BIOMETRIC WATERMARKING AND RECOGNITION USING IRIS

SUBMITTED BY:

ASTHA AGARWAL (081297)
DEEPIKA LOHANI (081266)
NEHA TOLANI (081307)

UNDER THE GUIDANCE OF:

MR. RAVIKANT VERMA
(Lecturer, JUIT)
MRS. MEENAKSHI ARYA
(Lecturer, JUIT)

MAY-2012

Submitted in partial fulfilment of the Degree of

Bachelor of Technology

DEPARTMENT OF COMUTER SCIENCE

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

TABLE OF CONTENTS

CERTIFICATE

This is to certify that the work titled "**Biometric Watermarking and Recognition using Iris**" submitted by "**Astha Agarwal(081297), Deepika Lohani(081266), Neha Tolani(081307)**" in partial fulfilment for the award of degree of **B.Tech** of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor ……..

Name of Supervisor ……..

Designation ……..

Date ……..

ACKNOWLEDGEMENT

We would like to thank Mrs. Meenakshi Arya for her kind support at every step of this project. The supervision and support that she gave truly help the progression and smoothness of the project. The co-operation is much indeed appreciated.

Also, we would like to convey our thanks to Mr. Ravikant Verma for extending his support.

Last but not the least, we express gratitude towards Dr. Nitin (Project coordinator), Mr. Satya Prakash Ghrera (Head of Computer Science Department) and Mr. Balbir Singh (Director of Jaypee University of Information Technology).

_____	_____	_____
ASTHA AGARWAL	NEHA TOLANI	DEEPIKA LOHANI
DATE:	DATE:	DATE:

SUMMARY

A watermark is a form, image or text that is impressed onto paper, which provides evidence of its authenticity. Digital watermarking is an extension of the same concept. There are two types of watermarks: visible watermark and invisible watermark. In this project we have concentrated on implementing watermark in image. The main consideration for any watermarking scheme is its robustness to various attacks. Watermarking dependency on the original image increases its robustness but at the same time we need to make sure that the watermark is imperceptible. In this project an invisible watermarking technique is implemented.

Biometric Watermarking pertains to the integration of biometrics with watermarking. We present a novel technique of watermarking using iris of the person to which the work belongs, as biometric. In this project, we propose to use the combination of Zernike moments for feature extraction along with Lifting Wavelet Transform (LWT) and Singular value decomposition (SVD) for watermark embedding. The extracted iris image obtained by inverse of LWT and SVD is subjected to the procedure of extraction of features for template matching. K-means clustering is used to match the recovered watermark image with the set of all centred images in the training database.

This technique when tested for robustness gives appreciable results .The major geometric attacks which this technique survives are histogram equalization, salt and pepper noise, Gaussian noise and lossy JPEG compression.

This project aims to develop biometric digital watermarking software for authorization and copyright identification.

_____	_____	_____	_____
ASTHA AGARWAL	NEHA TOLANI	DEEPIKA LOHANI	**SUPERVISER**
DATE:	DATE:	DATE:	DR. NITIN
			DATE:

PROBLEM STATEMENT

The problem statements are as follows:

- Nowadays audio, video, image and many other things can be represented in digital form. It is much easier for someone to make a perfect copy, which will lead to extensive unauthorized copying. Even when encrypted for distribution, data can easily be decrypted and copied.

- Sometimes forgery digital media or documents become an important issue for identification, authentication and law enforcement.

- These concerns triggered many researches to find ways to hide the copyright messages and serial number into digital media.

- In this project, we aim to find a method in which we embed information into images in such a way that it is imperceptible to human observer but easily detected by computer algorithm.

- The information then detected by the user should be easily recognizable and lead back to the correct owner without showing any ambiguity.

- The algorithm should be robust to several threats posed by the internet users i.e. the extracted information from the image should lead back to its original user despite various attacks by hackers or other users. Watermarking dependency on the original image increases its robustness but at the same time we need to make sure that the watermark is imperceptible.

- The method of embedding information in the images should be non-detectable, non-perceptible, robust, undeletable and unambiguous.

OBJECTIVE AND SCOPE OF THE PROJECT

Project Objectives:

1. Investigate feasible watermarking techniques that can be implemented in real time for document and image applications.
2. Implement technique of feature extraction which is robust to several geometric attacks such as translation, transformation and rotation.
3. A suitable matching technique, which recognizes the correct owner of the digital document or image.
4. An efficient method of information retrieval from a collection of iris images in the database.
5. The algorithm should be robust to any attack posed by users.

Project Scope:

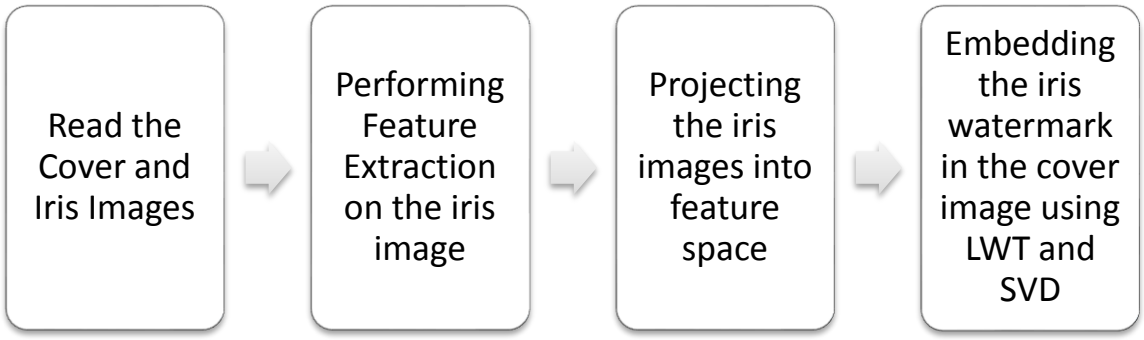
Digital watermarking may be used for a wide range of applications, such as:

1. Copyright protection
2. Source tracking (different recipients get differently watermarked content)
3. Broadcast monitoring (television news often contains watermarked video from international agencies)
4. Covert communication

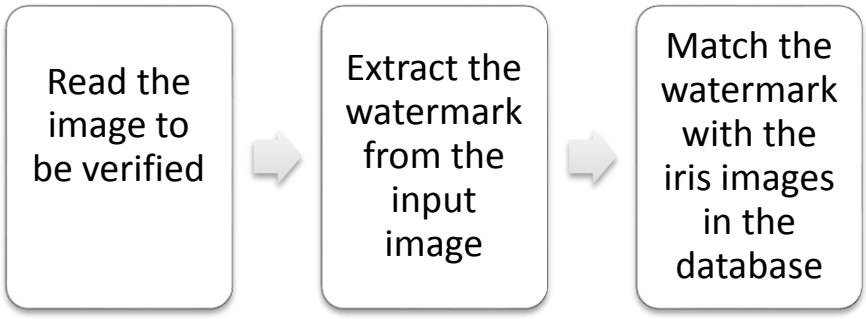
METHODOLOGY

The overall project has been divided into two phases:

1. Phase one consists of feature extraction using Zernike Moments and watermark embedding in cover image using Lifting Wavelet Transform(LWT) and Singular Value Decomposition(SVD).



2. Phase two consists of extraction of watermark from the cover image and matching it with the images in database.



We are using Matlab for the project coding because of several advantages discussed later.

LIST OF FIGURES

CHAPTER 1 - INTRODUCTION

1.1 Brief Overview

The ever increasing rate of data exchange over internet in the form of audio, video and images, is creating an indispensable need for copyright protection. Watermarking is the best remedy for protecting and preserving the rightful ownership of any useful data. But, the rise in technology brings with it a rise in fraudulent ways of acquiring, manipulating and tampering data. Thus, watermarking should be robust and should be able to survive a variety of attacks.

Embedding of a watermark into a host image can be done through a variety of techniques. They can be broadly classified into spatial domain and frequency domain techniques. Spatial domain involves technique random selection of subsets of host image matrix that is modified in order to hide the watermark. Although the complexity of spatial domain algorithm is low but it is not robust. Whereas frequency domain techniques are considered more robust against attacks as the watermark is inserted into the coefficients of a transformed image. DFT, DWT and DCT can be used for transforming images.

So far, research has been done using LWT in combination with SVD to achieve a hybrid and robust digital image watermarking technique which is resistant to various geometric attacks [1].Many researchers have proposed watermarking algorithms using SVD and Zernike moments [2][3][4][5][6] .Zernike moments has also been applied for biometric based recognition using fingerprint[9] and iris [7][8]. Zernike moments were used by GAO [10] along with LWT and neural networks to propose a watermarking algorithm and accurate reconstruction of an attacked image.

The use of biometric enables a more secure and convenient way of personal identification than password or card based identification of a person. Biometric based identification includes distinct physical characteristics of a person and therefore authenticates an individual uniquely. Such identifications are more reliable as they cannot be stolen, forgotten, lost or attached unlike the identification techniques mentioned earlier.

Therefore a lot of work is going on in the field of biometric watermarking using biometrics like fingerprint, face, voice and eyes. Using Iris as a biometric provides an edge as its fine and intricate structure allows a high level of accuracy, thus lowering the false match rates. The highly stable iris structure has grabbed a wide attention and is being researched upon extensively. It has various applications like securing bank accounts, national border controls, internet security, premises access control, entitlements and benefits authorization.

This implementation requires a huge database to be stored and processed. In order to reduce the computational complexity and memory requirement, it has become essential to reduce the available data to a subset containing more relevant information. For this purpose feature extraction technique to remove redundant and less significant information is used which simplifies it to a reduced representation set of features also known as feature vector.

Moments are used to represent values at locations relative to some reference point or axis. Various moment functions have been proposed for feature extraction, but Zernike moments due to their ability to show an additional feature of rotational invariance serves as a better alternative. Also, with relatively few data points, it provides an accurate description, better image reconstruction capability and insensitivity to image noise.

1.2 Watermarking Overview

Hold a Rs100 note up or your offer letter up to light. What you will see is a picture of Mahatma Gandhi or company's logo respectively. This is what is known as a watermark mainly used to prove the ownership (in case of offer letter, watermark prove that the document is official document of company meant for official work) or authenticity (in case of Rs 100, watermark rule out the forgery and authenticate the piece of paper of its worth).The watermark on the Rs100 (Figure2.1), just like most paper watermarks today, has two properties. First, the watermark is hidden from view during normal use, only becoming visible as a result of a special viewing process (in this case, holding the bill up to the light). Second, the watermark carries information about the object in which it is hidden (in this case, the watermark indicates the authenticity of the bill).

Fig 1.1 Image showing an INR 100 note having watermark at its left side which is considerably visible when note hold under light.

In addition to paper, watermarking can be applied to other physical objects and to electronic signals. Fabrics, garment labels, and product packaging are examples of physical objects that can be watermarked using special invisible dyes and inks Electronic representations of music, photographs, and video are common types of signals that can be watermarked. Thus, watermarking is defined as,

"the practice of imperceptibly altering a Work to embed a message about that Work."

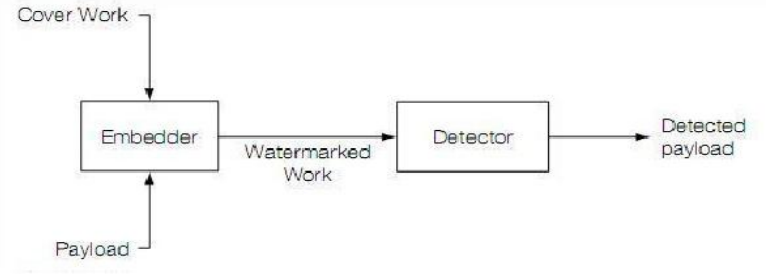


Fig 1.2 A Generic Watermarking System

As is clear from the figure, digital watermarking model consist of an embedder and a detector.

The embedder takes two inputs. One is the payload we want to embed (the watermark), and the other is the cover work in which we want to embed the payload. The output of the embedder is typically transmitted or recorded. Later, that Work (or some other Work that has not been through the embedder) is presented as an input to the detector. Most detectors try to determine whether a payload is present, and if so, output the message encoded by it. The watermarking model is analogous to a communication model in which sender encode a message before transmitting it over communication channel and on receiving, receiver decode the encoded message.

1.3 History Of Watermarking

Although the art of papermaking was invented in China over one thousand years earlier, paper watermarks did not appear until about 1282, in Italy. The marks were made by adding thin wire patterns to the paper moulds. The paper would be slightly thinner where the wire was and hence more transparent. The meaning and purpose of the earliest watermarks are uncertain. They may have been used for practical functions such as identifying the moulds on which sheets of papers were made, or as trademarks to identify the paper maker. On the other hand, they may have represented mystical signs, or might simply have served as decoration.

By the eighteenth century, watermarks on paper made in Europe and America had become more clearly utilitarian. They were used as trademarks, to record the date the paper was manufactured,

and to indicate the sizes of original sheets. It was also about this time that watermarks began to be used as anti-counterfeiting measures on money and other documents. The term watermark seems to have been coined near the end of the eighteenth century and may have been derived from the German term wassermarke (though it could also be that the German word is derived from the English). The term is actually a misnomer, in that water is not especially important in the creation of the mark. It was probably given because the marks resemble the effects of water on paper. About the time the term watermark was coined, counterfeiters began developing methods of forging watermarks used to protect paper money.

Counterfeiting prompted advances in watermarking technology. William Congreve, an Englishman, invented a technique for making colour watermarks by inserting dyed material into the middle of the paper during papermaking. The resulting marks must have been extremely difficult to forge, because the Bank of England itself declined to use them on the grounds that they were too difficult to make. A more practical technology was invented by another Englishman, William Henry Smith. This replaced the fine wire patterns used to make earlier marks with a sort of shallow relief sculpture, pressed into the paper mould. The resulting variation on the surface of the mould produced beautiful watermarks with varying shades of gray. This is the basic technique used today for the face of President Jackson on the $20 bill. Four hundred years later, in 1954, Emil Hembrooke of the Muzak Corporation filed a patent for “watermarking” musical Works. An identification code was inserted in music by intermittently applying a narrow notch filter centred at 1 kHz. The absence of energy at this frequency indicated that the notch filter had been applied and the duration of the absence used to code either a dot or a dash. The identification signal used Morse code. It is difficult to determine when digital watermarking was first discussed. In 1979, Szepanski described a machine-detectable pattern that could be placed on documents for anti-counterfeiting purposes. Nine years later, Holt described a method for embedding an identification code in an audio signal. However, it was Komatsu and Tominaga, in 1988, which appear to have first used the term digital watermark. Still, it was probably not until the early1990s that the term digital watermarking really came into vogue. About 1995, interest in digital watermarking began to mushroom. In addition, about this time, several organizations began considering watermarking technology for inclusion in various standards. The Copy Protection Technical Working Group (CPTWG) tested watermarking systems for protection of video on DVD disks. The Secure Digital Music Initiative (SDMI) made watermarking a central component of their system for protecting music. Two projects sponsored by the European Union, VIVA [110] and Talisman, tested watermarking for broadcast monitoring. The International Organization for Standardization (ISO) took an interest in the technology in the context of designing advanced MPEG standards. In the late 1990s several companies were established to market watermarking products. Technology from the

Verance Corporation was adopted into the first phase of SDMI and was used by Internet music distributors such as Liquid Audio. In the area of image watermarking, Digimarc bundled its watermark embedder and detectors with Adobe's Photoshop. More recently, a number of companies have used watermarking technologies for a variety of applications.

1.4 Requirements For Watermarking Algorithms:

A watermarking algorithm should be consistent over following properties and parameters:

1. **Transparency:** The most fundamental requirement for any Watermarking method shall be such that it is transparent to the end user. The watermarked content should be consumable at the intended user device without giving annoyance to the user. Watermark only shows up at the watermark-detector device.
2. **Security:** Watermark information shall only be accessible to the authorized parties. Only authorized parties shall be able to alter the Watermark content. Encryption can be used to prevent unauthorized access of the watermarked data.
3. **Ease of embedding and retrieval:** Ideally, Watermarking on digital media should be possible to be performed "on the fly". The computation need for the selected algorithm should be minimum.
4. **Robustness:** Watermarking must be robust enough to withstand all kinds for signal processing operations, "attacks" or unauthorized access. Any attempt, whether intentional or not, that has a potential to alter the data content is considered as an attack. Robustness against attack is a key requirement for Watermarking and the success of this technology for copyright protection depends on this.
5. **Effect on bandwidth:** Watermarking should be done in such a way that it doesn't increase the bandwidth required for transmission. If Watermarking becomes a burden for the available bandwidth, the method will be rejected.
6. **Interoperability:** Digitally watermarked content shall still be interoperable so that it can be seamlessly accessed through heterogeneous networks and can be played on various play out devices that may be watermark aware or unaware.

1.5 Iris as a watermark

The iris of the eye has been described as the ideal part of the human body for biometric identification for several reasons:

- It is an internal organ that is well protected against damage and wear by a highly transparent and sensitive membrane (the cornea). This distinguishes it from fingerprints, which can be difficult to recognize after years of certain types of manual labour.

- The iris is mostly flat, and its geometric configuration is only controlled by two complementary muscles (the sphincter pupillae and dilator pupillae) that control the diameter of the pupil. This makes the iris shape far more predictable than, for instance, that of the face.

- The iris has a fine texture that—like fingerprints—is determined randomly during embryonic gestation. Like the fingerprint, it is very hard (if not impossible) to prove that the iris is unique. However, there are so many factors that go into the formation of these textures (the iris and fingerprint) that the chance of false matches for either is extremely low. Even genetically identical individuals have completely independent iris textures.

- An iris scan is similar to taking a photograph and can be performed from about 10 cm to a few meters away. There is no need for the person being identified to touch any equipment that has recently been touched by a stranger, thereby eliminating an objection that has been raised in some cultures against fingerprint scanners, where a finger has to touch a surface, or retinal scanning, where the eye must be brought very close to an eyepiece (like looking into a microscope).

- While there are some medical and surgical procedures that can affect the colour and overall shape of the iris, the fine texture remains remarkably stable over many decades. Some iris identifications have succeeded over a period of about 30 years.

1.6 Importance Of Digital Watermarking

The sudden increase in watermarking interest is most likely due to the increase in concern over copyright protection of content. The Internet had become user friendly with the introduction of Marc Andreessen's Mosaic web browser in November 1993, and it quickly became clear that people wanted to download pictures, music, and videos. The Internet is an excellent distribution system for digital media because it is inexpensive, eliminates warehousing and stock, and delivery

is almost instantaneous. However, content owners (especially large Hollywood studios and music labels) also see a high risk of piracy. This risk of piracy is exacerbated by the proliferation of high-capacity digital recording devices. When the only way the average customer could record a song or a movie was on analog tape, pirated copies were usually of a lower quality than the originals, and the quality of second-generation pirated copies (i.e., copies of a copy) was generally very poor. However, with digital recording devices, songs and movies can be recorded with little, if any, degradation in quality. Using these recording devices and using the Internet for distribution, would-be pirates can easily record and distribute copyright-protected material without appropriate compensation being paid to the actual copyright owners. Thus, content owners are eagerly seeking technologies that promise to protect their rights. The first technology content owners turn to is cryptography. Cryptography is probably the most common method of protecting digital content. It is certainly one of the best developed as a science. The content is encrypted prior to delivery, and a decryption key is provided only to those who have purchased legitimate copies of the content. The encrypted file can then be made available via the Internet, but would be useless to a pirate without an appropriate key. Unfortunately, encryption cannot help the seller monitor how a legitimate customer handles the content after decryption. A pirate can actually purchase the product, use the decryption key to obtain an unprotected copy of the content, and then proceed to distribute illegal copies. In other words, cryptography can protect content in transit, but once decrypted, the content has no further protection. Thus, there is a strong need for an alternative or complement to cryptography: a technology that can protect content even after it is decrypted. Watermarking has the potential to fulfil this need because it places information within the content where it is never removed during normal usage. Decryption, reencryption, compression, digital-to-analog conversion, and file format changes a watermark can be designed to survive all of these processes.
Watermarking has been considered for many copy prevention and copyright protection applications. In copy prevention, the watermark maybe used to inform software or hardware devices that copying should be restricted. In copyright protection applications, the watermark may be used to identify the copyright holder and ensure proper payment of royalties. Although copy prevention and copyright protection have been major driving forces behind research in the watermarking field, there is a number of other applications for which watermarking has been used or suggested. These include broadcast monitoring, transaction tracking, authentication (with direct analogy to our Rs100 example), copy control, and device control.

1.7 Applications Of Digital Watermarking

Digital Watermarks are potentially useful in many applications, including:

1. **Ownership assertion:** Watermarks can be used for ownership assertion. To assert ownership of an image, Alice can generate a watermarking signal using a secret private key, and then embed it into the original image. She can then make the watermarked image publicly available. Later, when Bob contends the ownership of an image derived from this public image, Alice can produce the unmarked original image and also demonstrate the presence of her watermark in Bob's image. Since Alice's original imageis unavailable to Bob, he cannot do the same. For such a scheme to work, the watermark has to survive image processing operations aimed at malicious removal. In addition, the watermark should be inserted in such a manner that it cannot be forged as Alice would not want to be held accountable for an image that she does not own.
2. **Fingerprinting**: In applications where multimedia content is electronically distributed over a network, the content owner would like to discourage unauthorized duplication and distribution by embedding a distinct watermark (or a fingerprint) in each copy of the data. If, at a later point in time, unauthorized copies of the data are found, then the origin of the copy can be determined by retrieving the fingerprint. In this application the watermark needs to be invisible and must also be invulnerable to deliberate attempts to forge, remove or invalidate. Furthermore, and unlike the ownership assertion application, the watermark should be resistant to collusion. That is, a group of k users with the same image but containing different fingerprints should not be able to collude and invalidate any fingerprint or create a copy without any fingerprint.
3. **Copy prevention or control:** Watermarks can also be used for copy prevention and control. For example, in a closed system where the multimedia content needs special hardware for copying and/or viewing, a digital watermark can be inserted indicating the number of copies that are permitted. Every time a copy is made the watermark can be modified by the hardware and after a point the hardware would not create further copies of the data. An example of such a system is the Digital Versatile Disc (DVD). In fact, a copy protection mechanism that includes digital watermarking at its core is currently being considered for standardization and second generation DVD players may well include the ability to read watermarks and act based on their presence or absence. Another example is in digital cinema, where information can be embedded as a watermark in every frame or a sequence of frames to help investigators locate the scene of the piracy more quickly and point out weaknesses in security in the movie's distribution. The information could include data such as the name of the theatre and the date and time of the screening. The technology would be most useful in fighting a form of piracy that's surprisingly

common, i.e., when someone uses a camcorder to record the movie as it's shown in a theatre, then duplicates it onto optical disks or VHS tapes for distribution.

4. **Fraud and tamper detection:** When multimedia content is used for legal purposes, medical applications, news reporting, and commercial transactions, it is important to ensure that the content was originated from a specific source and that it had not been changed, manipulated or falsified. This can be achieved by embedding a watermark in the data. Subsequently, when the photo is checked, the watermark is extracted using a unique key associated with the source, and the integrity of the data is verified through the integrity of the extracted watermark. The watermark can also include information from the original image that can aid in undoing any modification and recovering the original. Clearly a watermark used for authentication purposes should not affect the quality of an image and should be resistant to forgeries. Robustness is not critical as removal of the watermark renders the content inauthentic and hence of no value.
5. **ID card security:** Information in a passport or ID (e.g., passport number, person's name, etc.) can also be included in the person's photo that appears on the ID. By extracting the embedded information and comparing it to the written text, the ID card can be verified. The inclusion of the watermark provides an additional level of security in this application. For example, if the ID card is stolen and the picture is replaced by a forged copy, the failure in extracting the watermark will invalidate the ID card. The above represent a few example applications where digital watermarks could potentially be of use. In addition there are many other applications in rights management and protection like tracking use of content, binding content to specific players, automatic billing for viewing content, broadcast monitoring etc. From the variety of potential applications exemplified above it is clear that a digital watermarking technique needs to satisfy a number of requirements. Since the specific requirements vary with the application, watermarking techniques need to be designed within the context of the entire system in which they are to be employed. Each application imposes different requirements and would require different types of invisible or visible watermarking schemes or a combination thereof. In the remaining sections of this chapter we describe some general principles and techniques for invisible watermarking. Our aim is to give the reader a better understanding of the basic principles, inherent trade-offs, strengths, and weakness, of digital watermarking. We will focus on image watermarking in our discussions and examples. However as we mentioned earlier, the concepts involved are general in nature and can be applied to other forms of content such as video and audio.
6. **Broadcasting Monitoring**: Commercials are aired by broadcasting channels and stations. For this advertising firm purchase airtime from broadcasting channel. There are several organizations and individuals interested in broadcasting monitoring, viz. advertiser, to ensure if

his commercial is broadcasted for all of his purchased airtime, performers, who want to ensure that they get the royalties due to them from advertising firm and owners of copyrighted works, who want to ensure that their property is not illegally rebroadcasted by pirate stations. One solution to the problem is human observers watching the broadcasting which is neither a feasible nor practically possible solution. The other solution is to match the signal with the signals present in databases to ascertain advertisers that messages are broadcasted. But matching signals from databases is very complex process and require large amount of time and money. The last solution is using watermarking techniques. It has advantage of existing within content itself, rather than exploiting a particular segment of the broadcast signal, and is therefore completely compatible with the installed base of broadcast equipment, including both digital and analog transmission.

1.8 Attacks On Watermarked Work

Below are some significant known attacks: -

1. **Scrambling Attacks:** A scrambling attack is a system-level attack in which the samples of a Work are scrambled prior to presentation to a watermark detector and then subsequently descrambled. The type of scrambling can be a simple sample permutation or a more sophisticated pseudo-random scrambling of the sample values. The degree of scrambling necessary depends on the detection strategy. A well-known scrambling attack is the mosaic attack, in which an image is broken into many small rectangular patches, each too small for reliable watermark detection. These image segments are then displayed in a table such that the segment edges are adjacent. The resulting table of small images is perceptually identical to the image prior to subdivision. This technique can be used in a web application to evade a web-crawling detector. The scrambling is simply the subdivision of the image into sub images, and the descrambling is accomplished by the web browser itself.
2. **Pathological Distortions:** For a watermark to be secure against unauthorized removal, it must be robust to any process that maintains the fidelity of the Work. This process may be a normal process, in which case we are requiring that a secure watermark be robust. However, it may also be a process unlikely to occur during the normal processing of the Work. Any process that maintains the fidelity of the Work could be used by an adversary to circumvent the detector by masking or eliminating the watermark. The two most common categories of such pathological distortions, geometric/temporal distortions (attacks on synchronization) and noise removal distortions.

- Synchronization Attacks: Many watermarking techniques are sensitive to synchronization. By disturbing this synchronization, an adversary attempts to mask the watermark signal. Examples of simple synchronization distortions include delay and time scaling for audio and video, and rotation, scaling, and translation for images and video. These simple distortions can be implemented such that they vary over time or space. More complex distortions include pitch-preserving scaling and sample removal in audio, and shearing, horizontal reflection, and column or line removal in images. Even more complex distortions are possible, such as nonlinear warping of images.
- Linear filtering and noise removal attack: Linear filtering also can be used by an adversary in an attempt to remove a watermark. For example, a watermark with significant energy in the high frequencies might be degraded by the application of a low-pass filter. In addition, any watermarking system for which the added pattern is "noise like" is susceptible to noise-removal techniques.

3. **Copy Attacks:** A copy attack occurs when an adversary copies a watermark from one Work to another. As such, it is a form of unauthorized embedding. The copy attack attempts to thwart the effectiveness of such systems by estimating the watermark given in an originally watermarked piece of media, and then adding that watermark to an un-watermarked piece. In the first scenario listed above, this would allow an attacker to have an inauthentic image be declared authentic, since it contains a watermark. In the second scenario, an attacker could flood the market with content which ordinarily would allow a user to manipulate it as he saw fit, but due to the presence of the watermark, limitations would be imposed. In this way, schemes which sought to limit use of watermarked media may prove to be too unpopular for wide distribution.
4. **Ambiguity Attacks:** Ambiguity attacks create the appearance that a watermark has been embedded in a Work when in fact no such embedding has taken place. An adversary can use this attack to claim ownership of a distributed Work. He or she may even be able to make an ownership claim on the original Work. As such, ambiguity attacks can be considered a form of unauthorized embedding. However, they are usually considered system attacks.

CHAPTER 2 - THEORETICAL FRAMEWORK

2.1 Zernike Moments

The basic reason of using Zernike Moment here is due to its significant advantage of being robust to noise, rotation invariance and computational efficiency. Zernike moments are calculated by mapping an image function onto a basis set of orthogonal functions which is nothing but a unit disc in the polar coordinates .The basis function of Zernike Moment is calculated by

$$V_{n,m}(x,y) = R_{n,m}(\rho)e^{jm\theta}$$

where we have n as the order, m as the repetition, both m and n are non-negative integers following the condition n-|m| is even and greater than equal to 0, ρ is the vector length of (x, y) from the origin and Θ is the angle between the vector ρ and the x-axis taken counterclockwise. R is the radial polynomial given by

$$\mathrm{R}_{n,m}(\rho) = \sum_{k=|m|,n-k=even}^{n} \frac{(-1)^{n-k/2}\,\frac{n+k}{2}!}{\frac{n-k}{2}!\frac{k+m}{2}!\frac{k-m}{2}!}$$

Following this, we calculate Zernike moment as

$$Z_{n,m} = \frac{n+1}{\pi}\sum\sum_{x^2+y^2\le 1} f(x,y)V^{*}_{n,m}(x,y)$$

2.2 The Wavelet Transform

An oscillating function of time or space which when has a periodic nature is called a wave. The wavelet transform provides a time-frequency representation of a signal which may be either 1-D or 2-D. Simply stated, the wavelet transform allows a signal to be decomposed into a coarser and finer approximation such that each decomposed approximation can be represented as the sum of coarser signal components plus a detailed signal coefficient. Let x (t) be a 1-D signal, its wavelet transform is given by:

$$W_f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\varphi^*\left(\frac{x-b}{a}\right) dt$$

$$W_f(a,b) = \int_{-\infty}^{\infty} s(t)h^*{}_{ab}(t)dt$$

$$\varphi_{ab}(x) = \frac{1}{\sqrt{a}}\varphi\left(\frac{x-b}{a}\right)$$

$\varphi(t)$ is in general the mother wavelet and the basis function called daughter wavelets are given by eq.3

From equation 1, 2 and 3 we can deduct that

$$\varphi_{ab}(x) = f(\varphi(t), \alpha, \beta)$$

Where α is the parameter used either for dilation or compression (also called as the scaling parameter). Its value varies from 0 to ∞.

If, α<1, signal is compressed.

α>=1, signal is dilated.

b is known as the shift parameter and is used to achieve temporal translation.

Extending the above equations to 2-D. Let f (x,y) be a 2-D image, its wavelet transform is given by

$$\frac{1}{\sqrt{a}} \iint f(x,y)\varphi\left(\frac{x-m}{a}, \frac{y-n}{b}\right) dx\, dy$$

Where m and n are shifting parameters.

The kernel of the wavelet transform can be either continuous obtained by dilation and translation of a bandpass filter or discrete obtained by discretizing the dilation and translation parameters. Computation of DWT can be carried out either using convolution based (filter bank) procedures or lifting based procedures as proposed by Swelden [11].

The lifting wavelet transform is widely used in signal processing because of its efficient implementation with low memory and computational complexity. [12]The process of lifting wavelet transform and inverse lifting wavelet transform can be represented as

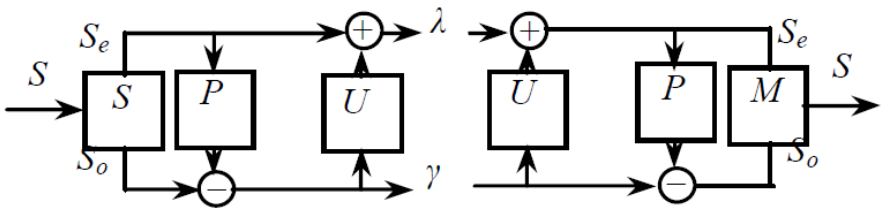


Fig 2.1 Lifting Wavelet Transform

The lifting algorithm consists of three phases: Split, Predict and Update. The 2-D image S(m,n) is split into its wavelet coefficients.

- Split: The signal is divided into two parts which are half the size of original. The two sets, even series and odd series are disjoint.

$$S_e(m,n) = S(m,2n)$$
$$S_o(m,n) = S(m,2n+1)$$

- Predict: Odd coefficients are predicted by their neighbouring even coefficients using a predict operator P, and the error produced are called high pass wavelet coefficient.

$$\gamma(m,n) = S_o(m,n) - P[S_e(m,n)]$$

The odd samples can be recovered in inverse wavelet transform by

$$S_o(m,n) = \gamma(m,n) + P[S_e(m,n)]$$

- Update: The even series is updated with the value U(m,n) to produce a low pass wavelet coefficient , λ(m,n)

$$\lambda(\mathrm{m},\mathrm{n}) = S_e(m,n) + U(m,n)$$

In inverse transform, even series is acquired by inverse updating.

$$S_e(m,n) = \lambda(\mathrm{m},\mathrm{n}) - U(m,n)$$

2.3 Singular Value Decomposition

SVD is a method for data reduction which exposes the underlying structure of the matrix under consideration by factorizing it into a series of linear approximations.

The above objective is achieved by reducing a high dimensional highly variable set of data points to a lower dimensional space in a manner that the substructure of the original data points is exposed more clearly and ordering is done from most variant data to least variant one. It enables us to represent data with a smaller value set in such a way that the most distinguishing and useful data points features are preserved and the storage space required is reduced considerably.

This introduction to SVD leads us to three views of this technique.

A) It is a mathematical tool used for exposing the various data relations by bringing out the non-correlation among the data points which are originally correlated.
B) It provides a way of slicing and dicing the data in a manner such that the data points within a slice exhibit least variation and within multiple slices exhibits most variation.
C) Once maximum variation point is identified, SVD helps in approximating the data into fewer dimensions.

Mathematically, SVD is based on the theorem that any matrix $\alpha \in R^{m*n}$ can be broken down as the product of three matrices.

$$X_{m*n} = U_{m*m} S_{m*n} V^T_{n*n}$$

Where $U \in R^{m*m}$ and $V \in R^{n*n}$ are orthogonal matrices such that the columns of U are orthonormal Eigen vectors of AA^T whereas the columns of V are orthonormal eigenvector of $A^T A$.

$S \in R^{m*n}$ is a diagonal matrix denoted as σ_i, i=1.....p, where p=min(n,m) and contains the square roots of the non zero Eigen values of both U and V and satisfying the criteria:

$$\sigma_1 \geq \sigma_2 \geq .....6p$$

CHAPTER 3 - METHODOLGY

3.1 Feature Extraction

The central idea is to extract the features from a constraint set of image values i.e.

If g(ρ, Θ) ={ f(x, y): ρ=$\sqrt{x^2 + y^2}$ and Θ =$\tan^{-1}(\frac{y}{x})$}, then Zn, m={h(g(ρ, Θ)): ρ<=1}.H is some function of g.

The proposed algorithm for calculating Zernike Moment is as follows:

- The image is converted to a binarized image. Binarization can be represented as:

  If the threshold value=t, then the binarization function b(x, y) for the input image function f(x, y) will be

  B(x, y) =1 if f(x, y)>=t and 0 if B(x, y)= f(x, y)< t

  The result of binarization is as follows:

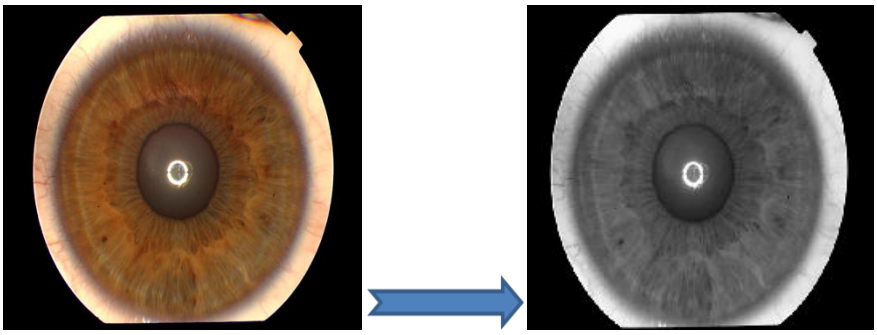


Fig 3.1 Binarization of image

- The result image is mapped onto a unit disc in polar coordinates in such a way so that the center of the image coincides with the center of the unit disc. Our calculations are limited to the area within this unit circle.

  The function used here will be mapped as (x, y) -> (ρ, Θ) where ρ<=1 and 0<= Θ<=360 (degrees) and is represented as

  g(ρ, Θ) ={ f(x, y): ρ=$\sqrt{x^2 + y^2}$

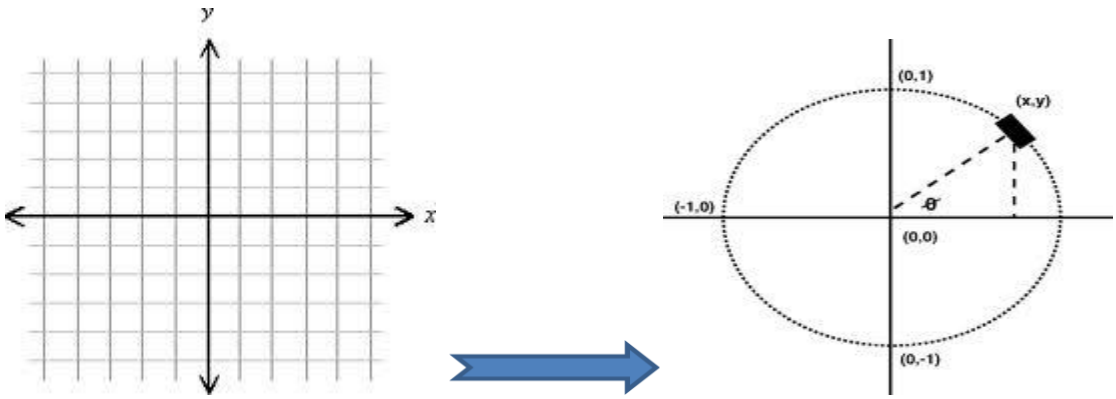


Fig 3.2 Mapping into polar coordinates

- The image is then normalized by re-sampling the image to a preset size.

  If $\max(g(\rho, \Theta))= m_1$ and $\min(g(\rho, \Theta))= m_2$, then the normalization function $n(\rho, \Theta)$ will be

  $n(\rho, \Theta) =(g(\rho, \Theta)-m_2)/m_1$

- Now this image is used to extract the Zernike moments as follows:
  - The basis function is calculated with the help of radial polynomial
  - Zernike Moment is obtained by projecting image function on this polynomial and the magnitudes thus obtained represent the features of the input image.

3.2 Watermark Embedding

The host image (H) of size (m*m) is used to embed an iris watermark (W) of size (x*y).The proposed watermark embedding procedure is depicted as follows:

- The host image is decomposed into sub-bands using lifting wavelet transform (LWT). The number of selected decomposition level (l), determines the size of sub-bands i.e. $m/2^l$ .Subsequently, the image is divided into high and low frequency components depicting the edges and smooth areas, respectively.
- The iris image, W of size (x*y) is resized to (m*m), i.e. the size of watermark image is made equal to the host image.

$$size(W) = size(H)$$

- The watermark image is also decomposed into sub-bands using lifting wavelet transform.
- The original host image and watermark iris image is decomposed into two levels, i.e. l=2.

- The sub-band with horizontal details is selected and Singular Value Decomposition (SVD) is applied to both the host and watermark image.
- A modified LWT coefficient is attained at the second level of decomposition by altering the Singular Values of the host image by watermark iris image.

$$[I(2,h)]_{singular} = [H(2,h)]_{singular} + p*[W(2,h)]_{singular}$$

- The embedded image is constructed by inverse Lifting Wavelet Transform of the modified image.

3.3 Watermark Extraction

This scheme of watermarking is non-blind as it uses singular values of the original host image for extraction as key. The proposed extraction procedure is described as follows:

- The watermarked image is decomposed to sub-bands with previously selected, levels of decomposition (l).
- The horizontal sub-band is used to extract the singular values of the iris image.

$$\sum S_{wm} = (I(2,h)_{singular} - H(2,h)_{singular})/p$$

- The singular value thus obtained is used to recover the watermark iris image.
- Inverse lifting wavelet transform is done to obtain the watermark iris image.

3.4 Hardware Requirements

We are using **Matlab 7.6** .

Advantages:

- Excellent graphics
- Image as matrix: operation of images without "for" loop
- Independent of platform
- Mathematical transformation and analysis, combined with Image Processing Toolbox and Signal Processing Toolbox

CHAPTER 4 - RECOGNITION

Overview

The extracted iris image is subjected to the procedure of extraction of features for template matching. The database used here contains the iris images of 16 persons, 6 images of each-3 of left eye and 3 of right eye. The steps of feature extraction are applied on the entire database generating a feature vector for each image. The same algorithm extracts the features of the iris image used to watermark. The feature vector so generated is compared with the feature vector of the database .K-means clustering is used to match the recovered watermark image with the set of all centered images in the training database. The cluster having the minimum distance with watermarked image is calculated and the image index corresponding to the minimum distance within the cluster is recorded. This image is read from the database and the user is verified.

4.1 Theoretical Framework

In data mining, k-means clustering is a method of cluster analysis which aims
to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early grouping is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.
Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centres.

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centres. The k-means algorithm can be run multiple times to reduce this effect.

4.2 Proposed Technique

- Randomly choose k items and make them as initial centroids.
- For each point, find the nearest centroid and assign the point to the cluster associated with the nearest centroid.
- Update the centroid of each cluster based on the items in that cluster. Typically, the new centroid will be the average of all points in the cluster.
- Repeats steps 2 and 3, till no point switches clusters.
- Store each cluster in a different database.
- Find the centroid of each cluster.
- Find the cluster which is closest to the iris image which has to be matched.
- Inside that cluster, find the iris image closest to the iris image to be matched.
- Return that image.

CHAPTER 5 - ATTACKS ON WATERMARKED IMAGE

The attack methodology on the watermarked image is based on the idea that an attacker does not have any access to the original image or the watermark image. The attacks are, therefore, done on the watermarked image using only the watermarked image as input. The individual, i.e. attacker, likely has no idea if the attack worked or not so the results are not known to the attacker.

5.1 Histogram Equalization

Histogram modeling techniques (e.g. histogram equalization) provide a sophisticated method for modifying the dynamic range and contrast of an image by altering that image such that its intensity histogram has a desired shape. Unlike contrast stretching, histogram modeling operators may employ non-linear and non-monotonic transfer functions to map between pixel intensity values in the input and output images. Histogram equalization employs a monotonic, non-linear mapping which re-assigns the intensity values of pixels in the input image such that the output image contains a uniform distribution of intensities (i.e. a flat histogram). This technique is used in image comparison processes (because it is effective in detail enhancement) and in the correction of non-linear effects introduced by, say, a digitizer or display system.

This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique and an invertible operator. So in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal.

In scientific imaging where spatial correlation is more important than intensity of signal (such as separating DNA fragments of quantized length), the small signal to noise ratio usually hampers visual detection.

There are two ways to think about and implement histogram equalization, either as image change or as palette change. The operation can be expressed as P(M(I)) where I is the original image, M is histogram equalization mapping operation and P is a palette. If we define new palette as P'=P(M) and leave image I unchanged then histogram equalization is implemented as palette change. On the other hand if palette P remains unchanged and image is modified to I'=M(I) then the implementation is by image change. In most cases palette change is better as it preserves the original data.

Histogram equalization also seems to be used in biological neural networks so as to maximize the output firing rate of the neuron as a function of the input statistics. This has been proved in particular in the fly retina. Histogram equalization is a specific case of the more general class of histogram remapping methods. These methods seek to adjust the image to make it easier to analyze or improve visual quality (e.g. retinex).

Let f be a given image represented as a m_r by m_c matrix of integer pixel intensities ranging from 0 to L − 1. L is the number of possible intensity values, often 256. Let p denote the normalized histogram of f with a bin for each possible intensity. So

$$pn = \frac{\text{number of pixels with intensity n}}{\text{total number of pixels}} \qquad n = 0,1,2,\ldots . L-1$$

The histogram equalized image g will be defined by

$$g(i,j) = floor((L-1)\sum_{n=0}^{f(i,j)} pn)$$

(1)

where floor() rounds down to the nearest integer. This is equivalent to transforming the pixel intensities, k, of f by the function

$$T(k) = floor((L-1)\sum_{n=0}^{k} pn)$$

The motivation for this transformation comes from thinking of the intensities of f and g as continuous random variables X, Y on [0, L − 1] with Y defined by

$$Y=T(X) = floor(L-1)\int_0^X px(x)dx$$

(2)

where p(X) is the probability density function of f. T is the cumulative distributive function of X multiplied by (L − 1). Assume for simplicity that T is differentiable and invertible. It can then be shown that Y defined by T(X) is uniformly distributed on [0, L − 1], namely that $py(y) = \frac{1}{L-1}$

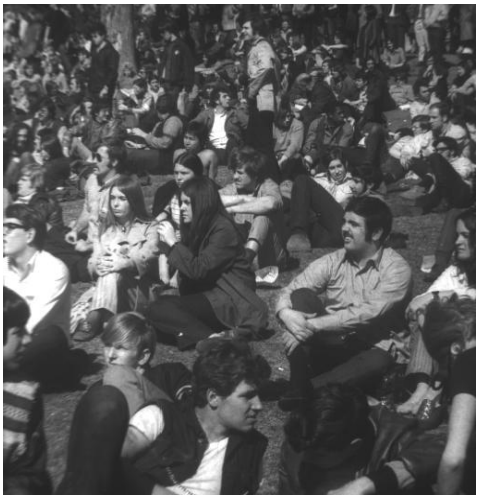
$$\int_0^y py(z)dz = probability\ that\ 0 \le Y \le y$$

$= probability\ that\ 0 \le X \le T^{-1}(y)$

$= \int_0^{T^{-1}(y)} px(w)dw$

$$\frac{d}{dy}\left(\int_0^y py(z)dz\right) = py(y) = px\left(T^{-1}(y)\right)\frac{d}{dy}(T^{-1}(y))$$

Original image



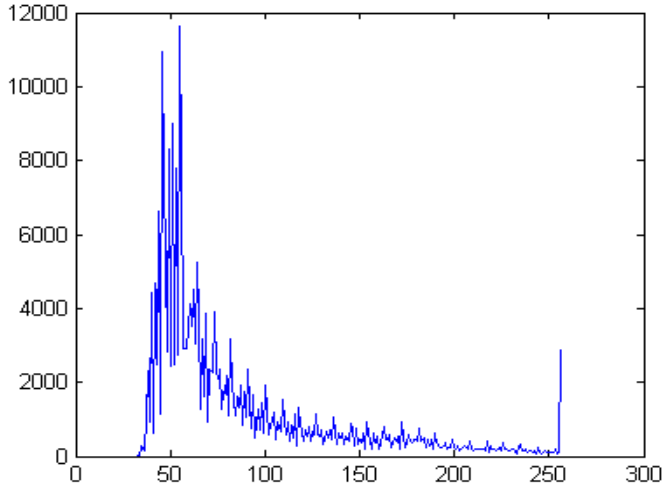


Fig 5.1(a) Original Image

Fig 5.1(b) Histogram

Transformed image

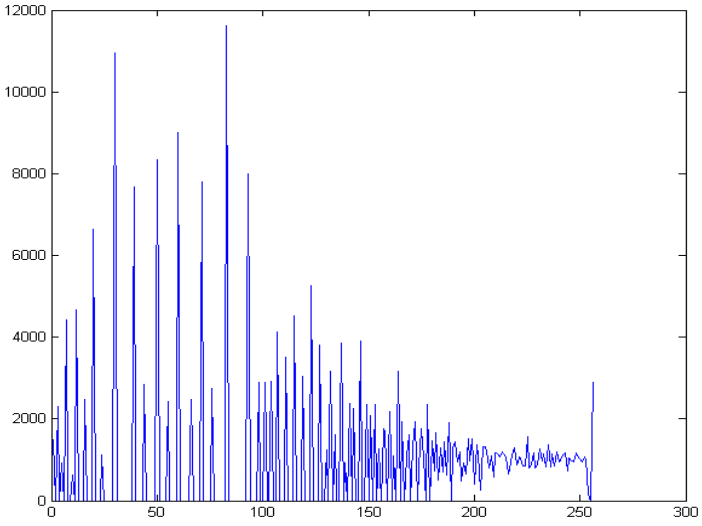


Fig 5.2(a) Transformed Image

Fig 5.2(b) Histogram

Histogram Equalization applied to low contrast image

Note that our discrete histogram is an approximation of px(x) and the transformation in Equation 1 approximates the one in Equation 2. While the discrete version won't result in exactly flat histograms, it will flatten them and in doing so enhance the contrast in the image.

The main idea is to redistribute the gray-level values uniformly.

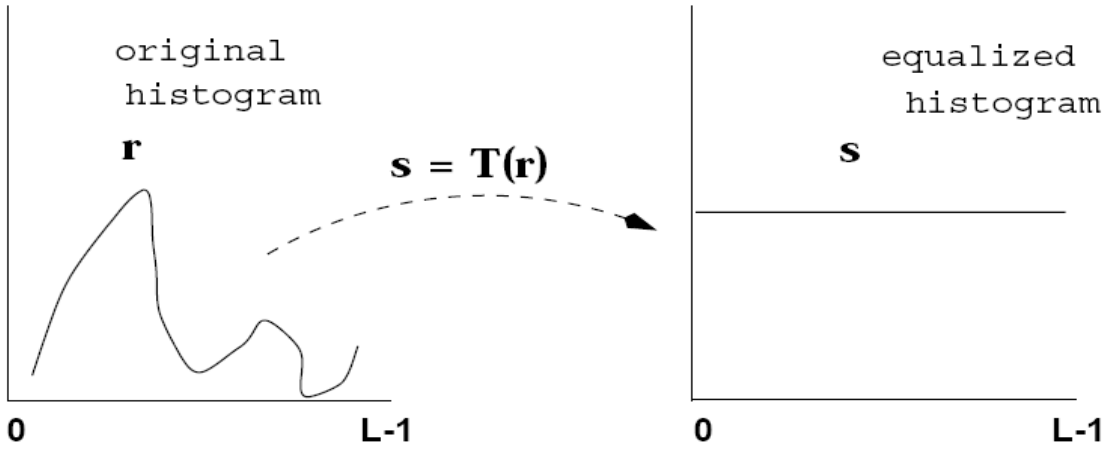


Fig 5.3 Histogram Equalization

5.2 Median Filtering

Mean and median filters are simple burring functions of image processing software. The resistance of a watermarked algorithm against mean and median filter depends largely on where the watermark information is embedded. High frequency edge embedding will likely suffer from mean and median filters while low frequency intensity embedding will remain relatively resistant to such filter attacks. This study will verify this hypothesis by looking at algorithms that embed in the high frequency versus lower frequency.

The ith order statistic of a set of n elements is defined as the ith smallest element. Assuming n is odd the median is the (n + 1) =2 smallest element. Median filtering is done by replacing the value of each element by the median found in a window around the element. This question leads us to the issue of robust estimation. When estimating a location value we would like our estimate to be robust. An indicator of an estimator's robustness is its breakdown point. The breakdown point is defined as the smallest fraction of the observations which when replaced with outliers will throw the estimator outside of reasonable bounds.

By calculating the median value of a neighbourhood rather than the mean filter, the median filter has two main advantages over the mean filter:

The median is a more robust average than the mean and so a single very unrepresentative pixel in a neighbourhood will not affect the median value significantly.

Since the median value must actually be the value of one of the pixels in the neighbourhood, the median filter does not create new unrealistic pixel values when the filter straddles an edge. For this reason the median filter is much better at preserving sharp edges than the mean filter.

5.3 Jpeg Compression

If the watermarked image is not already in JPEG format, the attacker can simply convert the watermarked image into a JPEG, varying the "quality factor" of JPEG compression to as low as he can before the features he needs on the image deteriorates. Even if the watermarked image is already a JPEG, the attacker can resave as a JPEG using a lower quality factor. This attack is a simple one without the need for complicated image processing software and many image viewers, available online, is able to save JPEG files using different quality factors. Because of how common and easy the JPEG attack is, resistance to JPEG compression is treated as the most important criteria in this assessment of robustness.

JPEG compression algorithms may fall in to one of several categories depending on how the compression is actually performed:

a. Baseline/Sequential
b. Lossless
c. Progressive
d. Hierarchical
e. "Motion JPEG" - Baseline JPEG applied to each image in a video.

The Major Steps in JPEG Compression involve:

1. DCT (Discrete Cosine Transformation)
2. Quantization
3. Zigzag Scan
4. DPCM on DC component
5. RLE on AC Components
6. Entropy Coding

1. DCT

The basic operation of the DCT is as follows:

The input image is N by M. f(i,j) is the intensity of the pixel in row i and column j. F(u,v) is the DCT coefficient in row k1 and column k2 of the DCT matrix. For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT. Compression is achieved since the lower right values represent higher frequencies, and are often small - small enough to be neglected with little visible distortion. The DCT input is an 8 by 8 array of integers. This array contains each pixel's gray scale level 8 bit pixels have levels from 0 to 255.Therefore an 8 point DCT would be:

$$\lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & for\ \xi = 0, otherwise \\ 1 \end{cases}$$

F[0,0] define DC and AC components. The output array of DCT coefficients contains integers; these can range from -1024 to 1023.It is computationally easier to implement and more efficient to regard the DCT as a set of basis functions which given a known input array size (8 x 8) can be precomputed and stored. This involves simply computing values for a convolution mask (8 x8 window) that get applied.

2. QUANTIZATION

Uniform Quantization

Divide by constant *N* and round result (*N* = 4 or 8 in examples above).Non powers-of-two gives fine control (e.g., *N* = 6 loses 2.5 bits).

Quantization Tables

In JPEG, each F[u,v] is divided by a constant q(u,v).Table of q(u,v) is called *quantization table*.

16 11 10 16 24 40 51 61

12 12 14 19 26 58 60 55

14 13 16 24 40 57 69 56

14 17 22 29 51 87 80 62

18 22 37 56 68 109 103 77

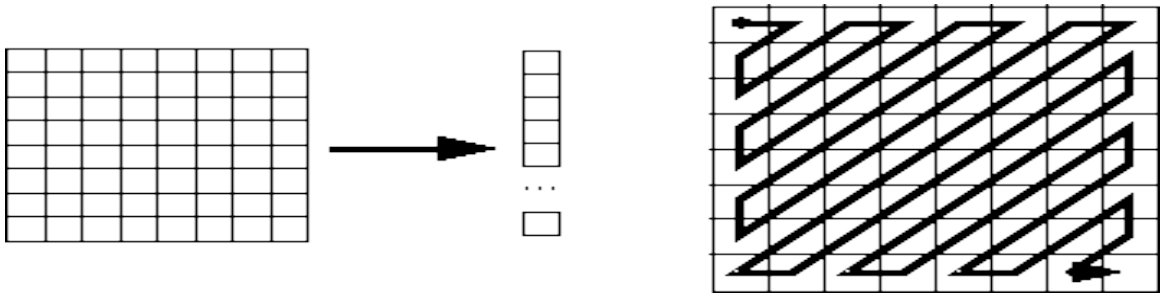
24 35 55 64 81 104 113 92

49 64 78 87 103 121 120 101

72 92 95 98 112 100 103 99

Eye is most sensitive to low frequencies (upper left corner), less sensitive to high frequencies (lower right corner).Standard defines 2 default quantization tables, one for luminance (above), one for chrominance. Quality factor in most implementations is the scaling factor for default quantization tables. Custom quantization tables can be put in image/scan header.

3. ZIGZAG SCAN



4. DIFFERENTIAL PULSE CODE MODULATION ON DC COMPONENT

Here we see that besides DCT another encoding method is employed: DPCM on the DC component at least. DC component is large and varied, but often close to previous value (like lossless JPEG).Encode the difference from previous 8x8 blocks – DPCM.

5. RUN LENGTH ENCODE ON AC COMPONENT

1x64 vector has lots of zeros in it. Encode as (*skip, value*) pairs, where *skip* is the number of zeros and *value* is the next non-zero component. Send (0,0) as end-of-block sentinel value.

6.ENTROPY ENCODING

DC and AC components finally need to be represented by a smaller number of bits. Categorize DC values into SSS (number of bits needed to represent) and actual bits.

Value SSS

0 0

-1,1 1

-3,-2,2,3 2

-7..-4,4..7 3

Example: if DC value is 4, 3 bits are needed.

Send off SSS as Huffman symbol, followed by actual 3 bits. For AC components (*skip, value*), encode the composite symbol (*skip,SSS*) using the Huffman coding. Huffman Tables can be custom (sent in header) or default.

5.4 Gaussian Noise

It refers to introducing Gaussian noise into the watermarked image. Gaussian noise is additive in nature, is independent at each pixel and independent of the signal intensity, It is caused primarily by Johnson–Nyquist noise (thermal noise), including that which comes from the reset noise of capacitors ("kTC noise"). In color cameras where more amplification is used in the blue color channel than in the green or red channel, there can be more noise in the blue channel.

The probability distribution function of Gaussian noise describes stochastic processes as the random voltage variations in a carbon resistor due to thermal motion, or the so-called Brownian motion discovered The formula for the distribution implies that large deviations from the mean become less probable according to exp(-x2). When an electrical variation obeys a Gaussian distribution, such as in the case of thermal motion cited above, it is called Gaussian noise or Random noise. Other examples occur with some types of radio tubes or semi-conductors where the noise may be amplified to produce a noise generator. Note that in all of these cases; it is only the signal's amplitude fluctuating randomly that result in its being classified as Gaussian noise.

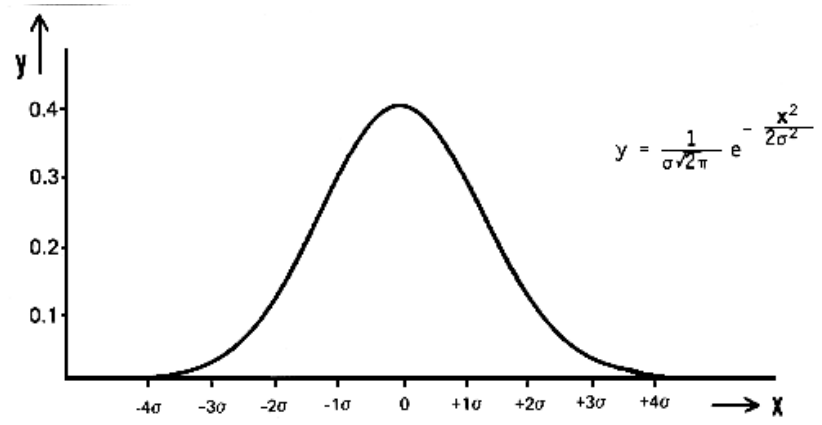


Fig 5.4 Probablilty Distribution Function of Gaussian noise

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

The parameter μ is the mean or expectation (location of the peak) and σ^2 is the variance. σ^2 is known as the standard deviation. The distribution with μ = 0 and σ = 1 is called the standard normal distribution or the unit normal distribution.

5.5 Salt And Pepper Noise

Another common form of noise is data drop-out noise (commonly referred to as intensity spikes, speckle or salt and pepper noise). The corrupted pixels are either set to the maximum value (which looks like snow in the image) or have single bits flipped over. In some cases, single pixels are set alternatively to zero or to the maximum value, giving the image a `salt and pepper' like appearance. Unaffected pixels always remain unchanged. The noise is usually quantified by the percentage of pixels which are corrupted. SALT-AND-PEPPER noise is a special case of impulse noise, where a certain percentage of individual pixels.

In digital image are randomly digitized into two extreme intensities. Normally, these intensities being the maximum and minimum intensities. The contamination of digital image by salt-and-pepper noise is largely caused by error in image acquisition and/or recording. For example, faulty memory locations or impaired pixel sensors can result in digital image being corrupted with salt-and-pepper noise.

The addition of salt-and-pepper noise in an image results in great difficulty in certain image processing tasks such as edge detection or segmentation is carried out. This is because the occurrence of salt-and-pepper noise can severely damage the information or data embedded in the original image.

For an 8bit/pixel image, the typical intensity value or pepper noise is close to 0 and for salt noise is close to 255.

I(t) = (1 − e)S(t) + eN(t) where e = {0, 1} with a probability P

$$PDF = \begin{cases} A\ for\ g = a(\text{"}pepper\text{"}) \\ B\ for\ g = b(\text{"}salt\text{"}) \end{cases}$$

5.6 Image Cropping Attack

Cropping: In some cases, infringers are just interested by the “central" part of the copyrighted material, moreover more and more Web sites use image segmentation, which is the basis of the “Mosaic" attack. This is of course an extreme case of cropping.

Scaling: This happens when a printed image is scanned or when a high resolution digital image is used for electronic applications such as Web publishing. This should not be neglected as we move more and more toward Web publishing. Scaling can be divided into two groups, uniform and non-uniform scaling. Under uniform scaling we understand scaling which is the same in horizontal and vertical direction. Non-uniform scaling uses different scaling factors in horizontal and vertical direction (change of aspect ratio). Very often digital watermarking methods are resilient only to uniform scaling.

5.7 Sharpening Attack

Sharpening an image increases the contrast between bright and dark regions to bring out features.

The sharpening process is basically the application of a high pass filter to an image. The following array is a kernel for a common high pass filter used to sharpen an image:

$$\begin{bmatrix} -1/9 & -1/9 & -1/9 \\ -1/9 & 1 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{bmatrix}$$

5.8 Image Noise Measures

PSNR:

The phrase peak signal-to-noise ratio, often abbreviated PSNR, is A term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

The PSNR is most commonly used as a measure of quality of reconstruction of lossy compression codecs (e.g., for image compression). The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs it is used as an approximation to human perception of reconstruction quality, therefore in some cases one reconstruction may appear to be closer to the original than another, even though it has a lower PSNR (a higher PSNR would normally indicate that the reconstruction is of higher quality). One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content.

The PSNR is defined as:

$$PSNR = 20.\log_{10}(MAX_I) - 10.\log_{10}(MSE)$$

Here, MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linear PCM with B bits per sample, MAXI is $2^B - 1$. For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three. Alternately, for color images the image is converted to a different color space and PSNR is reported against each channel of that color space, e.g., YCbCr or HSL.

Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, where higher is better. Acceptable values for wireless transmission quality loss are considered to be about 20 dB to 25 dB. When the two images are identical, the MSE will be zero. For this value the PSNR is undefined (see Division by zero).

SSIM:

The structural similarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index is a full reference metric, in other words, the measuring of image quality based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods like peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proved to be inconsistent with human eye perception.

The difference with respect to other techniques mentioned previously such as MSE or PSNR, is that these approaches estimate perceived errors on the other hand SSIM considers image degradation as perceived change in structural information. Structural information is the idea that the

pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene.

The SSIM metric is calculated on various windows of an image. The measure between two windows x and y of common size N×N is:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

with

μ_x the average of x;

μ_y the average of y;

σ_x^2 the variance of x;

σ_y^2 the variance of y;

σ_{xy} the covariance of x and y;

$c_1=(k_1 L)^2$, $c_2=(k_2 L)^2$ two variables to stabilize the division with weak denominator;

L the dynamic range of the pixel-values (typically this is $2^{\#bits\ per\ pixel}-1$);

$k_1=0.01$ and $k_2=0.03$ by default.

In order to evaluate the image quality this formula is applied only on luma. The resultant SSIM index is a decimal value between -1 and 1, and value 1 is only reachable in the case of two identical sets of data. Typically it is calculated on window sizes of 8×8. The window can be displaced pixel-by-pixel on the image but the authors propose to use only a subgroup of the possible windows to reduce the complexity of the calculation.

Structural dissimilarity (DSSIM) is a distance metric derived from SSIM.

$$\mathrm{DSSIM}(x,y) = \frac{1}{1 - \mathrm{SSIM}(x,y)}$$

CHAPTER 6 – RESULT AND CONCLUSION

6.1 Result

- We have taken a database consisting of 48 iris images in total of 16 persons, 6 images of each-3 of left eye and 3 of right eye.
- The result obtained after applying Zernike moment on an iris image is a feature matrix of reduced size 60*30 containing complex values.
- The result after performing watermarking on a Lena image using Zernike moment is as follows:

a. Before watermarking b. After watermarking

Fig 6.1 Watermarking

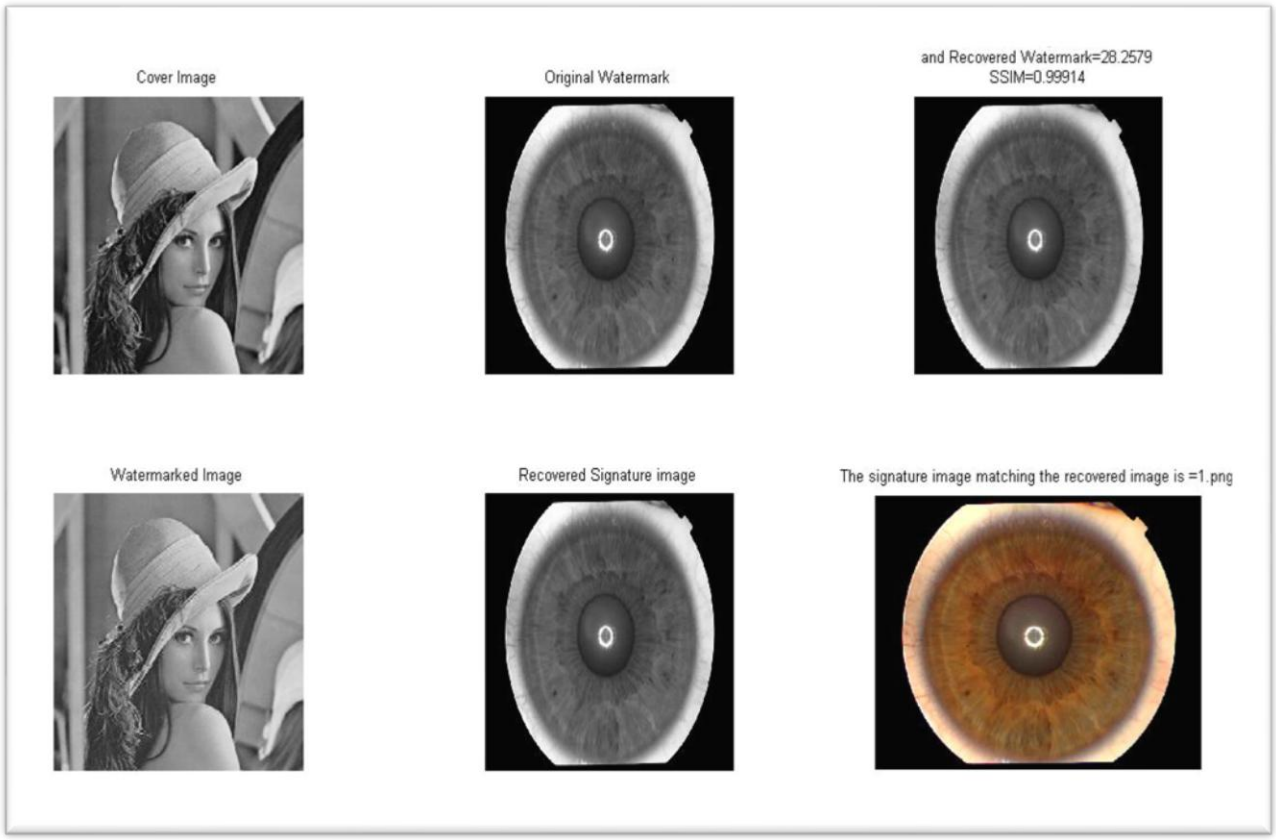


Fig 6.2 Screenshot

- PSNR and SSIM between each iris image and extracted iris image is calculated and plotted. The PSNR of images ranged between 27dB to 32dB. The SSIM of images are all above 0.993.

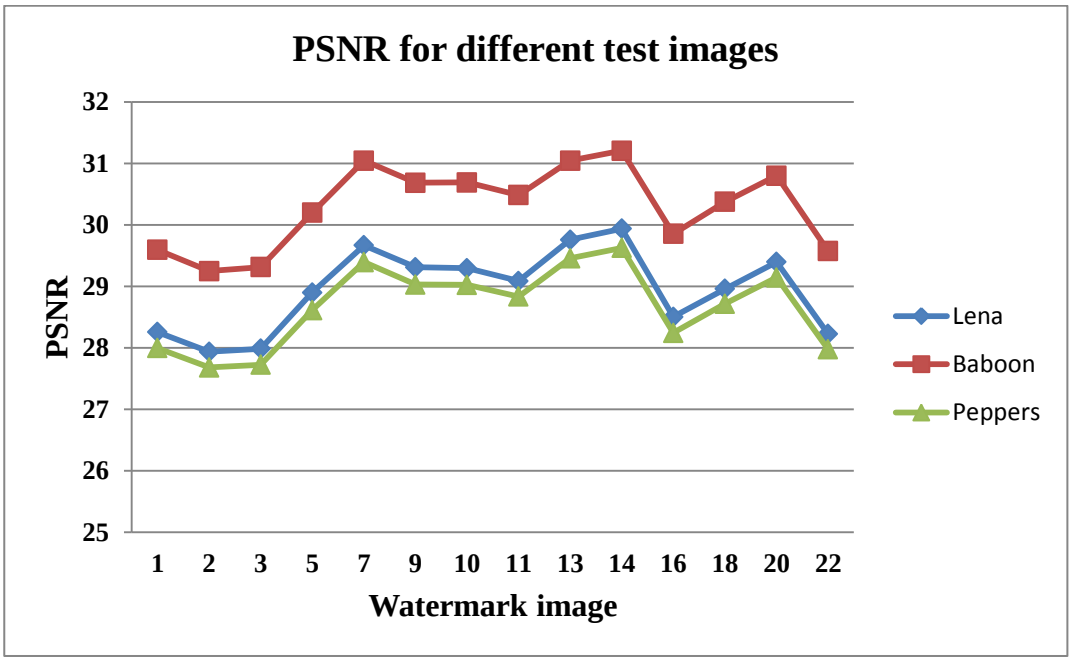


Fig 6.3 PSNR for different test images

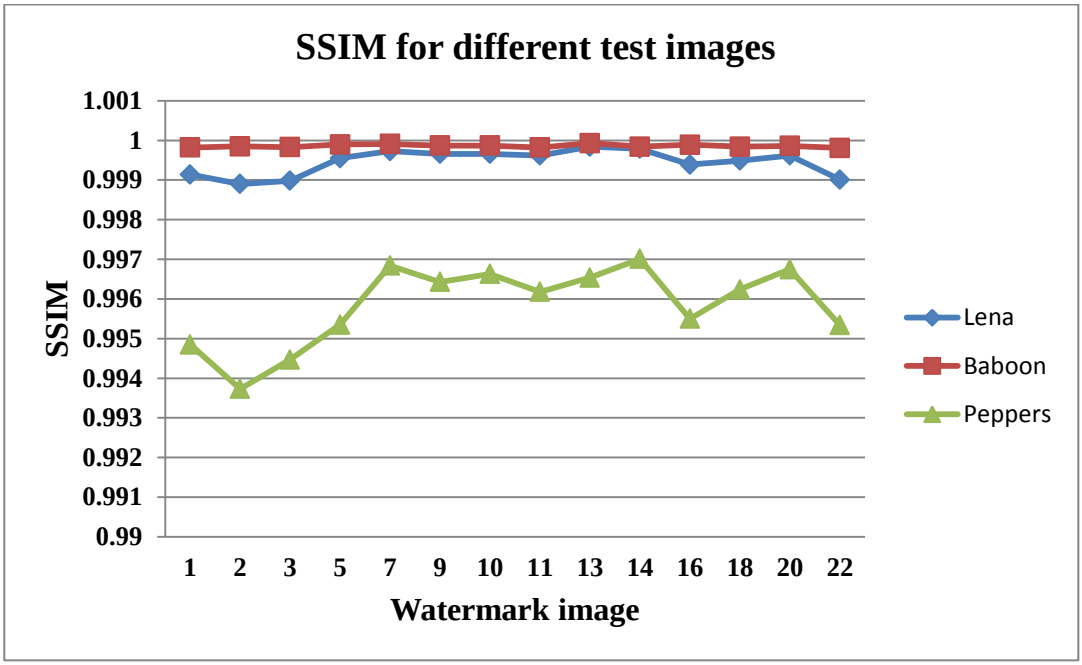


Fig 6.4 SSIM for different test images

Testing

- Attacks applied on our watermarked image are:
  - Histogram Equalization
  - Median Filtering
  - Salt and Pepper Noise
  - Gaussian Noise
  - JPEG Compression

- The following are the results in tabular and graphical representation:

Attack Simulated	Lena	Baboon	Peppers
Histogram Equalization	100	14.23	100
Median Filtering	36.8	11.76	52.63
Salt and Pepper Noise	41.2	70.59	52.94
Gaussian Noise	36.8	61.11	31.14

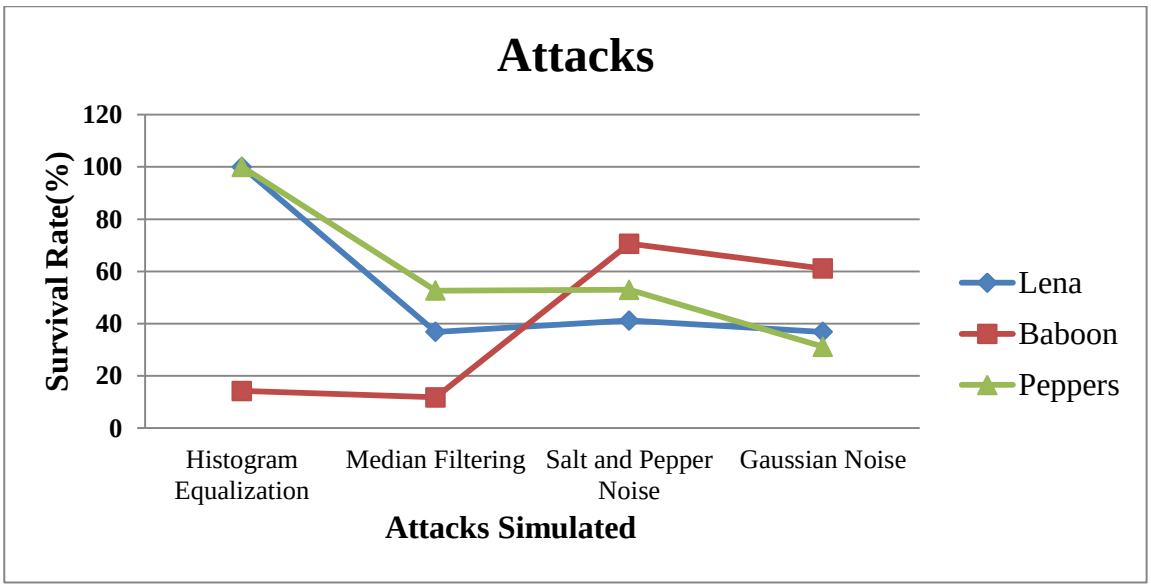


Fig 6.5 Attacks

- JPEG Compression

QUALITY FACTOR	LENA	BABOON	PEPPERS
10	90	70	80
20	100	100	100
30	100	91.67	100
40	100	100	100
50	100	100	100

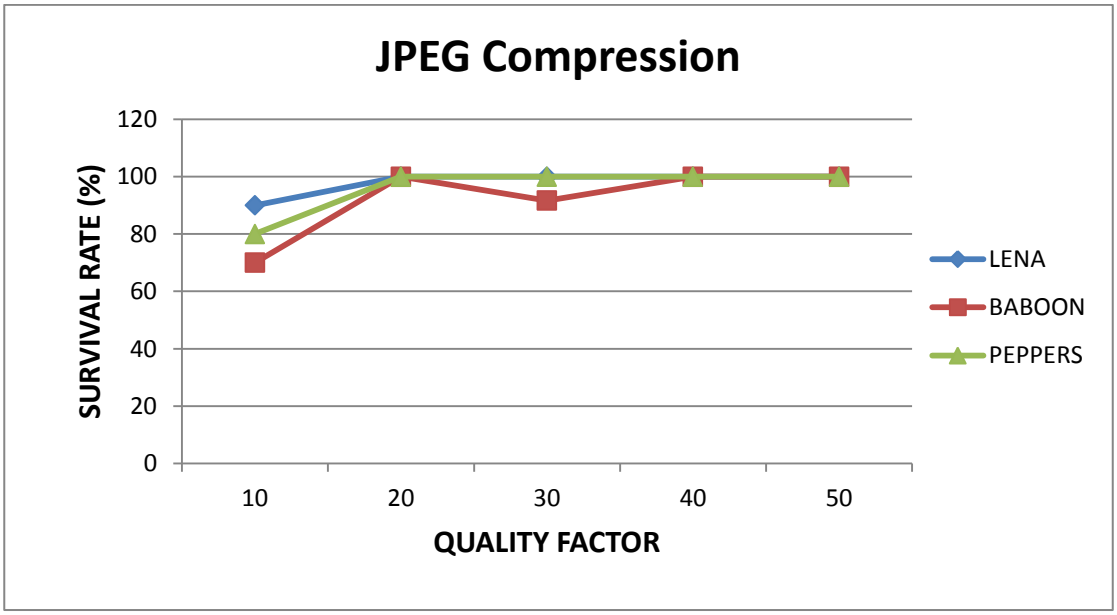


Fig 6.6 JPEG Compression

6.2 Conclusion

The tabulated results show that the technique proposed in this paper is semi-fragile, and provides resistance against most of the geometric attacks, majorly including histogram equalization, salt and pepper noise, Gaussian noise and lossy JPEG compression.

6.3 Scope for further improvement

We can further improve the watermark embedding process by using continuous wavelet transform instead of lifting wavelet transform as the former provides more detailed representation of the image and retains more information. Further, a better technique can be used for watermark recognition in place of K-means clustering to enhance the performance and improve the robustness further.

BIBLIOGRAPHY

1. Meenakshi Arya and Rajesh Siddavatam. A Novel Biometric Watermaking Approach Using LWT- SVD.INFORMATION TECHNOLOGY AND MOBILE COMMUNICATION in Computer and Information Science, 2011, Volume 147, Part 1, 123-131, DOI: 10.1007/978-3-642-20573-6_20.

2. Haifeng Li, Shuxun Wang, Weiwei Song and Quan Wen.A Novel Watermarking Algorithm Based on SVD and Zernike Moments.INTELLIGENCE AND SECURITY INFORMATICS Lecture Notes in Computer Science, 2005, Volume 3495/2005, 251-309, DOI: 10.1007/11427995_41

3. Say Wei Foo, Qi Dong.A Normalization-based Robust Watermarking Scheme Using Zernike Moments .International Journal of Computer and Information Engineering 3:4 2009

4. . Qiang Li; Chun Yuan; Yu-Zhuo Zhong;

Adaptive DWT-SVD Domain Image Watermarking Using Human Visual Model

Advanced Communication Technology, The 9th International Conference ,2007

5. Jianzhong Li; Yinghui Zhu; Dept. of Math. & Inf. Technol., Hanshan Normal Univ., Chaozhou, China .A geometric robust image watermarking scheme based on DWT-SVD and Zernike moments. Computer Science and Information Technology (ICCSIT), 2010

6. Zhong Wang,Xiong-fei Ye,Nian Xiao.Robust Watermarking Based on Norm Quantization Singular Value Decomposition and Zernike Moments.PACIIA '08 Proceedings of the 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application - Volume 02

7. Chenhong Lu and Zhaoyang Lu.Zernike Moment Invariants Based Iris Recognition.ADVANCES IN BIOMETRIC PERSON AUTHENTICATION Lecture Notes in Computer Science, 2005, Volume 3338/2005, 3-48, DOI: 10.1007/978-3-540-30548-4_63

8. Juan Reyes-Lopez, Sergio Campos and H ´ ector Allende,Rodrigo Salas.Zernike's feature descriptors for Iris Recognition with SVM.

9. Hasan Abdel Qader,Abdul Rahman Ramli,and Sayed Al-Haddad.Fingerprint Recognition Using Zernike Moments.The International Arab Journal of Information Technology, Vol 4 ,No. 4,October 2007.

10. Guangyong GAO1,2,†, Anyuan DENG1, Mingfa ZHOU1, Jinlin GONG1 . A Robust Grayscale Watermark Algorithm Against Geometric Attacks. Journal of Computational Information Systems 6:9 (2010) 3117-3125.

11. W. Sweldens, "The lifting scheme: A construction of second generation wavelets", SIAM Journal on Mathematical Analysis, 29(2), pp. 511546,1997.

12. Loukhaoukha, K.; Chouinard, J.-Y.; Hybrid watermarking algorithm based on SVD andlifting wavelet transform for ownership verification.

13. Internet- Google book results, etc.

14. Wikipedia

15. IEEE Xplore

16. **Book**-Digital Watermarking by Ingemar J. Cox,Mathew L. Miller and Jeffrey A. Bloom

17. **Book**-Biometric Systems by James Wayman, Anil Jain, Davide Maltoni, Darid Maio.

18. **Book**-Biometrics: Personal Identification in Networked Society by Anil Jain, Ruud Bolle and Sharath Pankanti

APPENDIX-1

Main Code:

```
%Project:   Biometric watermarking and recognition using iris%

clear all;
clc;
close all;

%************************* Save start time*********************************%
start_time=cputime;

%********************* Set the gain factor for embedding*******************%


k=0.2;

%**************************Read the Cover Image*****************************%

h=input('Enter the cover image name:','s');
[I,map]=imread(h);
cover_object=double(I);
[Mc Nc]=size(cover_object); %Height and width of host image

%************************Read the Signature Image************************%
SignDatabase = ('Traindatabase');
%SignDatabase = ('Traindatabase_New');
prompt = {'Enter the signature image to be used as a watermark:'};
dlg_title = 'Image to be watermarked';
num_lines = 1;
def = {'.png'};
Watermark_Image = inputdlg(prompt,dlg_title,num_lines,def);
Watermark_Image= strcat(SignDatabase,'\',char(Watermark_Image));
```

```
imrgb = imread(Watermark_Image);
im=rgb2gray(imrgb);
message=imresize(im,[60 30]);

%[r_message,l_message]=size(message);
wm1=im2bw(message);
%wm_1D=reshape(wm,1,r_message*l_message);
wm1=double(wm1);
[Mm Nm]=size(wm1);

%wm_1D=round(reshape(wm,Mm*Nm,1)./256);
wm_1D=round(reshape(wm1,Mm*Nm,1));

%********Performing Feature Extraction on the signature database**********%

Database =  ReadSigImages(SignDatabase);
%a=abs(Database);
[centroid,cl1,cl2,cl3,cl4,cl1_index,cl2_index,cl3_index,cl4_index]=clus(Database);
ProjectedImages=Database;
No_of_images = size(ProjectedImages,2);

%************************Beginning of watermarking process***************%

h=waitbar(0,'Embedding the signature in the image');
wm=imresize(im,[Mc Nc]);
message=double(wm);
wm=message;
[P Q]=size(wm);
%figure,imshow(I);
%title('Watermark Image');
I2=im2double(I);
% Perform integer LWT of the same image.
lshaar = liftwave('bior1.1');

% Add a primal ELS to the lifting scheme.
```

```
%els = {'p',[-0.125 0.125],0};
els = {'p',[-0.5 0.5],0};
lsnew = addlift(lshaar,els);
[cA1,cH1,cV1,cD1] = lwt2(cover_object,lsnew);
[cA2,cH2,cV2,cD2] = lwt2(cA1,lsnew);

[U,S,V] = svd(cH2);
%S=diag(S);
[wA1,wH1,wV1,wD1] = lwt2(message,lsnew);
[wA2,wH2,wV2,wD2] = lwt2(wA1,lsnew);
[A,B,C] = svd(wH2);
%B=diag(B);
%D=zeros(length(S));
D=S+ k*B;
[u1,D1,v1]=svd(D);
cH2=U*D1*V';
cA1=ilwt2(cA2,cH2,cV2,cD2,lsnew);
  %  watermarked_image=U*D1*V';
   watermarked_image=ilwt2(cA1,cH1,cV1,cD1,lsnew);

%watermarked_image = [U,D,V];
watermarked_image_uint8=uint8(watermarked_image);
imwrite(watermarked_image_uint8,'dwt_watermarked.bmp','bmp');
close(h);

%**********************RECOVERING THE
WATERMARK**************************%

file_name='dwt_watermarked.bmp';
watermarked_image=double(imread(file_name));

% determine size of watermarked image
Mw=size(watermarked_image,1);          %Height
Nw=size(watermarked_image,2);          %Width
```

```
[cA1,cH1,cV1,cD1] = lwt2(watermarked_image,lsnew);
[cA2,cH2,cV2,cD2] = lwt2(cA1,lsnew);
[X,Y,Z]=svd(cH2);

B=(Y-S)/k;
wH2=A*B*C';

wA1=ilwt2(wA2,wH2,wV2,wD2,lsnew);
out_image=ilwt2(wA1,wH1,wV1,wD1,lsnew);
out_image=imresize(out_image,[P Q]);
out_image_uint8=uint8(out_image);
wm_uint8=uint8(wm);
cover_object_uint8=uint8(cover_object);
%[E] = psnr(cover_object_uint8,watermarked_image_uint8,Mc,Nc);
[E] = snr(cover_object_uint8,watermarked_image_uint8);
[F] = ssim_index(wm_uint8,out_image_uint8);



%******Extracting the zernike features from recovered watermark image*********%

InputImage = out_image_uint8;
temp = InputImage(:,:,1);
%temp=rgb2gray(temp1);
temp=imresize(temp,[60 30]);
[res] = try1(temp);
[irow icol] = size(res);
InImage = reshape(res',irow*icol,1);
ProjectedTestImage=InImage;
%Difference = double(InImage)-mean_D ;              % Centered test image
%ProjectedTestImage = Prin_Coeff'*Difference;     %Test image feature vector

%********************K- means clustering**********************
% Euclidean distances between the recovered watermark coefficients and the
% projection of all centered training images is calculated. Test image is
```

```
% supposed to have minimum distance with its corresponding image in the
% training database.
centroid=centroid';
cl1=cl1';
cl2=cl2';
cl3=cl3';
cl4=cl4';
Euc_dist = [];
%size(ProjectedTestImage)
%size(centroid)
for i = 1 : 4
    q = centroid(:,i);
    temp = ( norm( ProjectedTestImage - q ) )^2;
    Euc_dist = [Euc_dist temp];
end
[Euc_dist_min , Recognized_index] = min(Euc_dist);
Euc_dist = [];
if(Recognized_index==1)
    [a b]=size(cl1);
    for i = 1 : b
        q = cl1(:,i);
        temp = ( norm( ProjectedTestImage - q ) )^2;
        Euc_dist = [Euc_dist temp];
    end
    [Euc_dist_min , Recognized_index] = min(Euc_dist);
    OutputName = strcat(int2str(cl1_index(Recognized_index)),'.png');
elseif(Recognized_index==2)
        [a b]=size(cl2);
        for i = 1 : b
            q = cl2(:,i);
            temp = ( norm( ProjectedTestImage - q ) )^2;
            Euc_dist = [Euc_dist temp];
        end
        [Euc_dist_min , Recognized_index] = min(Euc_dist);
        OutputName = strcat(int2str(cl2_index(Recognized_index)),'.png');
```

```
elseif(Recognized_index==3)
        [a b]=size(cl3);
        for i = 1 : b
            q = cl3(:,i);
            temp = ( norm( ProjectedTestImage - q ) )^2;
            Euc_dist = [Euc_dist temp];
        end
        [Euc_dist_min , Recognized_index] = min(Euc_dist);
        OutputName = strcat(int2str(cl3_index(Recognized_index)),'.png');
 else
        [a b]=size(cl4);
        for i = 1 : b
            q = cl4(:,i);
            temp = ( norm( ProjectedTestImage - q ) )^2;
            Euc_dist = [Euc_dist temp];
        end
        [Euc_dist_min , Recognized_index] = min(Euc_dist);
        OutputName = strcat(int2str(cl4_index(Recognized_index)),'.png');
 end
SelectedImage = strcat(SignDatabase,'\',OutputName);
SelectedImage = imread(SelectedImage);


figure()
subplot(2,3,1),imshow(cover_object_uint8,[]);
title('Cover Image');
subplot(2,3,4),imshow(watermarked_image_uint8,[]);
title('Watermarked Image');
subplot(2,3,2),imshow(wm,[]);
title('Original Watermark');
subplot(2,3,3),imshow(out_image_uint8,[]);
title({'PSNR between Original Watermark';['and Recovered
Watermark=',num2str(E)];['SSIM=',num2str(F)]});
subplot(2,3,5), imshow(out_image_uint8,[]);
title('Recovered Signature image');
```

```
str = strcat('The signature image matching the recovered image is =  ',OutputName);
subplot(2,3,6),imshow(SelectedImage)
title(str);

 c=3;
 reply = input('Do you want to simulate attacks? Y/N [Y]: ','s');
 if(reply=='y'|| reply=='Y')
    watermarked_image1_uint8=uint8(watermarked_image);
    fprintf('Press \n');
    fprintf('C for Image Cropping Attack \n');
    fprintf('H for Histogram Equalization Attack \n');
    fprintf('M for Median Filtering Attack \n');
    fprintf('P for Salt n Pepper attack \n');
    fprintf('G for Gaussian Noise Attack \n');
    fprintf('W for Weiner Filtering Attack \n');
    fprintf('S for Sharpening Attack \n');
    fprintf('J for Lossy JPEG compression \n');

choice=input('Enter your choice=','s');

switch lower(choice)

%*********************Image Cropping Attack*********************************%
   case 'c'
      a=' Cropping ';
%Cropping four Corners of image
   watermarked_image1_uint8(1:50,1:50)=128;
   watermarked_image1_uint8(1:50,206:256)=128;
   watermarked_image1_uint8(1:50,462:512)=128;
   watermarked_image1_uint8(206:256,1:50)=128;
   watermarked_image1_uint8(462:512,1:50)=128;
   %watermarked_image_uint8(206:256,206:256)=128;
   %watermarked_image_uint8(462:512,462:512)=128;
imwrite(watermarked_image1_uint8,'dwt_watermarked.jpeg','jpeg');
```

```
% **********Histogram Equalization**********%

  case 'h'
 a=' Histogram Equalization ';
 watermarked_image1_uint8=histeq(watermarked_image_uint8);
imwrite(watermarked_image1_uint8,'dwt_watermarked.jpeg','jpeg');
%[snr]=snr(cover_object,watermarked_image)            %Calculating the SNR

%********** Attack of median Filtering**********%
  case 'm'
     a=' Median Filtering ';
 watermarked_image1_uint8= medfilt2(watermarked_image_uint8,[5 5]);
imwrite(watermarked_image1_uint8,'dwt_watermarked.jpeg','jpeg');

% ********** salt and pepper noise**********%
case 'p'
   a=' Salt n Pepper ';
watermarked_image1_uint8=imnoise(watermarked_image_uint8,'salt & pepper',0.02);
imwrite(watermarked_image1_uint8,'dwt_watermarked.jpeg','jpeg');

 % ********** Attack of Gaussian Noise**********%
case 'g'
   a=' Gaussian Noise ';
watermarked_image1_uint8=imnoise(watermarked_image_uint8,'gaussian',0.02);
imwrite(watermarked_image1_uint8,'dwt_watermarked.jpeg','jpeg');

%********** Attack of Wiener Filtering**********%
  case 'w'
    a=' Weiner Filtering ';
watermarked_image1_uint8= wiener2(watermarked_image_uint8,[5 5]);
imwrite(watermarked_image1_uint8,'dwt_watermarked.jpeg','jpeg');

% **********Sharpening**********%
```

```
     case 's'
        a=' Sharpening ';
    H = fspecial('unsharp',0.1);
  watermarked_image1_uint8 = imfilter( watermarked_image_uint8,H);
imwrite(watermarked_image1_uint8,'dwt_watermarked.jpeg','jpeg');

case 'j'
        a=' JPEG Compression ';
    Q = input('Quality factor');

imwrite(watermarked_image1_uint8,'dwt_watermarked.jpeg','jpeg','quality',Q);
  otherwise
     disp('Incorrect choice')
     reply = input('Do you want to simulate attacks? Y/N [Y]: ','s');
end


%**********************RECOVERING THE
WATERMARK**************************%

file_name='dwt_watermarked.jpeg';
watermarked_image1=double(imread(file_name));

% determine size of watermarked image
Mw=size(watermarked_image1,1);          %Height
Nw=size(watermarked_image1,2);          %Width

[cA1,cH1,cV1,cD1] = lwt2(watermarked_image1,lsnew);
[cA2,cH2,cV2,cD2] = lwt2(cA1,lsnew);
[X,Y,Z]=svd(cH2);

B=(Y-S)/k;
wH2=A*B*C';
wA1=ilwt2(wA2,wH2,wV2,wD2,lsnew);
out_image=ilwt2(wA1,wH1,wV1,wD1,lsnew);
```

```
end

 %******Extracting the zernike features from recovered watermark image*********%

%InputImage = out_image_uint8;
InputImage1 = out_image;
temp = InputImage1(:,:,1);
%temp=rgb2gray(temp1);
temp=imresize(temp,[60 30]);
[res] = try1(temp);
[irow icol] = size(res);
InImage1 = reshape(res',irow*icol,1);
ProjectedTestImage1=InImage1;
%Difference = double(InImage)-mean_D ;                % Centered test image
%ProjectedTestImage = Prin_Coeff'*Difference;     %Test image feature vector

%*********************K Means Clustering ************************
% Euclidean distances between the recovered watermark coefficients and the
% projection of all centered training images is calculated. Test image is
% supposed to have minimum distance with its corresponding image in the
% training database.
Euc_dist = [];
%size(ProjectedTestImage)
%size(centroid)
for i = 1 : 4
    q = centroid(:,i);
    temp = ( norm( ProjectedTestImage1 - q ) )^2;
    Euc_dist = [Euc_dist temp];
end
[Euc_dist_min , Recognized_index] = min(Euc_dist);
Euc_dist = [];
if(Recognized_index==1)
    [a b]=size(cl1);
    for i = 1 : b
        q = cl1(:,i);
```

```
        temp = ( norm( ProjectedTestImage1 - q ) )^2;
        Euc_dist = [Euc_dist temp];
    end
    [Euc_dist_min , Recognized_index] = min(Euc_dist);
    OutputName = strcat(int2str(cl1_index(Recognized_index)),'.png');
elseif(Recognized_index==2)
        [a b]=size(cl2);
        for i = 1 : b
            q = cl2(:,i);
            temp = ( norm( ProjectedTestImage1 - q ) )^2;
            Euc_dist = [Euc_dist temp];
        end
        [Euc_dist_min , Recognized_index] = min(Euc_dist);
        OutputName = strcat(int2str(cl2_index(Recognized_index)),'.png');
elseif(Recognized_index==3)
            [a b]=size(cl3);
            for i = 1 : b
                q = cl3(:,i);
                temp = ( norm( ProjectedTestImage1 - q ) )^2;
                Euc_dist = [Euc_dist temp];
            end
            [Euc_dist_min , Recognized_index] = min(Euc_dist);
            OutputName = strcat(int2str(cl3_index(Recognized_index)),'.png');
 else
            [a b]=size(cl4);
            for i = 1 : b
                q = cl4(:,i);
                temp = ( norm( ProjectedTestImage1 - q ) )^2;
                Euc_dist = [Euc_dist temp];
            end
            [Euc_dist_min , Recognized_index] = min(Euc_dist);
            OutputName = strcat(int2str(cl4_index(Recognized_index)),'.png');
 end
SelectedImage = strcat(SignDatabase,'\',OutputName);
SelectedImage = imread(SelectedImage);
```

```
figure()
subplot(2,3,1),imshow(cover_object_uint8,[]);
title('Cover Image');
subplot(2,3,4),imshow(watermarked_image_uint8,[]);
title('Watermarked Image');
subplot(2,3,2),imshow(wm,[]);
title('Original Watermark');
subplot(2,3,3),imshow(out_image_uint8,[]);
title({'PSNR between Original Watermark';['and Recovered
Watermark=',num2str(E)];['SSIM=',num2str(F)]});
subplot(2,3,5), imshow(out_image_uint8,[]);
title('Recovered Signature image');
str = strcat('The signature image matching the recovered image is =  ',OutputName);
subplot(2,3,6),imshow(SelectedImage)
title(str);
```

APPENDIX-2

Zernike moments code:

```
function [zvalues] = try1(I)
I=im2double(I);
[X,Y]=size(I);

Inorm=normalize(I);
while (1)
  %n= input('enter n');
  %m= input('enter m');
  n=10;
  m=2;
  if(mod(n-m,2)==0)
      break;
   end
end
%t=abs(m);
count=0;
for x=1:X+1
   for y=1:Y+1

        a=x/X;
       xnorm(x)=a;
       b=y/Y;
       ynorm(y)=b;
       rho=sqrt(x*x+y*y);
        r=0;
        theta=atan(b/a);
      if((power(xnorm(x),2)+power(ynorm(y),2))<=1)
     count=count + 1;
         d=0;
      for l=0:n
        for k=0:l
```

```
        if(mod(l-k,2)==0)
        temp1=(n-k)/2;
        temp2=factorial((n+k)/2);
        temp3=factorial((k+m)/2);
        temp4=factorial((k-m)/2);
        temp5=power(-1,temp1);
        r(l+1,k+1)=((temp5*temp2*power(rho,k))/(factorial(temp1)*temp3*temp4));

        vconjugate(l+1,k+1)=r(l+1,k+1)*cos(k*theta)-i*r(l+1,k+1)*sin(k*theta);
        %vconjugate = cell2mat(vconjugate);
        d=d+vconjugate(l+1,k+1);
        end
        end
      end
zvalues(x,y)=((n+1)/3.14)*d*Inorm(x,y);
      end
   end
end


end
```

APPENDIX-3

K-means clustering code:

```
function [centroid,cl1,cl2,cl3,cl4,cl1_index,cl2_index,cl3_index,cl4_index]=clus(zvalues)
%k-means clustering
%euc_d = pdist(zvalues,'euclid');
k=1;
 clust=kmeans(zvalues',4,'emptyaction','singleton','start','uniform')
%store every cluster in a matrix
cl1=[];
cl2=[];
cl3=[];
cl4=[];
 cl1_index=[];
 cl2_index=[];
 cl3_index=[];
 cl4_index=[];
[a b]=size(clust);
for i=1:a
   if(clust(i)==1)
      cl1=[cl1 zvalues(:,i)];
      cl1_index=[cl1_index i];
   end
   if(clust(i)==2)
      cl2=[cl2 zvalues(:,i)];
      cl2_index=[cl2_index i];
   end
   if(clust(i)==3)
      cl3=[cl3 zvalues(:,i)];
       cl3_index=[cl3_index i];
   end
   if(clust(i)==4)
      cl4=[cl4 zvalues(:,i)];
```

```
        cl4_index=[cl4_index i];
    end
end
cl1=cl1';
cl2=cl2';
cl3=cl3';
cl4=cl4';
%find the centroid of each cluster
centroid=[];
[a b]=size(cl1);
for j=1:b
    sum=0;
   for i=1:a
       cl1(i,j);
       sum=(sum+cl1(i,j));
   end
   sum=(sum/a);
   centroid(1,j)=sum;
end
[a b]=size(cl2);
for j=1:b
    sum=0;
   for i=1:a
       sum=(sum+cl2(i,j));
   end
   sum=(sum/a);
   centroid(2,j)=sum;
end
[a b]=size(cl3);
for j=1:b
    sum=0;
   for i=1:a
       sum=(sum+cl3(i,j));
   end
   sum=(sum/a);
```

```
   centroid(3,j)=sum;
end
[a b]=size(cl4);
for j=1:b
    sum=0;
   for i=1:a
       sum=(sum+cl4(i,j));
   end
   sum=(sum/a);
   centroid(4,j)=sum;
end
end
```

CURRICULUM VITAE

ASTHA AGARWAL

Department of Computer Science and Engineering
Jaypee University of Information Technology,
Waknaghat, Solan (H.P) - 173234
Email: astha.juit@gmail.com
Phone (M): (+91)9805439561

Education

Standard	College/School	Year	CGPA/Percentage
B.Tech (CSE)	Jaypee University of Information Technology, Solan.	2012	8.1 (83%) (Up till 7th semester)
12th (ISC)	La Martiniere Girls' College, Lucknow	2008	90.60%
10th (ICSE)	La Martiniere Girls' College, Lucknow	2006	86%

Technical Skills

Programming Language	C,C++, Java , Assembly Language, basics of CUDA programming.
Operating Systems	Microsoft Windows XP, Vista, UNIX
Software Packages	Turbo C++ IDE, Matlab, Microsoft Office, Eclipse, Macromedia Flash, Prometheus Design Tool, Visual Studio

Projects

- FINAL YEAR: Biometric Watermarking and recognition using Iris
  Team Size: 3
  In this project we are working to develop a robust technique for iris watermarking and recognition. This technique is used for security purposes. We use 'Zernike moments' for iris feature extraction and 'K-means clustering' for iris recognition. We are using MATLAB 7.6 for the coding.
- THIRD YEAR: Online Polling System
  Team Size: 3
  Language: HTML, JavaScript

  In this project we had developed an online voting system to simplify the process of voting. We used HTML and JavaScript.

CURRICULUM VITAE

Deepika Lohani
Computer Science Engineering
Jaypee University of Information and Technology
Waknaghat, Distt – Solan (H.P.)
Contact no. : +91-9736082320
E-mail: deepikalohani23@gmail.com

OBJECTIVE

To work at a position where I can apply my skills to the best of my capabilities thereby gaining valuable experience and contributing to the organizational growth.

ACADEMIC QUALIFICATIONS

Course	School / College	Board / University	Year Of Passing	Percentage
B.TECH	Jaypee University of Information Technology,Waknaghat	JUIT	2008-2012	83.0% (CGPA 8.0)
CLASS XII (AISSCE)	STDB Vidyalaya,Gzb	C.B.S.E	2007	87.8%
CLASS X (AISSE)	STDB Vidyalaya,Gzb	C.B.S.E.	2005	90%

TECHNICAL SKILLS

Programming Language: C and C++
Operating System: Microsoft Windows 7, XP, Vista
Software Packages: Turbo C++, Matlab, Microsoft Office
Areas of Interest: Image Processing

PROJECTS UNDERTAKEN

1. **4th year project:** Working on Major Project Biometric based Watermarking and Recognition using Iris(VII-VIII semester)- Using Iris as a signature for watermarking and recognition of Iris watermark for improved authentication.
2. **Prototype of washing machine:** Using the microcontroller ATMega8 to control the direction, speed and time of rotation for the tumbler and implementing the basic features of washing machine.

CURRICULUM VITAE

Neha Tolani

Department of Computer Science and Engineering

Jaypee University Of Information Technology, Phone: +91-9805439258

Waknaghat, Solan (H.P.) - 173234 Email:neha.tolani67@gmail.com

Educational Credentials:

Year	Course	University/Board	Name of Institution	Percentage
2011	B.Tech(CSE)- (upto 7th semester)	Jaypee University of Information Technology	Jaypee University of Information Technology	87 (CGPA- 8.6)
2008	12th	State Board(Madhya Pradesh)	Nachiketa Higher Secondary School	85.33
2006	10th	State Board(Madhya Pradesh)	Nachiketa Higher Secondary School	86.6

Technical Skills:

- Programming Languages: C, C++, Java, J2EE
- Database: Oracle 10g, Mysql
- Web Development: PHP, ASP.Net , JavaScript, CSS, and AJAX
- Software Packages: Photoshop, Dreamweaver , Microsoft Visual Studio ,MATLAB
- Server: Apache Tomcat (for J2EE), Wamp Server(basics for PHP).
- Areas of Interest : Object Oriented Programming, Database Management , Operating Systems

Projects:

1. **4th year project-**Working on Major Project Biometric based Watermarking and Recognition using Iris(VII-VIII semester)- Using Iris as a signature for watermarking and recognition of Iris watermark for improved authentication.
2. **HACK** – It was an online event organized at the annual function Le-Fiestus as well as in the technical fest of the University.
3. **Online Quiz-** The basic aim of this project is to provide students with better learning facilities. This quiz module allows them to test their learning through a quiz .For those who are registered on this website (members as well as non-members of the institute), they can take timer-based tests related to computer programming languages.