured directly by a digital camera or indirectly by scanning a photograph with a scanner. They are displayed on the screen or printed. Digital images are composed of a collection of pixels that are arranged as a 2D matrix. This 2D or spatial representation is called the image resolution. Each pixel consists of three components: red (R), green (G) and blue (B). On a screen, each component of a pixel corresponds to a phosphor. A phosphor glows when excited by an electron gun. Various combinations of different RGB intensities produce different colors. The number of bits to represent a pixel is called the color depth, which decides the actual number of colors available to represent a pixel. Color depth is in turn determined by the size of the video buffer in the display circuitry.

The resolution and color depth determine the presentation quality and the size of image storage. The more pixels and the more colors there are means the better the quality and the larger the volume. To reduce the storage requirement, three different approaches can be used:

- ✓ Index color—This approach reduces the storage size by using a limited number of bits with a color lookup table (or color palette) to represent a pixel. Dithering can be applied to create additional colors by blending colors from the palette. This is a technique taking advantage of the fact that the human brain perceives the media color when two different colors are adjacent to one another. With palette optimization and color dithering, the range of the overall color available is still considerable, and the storage is reduced.
- ✓ Color subsampling—Humans perceive color as brightness, hue and saturation rather than as RGB components. Human vision is more sensitive to variation in the luminance (or brightness) than in the chrominance (or color difference). To take advantage of such differences in the human eye, light can be separated into the luminance and chrominance components instead of the RGB components. The color subsampling approach shrinks the file size by down-sampling the chrominance components, that is, using fewer bits to represent the chrominance components while having the luminance component unchanged.

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.500 \\ 0.500 & -0.41869 & -0.08131 \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix}, \quad \text{……..4.1}$$

- ✓ Spatial reduction—This approach, known as data compression, reduces the size by throwing away the spatial redundancy within the images.

4.1.2 Digital Video

Video is composed of a series of still-image frames and produces the illusion of movement by quickly displaying frames one after another. The Human Visual System (HVS) accepts anything more than 20 Frames per Second (fps) as smooth motion. Television and video are usually distinguished. Television is often associated with the concept of broadcast or

cable delivery of programs, whereas video allows more user interactivity, such as recording, editing and viewing at a user-selected time.

The biggest challenges posed by digital video are the massive volume of data involved and the need to meet the real-time constraints on retrieval, delivery and display. The solution entails the compromise in the presentation quality and video compression. As for the compromise in the presentation quality, instead of video with full frame, full fidelity and full motion, one may reduce the image size, use less bits to represent colors, or reduce the frame rate. To reduce the massive volume of digital video data, compression techniques with high compression ratios are required. In addition to throwing away the spatial and color similarities of individual images, the temporal redundancies between adjacent video frames are eliminated.

Digital audio systems are designed to make use of the range of human hearing. The frequency response of a digital audio system is determined by the sampling rate, which in turn is determined by the Nyquist theorem.

4.2 Distortions and Attacks

In practice, a watermarked object may be altered either on purpose or accidentally, so the watermarking system should still be able to detect and extract the watermark. Obviously, the distortions are limited to those that do not produce excessive degradations, since otherwise the transformed object would be unusable. These distortions also introduce degradation on the performance of the system. For intentional attacks, the goal of the attacker is to maximize the reduction in these probabilities while minimizing the impact that his/her transformation produces on the object; this has to be done without knowing the value of the secret key used in the watermarking insertion process, which is where all the security of the algorithm lies. Following are some of the best known attacks. Some of them may be intentional or unintentional, depending on the application.

✓ Additive Noise

This may stem in certain applications from the use of D/A and A/D converters or from transmission errors. However, an attacker may introduce perceptually shaped noise (thus,

imperceptible) with the maximum unnoticeable power. This will typically force to increase the threshold at which the correlation detector works.

✓ Filtering

Low-pass filtering, for instance, does not introduce considerable degradation in watermarked images or audio, but can dramatically affect the performance, since spread-spectrum-like watermarks have non negligible high-frequency spectral contents.

✓ Cropping

This is a very common attack since in many cases the attacker is interested in a small portion of the watermarked object, such as parts of a certain picture or frames of a video sequence. With this in mind, in order to survive, the watermark needs to be spread over the dimensions where this attack takes place.

✓ Compression

This is generally an unintentional attack which appears very often in multimedia applications. Practically all the audio, video and images that are currently being distributed via Internet have been compressed. If the watermark is required to resist different levels of compression, it is usually advisable to perform the watermark insertion task in the same domain where the compression takes place. For instance, DWT or DCT-domain image watermarking is more robust to JPEG/MPEG compression than spatial-domain watermarking. When the quality factor of the MPEG is low, the error of the extracted watermark is increased and the watermark is damaged significantly.

✓ Rotation and Scaling

This has been the true battle-horse of digital watermarking, especially because of its success with still images. Correlation based detection and extraction fail when rotation or scaling is performed on the watermarked image because the embedded watermark and the locally generated version do not share the same spatial pattern anymore. Obviously, it would be possible to do exhaustive search on different rotation angles and scaling factors

until a correlation peak is found, but this is prohibitively complex. Estimating the two parameters becomes simple when the original image is present, but, although the problem resembles synchronization for digital communications, the techniques applied there fail loudly.

✓ Statistical Averaging

An attacker may try to estimate the watermark and then 'unwatermark' the object by subtracting the estimate. This is dangerous if the watermark does not depend substantially on the data. Note that with different watermarked objects it would be possible to improve the estimate by simple averaging. This is a good reason for using perceptual masks to create the watermark.

✓ Frame Dropping

As a video contains a large amount of redundancies between frames, it may suffer attacks by frame dropping. Due to the redundancies few frames are lost during compression, since it leads little or no damage to the video signal.

✓ Attacks at Other Levels

There are a number of attacks that are directed to the way the watermark is manipulated. For instance, it is possible to circumvent copy control mechanisms discussed below by super-scrambling data so that the watermark is lost or to deceive web crawlers searching for certain watermarks by creating a presentation layer that alters the way data are ordered.

4.3 Human Visual System

The first problem that all data-embedding and watermarking schemes need to address is that of inserting data in the digital signal without deteriorating its perceptual quality. We must be able to retrieve the data from the edited host signal. Because the data insertion and data recovery procedures are intimately related, the insertion scheme must take into account the requirement of the data-embedding applications. Data insertion is possible because the digital medium is ultimately consumed by a human. The human hearing and

visual systems are imperfect detectors. Audio and visual signals must have a minimum intensity or contrast level before they can be detected by a human. These minimum levels depend on the spatial, temporal and frequency characteristics of the human auditory and visual systems. Most signal-coding techniques exploit the characteristics of the human auditory and visual systems directly or indirectly. Likewise, all data-embedding techniques exploit the characteristics of the human auditory and visual systems implicitly or explicitly. A diagram of a data-embedding algorithm is shown in Figure 4.1. The information is embedded into the signal using the embedding algorithm and a key. The dashed lines indicate that the algorithm may directly exploit perceptual analysis to embed information. In fact, embedding data would not be possible without the limitations of the human visual and auditory systems.

Data embedding and watermarking algorithms embed text, binary streams, audio, image or video in a host audio, image or video signal. The embedded data are perceptually inaudible or invisible to maintain the quality of the source data.

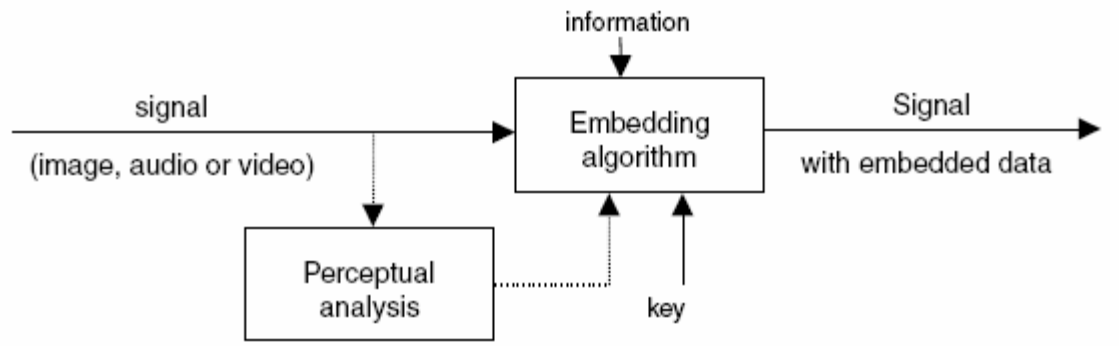


Figure 4.1 Block diagram of a data-embedding algorithm

In this paper, we propose a new video watermarking algorithm based on the human visual system (HVS) properties to find effective locations in video sequences for robust and imperceptible watermarks. In particular, we define a new HVS-optimized global masking map for hiding watermark signals by combining the spatial masking, and the motion masking effects of HVS.

4.3.1 The Global Masking Map

I. Spatial Masking:

The main purpose of the edge map in the proposed watermarking algorithm is to extract connected edges in each image frame. In this paper, we control the contrast of images before the edge detection operation to obtain a good spatial masking map from the following lightness function, proposed by Schreiber [5]:

$$S[x,y] = 1 + 99\frac{log(1 + I[x,y] \cdot a) - log(1+a)}{log(1+100a) - log(1+a)} \quad \text{.....4.2}$$

where I[x, y] is the luminance value of the original frame. Schreiber indicated that a=0.05 provides a well-adapted luminance scale. After we apply the lightness function, we extract important edges to find the spatial masking effect. The effect of spatial masking filter on two selected images is shown in fig 4.2.

Figure 4.2: Effect of spatial masking filter

II. Motion Masking:

A video watermarking method can exploit the structural characteristics of the video sequence. After we find displacement parts in the successive image frames, we can apply a suitable filter to extract image contours. The high values are assigned in the face part because of large motion changes which corresponds to high frequency data or the edges. The effect of motion masking filter on two selected images is shown in fig 4.3.

Figure 4.3: Effect of motion masking filter

III. Global Masking Map Modeling:

In this paper, we define the global masking map by combining the above spatial, and motion masking effects together after normalization. In other words, the global masking map G is obtained by:

$$G = S + M \qquad \text{…….4.3}$$

where S is the spatial masking and M is the motion masking, respectively.

Thus we can insert more watermarks effectively using the proposed global masking map. The effect of global masking filter on two selected images is shown in fig 4.4.

Figure 4.4: Effect of global masking filter

4.4 Feature Extraction Function

Extracting image features suitable for watermarking is one other key point when designing an algorithm. Numerous coefficient extraction functions, built around different selection criteria, have been tried and few of them are presented hereafter:

- ✓ Select all coefficients:

  These functions simply consider all coefficients as candidates for watermarking. They distinguish themselves by the order in which coefficients are scanned (raster, zigzag, etc).
- ✓ Select coefficients at random location:

  The frequency components to be watermarked are randomly chosen. This may be controlled by a density parameter adapted to the image content. Such a technique offers a good secretiveness, as only authorized parties know the location of all watermarked coefficients, but its robustness is limited as it does not ensure that selected coefficients have high perceptual capacities.
- ✓ Select high amplitude coefficients:

  When the frequency transform is orthogonal, these coefficients are carrying most of the information. The obtained scheme is more robust even if watermarked coefficients can be easily located (attempts to remove the watermark produce serious visual degradations). This function can also give satisfying results for nearly-orthogonal frequency transforms.
- ✓ Select coefficients from a specific color space:

  For color image, the choice of the decomposition color space may be very important. Indeed, components of some color spaces exhibit interesting perceptual capacities. For instance, describes a method that exclusively works with the blue image component.
- ✓ HVS based selection:

  The Human Visual System is very complex and still not well understood, even today. However few HVS properties (e.g., masking phenomena) have been modeled and can be considered to improve the scheme's performance. By marking preferably coefficients with higher perceptual capacities, one can increase the watermark energy and consequently obtain more robust schemes.

It is worth pointing out that, most of the time, efficient extraction functions are not simply based on one single selection criteria but usually combine several of them.

4.5 Watermarking Techniques

Different watermarking techniques have been proposed by various authors in the last few years. These proposed algorithms can be classified into two main classes on the basis of

the use of the original image during the detection phase: the algorithms that do not require the original image (blind scheme) and the algorithms where the original image is the input in the detection algorithms along with the watermarked image (nonblind scheme). Detectors of the second type have the advantage of detecting the watermarks in images that have been extensively modified in various ways.

Watermarking embedding can be done either in the spatial domain or in an appropriate transform domain, like a DCT domain, a wavelet transform domain (DWT) or a Fourier transform domain. In this algorithm, the imposed changes take into account the local image characteristics and the properties of the human visual system (perceptual masking) in order to obtain watermarks that are guaranteed to be invisible.

The DWT-based watermarking method has been developed for image watermarking that could survive several kinds of image processings and lossy compression. In order to extend the watermarking techniques into video sequences, the concept of temporal prediction exploited in MPEG is considered. For intraframe, the same techniques of image watermarking are applied, but for non-intraframe, the residual mask, which is used in image watermarking to obtain the spatially neighboring relationship, is extended into the temporal domain according to the type of predictive coding. In considering the JPEG-like coding technique, a DWT-based watermarking method is developed to provide an invisible watermark and also to survive the lossy compression.

The human eyes are more sensitive to noise in a lower frequency range than its higher frequency counterpart, but the energy of most natural images is concentrated in the lower frequency range. The quantization applied in lossy compression reflects the human visual system, which is less sensitive to quantization noise at higher frequencies. Therefore, to embed the watermark invisibly and to survive the lossy data compression, a reasonable trade-off is to embed the watermark into the middle-frequency range of the image. To prevent an expert from extracting the hidden information directly from the transform domain, the watermarks are embedded by modifying the relationship of the neighboring blocks of midfrequency coefficients of the original image instead of embedding by an additive operation.

CHAPTER # 5

ALGORITHM DESIGN

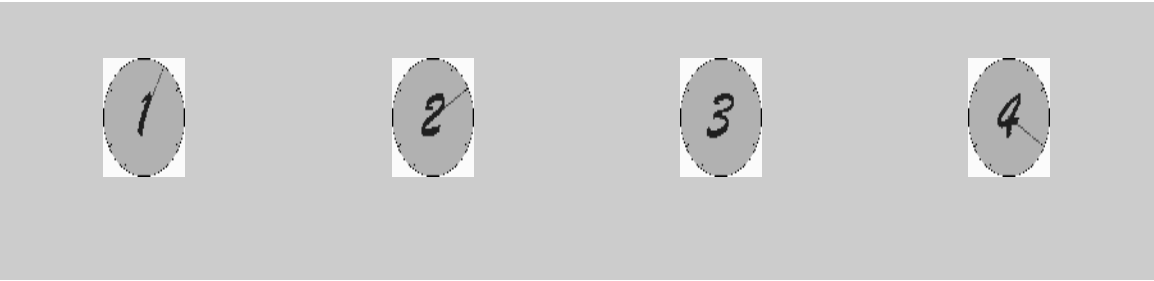
5.1 Watermarking Algorithm Design Issues

In this proposed algorithm a video sequence with four frames is used as a watermark. This watermark is then embedded in the uncompressed video using HVS properties. In particular, we define a new HVS-optimized global masking map for hiding watermark signals by combining the spatial masking, and the motion masking effects of HVS. The algorithm can be divided into four sections:

- ✓ Watermark Preprocess
- ✓ Video Preprocess
- ✓ Watermark Embedding
- ✓ Watermark Detection

5.1.1 Watermark Preprocess

The watermark video sequence is first converted in to frames. As these frames are in indexed format, they have to be converted in to RGB image. Instead of inserting the watermark directly, DWT of each image is taken and this DWT image is then inserted in the frames obtained from the video. The four images which were used as the watermark are shown in figure 1 and the level-3 DWT of fig 1(a) is shown in fig 2.



(a) (b) (c) (d)

Figure 5.1: Watermark frames

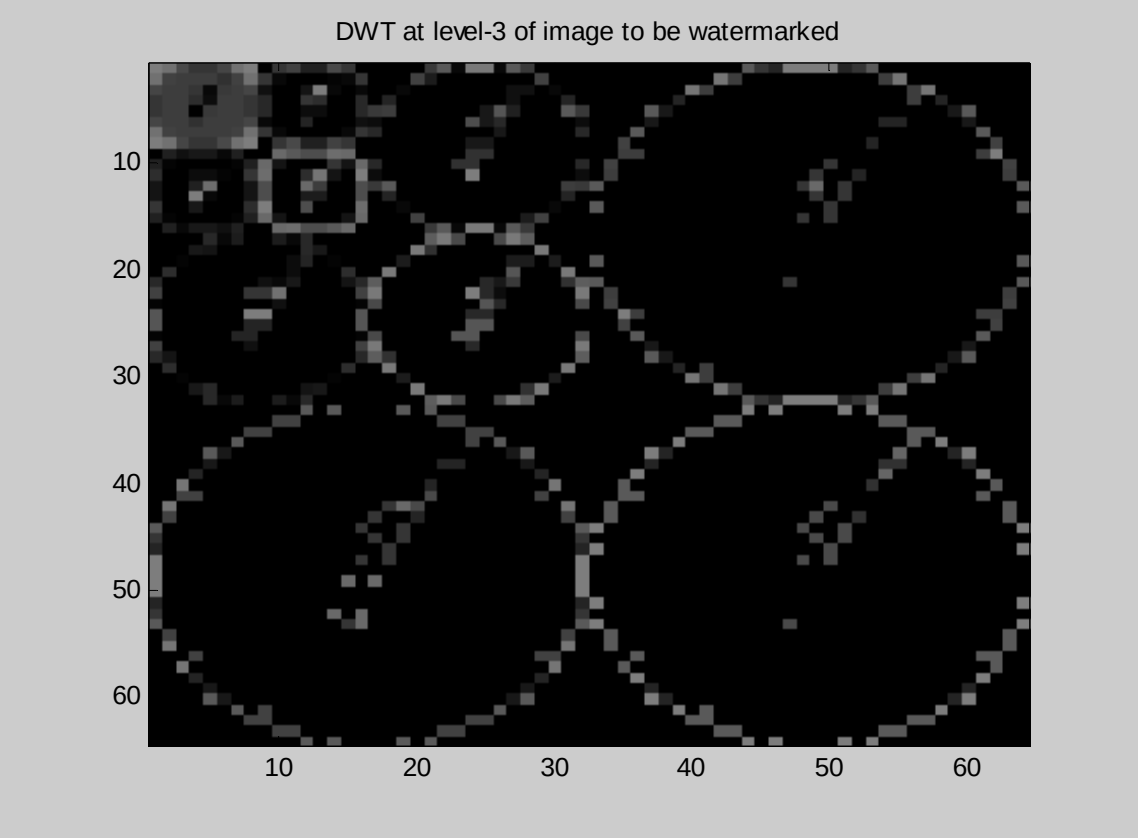


Figure 5.2: DWT of fig 5.1(a)

5.1.2 Video Preprocess

The video is first converted in to frames. And four frames are selected randomly for watermarking. These frames are converted into RGB and then in to YCbCr format. The Y component is also called the luminance component.

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.500 \\ 0.500 & -0.41869 & -0.08131 \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix},$$

The watermark is embedded in the Y component only. Thus DWT of this Y component is taken. The four frames which were selected randomly are shown in fig 3 and level-3 DWT of fig 3(b) is shown in fig 4.

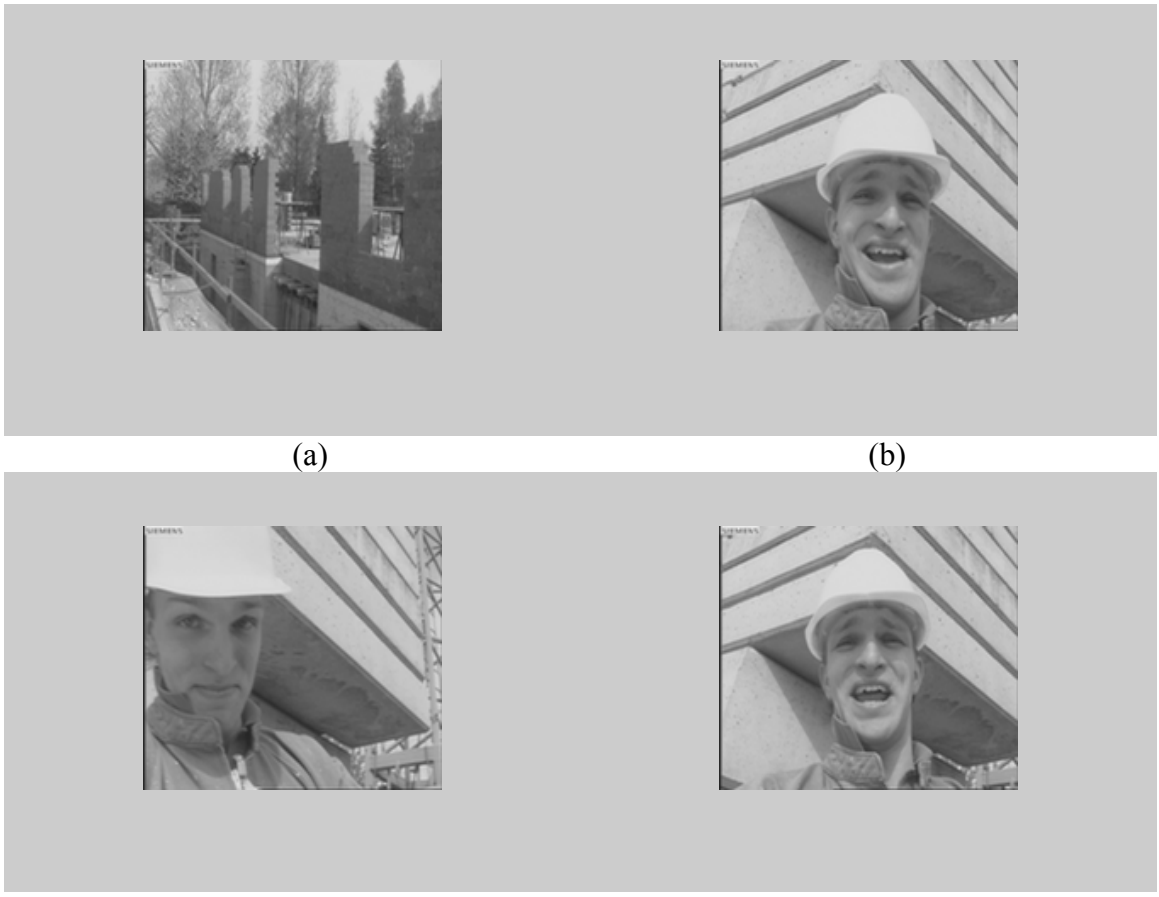


Figure 5.3: Four randomly selected frames from the video

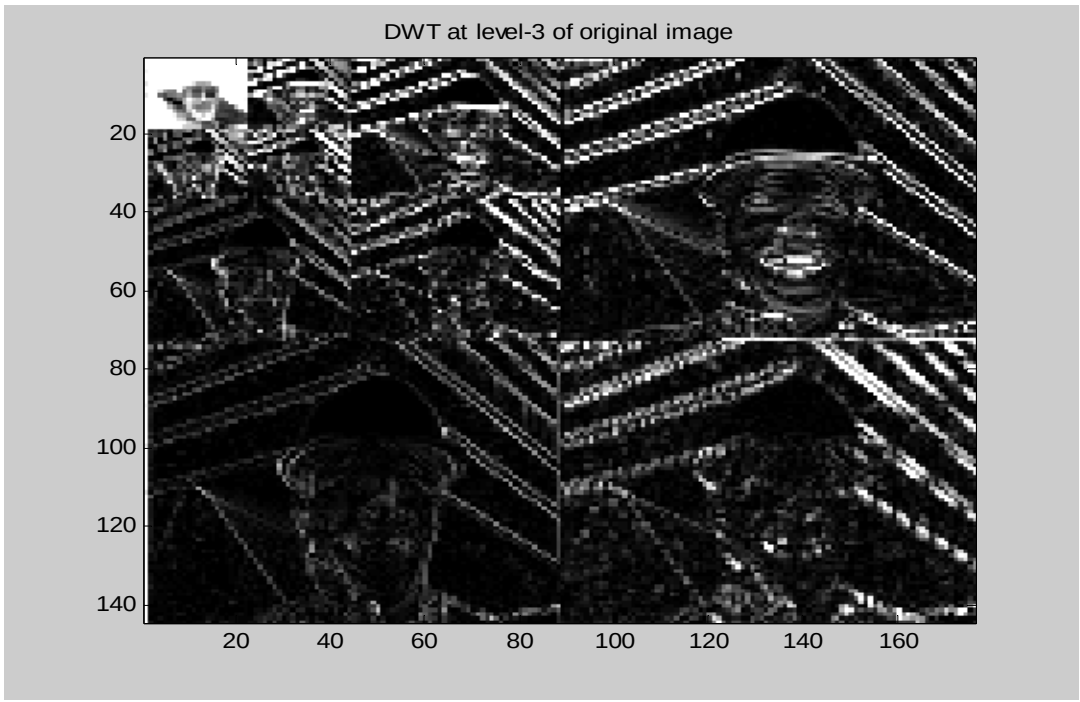


Figure 5.4: DWT of fig 5.3(b)

5.1.3 Watermark Embedding

The main objective of this algorithm is of inserting data in the digital signal without deteriorating its perceptual quality. We must be able to retrieve the data from the edited host signal. In this thesis, HVS-optimized global masking map for hiding watermark signals by combining the spatial masking, and the motion masking effects of HVS is used which have been described in previous chapter. The global masking map G is obtained by:

$$G = S + M \qquad \text{..... (5.1)}$$

where S is the spatial masking and M is the motion masking, respectively.

Thus a secret key is generated which stores the location of the pixel where watermark can be added without deteriorating its perceptual quality. The total number of pixels where watermark can be added should be greater than the number of pixels in the watermark otherwise the retrieved watermark will be distorted. The watermarking is done such that the coefficients of a particular subband of the DWT of the watermark image are embedded in to corresponding subband of the DWT of the video frame. The watermark is inserted by using the formula:

$$I' = I + \alpha \cdot G \cdot W \qquad \text{..... (5.2)}$$

where, the control parameter α is set such that the PSNR of the watermarked frame should not go below a certain threshold, G is the global masking map, I is the DWT coefficient of video frame and W is the DWT coefficient of watermark image.

After inserting the watermark inverse DWT is taken and the frames are combined to form a matlab movie. This matlab movie is then converted to an AVI file by using the matlab function *movie2avi()*. The watermarked frames are shown in fig 5.

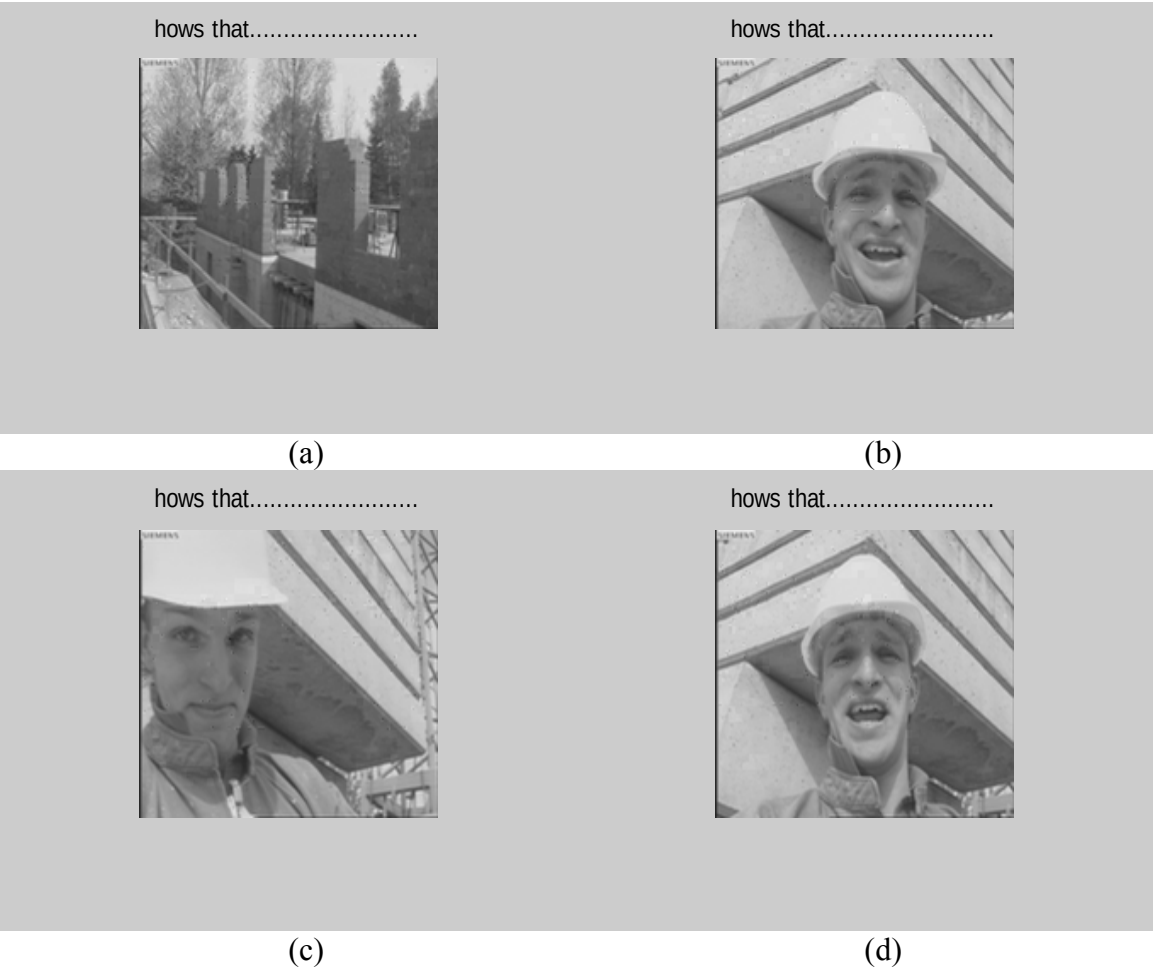


Figure 5.5: Watermarked frames

5.1.4 Watermark Detection

The watermarked video is converted in to frames and the four frames which were watermarked are selected. These frames are decomposed into DWT domain and the DWT image of the original frames are subtracted from them respectively. The coefficients thus obtained are scaled by 1/ α, so that we obtain the watermark.

$$I' - I = G \cdot W^* \qquad \ldots\ldots\ldots\ldots \quad (5.3)$$

where W* is the extracted watermark.

The extracted watermarks and their respective NC are shown in fig 6.

(a) (b) (c) (d)

NC = 99.0488 NC = 99.2091 NC = 98.5838 NC = 98.7917

Figure 5.6 : Extracted watermarks

5.2 Evaluating the Similarity of Watermarks

It is highly unlikely that the extracted mark W* will be identical to the original watermark W. Even the act of re-quantizing the watermarked document for delivery will cause W* to deviate from W. We measure the similarity of W* and W by

$$\delta = \frac{w^* \cdot w}{\|w^*\| \cdot \|w\|}.$$

..... (5.4)

Many other measures are possible, including the standard correlation coefficient. To decide whether W and W* match, one determines whether NC or **δ** >T, where T is some threshold. Setting the detection threshold is a classical decision estimation problem in which we wish to minimize both the rate of false negatives (missed detections) and false positives (false alarms). We can see that NC belongs [0, 100]. If we acquire the higher NC values, the embedded watermark is more similar to the extracted one.

5.3 Algorithm

✓ Watermark Embedding

1. Convert the video and the watermark into frames.
2. Select the frames of the video randomly for watermarking.
3. Obtain the spatial masking and the motion masking factors for each image block, respectively.

4. Generate the global masking map G by combining spatial masking and motion masking factors after normalization.
5. Decompose selected video frames and watermark frames in 2D-DWT domain into three hierarchical levels.
6. Add the watermark weighted by G to the original frame I:

$$I' = I + \alpha \cdot G \cdot W$$

where, the control parameter α is set such that the PSNR of the watermarked frame should not go below a certain threshold.

7. The coefficients of a particular subband of the DWT of the watermark image are embedded in to corresponding subband of the DWT of the video frame.
8. Take inverse DWT to obtain the watermarked frames.

✓ Watermarking Extraction System

1. Convert the watermarked video into frames.
2. Select the watermarked frames from them.
3. Decompose these frames in 2D-DWT domain into three hierarchical levels.
4. Subtract the DWT image of the original frames from the image obtained in step 3 to obtain the watermark.

$$I' - I = G \cdot W^*$$

where W* is the retrieved watermark.

CHAPTER # 6

RESULTS AND DISCUSSION

The sequences utilized in the simulations are Foreman.avi and clock.avi. In Foreman sequence each frame is of 176x144, while in clock sequence each frame is of 64x64. Only four frames of the Foreman sequence are selected randomly and the watermark is embedded only in Y component. The four frames of the clock sequence are used as a watermark.

The PSNR values between the original and watermarked frames are determined as 35.734549, 36.939590, 36.412936, and 36.588290 respectively for the four frames. The watermarked video is subjected to various attacks and the results are analyzed.

JPEG COMPRESSION:

QF: 100%

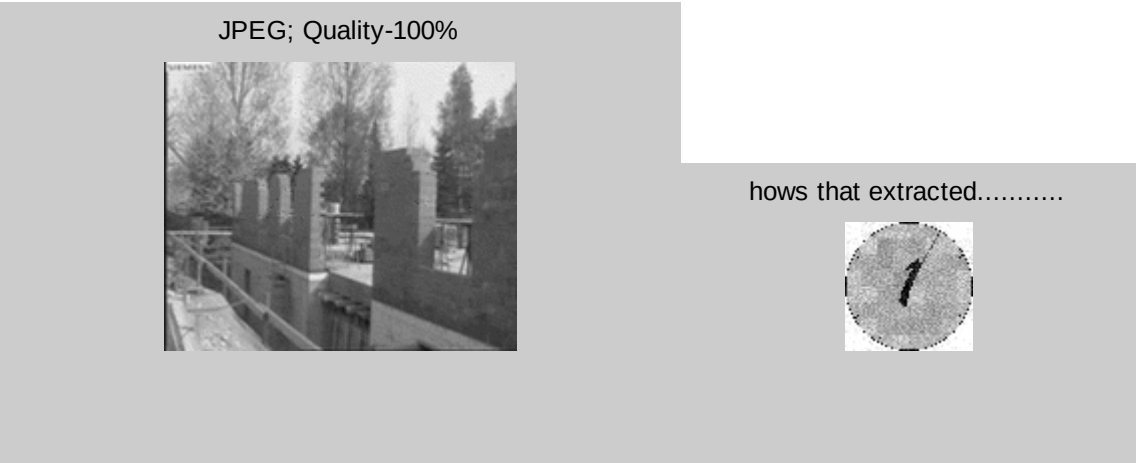


Fig 6.1: Retrieved watermark after JPEG compression with QF-100.

QF: 80%

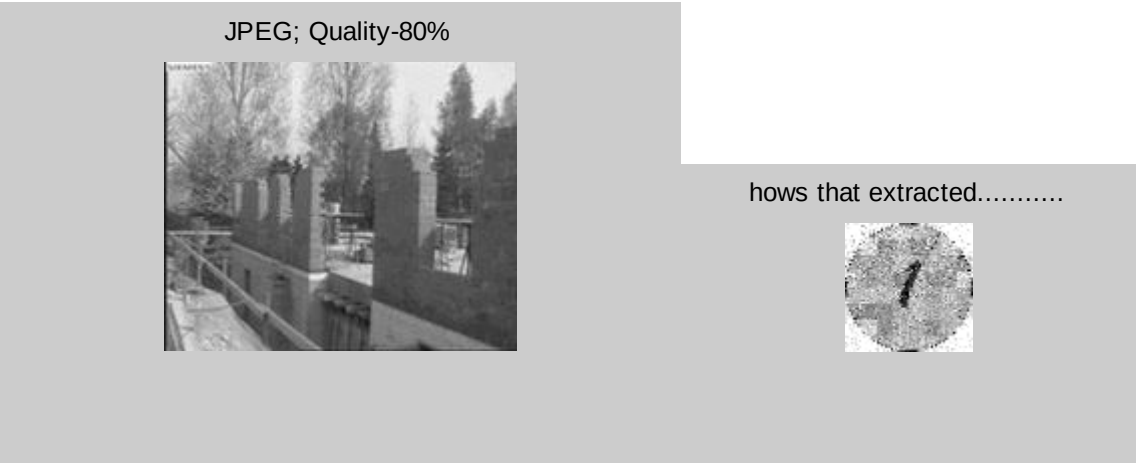


Fig 6.2: Retrieved watermark after JPEG compression with QF-80.

QF: 50%

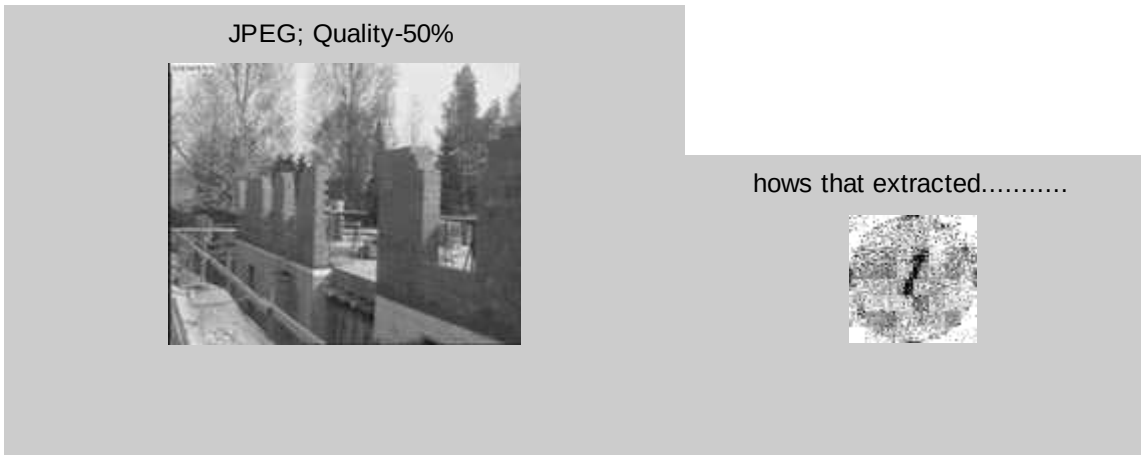


Fig 6.3: Retrieved watermark after JPEG compression with QF-50.

QF: 30%

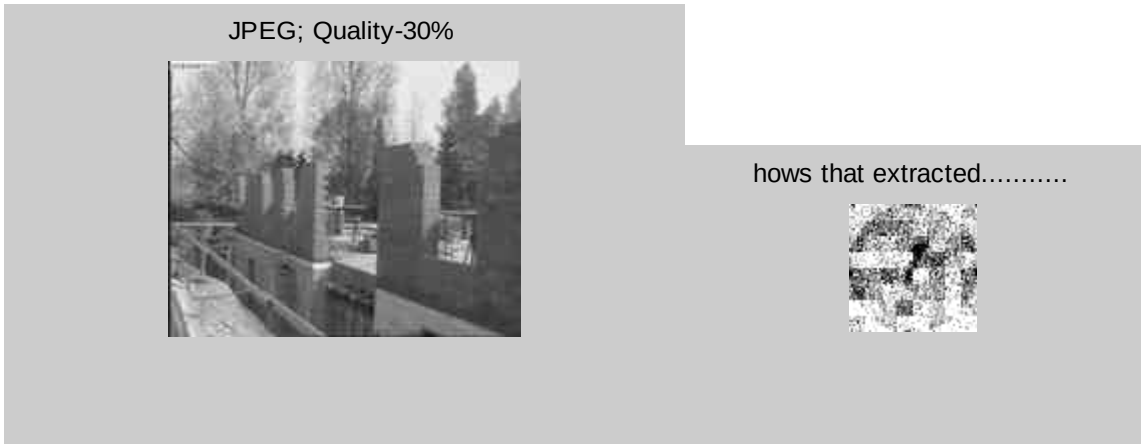


Fig 6.4: Retrieved watermark after JPEG compression with QF-30.

QF: 10%

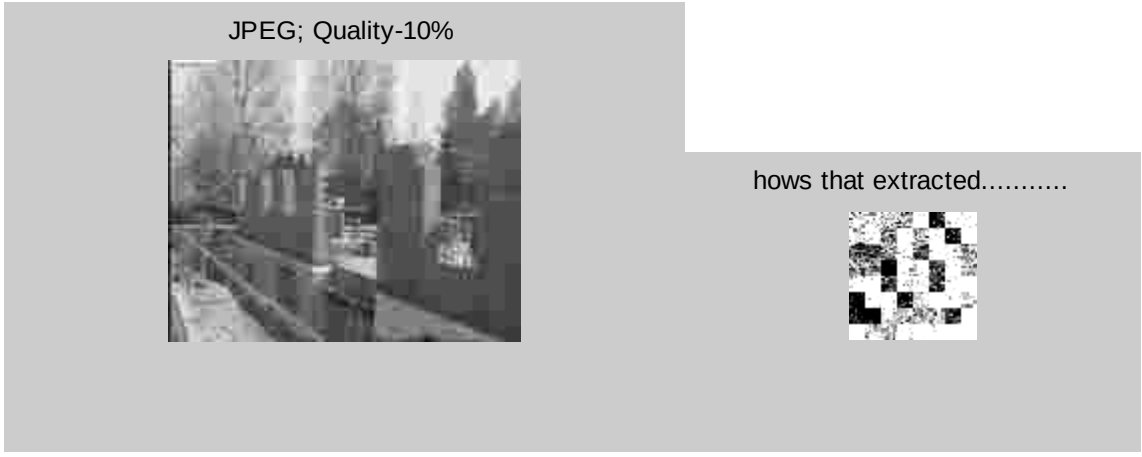


Fig 6.5: Retrieved watermark after JPEG compression with QF-10.

BLURRING ATTACK:

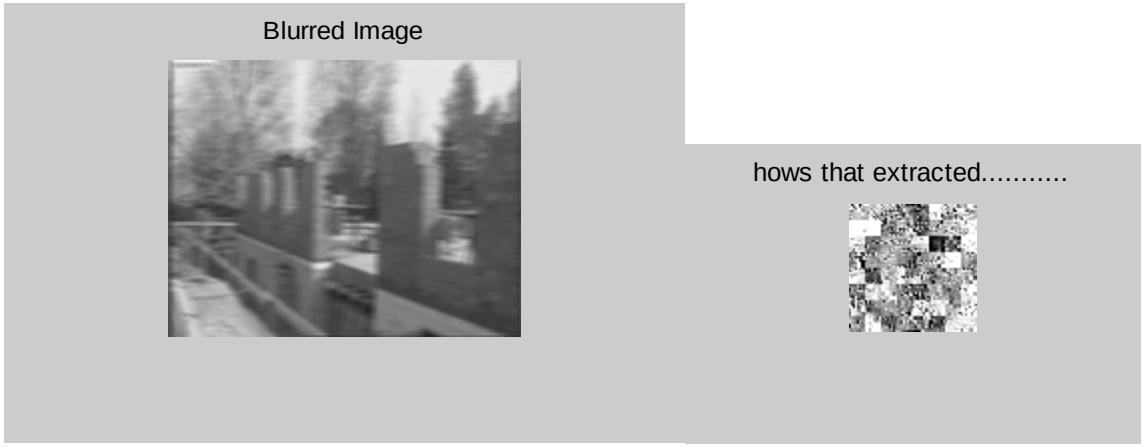


Fig 6.6: Retrieved watermark after blurring attack.

DEBLURRING:

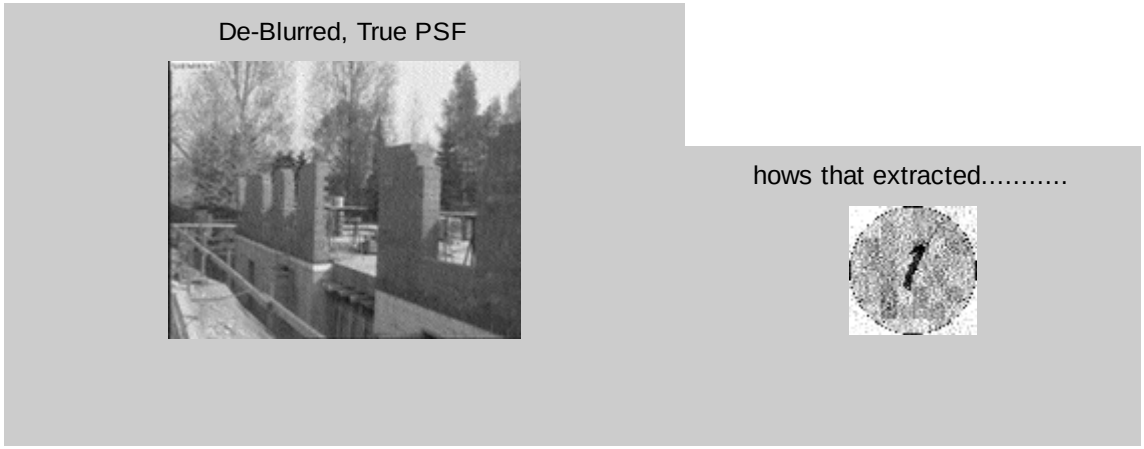


Fig 6.7: Retrieved watermark after deblurring.

ROTATION BY 3 DEGREES:

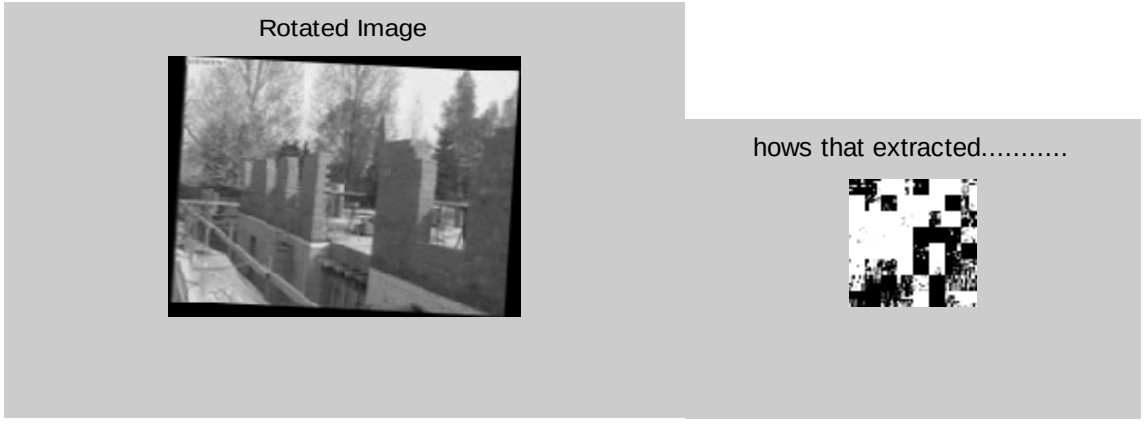


Fig 6.8: Retrieved watermark after rotation by 3 degrees.

AVERAGE FILTER:

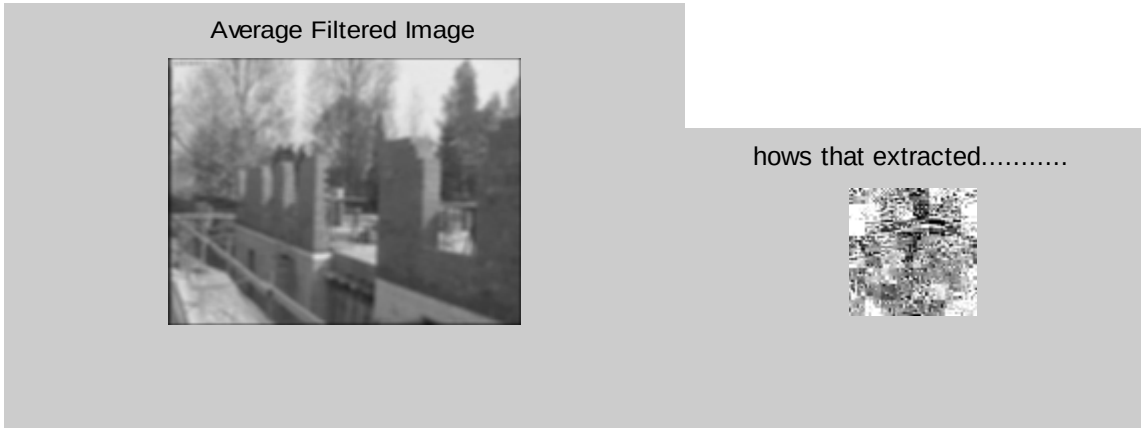


Fig 6.9: Retrieved watermark after average filtering.

SALT N PEPPER NOISE:

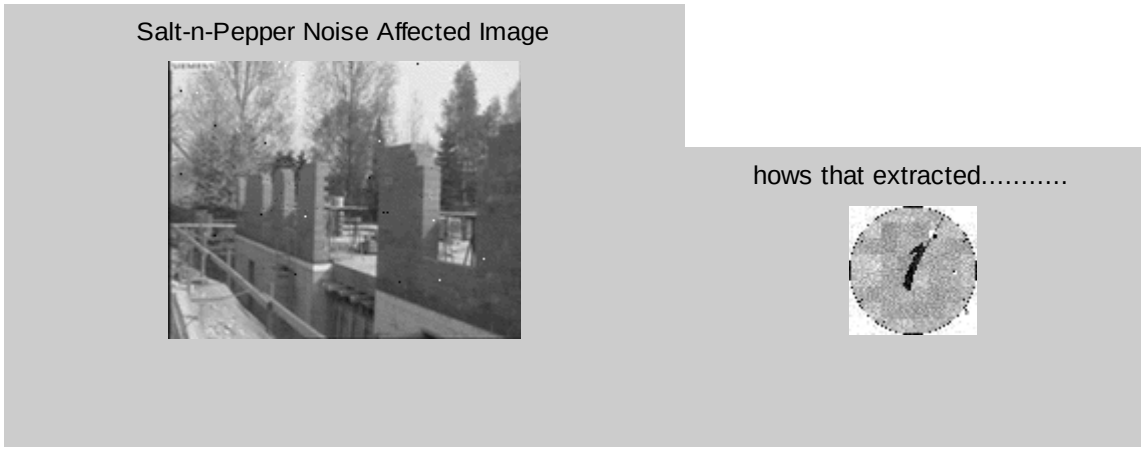


Fig 6.10: Retrieved watermark after salt and pepper attack.

AWGN NOISE:

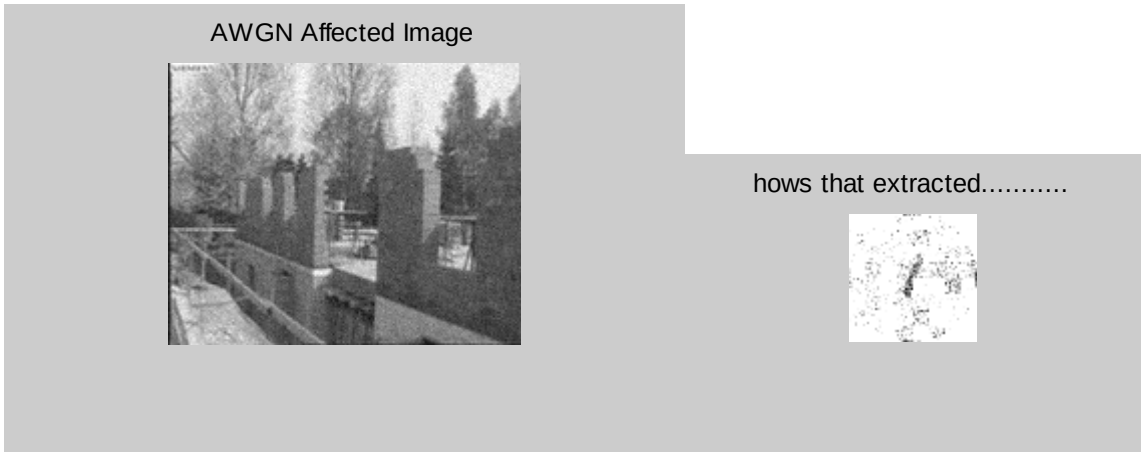


Fig 6.11: Retrieved watermark after AWGN noise attack.

DITHERING:

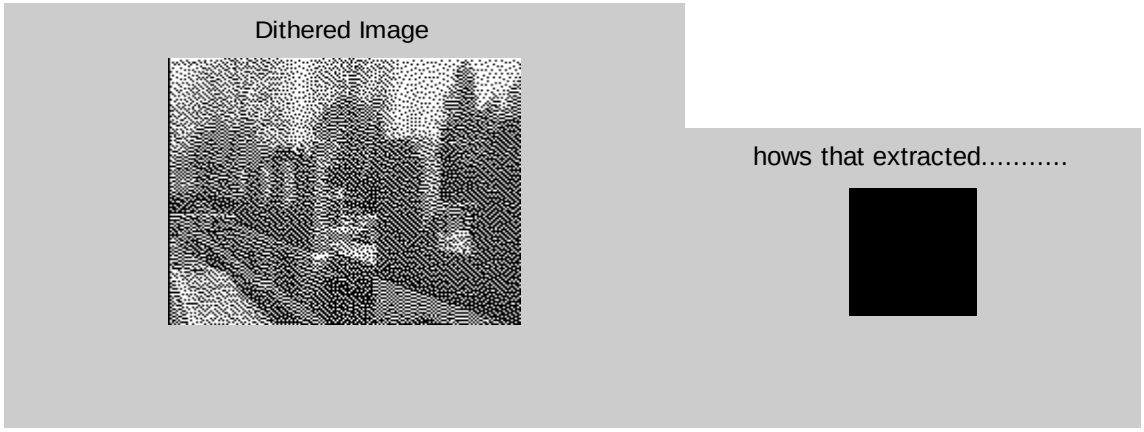


Fig 6.12: Retrieved watermark after dithered attack.

MEDIAN FILTER:

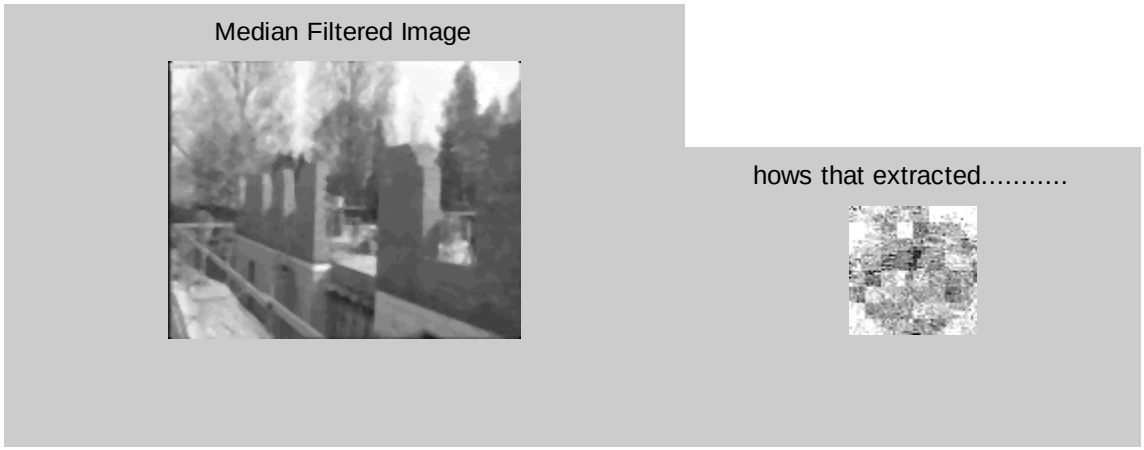


Fig 6.13: Retrieved watermark after median filtered image.

SHARP FILTERING:

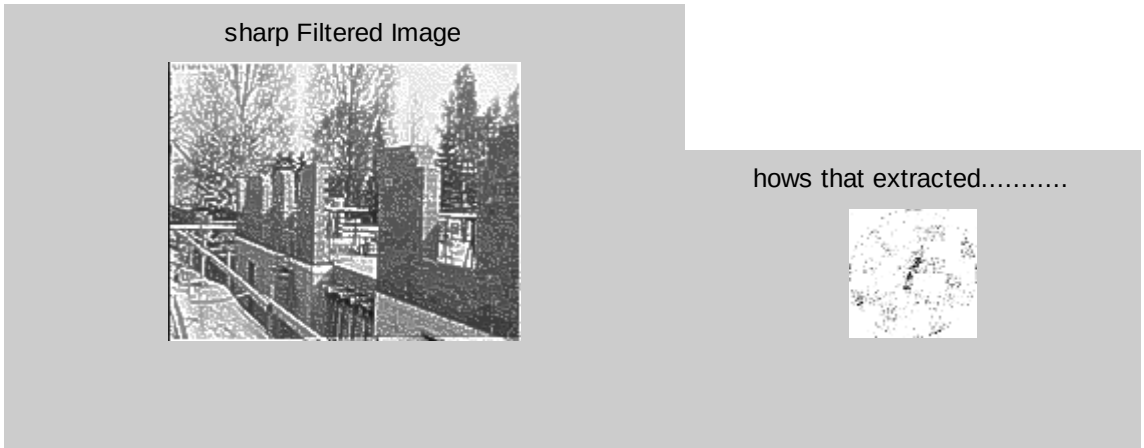


Fig 6.14: Retrieved watermark after sharp filtering.

TABLE 6.1: Comparison of Various Attacks

S. No.	ATTACKS	SNR	NC OF CH2
1	JPEG Compression, Q.F. – 100%	56.324723	99.2288
2	JPEG Compression, Q.F. –80%	33.183931	94.3445
3	JPEG Compression, Q.F – 50%	30.203403	90.1799
4	JPEG Compression, Q.F. – 30%	28.572850	79.7172
5	JPEG Compression, Q.F. – 10%	24.983540	74.0360
6	BLURRED	23.120760	72.9820
7	DEBLURRED	32.750930	91.1568
8	ROTATED BY 3 DEGREE	8.988185	62.8535
9	AVG FILTERED	24.111390	74.1902
10	SALT N PEPER NOISE	32.269387	99.1517
11	AWGN NOISE	26.432151	99.3059
12	DITHERED	1.087300	0.0000
13	MEDIAN	28.549271	89.3830
14	SHARPENED	24.471071	99.4859

CHAPTER # 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

As electronic distribution of copyright material becomes more prevalent a need for digital watermarking rises. In this project, the basic characteristics of a digital watermark are outlined; mainly including: fidelity preservation, robustness to common signal and geometric processing operations, robustness to attacks applicability to digital videos.

In this dissertation work, a new masking model for video watermarking based on the characteristics of the human visual system (HVS) is proposed. The process of this comprehensive video watermarking scheme, including watermark preprocessing, video preprocessing, watermark embedding, and watermark detection, is described in detail. In order to design the general watermarking scheme, the watermark signal is embedded in the uncompressed video sequence. In this thesis, an HVS-optimized global masking map for the best trade-off between invisibility and robustness is defined. The global masking map is generated by combining the spatial and the motion masking effects. After embedding the watermark signal using the information from the global masking map, the amount of watermarks is controlled with the control parameters. Experimental results show that the proposed method is imperceptible to human eyes, and also good in terms of watermark capacity. In addition, this method is robust against the various attacks. In this algorithm, the watermark under various attacks are extracted properly only with slight degradation of image quality.

7.2 Scope for Future Work:

1. The watermark can be scrambled through a well-known PN-sequence. Scrambling the logo image enhances the system security and provides a random distribution of original data.
2. Error correcting codes can be used for better results.
3. A hybrid approach can be applied by embedding audio watermark along with the video watermark.

CHAPTER # 8

BIBLIOGRAPHY

1. Ji-Young Moon and Yo-Sung Ho, “A Video Watermarking Algorithm Based on the Human Visual System Properties”, ISCIS 2003.
2. R. C.Gonzalez and R. E. Woods, “Digital image processing”, Pearson education, 2002.
3. Proakis and Manolakis “Digital signal processing – Principles, Algorithms, and applications,” 2nd edition Maxwell- Macmillan pub.
4. S. Craver et al., “Can Invisible Watermarks Resolve Rightful Ownership?”, IBM Research Report, RC205209, Jul25, 1996. http://www.research.ibm.com/.
5. M. D. Swanson, B. Zhu, A. H. Tewfik, “Multiresolution scene-based video watermarking using perceptual models”, IEEE journal on selected areas in communications, vol. 16, no. 4, may 1998.
6. D. Kundurand, D. Hatzinakos, “A Robust Digital Image Watermarking Method Using Wavelet-Based Fusion”, ICIP, Oct. 1997, vol. I.
7. F. Hartung and M. Kutter, “ Multimedia watermarking techniques,” Proc. IEEE, vol. 87, pp 1079-1107, July 1999.
8. Wolfgang R B, Podilchuk C I, Delp E J, “Perceptual watermarks for digital image and video”, Proceedings of IEEE, 1999, 87(7): 1108~1126.
9. V. Hernandez Guzman, M. N. Miyatake, H. M. P. Meana, “Analysis of a Wavelet-based Watermarking Algorithm”, IEEE Computer Society, 2004.
10. M. Hsieh, D. Chang, Y. Huang, “ Hiding Digital Watermark using Multiresolution Wavelet Transform ” IEEE Transaction on Industrial Electronics, vol. 48, No 5, October 2001, pp. 875-882.
11. Deepa Kundur, Dimitrios Hatzinakos, “ A Robust Digital Image Watermarking Method using Wavelet-Based Fusion ”, in Proc. 4th IEEE Int. Conf. Image Processing ’97, Santa Barbara CA, Oct. 26-29, 1997, pp. 544-547.
12. Mauro Barni, Franco Bartolini, Alessandro Piva, “Improved Wavelet-Based Watermarking Trough Pixel-Wise Masking” IEEE Transactions on Image Processing, Vol. 10, No. 5, May 2001.

13. C.V. Serdean, M.A. Ambroze, M. Tomlinson and J.G. Wade, “DWT based high capacity blind video watermarking, invariant to geometrical attacks”, IEE proc., vol. 150, no. 1, 2003.

14. A. S. Lewis, G. Knowles, “Image Compression Using the 2-D Wavelet Transform”, IEEE Transactions on Image Processing, Vol. 1, No. 2, April 1992.

15. Reza Safabakhsh, Shiva Zaboli and Arash Tabibiazar, “Digital Watermarking on Still Images Using Wavelet Transform”, IEEE Computer Society, 2004.

16. A. B. Watson Gloria Y. Yang, Joshua A. Solomon and J. Villasenor. “Visibility of wavelet quantization noise”. ICIP, vol. 6, no. 8, pp.1164-1175, Aug. 1997.

17. I. Cox and M. L. Miller, “A review of watermarking and the importance of perceptual modeling,” in Proc. SPIE, 1997, vol. 3016, pp. 92-99.

18. S Craver, et. al., “Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks and Implications”, IEEE Jou. Selected Areas in Communications, Vol. 16, No. 4, May 1998, pp. 573-586.

19. http://www.mathworks.com/

20. IEEE website: www.ieee.org.

APPENDIX A: SOURCE CODE

```
%====================main.m=========================%
clc;
clf;
close all;
clear all;

%============READ A MOVIE TAKE A FRAME================================%
watermark_frames =4;

water = aviinfo('for.avi')
watm = aviread('for.avi',1:12);

asd = aviinfo('foreman[1].avi')
mob = aviread('foreman[1].avi',1:300);

%----------------------IMAGE TO BE WATERMARKED ---------------------%
frame_no = rand(1,100);
frame_no = frame_no*1000;

counter = 0;

for frame=1:1:100

   if(frame_no(frame)<300  && counter < watermark_frames)
      counter=counter+1;
      n = fix(frame_no(frame));
      key(counter)=n;
      n1 = n+1;
   else
      continue;
   end

for lev = 1:1:3

aa = mob(n).cdata;
bb= mob(n).colormap;

pp=ind2rgb(aa,bb);
pp=rgb2ycbcr(pp);

IY = pp(:,:,1);
%figure(1);
%imshow(IY);
ICr = pp(:,:,2);
ICb = pp(:,:,3);

if(lev == 1)
   X=imresize(IY,[72,88],'nearest');
elseif(lev == 2)
   X=imresize(IY,[36,44],'nearest');
else
  X=imresize(IY,[18,22],'nearest');
end

%figure;clf;
%imshow(X);title('original image');
```

```
%====================SPATIAL MASKING TRIAL ========================%

[ROW,COL]=size(X);
S = zeros(ROW,COL);

a = 0.05;

for ii=1:ROW
   for jj=1:COL
            S(ii,jj) = 1+(99 * ( log10(1+X(ii,jj)*a)-log10(1+a))/(log10(1+100*a)-log10(1+a)));
   end
end

%figure(7);
%imshow(S);

edcont = edge (S ,'canny');
%figure();
%imshow(edcont);

%=========================MOTION MASKING EFFECT====================%

aa = mob(n).cdata;
bb= mob(n).colormap;

pp=ind2rgb(aa,bb);
pp=rgb2ycbcr(pp);

IYY = pp(:,:,1);
%figure(9);
%imshow(IYY);

aa = mob(n1).cdata;
bb = mob(n1).colormap;

pp=ind2rgb(aa,bb);
pp=rgb2ycbcr(pp);

IYZ = pp(:,:,1);
%figure(10);
%imshow(IYZ);

kk = imsubtract(IYY,IYZ);
kk = im2double(kk);

if(lev == 1)
  X=imresize(kk,[72,88],'nearest');
          elseif(lev == 2)
                    X=imresize(kk,[36,44],'nearest');
          else
                              X=imresize(kk,[18,22],'nearest');
end

%figure(9);
%imshow(X);
```

```
edmotion = edge (X ,'canny');
%figure();
%imshow(edmotion);

%=====================================================================%

S =im2double(edcont);
M =im2double(edmotion);

G = S+M;
%figure();
%imshow(G);

%=====================================================================%
[ROW,COL]= size(G);

c=0;
for ii=1:ROW
   for jj=1:COL
      if( G(ii,jj)==1)
        c = c+1;
        if(lev == 1)
        pixel_rlev1(counter,c)=ii;
        pixel_clev1(counter,c)=jj;
        elseif(lev == 2)
        pixel_rlev2(counter,c)=ii;
        pixel_clev2(counter,c)=jj;
        else
        pixel_rlev3(counter,c)=ii;
        pixel_clev3(counter,c)=jj;
        end
      end
   end
end

counter = counter;
cdd(counter,lev)=c;

end

%==================Inserting of water mark ==========================%

if(cdd(counter,1)<1100 && cdd(counter,1)~=0)
   counter=counter-1;
elseif(cdd(counter,2)<256 && cdd(counter,2)~=0)
 counter=counter-1;
elseif(cdd(counter,3)<64 && cdd(counter,3)~=0)
 counter=counter-1;
end
end

[orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2,
orig_data_cd2,orig_data_cv2,orig_data_ch3,orig_data_cv3, orig_data_cd3, orig_data_ca3]=
embed_video(counter, key, mob, watm, pixel_rlev1, pixel_clev1, pixel_rlev2, pixel_clev2, pixel_rlev3,
pixel_clev3);
```

```
extract_video(counter,key,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,pixel_rlev3,pixel_clev3,or
ig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2, orig_data_cv2
,orig_data_ch3,orig_data_cv3,orig_data_cd3,orig_data_ca3);

%=================================================================%
 sdf  = aviinfo('forwa.avi');
 mobb = aviread('forwa.avi',1:300);

 aa   = mobb(key(1)).cdata;
 bb   = mobb(key(1)).colormap;

 temp=ind2rgb(aa,bb);
 pp=rgb2ycbcr(temp);

 a1  = pp(:,:,1);

 imwrite(aa, 'J100%.jpg','quality',100);
 J  =  imread('J100%.jpg');
 k=ind2rgb(J,bb);
 pp=rgb2ycbcr(k);


 f1  = pp(:,:,1);

 figure;clf;
 imshow(J,'truesize');title('JPEG; Quality-100%');

ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2, orig_data_ch3, orig_data_cv3, orig_data_cd3,orig_data_ca3)

 imwrite(aa, 'J80%.jpg','quality',80);
 J = imread('J80%.jpg');
 k=ind2rgb(J,bb);
 pp=rgb2ycbcr(k);

 f1  = pp(:,:,1);
 figure;clf;
 imshow(J,'truesize');title('JPEG; Quality-80%');

ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2, orig_data_ch3, orig_data_cv3, orig_data_cd3, orig_data_ca3)

 imwrite(aa, 'J50%.jpg','quality',50);

 J = imread('J50%.jpg');

 k=ind2rgb(J,bb);
 pp=rgb2ycbcr(k);

 f1  = pp(:,:,1);
 figure;clf;
 imshow(J,'truesize');title('JPEG; Quality-50%');
```

```
ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3,orig_data_cd3,orig_data_ca3)

  imwrite(aa, 'J30%.jpg','quality',30);
  J = imread('J30%.jpg');
  k=ind2rgb(J,bb);
  pp=rgb2ycbcr(k);

  f1  = pp(:,:,1);
  figure;clf;
  imshow(J,'truesize');title('JPEG; Quality-30%');

ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3,orig_data_cd3,orig_data_ca3)

  imwrite(aa, 'J10%.jpg','quality',10);
  J = imread('J10%.jpg');
  k=ind2rgb(J,bb);
  pp=rgb2ycbcr(k);

  f1  = pp(:,:,1);

  figure;clf;
  imshow(J,'truesize');title('JPEG; Quality-10%');
ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3,orig_data_cd3,orig_data_ca3)

  %Blurring - De-blurring Attack:
  % create PSF
        LEN = 5;
        THETA = 2;
        PSF = fspecial('motion',LEN,THETA);

        % Blur the image
        Blurred = imfilter(aa,PSF,'circular','conv');
        imwrite(Blurred,'BLURR.tif');

        figure;clf;
        imshow(Blurred,'truesize');title('Blurred Image');

        J=imread('BLURR.tif');
        k=ind2rgb(J,bb);
        pp=rgb2ycbcr(k);

        f1  = pp(:,:,1);

        s = snr_cal(a1,f1);
        fprintf('SNR of blurred image = %f\n',s);
         ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3, orig_data_cd3,orig_data_ca3)
```

```
        % Deblur the image
        h10 = deconvwnr(Blurred,PSF);
        imwrite(h10, 'DE_BLURRED.tif');

        figure; clf;
        imshow(h10,'truesize');title('De-Blurred, True PSF');

        J=imread('DE_BLURRED.tif');

        k=ind2rgb(J,bb);
        pp=rgb2ycbcr(k);

 f1  = pp(:,:,1);
 s = snr_cal(a1,f1);
 fprintf('SNR of de-blurred image = %f\n',s);
         ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3,orig_data_cd3,orig_data_ca3)

%-------------------------------------------------------------------------

         %Rotate image by 3 degrees
         [Row_n,Col_n] = size(aa);
         WM_Ir = imrotate(aa,-3,'bilinear');
         WM_Ir2 = imresize(WM_Ir,[Row_n Col_n]);
         imwrite(WM_Ir2,'rot4.tif');

         figure;clf;
         imshow(WM_Ir2,[],'truesize') ;title('Rotated Image');

         J=imread('rot4.tif');
         k=ind2rgb(J,bb);
        pp=rgb2ycbcr(k);

 f1  = pp(:,:,1);
 s = snr_cal(a1,f1);
 fprintf('SNR of ROTATED image = %f\n',s);
         ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3,orig_data_cd3,orig_data_ca3)

          %Average Filtering

          F_h = ones(3,3)/9;
          h12 = imfilter(aa,F_h);
          imwrite(h12, 'Fltr_avg.tif');

          figure;clf;
          imshow(h12,'truesize');title('Average Filtered Image');

          J=imread('Fltr_avg.tif');
          k=ind2rgb(J,bb);
        pp=rgb2ycbcr(k);

 f1  = pp(:,:,1);
 s = snr_cal(a1,f1);
 fprintf('SNR of AVG-FILTERED image = %f\n',s);
```

```
ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,pixel_rlev3,pixel_clev3,orig_data_c
h1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2, orig_data_cv2
,orig_data_ch3,orig_data_cv3,orig_data_cd3,orig_data_ca3)

%-----------------------------------------------------------------

  %Adding 'Noise' (Salt & Pepper noise)
  imwrite(aa, 'unattacked.tif');
  h_1=imread('unattacked.tif');

  h1=uint8(h_1);

  I = imread('unattacked.tif');
  h13 = imnoise(I,'salt & pepper', 0.001);
  imwrite(h13,'Noisy_SNP.tif');

  figure;clf;
  imshow(h13,'truesize');title('Salt-n-Pepper Noise Affected Image');

  J=imread('Noisy_SNP.tif');
  k=ind2rgb(J,bb);
  pp=rgb2ycbcr(k);

 f1  = pp(:,:,1);
 s = snr_cal(a1,f1);
 fprintf('SNR of SALT & PEPER NOISE image = %f\n',s);
ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3,orig_data_cd3,orig_data_ca3)

%-----------------------------------------------------------------

          % AWG-Noise attack

          h14 = imnoise(aa,'gaussian',.01,.001);
          imwrite(h14,'Noisy_AWGN.jpg');
          figure;clf;
          imshow(h14,'truesize');title('AWGN Affected Image');

          J=imread('Noisy_AWGN.jpg');
         k=ind2rgb(J,bb);
        pp=rgb2ycbcr(k);

 f1  = pp(:,:,1);
 s = snr_cal(a1,f1);
 fprintf('SNR of AWGN NOISE image = %f\n',s);
           ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3,orig_data_cd3,orig_data_ca3)

% Apply Dithering Attack

          h15 = dither(a1);
          imwrite(h15,  'Im_dithered.tif');

          figure;clf;
```

```
            imshow(h15, 'truesize');title('Dithered Image');

            J=imread('Im_dithered.tif');
            k=ind2rgb(J,bb);
            pp=rgb2ycbcr(k);

   f1  = pp(:,:,1);
   s = snr_cal(a1,f1);
   fprintf('SNR of IM_DITHERED image = %f\n',s);
          ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3,orig_data_cd3,orig_data_ca3)

% Apply Median Filtering

       I_ym = aa;
      h16 = medfilt2(I_ym,[3 3]);
      J_mf = h16;

      h_16 = uint8(J_mf);
      imwrite(h_16, 'Med_fltr.tif');

      figure;clf;
      imshow(h_16,[], 'truesize');title('Median Filtered Image');

            J=imread('Med_fltr.tif');
            k=ind2rgb(J,bb);
            pp=rgb2ycbcr(k);

   f1  = pp(:,:,1);
   s = snr_cal(a1,f1);
   fprintf('SNR of MEDIAN FILTERED image = %f\n',s);
             ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3,orig_data_cd3,orig_data_ca3)

              %sharpening  Filtering

              F_h = [-1,-1,-1;-1,8,-1;-1,-1,-1]/1;

              h12 = imfilter(aa,F_h);
              h12 = imadd(aa,h12);
              imwrite(h12, 'Fltr_sharp.tif');

              figure;clf;
              imshow(h12,'truesize');title('sharp Filtered Image');

              J=imread('Fltr_sharp.tif');
              k=ind2rgb(J,bb);
              pp=rgb2ycbcr(k);

              f1  = pp(:,:,1);
              s = snr_cal(a1,f1);
              fprintf('SNR of SHARPENED image = %f\n',s);
              ex_att_video(f1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 ,orig_data_ch3, orig_data_cv3,orig_data_cd3,orig_data_ca3)
```

```
%==================embed_video.m=====================%

function [ orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2, orig_data_cv2
,orig_data_ch3,orig_data_cv3, orig_data_cd3,orig_data_ca3] = embed_video(counter,key,mob,watm,
pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,pixel_rlev3,pixel_clev3);

for wcou = 1:1:counter

aa = mob(key(wcou)).cdata;
bb = mob(key(wcou)).colormap;

pp=ind2rgb(aa,bb);
pp=rgb2ycbcr(pp);

IY = pp(:,:,1);
figure(100+wcou);
imshow(IY);
ICr = pp(:,:,2);
ICb = pp(:,:,3);

nbcol =128;
      [ca1,ch1,cv1,cd1] = dwt2(IY,'haar');
      cod_ca1 = wcodemat(ca1,nbcol);
      cod_ch1 = wcodemat(ch1,nbcol);
      cod_cv1 = wcodemat(cv1,nbcol);
      cod_cd1 = wcodemat(cd1,nbcol);

      [ca2,ch2,cv2,cd2] = dwt2(ca1,'haar');
      cod_ca2 = wcodemat(ca2,nbcol);
      cod_ch2 = wcodemat(ch2,nbcol);
      cod_cv2 = wcodemat(cv2,nbcol);
      cod_cd2 = wcodemat(cd2,nbcol);
      cod_ca1 = [cod_ca2,cod_ch2;cod_cv2,cod_cd2];

      [ca3,ch3,cv3,cd3] = dwt2(ca2,'haar');
      cod_ca3 = wcodemat(ca3,nbcol);
      cod_ch3 = wcodemat(ch3,nbcol);
      cod_cv3 = wcodemat(cv3,nbcol);
      cod_cd3 = wcodemat(cd3,nbcol);
      cod_ca2 = [cod_ca3,cod_ch3;cod_cv3,cod_cd3];
      cod_ca1 = [cod_ca2,cod_ch2;cod_cv2,cod_cd2];

      figure;clf;
      image([cod_ca1,cod_ch1;cod_cv1,cod_cd1]);
      title('DWT at level-3 of original image');
      colormap(gray);

aaa = watm(wcou).cdata ;
 bbb = watm(wcou).colormap;
 RGB = ind2rgb(aaa,bbb);
 A=rgb2gray(RGB);
 WA=im2double(A);

 figure();
 imshow(WA);
 [WR,WC]=size(WA);

nbcol =32;
```

```
        [wca1,wch1,wcv1,wcd1] = dwt2(WA,'haar');
        wcod_ca1 = wcodemat(wca1,nbcol);
        wcod_ch1 = wcodemat(wch1,nbcol);
        wcod_cv1 = wcodemat(wcv1,nbcol);
        wcod_cd1 = wcodemat(wcd1,nbcol);

        [wca2,wch2,wcv2,wcd2] = dwt2(wca1,'haar');
        wcod_ca2 = wcodemat(wca2,nbcol);
        wcod_ch2 = wcodemat(wch2,nbcol);
        wcod_cv2 = wcodemat(wcv2,nbcol);
        wcod_cd2 = wcodemat(wcd2,nbcol);
        wcod_ca1 = [cod_ca2,cod_ch2;cod_cv2,cod_cd2];

        [wca3,wch3,wcv3,wcd3] = dwt2(wca2,'haar');
        wcod_ca3 = wcodemat(wca3,nbcol);
        wcod_ch3 = wcodemat(wch3,nbcol);
        wcod_cv3 = wcodemat(wcv3,nbcol);
        wcod_cd3 = wcodemat(wcd3,nbcol);
        wcod_ca2 = [wcod_ca3,wcod_ch3;wcod_cv3,wcod_cd3];
        wcod_ca1 = [wcod_ca2,wcod_ch2;wcod_cv2,wcod_cd2];

        figure;clf;
        image([wcod_ca1,wcod_ch1;wcod_cv1,wcod_cd1]);
        title('DWT at level-3 of image to be watermarked');
        colormap(gray);
c=0;
for ii=1:1:32
   for jj=1:1:32
        c=c+1;
        orig_data_ch1(wcou,c)=ch1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c));
        ch1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c)) = ch1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c))+
wch1(ii,jj)/8;
        orig_data_cd1(wcou,c)=cd1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c));
        cd1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c)) =   cd1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c))+
wcd1(ii,jj)/8;
        orig_data_cv1(wcou,c)=cv1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c));
        cv1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c)) =
cv1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c))+wcv1(ii,jj)/8;
    end
end

c=0;
for ii=1:1:16
   for jj=1:1:16
        c=c+1;
        orig_data_ch2(wcou,c)=ch2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c));
        ch2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c))=  ch2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c))+
wch2(ii,jj)/8;
     orig_data_cd2(wcou,c)=cd2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c));
        cd2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c)) =  cd2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c))+
wcd2(ii,jj)/8;
     orig_data_cv2(wcou,c)=cv2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c));
        cv2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c)) =
cv2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c))+wcv2(ii,jj)/8;
   end
end

c=0;
```

```
for ii=1:1:8
   for jj=1:1:8
      c=c+1;
      orig_data_ch3(wcou,c)=ch3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c));
      ch3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c)) =
ch3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))+wch3(ii,jj)/8;
      orig_data_cd3(wcou,c)=cd3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c));
      cd3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c)) =
cd3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))+wcd3(ii,jj)/8;
      orig_data_cv3(wcou,c)=cv3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c));
      cv3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c)) =
cv3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))+wcv3(ii,jj)/8;
   end
end

c=0;
for ii=1:1:8
   for jj=1:1:8
      c=c+1;
      orig_data_ca3(wcou,c)=ca3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c));
      ca3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c)) =
ca3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))+wca3(ii,jj)/64;
   end
end

X2=idwt2(ca3,ch3,cv3,cd3,'haar');
X1=idwt2(X2,ch2,cv2,cd2,'haar');
X=idwt2(X1,ch1,cv1,cd1,'haar');

figure(1000+wcou);
imshow(X);
title('hows that.....');

pp(:,:,1)=X;
pp(:,:,2)=ICb;
pp(:,:,3)=ICr;
pp=ycbcr2rgb(pp);

[mob(key(wcou)).cdata , mob(key(wcou)).colormap]=rgb2ind(pp,mob(key(wcou)).colormap);
%AA = mob(n).cdata;
%BB =mob(n).colormap;
%figure(20);
%imshow(mob(n).cdata , mob(n).colormap);

end

movie2avi(mob,'forwa.avi','colormap',mob(1).colormap,'compression','none','fps',25);
sdf=aviinfo('forwa.avi')
mobb = aviread('forwa.avi',1:300);
%movie(mobb,1,5);
```

```
%================extract_video.m======================%

function []= extract_video(counter,key,watm,pixel_rlev1,pixel_clev1,pixel_rlev2, pixel_clev2,
pixel_rlev3,pixel_clev3,orig_data_ch1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2,
orig_data_cv2 , orig_data_ch3,orig_data_cv3,orig_data_cd3,orig_data_ca3)

sdf = aviinfo('forwa.avi')
mob = aviread('forwa.avi',1:300);

for wcou = 1:1:counter

aaa = watm(wcou).cdata;
bbb = watm(wcou).colormap;
RGB = ind2rgb(aaa,bbb);
A   = rgb2gray(RGB);
[WR,WC] = size(A);

orig_W  = im2bw(A);

aa = mob(key(wcou)).cdata;
bb = mob(key(wcou)).colormap;

pp=ind2rgb(aa,bb);
pp=rgb2ycbcr(pp);

IY = pp(:,:,1);
figure(200+wcou);
imshow(IY);
ICr = pp(:,:,2);
ICb = pp(:,:,3);

nbcol =128;
      [ca1,ch1,cv1,cd1] = dwt2(IY,'haar');
      cod_ca1 = wcodemat(ca1,nbcol);
      cod_ch1 = wcodemat(ch1,nbcol);
      cod_cv1 = wcodemat(cv1,nbcol);
      cod_cd1 = wcodemat(cd1,nbcol);

      [ca2,ch2,cv2,cd2] = dwt2(ca1,'haar');
      cod_ca2 = wcodemat(ca2,nbcol);
      cod_ch2 = wcodemat(ch2,nbcol);
      cod_cv2 = wcodemat(cv2,nbcol);
      cod_cd2 = wcodemat(cd2,nbcol);
      cod_ca1 = [cod_ca2,cod_ch2;cod_cv2,cod_cd2];

      [ca3,ch3,cv3,cd3] = dwt2(ca2,'haar');
      cod_ca3 = wcodemat(ca3,nbcol);
      cod_ch3 = wcodemat(ch3,nbcol);
      cod_cv3 = wcodemat(cv3,nbcol);
      cod_cd3 = wcodemat(cd3,nbcol);
      cod_ca2 = [cod_ca3,cod_ch3;cod_cv3,cod_cd3];
      cod_ca1 = [cod_ca2,cod_ch2;cod_cv2,cod_cd2];

      %figure;clf;
      %image([cod_ca1,cod_ch1;cod_cv1,cod_cd1]);
      %title('DWT at level-3 of original image');
      %colormap(gray);
```

```
c=0;
for ii=1:1:32
   for jj=1:1:32
      c=c+1;
      temp1 = ch1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c))-orig_data_ch1(wcou,c);
      wch1(ii,jj)=temp1*8;
      temp1=cd1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c))-orig_data_cd1(wcou,c);
      wcd1(ii,jj)= temp1*8;
      temp1=cv1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c))-orig_data_cv1(wcou,c);
      wcv1(ii,jj)=temp1*8;
    end
end

c=0;
for ii=1:1:16
   for jj=1:1:16
      c=c+1;
      temp1 = ch2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c))-orig_data_ch2(wcou,c);
      wch2(ii,jj)=temp1*8;
      temp1=cd2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c))-orig_data_cd2(wcou,c);
      wcd2(ii,jj)= temp1*8;
      temp1=cv2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c))-orig_data_cv2(wcou,c);
      wcv2(ii,jj)=temp1*8;
   end
end

c=0;
for ii=1:1:8
   for jj=1:1:8
      c=c+1;
      temp1 = ch3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))-orig_data_ch3(wcou,c);
      wch3(ii,jj)=temp1*8;
      temp1=cd3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))-orig_data_cd3(wcou,c);
      wcd3(ii,jj)= temp1*8;
      temp1=cv3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))-orig_data_cv3(wcou,c);
      wcv3(ii,jj)=temp1*8;
   end
end

c=0;
for ii=1:1:8
   for jj=1:1:8
      c=c+1;
      temp1 = ca3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))-orig_data_ca3(wcou,c);
      wca3(ii,jj)=temp1*64;
   end
end

X2=idwt2(wca3,wch3,wcv3,wcd3,'haar');
X1=idwt2(X2,wch2,wcv2,wcd2,'haar');
X=idwt2(X1,wch1,wcv1,wcd1,'haar');

figure(2000+wcou);
imshow(X);
title('hows that extracted.....');

Z = im2uint8(X);
[ex_water(wcou).cdata,ex_water(wcou).colormap] = gray2ind(Z,256);
```

```
W_4  = im2bw(X);

%--------------NORMALIZED CROSS CORRELATION -------------%
w=0;
r=0;

for ii=1:1:WR
   for jj = 1:1:WC
       w = w + (orig_W(ii,jj) * W_4(ii,jj));
       r = r + (orig_W(ii,jj) * orig_W(ii,jj));
   end
end
NC = (w / r)*100
end

movie2avi(ex_water,'forwave.avi','colormap',ex_water(1).colormap, 'compression','none','fps',1);
sss=aviinfo('forwave.avi')
end
```

%====================ex_att_video.m====================%

```
function []=
ex_att_video(a1,watm,pixel_rlev1,pixel_clev1,pixel_rlev2,pixel_clev2,pixel_rlev3,pixel_clev3,orig_data_c
h1, orig_data_cd1, orig_data_cv1, orig_data_ch2, orig_data_cd2, orig_data_cv2
,orig_data_ch3,orig_data_cv3,orig_data_cd3,orig_data_ca3)

%for wcou = 1:1:counter
wcou = 1;
aaa = watm(1).cdata;
bbb = watm(1).colormap;
RGB = ind2rgb(aaa,bbb);
A   = rgb2gray(RGB);
[WR,WC] = size(A);

orig_W  = im2bw(A);

nbcol = 128;
      [ca1,ch1,cv1,cd1] = dwt2(a1,'haar');
      cod_ca1 = wcodemat(ca1,nbcol);
      cod_ch1 = wcodemat(ch1,nbcol);
      cod_cv1 = wcodemat(cv1,nbcol);
      cod_cd1 = wcodemat(cd1,nbcol);

      [ca2,ch2,cv2,cd2] = dwt2(ca1,'haar');
      cod_ca2 = wcodemat(ca2,nbcol);
      cod_ch2 = wcodemat(ch2,nbcol);
      cod_cv2 = wcodemat(cv2,nbcol);
      cod_cd2 = wcodemat(cd2,nbcol);
      cod_ca1 = [cod_ca2,cod_ch2;cod_cv2,cod_cd2];

      [ca3,ch3,cv3,cd3] = dwt2(ca2,'haar');
      cod_ca3 = wcodemat(ca3,nbcol);
      cod_ch3 = wcodemat(ch3,nbcol);
      cod_cv3 = wcodemat(cv3,nbcol);
      cod_cd3 = wcodemat(cd3,nbcol);
      cod_ca2 = [cod_ca3,cod_ch3;cod_cv3,cod_cd3];
      cod_ca1 = [cod_ca2,cod_ch2;cod_cv2,cod_cd2];

      %figure;clf;
      %image([cod_ca1,cod_ch1;cod_cv1,cod_cd1]);
      %title('DWT at level-3 of original image');
      %colormap(gray);

c=0;
for ii=1:1:32
   for jj=1:1:32
      c=c+1;
      temp1 = ch1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c))-orig_data_ch1(wcou,c);
      wch1(ii,jj)=temp1*8;
      temp1=cd1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c))-orig_data_cd1(wcou,c);
      wcd1(ii,jj)= temp1*8;
      temp1=cv1(pixel_rlev1(wcou,c),pixel_clev1(wcou,c))-orig_data_cv1(wcou,c);
      wcv1(ii,jj)=temp1*8;
    end
end

c=0;
```

```
for ii=1:1:16
   for jj=1:1:16
      c=c+1;
      temp1 = ch2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c))-orig_data_ch2(wcou,c);
      wch2(ii,jj)=temp1*8;
      temp1=cd2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c))-orig_data_cd2(wcou,c);
      wcd2(ii,jj)= temp1*8;
      temp1=cv2(pixel_rlev2(wcou,c),pixel_clev2(wcou,c))-orig_data_cv2(wcou,c);
      wcv2(ii,jj)=temp1*8;
   end
end

c=0;

for ii=1:1:8
   for jj=1:1:8
      c=c+1;
      temp1 = ch3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))-orig_data_ch3(wcou,c);
      wch3(ii,jj)=temp1*8;
      temp1=cd3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))-orig_data_cd3(wcou,c);
      wcd3(ii,jj)= temp1*8;
      temp1=cv3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))-orig_data_cv3(wcou,c);
      wcv3(ii,jj)=temp1*8;
   end
end

c=0;
for ii=1:1:8
   for jj=1:1:8
      c=c+1;
      temp1 = ca3(pixel_rlev3(wcou,c),pixel_clev3(wcou,c))-orig_data_ca3(wcou,c);
      wca3(ii,jj)=temp1*64;
   end
end

X2=idwt2(wca3,wch3,wcv3,wcd3,'haar');
X1=idwt2(X2,wch2,wcv2,wcd2,'haar');
X=idwt2(X1,wch1,wcv1,wcd1,'haar');

figure;
imshow(X);
title('hows that extracted.....');

W_4  = im2bw(X);

%--------------NORMALIZED CROSS CORRELATION ------------%
w=0;
r=0;
for ii=1:1:WR
   for jj = 1:1:WC
       w = w + (orig_W(ii,jj) * W_4(ii,jj));
       r = r + (orig_W(ii,jj) * orig_W(ii,jj));
   end
end

NC = (w / r)*100
```

%==========end==========%