ROBUST DIGITAL WATERMARKING IN THE CURVELET DOMAIN

by

PEINING TAO

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2008

UMI Number: 3296983

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

The manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirements for the degree of Doctor of Philosophy

Prof. Scott Dexter

Date — Chair of Examining Committee

Prof. Ted Brown

Date — Executive Officer

Prof. Michael Anshel

Dr. Candemir Toklu
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT

ROBUST DIGITAL WATERMARKING IN THE CURVELET DOMAIN

by

PEINING TAO

Advisor: Professor Scott Dexter

Watermarking is a method in computer security by which identifiers of sources or copyright owners of digital signals are embedded into the respective signals themselves in order to keep track of where a signal comes from or who the copyright owners are. In general, a watermarking system must have two characteristics: perceptual transparency and robustness. This thesis proposes a method for transparently and robustly embedding a watermark into the curvelet transform of grayscale images. The image is partitioned into small blocks; Fast Discrete Curvelet Transform (FDCT) via Unequally-Spaced Fast Fourier Transforms (USFFT) is employed to decompose each block into curvelet domain. We embed the watermark into the selected blocks, scale and curvelet coefficients based on the edge map of the cover image. The embedding strength is adjusted by a Just Noticeable Distortion (JND) model computed for each curvelet coefficient. Robustness is tested against a variety of types of image attacks. Since the curvelet transform enables most of the energy of the object to be localized in just a few coefficients, the optimally sparse representations of image edges allows for the embedded watermarks to recover from severe image degradation. However, the block-based watermarking algorithm in curvelet domain provides low robustness against geometrical distortion because geometrical distortion (e.g. rotation) desynchronizes the embedding location in the cover work. A scheme relying on the radon transforms and edge detection is developed to

synchronize embedding location before the watermark detector is applied. The proposed scheme estimates the geometrical distortion the cover image was subjected to and restores the distorted image to its original state. Thus, the improved watermarking system provides high tolerance to geometric attacks as well as normal image processing. We also propose a technique for selecting the threshold for watermarking detection based on statistical analysis over host signals and embedding schemes. Experiments show our scheme is capable of keeping the probability of false positive and false negative both low and is generally robust against a wide range of image attacks. Finally, we present a new quality measure, M-SVD, which expresses the perceived distortion of image/video. We show our measure to be strongly correlated with evaluations by the Human Visual System. The quality of the watermarked images marked by our proposed curvelet based algorithm is evaluated with this approach. The evaluation demonstrates the transparency of our watermarking system, the performance of JND modeling is also confirmed with M-SVD.

ACKNOWLEDGEMENTS

I would like to first thank my former advisor Professor Eskicioglu who educated and inspired me to become interested into digital watermarking. He closely supervised my PhD study, offered many helpful suggestions and kind support. Under his guidance, I have learned the methodology of conducting research and converted my work into paper.

Special thanks go to my current mentor Professor Dexter for being so supportive and encouraging. He spent a lot of time scrutinize my thesis and offered tremendous amount of valuable comments. He provided close supervision during my thesis writing and took many efforts to help me in accomplishment of my thesis. I was impressed by his earnest attitude towards work.

Thanks also to my dear committee members Professor Brown and Professor Anshel for their advice benefiting in completion of my PhD study, especially when my advisor Professor Eskicioglu is away. Thanks to Professor Brown, The Graduate Center, CUNY and US Air Force for funding my project in part.

Last but not least, I want to thank my husband dearly for loving me, helping me and supporting me all the time. I am very grateful to my family, my parents, my husband Guangwei, my son Kevin for making me happy in this long journey.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1
Background

Digital watermarking has been proposed as a method for discouraging illegal copying and distribution of copyrighted material. A well designed watermarking system must provide two properties: perceptual transparency and robustness. This thesis introduces a method for transparently and robustly embedding a watermark into the block-based curvelet domain of grayscale images. The perceptual quality of watermarked images is measured by PSNR, UIQI and the new metric M-SVD. The robustness is demonstrated against variety types of image attacks. A scheme relying on the radon transform and edge detection is used to estimate geometrical distortion before applying watermark detector. We also propose a technique for selecting threshold for watermarking detection. Algorithms and experimental results are given in the following chapters.

In Chapter 1, we review the background of the digital image watermarking technology. This chapter provides an overview to watermarking motivation, watermarking applications and the general framework and classification of watermarking systems. We will explore a taxonomy of watermarking techniques based on domain, algorithm and the information required in watermark detection/extraction procedures. This chapter also covers a variety of types of image attacks to which watermarks might be subjected and presents the common measures used to evaluate the performance of watermarking systems.

1.1 Importance of Digital Watermarking

Due to the fast and extensive growth of network technology, digital information can be distributed with no quality loss, low cost and nearly instantaneous delivery. Protection of multimedia content has recently become an important issue because of consumers' insufficient cognizance of the ownership of intellectual property. Thus, content owners are eagerly seeking technologies that promise to protect their rights. Two fundamental groups of technologies have been identified with the purpose of discouraging unauthorized consumption and duplication: encryption and watermarking [1,2,3].

Encryption makes multimedia content unintelligible through a reversible mathematical transformation and is probably the most common method of protecting digital content. 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, encryption 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 encryption: a technology that can protect content even after it is decrypted. Watermarking has been considered to fulfill this need because it embeds data directly into a multimedia element such as an image, audio or video file where it is hard to remove during normal processing. A

watermark can be designed to survive after the content has undergone some common signal processing operations which include decryption, re-encryption, compression, filtering, digital-to-analog conversion, and file format changes. There are two types of watermarking techniques: visible and invisible. The visible watermark can be seen by naked eyes, like the watermark on dollar bills. In this study, we focus on invisible watermarking techniques, i.e., hiding the information into the host image in a way that causes an imperceptible distortion.

1.2 Watermarking Applications

Watermarking can be used in a wide variety of applications [4,87]. In general, if it is useful to associate some additional information with a multimedia element, this associated information can be embedded as a watermark. The following are several proposed or actual watermarking applications: copyright protection, fingerprinting, content authentication, broadcast monitoring, and copy control [87].

Copyright Protection*:* One of the oldest application fields of watermarking is copyright protection. The goal of watermarking for copyright protection is to embed a "mark" into the content that can identify the copyright holder of the work. The mark can be a registered number, a text message a graphical logo or some unique pattern. The copyright owner may embed a watermark representing copyright information into digital content; later it may be used as a proof of ownership in disputes over copyright infringement.

Fingerprinting*:* Fingerprinting is an approach used in tracing the distribution of illegal copies. The watermark might record the recipient in each legal sale or distribution of the

work. The owner or producer of the work would place a different watermark in each copy. If the work is subsequently misused (e.g., leaked to the press or distributed to third party), the owner could find out who was responsible.

***Content Authentication**:* It is becoming easy to tamper with digital images in ways that are difficult to detect due to sophisticated image processing software. Digital photographs are used more and more often as court evidence. Thus, it is critical to verify the originality of a digital image which might be used as a piece of evidence in a legal case or police investigation. Watermarks can be used here as a means to verify that an image is genuine. Watermarks for verification purpose are required to be fragile [91], so that any modification to the image would be reflected in a corresponding error in the watermark (Fragile watermarks are designed to be sensitive to any form of modification applied to the cover image).

***Broadcast Monitoring**:* Owners of copyrighted works want to ensure that their property is not illegally rebroadcast by pirate stations. Advertisers want to ensure that they receive all of the air time they purchase from broadcasters. Thus, watermarks existing within the content itself are information an automated monitoring system can rely on to verify the broadcaster is fulfilling its contractual obligations.

Copy Control: Encryption is a methodology that protects digital content from unauthorized recording. But once it is decrypted, the content has no further protection. Watermarks embedded in the content itself might provide a better method of implementing copy control. If every recording device were fitted with a watermark

detector, the devices could be made to prohibit recording whenever a never-copy watermark is detected at its input [92].

1.3 Properties of Watermarking System

The basic idea of digital image watermarking is to embed data into a host image. In general, a watermarking system should have the capability to support several important properties which include invisibility, robustness, security, unambiguousness, and high data capacity. The relative importance of these properties depends on the requirements of a given application [88]. In fact, one property may conflict with another property. For example, if a mark is hidden in the unperceivable part of image signals then invisibility is improved, however the mark may have low robustness against various attacks. We can modify a large amount of image signals or embed mark to perceivable part of an image to achieve better capacity and robustness, but the mark is likely to be visible in this case. Therefore, there is a trade off among these properties.

- *Invisibility*

- ***Perceptual Invisibility*:** An embedded watermark should not introduce a significant degree of distortion in the cover image. The perceived degradation of the watermarked image should be imperceptible so as not to affect the viewing experience of the image. For this purpose, the characteristics of the human visual system (HVS) [46,47] for images are exploited in the watermark embedding process. However, this requirement conflicts with other requirements such as robustness, which is an important requirement when facing watermarking attacks.

- ***Statistical Invisibility*****:** An unauthorized person should not detect the watermark by means of statistical methods. For example, the availability of a great number of digital works watermarked with the same mark should not allow the extraction of the embedded mark by applying statistically based attacks. One solution to resist this kind of attack is choosing a watermark independent from the content to be protected.

- ***Robustness***

Digital images commonly are subject to many types of distortions (so called attacks), such as lossy compression, filtering, resizing, contrast enhancement, cropping, rotation and so on. The mark should be detectable even after such distortions have occurred. Robustness against signal distortion is better achieved if the watermark is placed in high frequency parts of the image signal. For example, a watermark hidden among high frequency data is likely not to survive lossy compression. Moreover, improving the resistance to geometric manipulations, such as translation, resizing, rotation and cropping is still an open issue.

- ***Security***

A hostile attack is any process specifically intended to thwart the watermark's purpose. Some of the techniques use the original non-marked image in the extraction process. Therefore, one often uses a secret key to generate the watermark for security purpose. The effectiveness of a watermark algorithm cannot be based on the assumption that possible attackers do not know the process through which the watermark is embedded. It is assumed that the attackers have full knowledge about the applied watermark procedure;

however, they have no knowledge of the secret key. Therefore, an attacker will try to manipulate the data to destroy the watermark. The *security* of a watermark refers to its ability to resist hostile attacks.

- ***Capacity***

Data capacity refers to the amount of data that can be embedded. A watermarking system should be able to embed a relatively high amount of data without affecting perceptual transparency. It is common to use *data payload* to describe the number of bits a watermark encodes within an image for watermarking system.

- ***Unambiguousness***

The watermark should unambiguously identify the owner. It is desired that the difference between the extracted and the original watermark is as low as possible. The accuracy of identification should degrade gracefully irrespective of the type of attack.

1.4 A General Image Watermarking Framework

The general process of image watermarking is depicted in Figure 1.1. Generally, watermarking systems for digital media involve two distinct stages: (1) watermark embedding to indicate copyright and (2) watermark extraction/detection to identify the owner [5].

- ***Watermark Embedding***

In order to combine a watermark with a digital image, we need an image, a mark and an encoding algorithm to create a watermarked image. Let I denote the original image

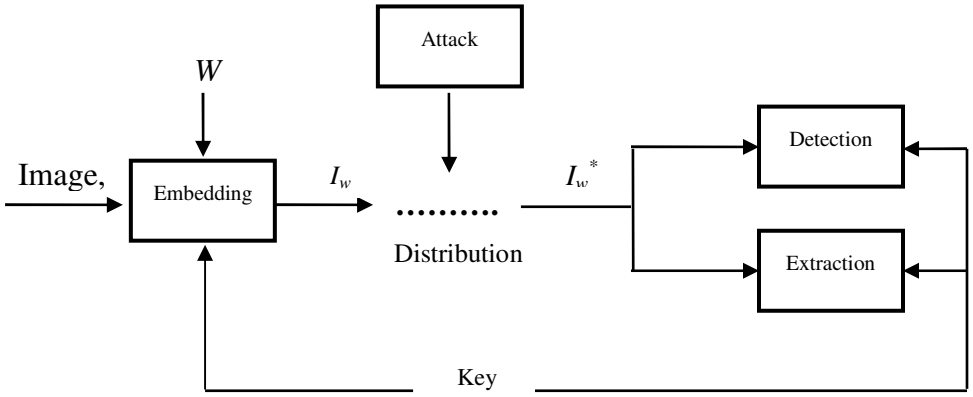


Figure 1.1. Generic Watermarking System

before watermarking, let W denote the watermark and I_w denote the signal with the embedded watermark. The mark can be a unique pattern, a pseudo random sequence with zero mean unit variance or a meaningful visual logo. The encoder takes the watermark and the cover I, and generates the watermarked image, that is described as a function:

$$E(I, W) = I_w \tag{1.1}$$

In this case, secret or public keys and other parameters can be used to extend the watermarking encoder. The employment of keys in watermarking application provides more security to copyright protection. Many watermarking algorithm are designed to use secret keys in such way that it is not possible to detect the presence of a watermark in an image without knowledge of the key, even if the watermarking algorithm is known. For example, a seed used to generate the pseudo-random sequence (often used as watermark added to an image) can be considered a secret key. In some algorithms, keys are also used during the embedding and detection processes to identify the watermark embedding location in the contents.

- ***Watermark Detection/Extraction***

In order to extract/detect the watermark hidden in the digital image, we need the watermarked cover image (which may be distorted by attacks) and a decoding algorithm to detect/extract the hidden mark. In this case, the decoder *D* takes the watermarked, normal or distorted image I_w, and extracts/detects the hidden watermark *W*. Extraction is the process of reconstructing the mark from the marked cover image, thus the concrete watermark can be obtained for further processing. Detection, on the other hand, provides a measure to indicate whether or not a given mark is present in a cover image. In *non-blind* watermarking techniques, the decoder *D* loads the original image *I* to extract the watermarking information. The process can be described as

$$D(I, I_w) = W \tag{1.2}$$

Using *blind* watermarking techniques, the original image is not available at the decoding stage; the decoder D may require information such as the original watermark *W*, secret key *K* and/or other parameters *P* for watermark detection/extraction. The extractor/detector extracts the watermark or just outputs "Yes" if the mark is present or "No" otherwise:

$$D(W/K/P, I_w) = W/(Y/N) \tag{1.3}$$

The blind watermarking approach is essential for applications such as copy control or monitoring of the broadcast audio/video program where the original signals are not accessible during watermark detection/extraction. In general, *non-blind* methods are more

robust against various attacks because the noise from host signals that interfere with watermark extraction/detection are compensated by the given original image.

1.5 Classification of Watermarking Systems

Watermarking systems can be classified according to many different criteria. For example, watermarking systems can be classified based on how a watermark gets merged into the cover work to create the watermarked image, whether the original cover image is needed to extract\detect the watermark, whether watermarks are manipulated in spatial or transform domains. The classification summary is shown in the Table 1.1. Other classifications are possible as well.

Table 1.1. Classification of image watermarking systems

Criterion	Class	Brief description
Domain type	Spatial [6,7,8,9,10,11]	Pixel values are modified to embed the watermark.
	Transform [12,13,14,15,16]	Transform coefficients are modified to embed the watermark. Recent popular transforms are Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Discrete Fourier Transform (DFT).
Information type	Non-blind [7,12,17]	Watermarking system that uses the original cover image in the watermark decoding process.
	Blind or oblivious[18,19,20,21]	Watermarking system that does not require the original cover image to be able to detect the embedded watermark.
Algorithm type	Additive Algorithm [12,22,17,23]	Additive algorithm performs linear modification of the host image and the correlative processing in the detection process.
	Quantization Algorithm [24,25]	Quantization algorithm performs non-linear modification of the host signals, quantizes and maps the received samples to nearest reconstruction points.

1.5.1 Type of Domain

Watermarking techniques can be classified according to whether they embed marks in the spatial domain or the transform domain.

1.5.1.1 Embedding in Spatial Domain

The most straight forward approach for hiding a watermark is to modify the host image pixel values directly. Although spatial domain techniques are easy to implement, in general they are not as robust as transform domain techniques.

LSB (least significant bit) embedding is the earliest and also the simplest spatial embedding technique. Since the last binary bits are the least significant bits, their modification will not perceived by human eyes. However, the information carried in the least significant bit rarely survives various attacks. Schyndel et al [6] proposed two methods using least significant bit embedding. In the first method, they compress the original 8 bits to 7 bits by adaptive histogram manipulation so as to enable the LSB to carry the watermark information. The watermark can be decoded by comparing the LSB bit pattern with a stored counterpart. In the second method, they use LSB addition for embedding the watermark. The decoding process is more complex, relying on a unique optimal autocorrelation function.

W. Bender et al [8] provided two methods, Patchwork and Texture Block Coding, which change the data directly in host image. In the first approach, two patches in host image are chosen pseudo randomly, and the data in each patch are lightened and darkened respectively. Watermark detection relies on statistical analysis. The second approach,

Texture Block Coding, is implemented by copying a region from a random texture pattern found in a picture to an area that has similar texture. This results in a pair of identically textured regions in the image. These regions can be detected based on autocorrelation of the watermarked and original images. Such approaches tend to degrade the cover image visibly and are vulnerable to a set of intentional and unintentional attacks.

Pitas [9] presented a technology for casting digital watermarks on images by embedding a predetermined small luminance value in randomly selected image pixels. The luminance values are small enough to be undetected by the human eye. The watermark is essentially the seed to a random pixel generator. The decoding scheme is based upon statistical detection theory criteria. The embedded watermark is proven to be resistant to subsampling but not robust to compression and filtering.

1.5.1.2 Embedding in Transform Domains

Most of the transform domain techniques embed the information into the transform coefficients of the cover image. Three popular transforms used for this purpose are the DCT, DWT, and DFT. After the modification of the coefficients, the image is converted back to the spatial domain. This procedure needs a certain amount of computation, but it has more potential to prevent watermark destruction by a malicious attack. Thus, transform domain techniques dominate the current literature.

1.5.1.2.1 Discrete Cosine Transform

The DCT is an important transform in 2-dimensional signal processing. It is known to be close to optimal in terms of its energy compaction capabilities and can be computed via a

fast algorithm. The DCT is used in two international image/video compression standards, Joint Photographic Experts Group (JPEG), and Motion Picture Experts Group (MPEG). To compute two-dimensional discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT), we use the following pair of formulas:

$$X[k_1,k_2]=\alpha[k_1]\alpha[k_2]\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}x[n_1,n_2]\cos\left(\frac{\pi(2n_1+1)k_1}{2N_1}\right)\cos\left(\frac{\pi(2n_2+1)k_2}{2N_2}\right) \quad (1.4)$$

$$x[n_1,n_2]=\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}\alpha[k_1]\alpha[k_2]X[k_1,k_2]\cos\left(\frac{\pi(2n_1+1)k_1}{2N_1}\right)\cos\left(\frac{\pi(2n_2+1)k_2}{2N_2}\right) \quad (1.5)$$

$$for \quad k_1=0,1,...,N_1-1 \quad and \quad k_2=0,1,...,N_2-1$$

$$\alpha[k]=\begin{cases}\sqrt{\frac{1}{N}} & for \quad k=0\\ \sqrt{\frac{2}{N}} & for \quad k=1,2,...,N-1\end{cases}$$

In formula (1.4), X returns the two-dimensional DCT of x where N_1 and N_2 are the row and column sizes of X, respectively. The matrix X is the same size as x, and contains the DCT coefficients $X[k_1,k_2]$. In formula (1.5), $x[n_1,n_2]$ is the two-dimensional inverse DCT of $X[k_1,k_2]$.

Many watermarking algorithms use either a block-based or global DCT. In 1997, Cox et al proposed a watermarking technique based on global DCT [12]. Barni et al [26] provided a watermarking algorithm for digital images which embeds a pseudo-random

sequence of real numbers in a selected set of DCT coefficients. Koch and Zhao [27] pseudo-randomly select 8×8 DCT blocks as embedding area in their algorithm.

1.5.1.2.2 Discrete Wavelet Transform

The wavelet transform has been extensively studied in the last decade. Many applications of the wavelet transform have been found. The basic idea of the DWT for a one dimensional signal is as follows. A signal is split into two parts, usually high frequencies and low frequencies. The edge components of the signal are largely confined in the high frequency part. This process is continued until signal has been entirely decomposed or stopped by the given application. The original signal can be reconstructed from the DWT coefficients. The reconstruction process is called the inverse DWT (IDWT).

Mathematically, the DWT and IDWT can be stated as follows. Let

$$H(w) = \sum_k h_k \cdot e^{-jkw}, \tag{1.6}$$

and

$$G(w) = \sum_k g_k \cdot e^{-jkw} \tag{1.7}$$

be a low-pass and a high-pass filter, respectively, which satisfy the orthogonal condition

$$\left|H(w)\right|^2 + \left|G(w)\right|^2 = 1 \tag{1.8}$$

An example of such $H(w)$ and $G(w)$ is given by

$$H(w) = \frac{1}{2} + \frac{1}{2}e^{-j\omega} \qquad \text{and} \qquad G(w) = \frac{1}{2} - \frac{1}{2}e^{-j\omega} \tag{1.9}$$

which is known as the Haar wavelet filter. Other common filters used in image processing are the family of Daubechies orthogonal and bi-orthogonal filters. A signal $F(n)$ can be decomposed recursively as

$$f_{j-1}^{low}(k) = \sum_n h_{n-2k} f_j(n) \tag{1.10}$$

and

$$f_{j-1}^{high}(k) = \sum_n g_{n-2k} f_j(n) \tag{1.11}$$

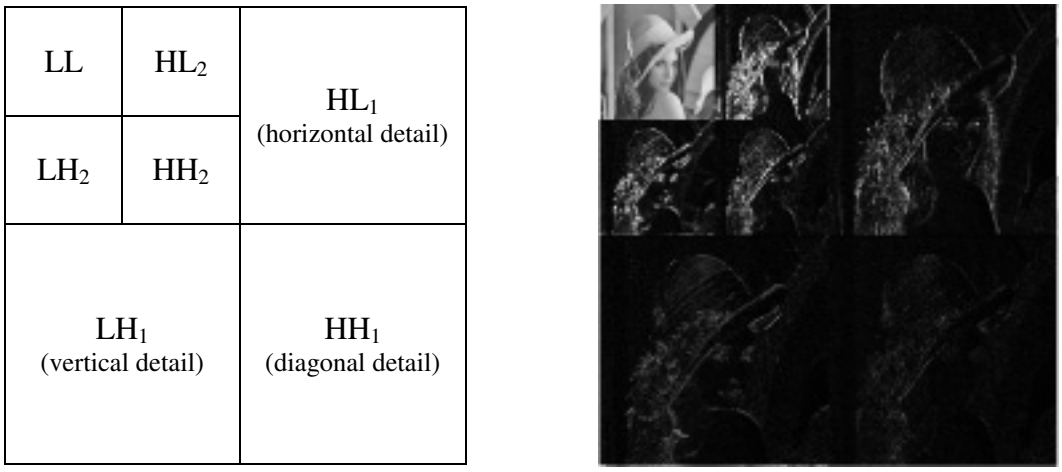
for $j = J+1, J, ..., J_0$ where $f_{J+1}(k) = F(f), k \in Z$. $J+1$ is the highest resolution level index and J_0 is the low resolution level index. The coefficients

$$f_{J_0}^{low}(k), f_{J_0}^{high}(k), f_{J_0+1}^{high}(k), ..., f_J^{high}(k) \tag{1.12}$$

are called the DWT coefficients of the signal $F(n)$, where $f_{J_0}^{low}(k)$ is the lowest resolution part of $F(n)$ (the approximation) and the $f_j^{high}(k)$ are the details of $F(n)$ at various bands of frequencies. The signal $F(n)$ can be reconstructed from the DWT coefficients recursively by

$$f_j^{low}(n) = \sum_k h_{n-2k} f_{j-1}^{low}(k) + \sum_k g_{n-2k} \cdot f_{j-1}^{high}(k) \tag{1.13}$$

The DWT and IDWT for a two dimensional image *F(m,n)* can be similarly defined by implementing the one dimensional DWT and IDWT for each dimension *m* and *n* separately, resulting in the pyramidal representation of an image shown in Figure 1.2.



(a) decomposition structure (b) decomposed image

Figure 1.2. The pyramidal two-level decomposition of an image

Watermarking techniques operating in the wavelet transform domain have become attractive because it has inherent robustness against JPEG-2000 lossy compression if the low frequency band is selected for watermark embedding, and, additionally, the wavelet transform provides a multiresolution representation of images, which can be exploited to build more efficient watermark detection schemes. Zhu et al [17] propose adding a mark, a Gaussian sequence of pseudo-random real numbers into all the high-pass bands in the wavelet domain according to the formula $v_i^{'} = v_i(1+\alpha_i x_i)$. An algorithm developed by Xia et al [22] utilize large DWT coefficients of the high and middle frequency bands to embed a random Gaussian distributed watermark sequence. Dugad et al [28] provide a method to embed a Gaussian sequence of pseudo-random real numbers into selected coefficients in all detail subbands with magnitude above a given threshold in three-level decomposition with Daubechies–8 filters.

In general, the watermark embedded in low pass bands of wavelet domain is robust against a group of attacks such as low pass filtering, adding Gaussian noise and lossy

compression and that in high pass bands is resistant to another set of attacks such as histogram equalization, intensity adjustment and gamma correction [29].

1.5.1.2.3 Discrete Fourier Transform

The discrete Fourier transform (DFT) is the primary tool of digital signal processing. The foundation of the DFT is the Fast Fourier Transform (FFT) algorithm. For a 2-dimensional signal of length *N*, the transform and its inverse are defined by:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)} \tag{1.14}$$

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M + vy/N)} \tag{1.15}$$

The discrete Fourier transform of an image is generally complex-valued, resulting in a magnitude and phase representation for the image. Adding a watermark to the phase of the DFT, as was proposed in [30], improves the robustness of the watermark because any modification of those visually important image components in an attempt to remove the watermark will significantly degrade the quality of the image. Adding a watermark to the DFT magnitude coefficients and ignoring the phase was proposed in [31]. It is reported that that all major compression schemes (JPEG, set partitioning in hierarchical trees (SPIHT), and MPEG) preserved both the DFT magnitude coefficients as well as the DFT phase.

Another reason for using the DFT magnitude domain for watermarking is its translation-invariant property. From the translation property of the Fourier transform, it is clear that

spatial shifts affect only the phase representation of an image. This leads to the well known result that the DFT magnitude is circular translation invariant. Image translation, as well as image scaling and rotation, desynchronize the image and thus make the embedded watermarks undetectable. It requires an exhaustive search over large space as a synchronization process before the watermark detector is applied. The basic idea of using DFT magnitude domain is to avoid a need for synchronization search by transforming the image into a new workspace that is invariant to specific geometrical transformations, and embedding the watermark in that workspace. In 2000, Caldelli et al [18] proposed a scheme that exploited the theory of geometric invariants by modifying the magnitudes of some DFT coefficients. The inserted watermark is robust to most geometric manipulations. O'Ruanaidh and Pun [32] proposed embedding the watermark in the Fourier–Mellin transform domain. They performed the Fourier transform of the image, resampling the Fourier magnitudes into log-polar coordinates, and then summing a function of the magnitudes along the log-radius axis. Such a scheme is robust against rotation, scaling, and translation.

1.5.2 Blind and Non-blind Techniques

As described before, watermarking techniques can be classified according to whether or not the original data is used in extraction/detection procedure. In 1997, Cox et al [12] proposed watermarking technique based on spread spectrum. They embed the watermark into the lower frequency coefficients in the DCT domain. Their method needs the original image and the embedding strength to detect the presence of the watermark. However, the original source might not be available in many applications. Barni et al [26] present a

method to overcome the non-blind watermark problem. They correlate the watermark sequence w directly with all N coefficients of the received image signal, and then compare the correlation value with some detection threshold. Only the watermark sequence and scaling factor are needed in watermark detection. This approach is widely utilized in the watermarking community. However, it turns out that blind techniques are less secure than non-blind methods.

1.5.3 Types of Algorithms in Transform Domain

Watermarking techniques can be classified according to whether they use embedding based on additive algorithms or quantization algorithms.

1.5.3.1 Additive Algorithms

Additive embedding strategies are characterized by the linear modification of the host image and correlative processing in the detection stage. A considerable number of image watermarking techniques share this architecture. In most algorithms, the signature data is a sequence of numbers w_i of length N that is embedded in a suitable selected subset of the host signal data coefficients. Three basic and commonly used embedding formulas are:

$$V_i^{'} = V_i(1 + \alpha \cdot w_i) \tag{1.16}$$

$$V_i^{'} = V_i + \alpha \cdot w_i \tag{1.17}$$

$$V_i^{'} = V_i \cdot e^{\alpha w_i} \tag{1.18}$$

where α is a weighting factor influences the robustness as well as the visibility and V' is the resulting modified host data coefficients carrying the watermark information. The

majority of watermarking systems presented in the literature fall into this class, differing chiefly in the signal design, the embedding, and the retrieval of the watermark content.

Extraction process is accomplished by applying the inverse embedding formulas. The extracted watermark sequence w^* is compared with the original watermark w using the normalized correlation of the sequences as a similarity measure

$$\delta = \frac{w^* \cdot w}{\left\|w^*\right\| \cdot \left\|w\right\|} \tag{1.19}$$

For a blind retrieval of the watermark, a statistical detector is proposed based on the following formula:

$$\delta = \frac{\Sigma^N V_i^* \cdot w_i}{N} \tag{1.20}$$

δ is retrieved by correlating the watermark sequence w directly with all N coefficients of the received image signal V^*. A large number of random sequences are tested, but only the sequence that was originally embedded yields a high correlation output. Therefore, we can conclude that the image has been watermarked with w, as shown in Figure 1.3. A detection threshold τ can be established to make the detection decision, $\delta > \tau$. The detection threshold can be derived either experimentally or analytically (see Chapter 5 for detail).

The algorithm developed by Dugad et al [28] in 1998 makes use of a sequence of pseudo-random Gaussian real numbers, matching the size of the detail subbands of wavelet domain. The authors performed a three-level decomposition with Daubechies–8 filters,

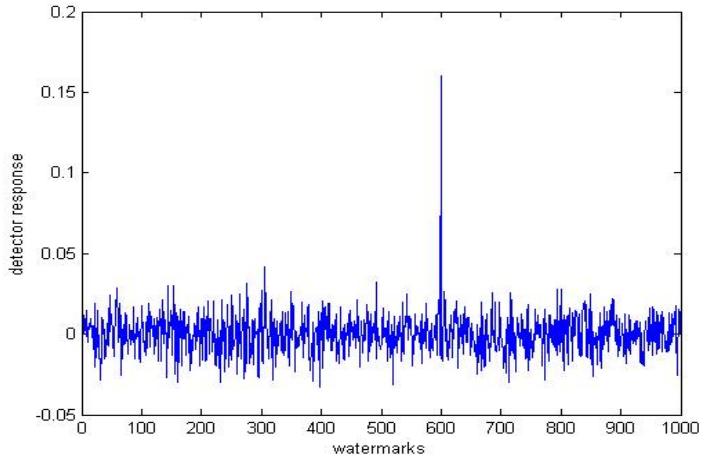


Figure 1.3. The detector response of embedded watermark against a large number of random sequences

and selected all coefficients in all detail subbands whose magnitude is above a given threshold T_1. The equation used for watermark embedding is described as

$$V_i^{'} = V_i + \alpha \cdot |V_i| \cdot w_i \tag{1.21}$$

The blind detection method depends on Equation 1.20. The correlation value δ is compared with detection threshold τ,

$$\tau = \frac{\alpha}{2 \cdot N} \sum_{i=1}^{N} \left| V_i^{*} \right| \tag{1.22}$$

where only the coefficients above the detection threshold $T_2 > T_1$ are considered. Experimental results demonstrated that the watermark is robust to many signal processing techniques.

1.5.3.2 Algorithms Based on Quantization

The quantization schemes perform non-linear modifications during embedding and detect

the embedded message by quantizing the received samples to map them to the nearest reconstruction point. Quantization is the process of mapping a large – possibly infinite – set of values to a much smaller set. A quantizer consists of an encoder mapping and a decoder mapping. The range of source values is divided into a number of intervals. The encoder represents each interval with a codeword assigned to that interval. The decoder is able to reconstruct a value for every codeword produced by the encoder. Scalar quantizers take scalar values as input and output a codeword that represents a single sample of the source output, while vector quantizers work with vectors of input sequences or blocks of the source input and represent one of them with a single codeword.

Watermarking by quantization index modulation (QIM), proposed by Chen and Wornell [33], is based on a set of N-dimensional quantizers. The algorithm is illustrated in Figure 1.4 in which the reconstruction points belonging to two quantizers are marked with x and o respectively. The message *m* that should be transmitted is the index for the quantizer used for quantizing the host-signal vector c_o. To embed one bit m, $m \in \{1,2\}$, the host

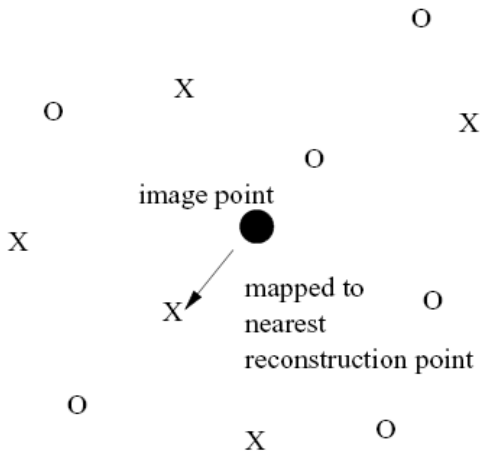


Figure 1.4. Quantization index modulation.

signal is mapped to the nearest reconstruction point x (for m=1) or o (for m=2). During watermark extraction, one evaluates a distance metric to all quantizers. The index of the quantizer with the smallest distance contributes to the message m.

To reduce the distortion caused by quantization and reconstruction, the distortion constraint has to be fulfilled: $E[e^2(u, c_o)] \leq D_s$ [33] where u is a sample variable generated from c_o via quantization, e is a deterministic function of (u, c_o) evaluating distortion between u and c_o. The expected squared error E is bounded by D_s (Ds is a function of vector length and the step size of quantizers[33]). To increase robustness, the codewords produced by different quntizers for a given input should be as far from each other as possible.

1.6 Attacks on Digital Watermarks

A watermarking system is designed to be robust against various image attacks. The watermark should survive after the image has undergone attacks either friendly or malicious. The friendly attack is generally described as an unintentional event where the user has no knowledge of the watermark and its embedding procedure. Conventional image or data operations applied in the normal use of computer technology such as lossy compression, gamma correction and contrast adjustment can be categorized into this group. The second kind of attack, the malicious attack, occurs with the intention of eliminating the watermark in the content. With partial knowledge of the watermark and the process of watermarking encoder, pirates might deliberately operate on the image in order to destroy the watermark information. In the literature, image attacks are classified into the following three groups.

1.6.1 Simple Attacks

Simple watermark attacks try to wipe out the watermark information by manipulating the whole image and its components. The attack does not isolate or identify the specific watermark information. The goal is to add distortion to the host image in order to render the watermark un-detectable or un-readable. The attack is successful when the watermarking information cannot be extracted or recognized anymore but the image is still intelligible and can be used for a particular purpose. Examples are common signal processing operations, such as lossy compression, addition of Gaussian noise, bluring, linear filtering (as in high-pass or low-pass filtering), nonlinear filtering (such as media filtering), color reduction, D/A-A/D conversions, resampling, requantization, and dithering distortion.

1.6.2 Geometrical Attacks

Geometrical attacks attempt to break the synchronization between the original and received watermarked images. Many proposed image watermarking techniques are sensitive to geometric distortions, such as rotation, scaling, translation, cropping and shearing. It is well known that a small amount of rotation and/or scaling can dramatically disable the receiver from detecting the watermark. For instance, it is evident that rotation by a certain angle will substantially lower the performance of watermarking applications for some block-based embedding algorithm. This is because the rotation removes the correspondence between the blocks of the original image and the blocks of the rotated image. Thus the detection of the watermark requires a synchronization step to locate the embedded watermark in the content.

Many efforts [32,35,36] are made to estimate and compensate for geometric distortion in order to synchronize the location of embedded watermark. Robustness to global geometric distortions often relies on the use of either a transform–invariant domain (Fourier-Melline), an additional template, or specially designed periodic watermarks whose auto-correlation allows estimation of the geometric distortion. More details will be discussed in Chapter 4.

1.6.3 Ambiguity Attacks or Removal Attacks

Ambiguity attacks disable the watermark by inserting a new, overlapping watermark in the source image (rewatermarking) or by averaging separately watermarked images (collusion). Removal attacks analyze the watermark, estimate the technique or watermark, and attempt to extract the watermark in order to delete it. Sophisticated removal attacks try to optimize operations like denoising or quantization to impair the embedded watermark as much as possible while keeping the quality of the attacked document high enough.

1.7 Evaluation of Watermarking System

Once a watermarking system has been designed and implemented, it is important to be able to evaluate its performance objectively.

1.7.1 Evaluation of Invisibility

The distortion of watermarked image can be represented as a measure of difference or distance between the original and the watermarked signal. One of the simplest distortion

measures is the mean squared error (MSE) function defined as

$$MSE = \frac{1}{N}\sum^{N}(F_i^{'} - F_i)^2, \tag{1.23}$$

which is the mean of term-by-term difference between the input signal (the original image, F) and the output signal (the watermarked image, $F^{'}$). The most popular distortion measures are signal-to-noise ratio (SNR) and peak-signal-to-noise ratio ($PSNR$). The SNR is defined as,

$$SNR = \frac{\frac{1}{N}\sum_i^N F_i^2}{MSE}, \tag{1.24}$$

which represents the size of the error relative to the input signal, alternatively,

$$SNR(dB) = 10\log_{10} SNR \tag{1.25}$$

can be used in units of decibels, and the PSNR is given by

$$PSNR(dB) = 10\log_{10}\frac{F_{peak}^2}{MSE}, \tag{1.26}$$

where F_{peak} is the peak value of the input signal (usually 255 for 8-bit grayscale images).

Those distortion metrics described above are simple and popular. One particular advantage is that they do not depend on subjective evaluations. Their disadvantage is that they are not correlated with human vision. In other words, a small metric distance between the original and the watermarked signal does not always guarantee high fidelity

of the watermarked signal. Wang and Bovik [34] have proposed a new quality metric called the *Universal Image Quality Index* (UIQI). The new index is mathematically defined as [34]:

$$Q = \frac{4\sigma_{xy}\mu_x\mu_y}{\left(\sigma_x^2 + \sigma_y^2\right)\left(\mu_x^2 + \mu_y^2\right)} \tag{1.27}$$

where x is the original image, y is a distorted version of x, and

$$\mu_x = \frac{1}{N-1}\sum x_i, \qquad \mu_y = \frac{1}{N}\sum y_i,$$
$$\sigma_x^2 = \frac{1}{N-1}\sum \left(x_i - \mu_x\right)^2, \qquad \sigma_y^2 = \frac{1}{N-1}\sum \left(y_i - \mu_y\right)^2$$
$$\sigma_{xy} = \frac{1}{N-1}\sum \left(x_i - \mu_x\right)\left(y_i - \mu_y\right)$$

The dynamic range of Q is [-1,1] with the best value achieved when $x_i = y_i$, i=1,2,…n. This index models any distortion as a combination of three factors-loss of correlation, mean distortion and variance distortion:

$$\frac{\sigma_{xy}}{\sigma_x\sigma_y}, \qquad \frac{2u_x u_y}{u_x^{\,2} + u_y^{\,2}} \qquad \text{and} \quad \frac{2\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2}$$

The authors claim that it performs significantly better than the widely used MSE metric in evaluation of perceived distortion.

Watermark perceptibility can also be measured using different techniques developed as a result of the Human Visual System (HVS) [46,47] studies. In general, modeling the HVS

is very complex and the resulting quality metrics have not shown any clear advantage over simple distortion metrics so far [34].

1.7.2 Analyzing Detection Errors

Errors are inevitable in even the best-designed watermarking systems. A *false positive error* occurs when the detector incorrectly indicates that a watermark is present. Conversely, a *false negative error* occurs when a detector incorrectly indicates the absence of a watermark[4]. It should be noted that whereas false positives depend only on the detection algorithm, false negatives also depend on the embedding algorithm and what kind of distortion introduced to the cover image. False-positive and false-negative errors are interrelated according to a selected threshold as shown in Figure 1.5. Curves in Figure 1.5 are distribution of the detector response of unwatermarked images and

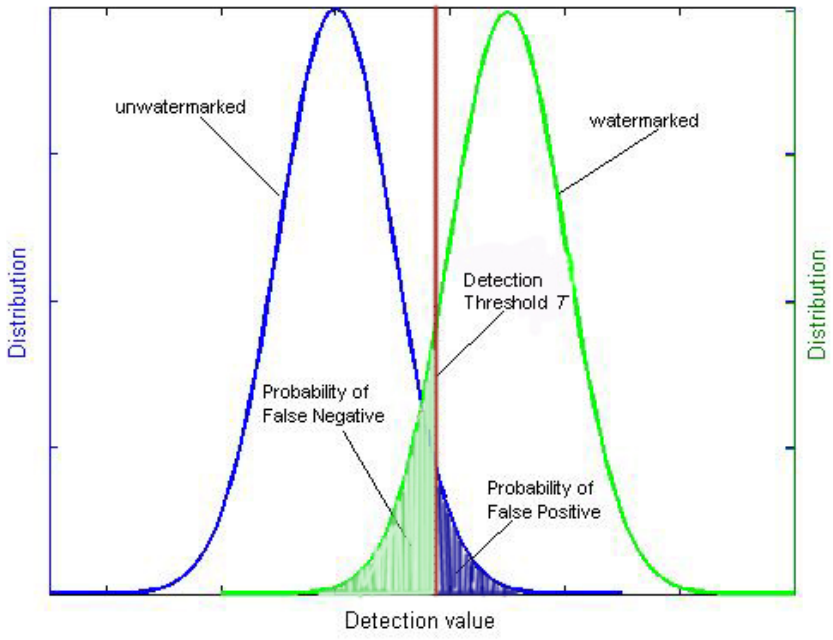


Figure 1.5. False-positive and false-negative errors are interrelated according to a selected threshold.

watermarked images respectively. For a selected threshold value *T*, the area to the right of threshold line under the left-hand curve (darkly shaded area) represents the probability of a false positive and the area to the left of threshold line under the right-hand curve (lightly shaded area) is the probability of false negative. Thus, it is not possible to minimize both probabilities (and error rates) simultaneously; those two errors should be measured and presented together — for example, using a receiver operating characteristics (ROC) curve. Therefore it is necessary to develop models with the purpose of minimizing errors, and designer would determine what error rates are acceptable in one particular stage of the system development. Such a model would allow us to select a detection threshold to meet the requirements and be confident that the specified error rates will not be exceeded. More detail about the model of statistically selecting detection threshold is discussed in Chapter 5.

1.7.3 Evaluation of Other Properties

Data capacity, also called data payload, refers to the amount of information embedded within a unit of host signal. Capacity is an important property because although high capacity is desirable it has a direct negative impact on watermark transparency. Capacity can be evaluated by calculating the ratio of capacity (e.g., payload size) to some parameter of reliability (e.g., error rate). Those results can then be used to estimate the theoretical maximum capacity of the watermarking system under consideration.

There are a number of benchmarking tools that have been created to standardize watermarking system evaluation processes. Stirmark[37,38] is a benchmarking tool for digital watermarking designed to test robustness. The following image attacks have been

implemented in Stirmark Version 3.1: cropping, flip, rotation, rotation scale, sharpening, Gaussian filtering, random bending, linear transformations, aspect ratio, scale changes, line removal, color reduction, and JPEG compression. Checkmark [39] is a benchmarking suite for digital watermarking developed on Matlab under UNIX and Windows. It has also been recognized as an effective tool for the evaluation and rating of the robustness of watermarking systems.

1.8 Structure of the Thesis

The thesis is organized as following. In Chapter 2 we propose a robust block-based watermarking scheme in curvelet domain and test its robustness against a variety of attacks. In Chapter 3 we present a perceptual data hiding method based on Barten's (1990) model so that we can adaptively embed watermark into host signals. The proposed watermarking scheme has weak robustness against geometrical distortion. Thus, we describe an effective method for detecting and recovering geometrical distortion in Chapter 4. A technique of threshold selection for watermark detection based on statistical analysis over host signals and embedding schemes is given in Chapter 5. Chapter 6 presents a new quality measure, M-SVD, can express the quality of distorted images/videos either numerically or graphically. Finally the quality measure M-SVD is applied to the watermarked images produced by our block-based watermarking algorithm in curvelet domain.

Chapter 2

Image Watermarking Scheme in the Curvelet Domain

In this chapter, we propose a robust watermarking scheme operating in the curvelet domain. The curvelet transforms take the curve as the basic representation element; it provides optimally sparse representations of objects along a general curve with bounded curvature – such as images with edge [44]. The image is partitioned into blocks; Fast Discrete Curvelet Transform via Unequally-Spaced Fast Fourier Transforms (FDCT-USFFT) is employed to transform each block into curvelet domain. We embed the watermark into the selected blocks, scale and curvelet coefficients based on the edge map of the cover image. As usual, the watermarks are blindly detected using a correlation detector. Experimental results demonstrate that the embedded watermark survives severe image attacks and shows advantages over the watermark in wavelet domain.

2.1 Introduction

In the past two decades, many researchers have proposed embedding watermark in the wavelet transform domain [15,16,17], which provides multiresolution representation [40] of the cover work. Embedding watermarks hierarchically, starting from the low-resolution subbands first, along to higher resolution subbands, makes watermarks inherently robust to JPEG-2000 lossy compression as well as other image attacks. Despite considerable success, intense research in the past few years has shown that

classical multiresolution ideas are far from ideal in image representation. For one dimensional signal, wavelet transform provides near optimal representation of one dimensional piecewisely smooth signal with point singularities. However wavelets are not well suited to efficiently providing a compact representation of high dimensional structures, it loses its advantages when dealing with two dimensional piecewisely smooth signal e.g, curves or edges with line singularities. The curvelet transform [41,42,43] was developed in the last few years in an attempt to overcome these inherent limitations of traditional multiscale representations.

"Conceptually, the curvelet transform is a multiscale pyramid with many directions and positions at each length scale, and needle-shaped elements at fine scales." [42]. Curvelet transform directly take the edge as the basic representation element; it provides optimally sparse representations of objects along edges. Such representations are far more sparse than the wavelet decomposition of the object [41,42,44]. "The implication in statistics is that one can recover such objects from noisy data by simple curvelet shrinkage and obtain a Mean Squared Error (MSE) order of magnitude better than what is achieved by more traditional methods."[44] Therefore, we want to take the advantages of the optimal sparse representation of edges in curvelet domain, hide information into the significant curvelet coefficients to achieve high robustness. We begin with an introduction to curvelet transform.

2.1.1 Curvelet Transform

Along with the wavelet transform and the ridgelet [45] transform, the curvelet transform theory is based on sparsity theory [45]. The following two sections summarize the

curvelet mathematical transform and digital implementation presented in [41,42,44].

2.1.1.1 Continuous-Time Curvelet Transform

The idea of curvelets is to compute the inner product between the signal or function and the curvelet function to realize the sparse representation of the signal or function. So the curvelet transform can be expressed as

$$c(j,\ell,k)=\left\langle f,\varphi_{j,\ell,k}\right\rangle \tag{2.1}$$

here, j =0, 1, 2, . . . is a scale parameter; $\ell = 0, 1, \ldots\ 2^{\lfloor j/2 \rfloor}-1$ is an orientation parameter; and $k=(k_1,k_2)\in Z^2$ is a translation parameter. In the frequency domain, curvelets are compactly supported; each element is localized near the symmetric wedge with the length about 2^j and width about $2^{j/2}$ .Such a symmetric wedge is illustrated in Figure 2.1 [44].

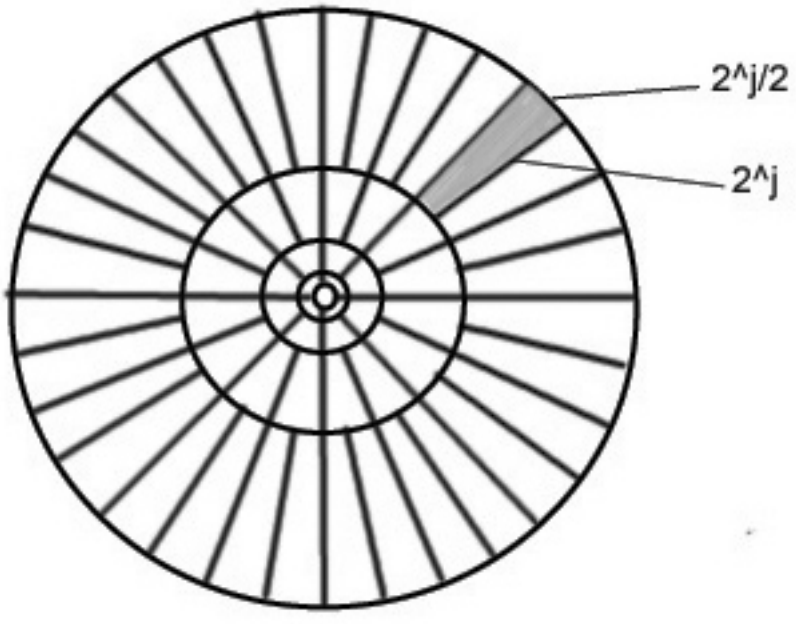


Figure 2.1. Structure of curvelet transform element in the frequency domain

This explains their oscillatory nature: at scale 2^{-j}, a curvelet is a little needle whose envelope is a specified 'ridge' of effective length $2^{-j/2}$ and width 2^{-j}. The curvelet waveform $\varphi_j(\omega)$ is defined by means of its Fourier transform $\hat{\varphi}_j(\omega) = U_j(\omega)$, U_j here is the frequency window defined in the polar coordinate system [44]:

$$U_j(r,\theta) = 2^{-3j/4} W(2^{-j} r) V\left(\frac{2\lfloor j/2 \rfloor \theta}{2\pi}\right). \tag{2.2}$$

where W and V are radial and angular windows respectively and will always obey certain admissibility conditions [44]:

$$\sum_{j=-\infty}^{\infty} W^2\left(2^j r\right) = 1, \quad r \in (3/4, 3/2) \quad and \quad \sum_{\ell=-\infty}^{\infty} V^2(t-\ell) = 1 \quad t \in (-1/2, 1/2) \tag{2.3}$$

With this notation, curvelets (as function of x=(x_1, x_2)) at scale 2^{-j}, orientation θ_ℓ and position $x_k^{(j,\ell)} = R_{\theta_\ell}^{-1}\left(k_1 \cdot 2^{-j}, k_2 \cdot 2^{-j/2}\right)$ can be expressed as

$$\varphi_{j,\ell,k}(x) = \varphi_j\left(R_{\theta_\ell}\left(x - x_k^{(j,\ell)}\right)\right), \tag{2.4}$$

where $\theta_\ell = 2\pi \cdot 2^{-\lfloor j/2 \rfloor} \cdot \ell$, $\ell = 0,1\ldots$, $0 \le \theta_\ell < 2\pi$, R_θ is the rotation by θ radians and R_θ^{-1} its inverse (also its transpose),

$$R_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \; R_\theta^{-1} = R_\theta^T = R_{-\theta}. \tag{2.5}$$

A curvelet coefficient is then simply the inner product between an two-dimensional object $f \in L^2(R^2)$ and a curvelet $\varphi_{j,\ell,k}$,

$$c(j,\ell,k) = \left\langle f, \varphi_{j,\ell,k} \right\rangle = \int_{R^2} f(x)\overline{\varphi_{j,\ell,k}(x)}dx\,. \tag{2.6}$$

Since digital curvelet transforms operate in the frequency domain, it can be expressed in the following form:

$$c(j,\ell,k) = \frac{1}{(2\pi)^2}\int \hat{f}(\omega)\overline{\hat{\varphi}_{j,\ell,k}(\omega)}d\omega = \frac{1}{(2\pi)^2}\int \hat{f}(\omega)U_j\left(R_{\theta_\ell}\omega\right)\exp\left(i\left\langle x_k^{(j,k)}, \omega\right\rangle\right)d\omega \tag{2.7}$$

2.1.1.2 Digital Curvelet Transform via FDCT-USFFT

The digital curvelet takes Cartesian arrays of the form $f[t_1, t_2]$, $0 \le t_1, t_2 < n$ as input, and outputs a collection of coefficients $c^D(j,\ell,k)$ expressed as

$$c^D(j,\ell,k) = \sum_{0\le t_1,t_2<n} f[t_1,t_2]\overline{\varphi^D_{j,\ell,k}[t_1,t_2]}, \tag{2.8}$$

where each $\varphi^D_{j,\ell,k}$ is a digital curvelet waveform. In the digital definition, the window U_j does not exactly extract frequencies near the dyadic corona $\{2^j \le r \le 2^{j+1}\}$ and near the angle $\{-\pi \cdot 2^{-j/2} \le \theta \le \pi \cdot 2^{-j/2}\}$, and must be adapted to Cartesian arrays as illustrated in Figure 2.2. The "Cartesian window" $\tilde{U}_j(\omega)$ is the product of radial and angular window such as

$$\tilde{U}_j(\omega) = \tilde{W}_j(\omega)V_j(\omega), \tag{2.9}$$

where $\tilde{W}_j(\omega)$ and $V_j(\omega)$ obey certain admissibility conditions [44].

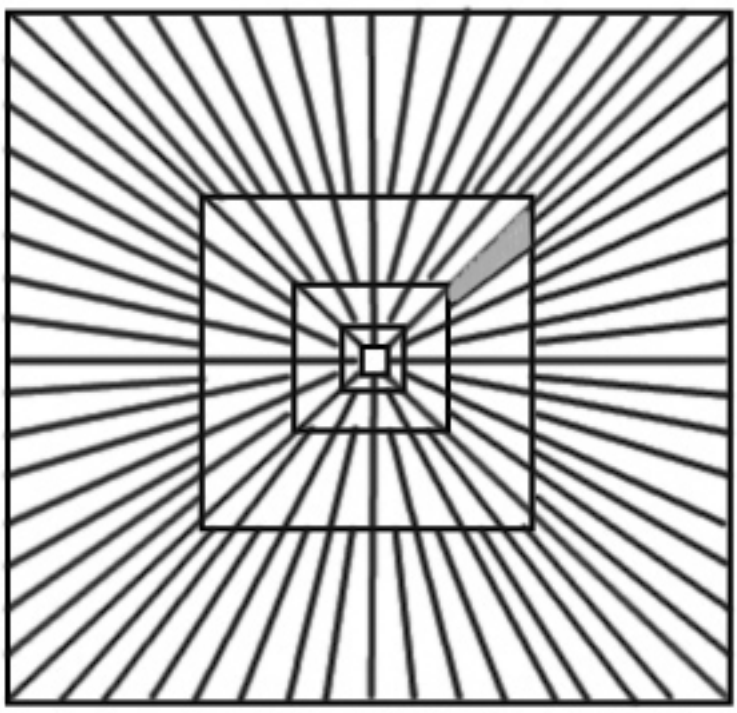


Figure 2.2. Basic digital simulation of curvelets

So given a Cartesian array $f[t_1, t_2]$, $0 \le t_1, t_2 < n$, and let $\hat{f}[n_1, n_2]$ denote its 2D discrete Fourier transform

$$\hat{f}[n_1, n_2] = \sum_{t_1, t_2}^{n-1} f[t_1, t_2] e^{-i 2\pi (n_1 t_1 + n_2 t_2)/n}, \quad -n/2 \le n_1, n_2 < n/2. \tag{2.10}$$

Then the parabolic window $\tilde{U}_j[n_1, n_2]$ is supported on some rectangle P_j of length $L_{1,j}$ and width $L_{2,j}$,

$$P_j = \{(n_1, n_2) : n_{1,0} \le n_1 < n_{1,0} + L_{1,j}, n_{2,0} \le n_2 < n_{2,0} + L_{2,j}\}, \tag{2.11}$$

where $(n_{1,0}, n_{2,0})$ is the index of the pixel at the bottom-left of the rectangle. Therefore the FDCT via USFFT simply evaluates

$$c^D(j,l,k) = \sum_{n_1,n_2 \in P_j} \hat{f}[n_1, n_2 - n_1 \tan\theta_l] U_j[n_1, n_2] \exp\left(i2\pi\left(k_1 n_1 / L_{1,j} + k_2 n_2 / L_{2,j}\right)\right) . \quad (2.12)$$

The implementation referred to as the FDCT via USFFT is roughly as follows [44]:

1. Perform the 2D FFT to a Cartesian array, and obtain Fourier samples $\hat{f}[n_1, n_2], -n/2 \le n_1, n_2 < n/2$

2. Resample (or interpolate) $\hat{f}[n_1, n_2]$ to obtain sampled values $\hat{f}[n_1, n_2 - n_1 \tan\theta_l]$ for $(n_1, n_2) \in P_j$ for each scale/angle pair *(j, l)*,

3. Multiply the interpolated (or sheared) object $\hat{f}$ with the parabolic window $\hat{U}_j$, obtain

$$\bar{f}_{j,l}[n_1, n_2] = \hat{f}[n_1, n_2 - n_1 \tan\theta_l] \hat{U}_j[n_1, n_2].$$

4. Apply the inverse 2D FFT to each $\bar{f}_{j,l}$, hence collecting the discrete coefficients

$c^D(j, l, k)$.

2.2 Proposed Watermarking System

In this chapter, we proposed a novel watermarking scheme in curvelet domain. The framework of the proposed watermarking system is depicted in Figure 2.3. The image is partitioned into small blocks, the Fast Discrete Curvelet Transform (FDCT) via Unequally-Spaced Fast Fourier Transforms (USFFT) is employed to decompose each block into curvelet domain. We embedded the watermark (a pseudo random sequence)

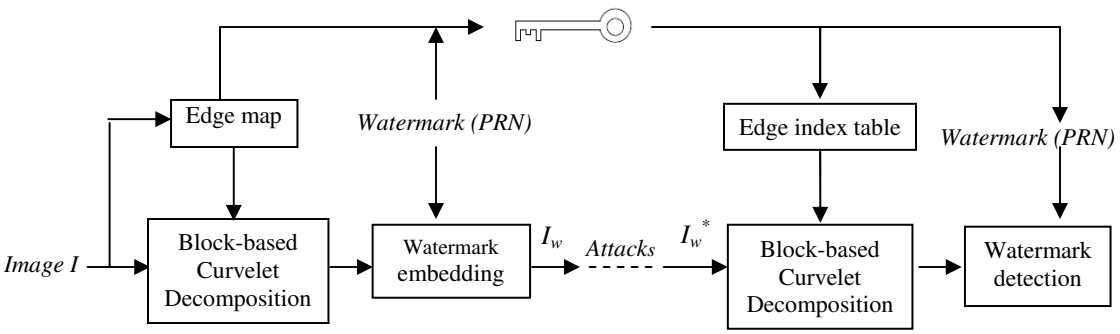


Figure 2.3. Framework of proposed watermarking system

into the selected blocks based on the edge map of cover image. The index table of embedding blocks is recorded and required in the detection process. Thus, a secret key (multi-bit sequence, e.g. 128 bits) is generated and consisted of two parts: The seed value (two large integers each has been allocated for 32 bits) to a pseudo-random number generator that produces the watermark is hidden as the first part in the secret key; the index table converted into a sequence of bits (64 bits) is hidden as the second part in the secret key. Then the cover image I_w is reconstructed, distributed to the public and possibly undergone various attacks. The cover image I_w^* and the secret key are then passed to receiver. As usual, the watermarks are blindly detected using a correlation detector. A real watermark will yield a high response indicating the image is marked. The watermarking algorithm is described in detail as below.

2.2.1 Block Participation and Classification

We use the Canny edge function [89] to detect edges of the original image *I(x,y)*. The resulting edge map is a binary image $B=\{b_{ij}\}$, $i=1,2,\ldots N$, $j=1,2,\ldots N$ containing 1 where

edges are detected and 0 elsewhere. A grid of edge strength is produced based on the edge map. We divide the edge map into small blocks; each receives edge strength as:

$$e_k = \sum_{i,j}^{n} b_{ij} \tag{2.13}$$

where n is the block size. Since changes in blocks with strong edge strength are less visible to human eyes, we only permit the watermark embedding to the blocks whose edge strength is greater than a selected threshold. An edge index table identifying those embedding blocks is constructed based on the edge map and the determined threshold. If

(a)

(b)

114	0	53	84	20	66	111	78
166	16	150	272	208	189	153	157
136	117	441	696	470	348	212	97
141	52	664	742	390	404	136	0
119	560	685	384	341	570	174	65
138	761	694	379	381	531	67	72
263	706	857	382	85	353	190	284
286	605	609	214	0	117	265	344

(c)

1	0	0	0	0	0	1	0
1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	0
1	0	1	1	1	1	1	0
1	1	1	1	1	1	1	0
1	1	1	1	1	1	0	0
1	1	1	1	0	1	1	1
1	1	1	1	0	1	1	1

(d)

Figure 2.4. An example of the edge strength map and the edge index table (a) Original image "Lena" (b) Edge map of "Lena" under Canny detection (c) Corresponding edge strength map with block size 64 (d) Edge index table with bit 1 identifying the embedding blocks whose edge strength is greater than 100

we have a 512×512 cover image and let the block size be 64, the edge index table is thus an 8×8 binary table. Hence, 8 bytes (64 bits) information will be stored into a secret key which is required in detection process. Figure 2.4 shows an example of edge strength map and edge index table with selected threshold 100.

2.2.2 Watermark Embedding

The original image *I(x,y)* is partitioned into non-overlapping blocks of $n \times n$ denoted by B_k, $k=1,2..N$, that is:

$$I(x,y) = \bigcup_k B_k = \bigcup_k I^k(x',y'), 0 \le x', y' \le n \tag{2.14}$$

We apply FDCT-USFFT [44] to the blocks with strong edges and collect curvelet coefficients. The curvelet transform uses concentric squares on separate scales, thus there is one such grid (a wedge-like grid as shown in Figure 2.2) per scale and angle. With the increase of the scales, rectangular grids expand along directions and frequency plane. We leave out the largest n terms in the lowest scale to ensure transparency and choose the significant coefficients in the next immediate scale to embed watermark. Then the watermark $W=\{w_1,w_2,\ldots w_m\}$, a pseudo random sequence of real numbers having normal distribution, zero mean and unit variance, is superimposed into curvelet coefficients according to:

$$c_{l,k}^W = c_{l,k} + a \cdot |c_{l,k}| . w_i \tag{2.15}$$

a is a scaling factor chosen experimentally that controls the embedding strength. We obtain the modified curvelet coefficients for selected blocks and compute the inverse transform to reconstruct the cover image.

2.2.3 Watermark Detection

We evaluate the correlation between watermark and curvelet coefficients to determine whether or not the watermark is present. We split the I_w^* into $n \times n$ non-overlapping blocks and edge map is computed. The secret key is examined. The seed value to random number generator is used to generate the embedded pseudo random sequence i.e. the watermark. The edge index table extracted from the secret key is employed to identify those blocks used for hiding watermark information. We extract coefficients $[C]^* = \{c_{k,i}^*\}$ with k=1,2,…,N and i=1,2,…M from *N* identified blocks. A detector ρ which evaluates the correlation between $[C]^*$ and the watermark $W=\{w_1,w_2,\ldots w_j\}$, is defined by

$$\rho = \frac{1}{NM} \sum_{k=1}^{N} \sum_{i=1}^{M} c_{k,i}^* \cdot w_j \ . \tag{2.16}$$

Using this correlation detector, we determine whether the mark is present or not by comparing ρ with the threshold defined as:

$$T_\rho = k \cdot \sqrt{\frac{1}{NM} \sum_{k=1}^{N} \sum_{i=1}^{M} \left(c_{k,i}^*\right)^2} \tag{2.17}$$

where *k* is a constant greater than 3, which allows for low false positive rate. More detail about statistically selecting this threshold for watermark detection is given in Chapter 5.

The mark that gives the correlation higher than the threshold has been detected.

2.3 Experimental Results

2.3.1 Performance of Proposed Watermarking System

We have tested our proposed watermarking scheme on five different images (Lena, Barbara, Boat, Airplane, Goldhill) of dimension of 512×512. They have been partitioned into blocks of n×n pixels with n=64, thus obtaining 64 blocks. The selected edge-rich blocks are decomposed into curvelet domain via FDCT-USFFT and we obtained curvelet coefficients. We leave out scale 1, add the watermark into all the significant coefficients in scale 2 and choose 0.4 as the scaling factor in Equation 2.15.

We have tested our algorithm using edge-strength threshold 0 (selecting all blocks), 100 and 450. Generally, if we are more concerned with watermark payload and the resistance of the watermark to various image attacks, we want to embed into as many blocks as possible. On the other hand, the quality of the watermarked image may be higher if only a few blocks with strong edge strength are modified. The original image “Lena” and watermarked “Lena”s are given in Figure 2.5. From top down, the edge strength threshold of the selected blocks for watermark embedding are 0, 100 and 450 respectively. The fidelity of cover images are well maintained according to the PSNR measurement. Of course, fewer blocks are modified if we choose a higher edge strength threshold (EST). Thus, as we can see in the third image, the watermark energy is compacted into a few blocks with strong edges, leaving most part of the image unmodified. The sparse

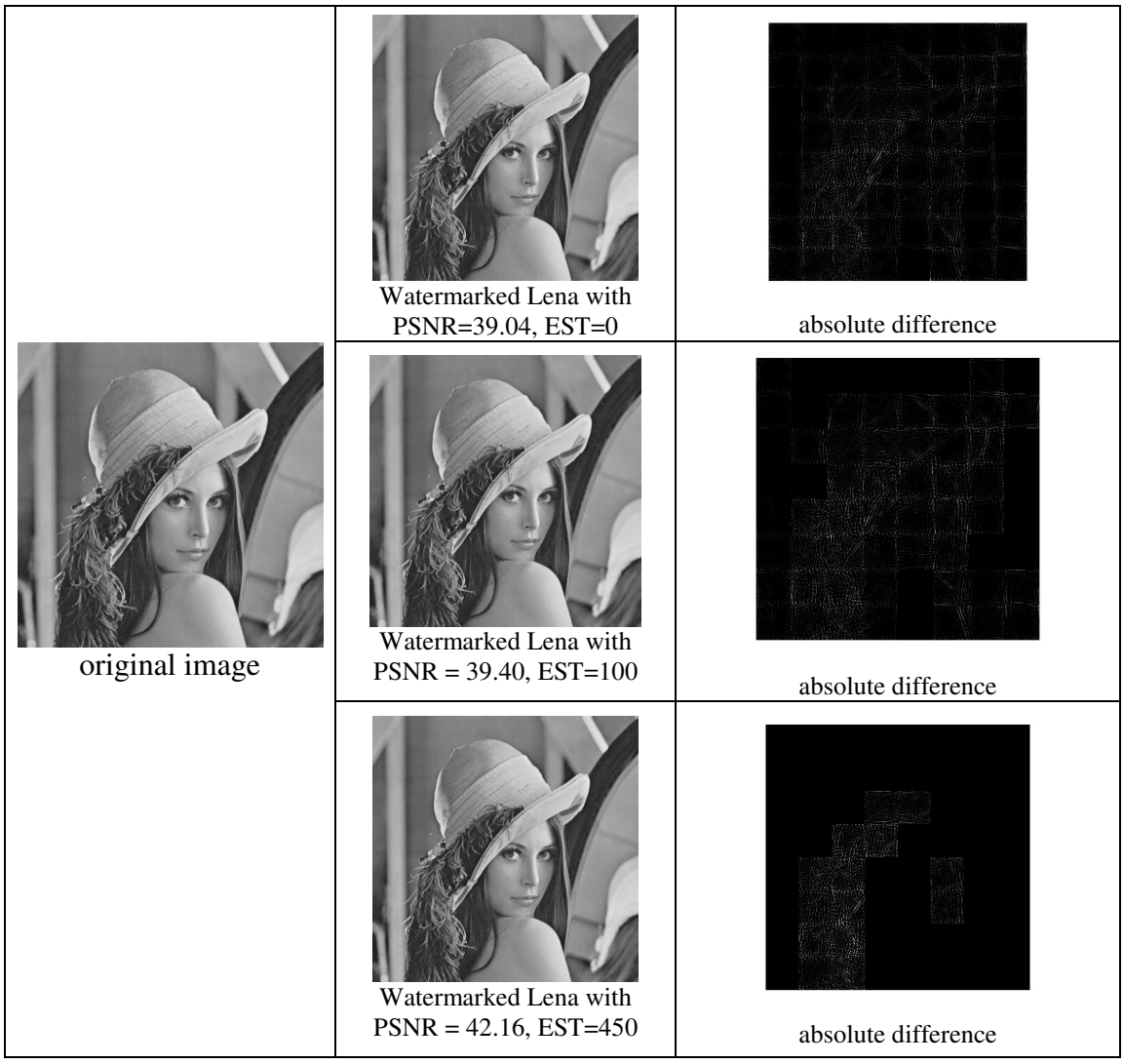


Figure 2.5. The original "Lena", watermarked "Lena"s and absolute difference between them

representation of edges in the curvelet transform allows watermark embedding into the significant component in those blocks with high textures, hence the distortion introduced is less perceivable. Also, because most of the energy of the watermark is imposed into the significant curvelet coefficients, it provides high robustness during detection process. We applied the correlation detector to the watermarked images shown in Figure 2.5 and tested the detector response against 999 fake watermarks. Figure 2.6, 2.7 and 2.8 show that the evaluated correlations (ρ) are 0.1604, 0.2180 and 0.3447 all well above the thresholds T_ρ, and the responses of fake watermarks are all below the thresholds.

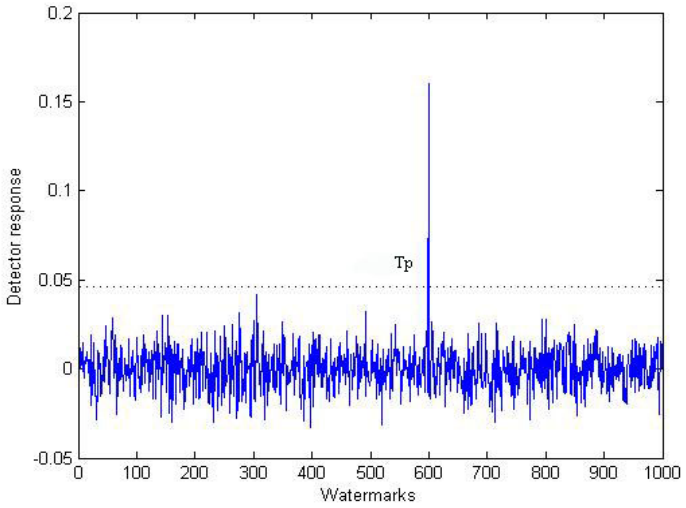


Figure 2.6. The detector responses of the watermarked "Lena" with EST = 0

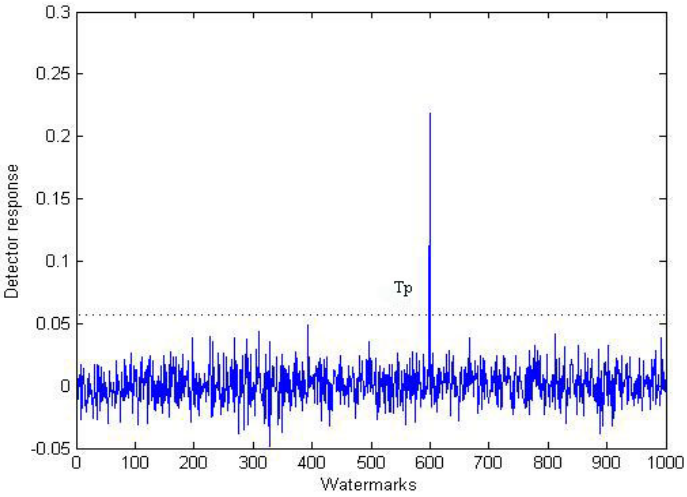


Figure 2.7. The detector responses of the watermarked "Lena" with EST = 100

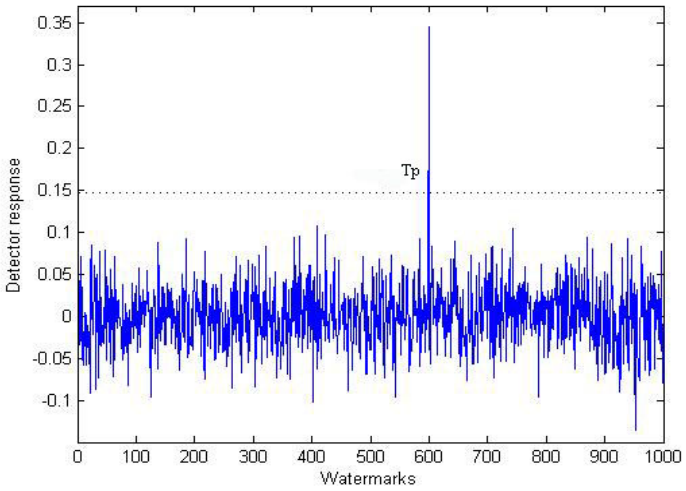


Figure 2.8. The detector responses of watermarked "Lena" with EST=450

Table 2.1 shows the performance of detector response of the watermarked "Lena" with increasing edge strength. ρ_w denotes the detector response of the real watermark while ρ_{maxs} denotes the max response among 999 fake watermarks. Z-value is a statistical measure defined by

$$z = \frac{\rho_w - m_s}{\sigma_s} \tag{2.18}$$

where m_s and σ_s are the mean and standard deviation of the detector responses among 999 fake watermarks. A higher ratio of ρ_w/ρ_{maxs} and Z-value indicate better watermark detecting performance.

The choice of embedding strength is essential for the performance of the watermarking system and also depends on the requirement of application. Generally, if we choose a low embedding strength, then a large amount of curvelet coefficients are included for

Table 2.1. The performance of detector response of watermarked "Lena" along with increasing edge strength threshold (EST)

EST	PSNR	Detector response ρ_w	Max response of fake watermarks ρ_{maxs}	Ratio of ρ_w/ρ_{maxs}	Z- value
0	39.04	0.1605	0.0416	3.8582	13.9427
50	39.18	0.1834	0.0394	4.6548	15.0451
100	39.40	0.2190	0.0487	4.4969	15.7674
150	39.54	0.2295	0.0532	4.3139	12.8207
200	39.88	0.2613	0.0603	4.3333	13.0003
250	40.34	0.2537	0.0659	3.8497	11.7476
300	40.63	0.2602	0.0757	3.4373	10.6863
350	40.87	0.2692	0.0815	3.3031	9.8655
400	41.71	0.3234	0.1200	2.6950	9.5533
450	42.16	0.3447	0.1076	3.2035	9.4129
500	42.22	0.3364	0.1248	2.6955	9.0366
550	42.56	0.3705	0.1269	2.9196	9.4131
600	43.11	0.3277	0.1749	1.8736	6.998

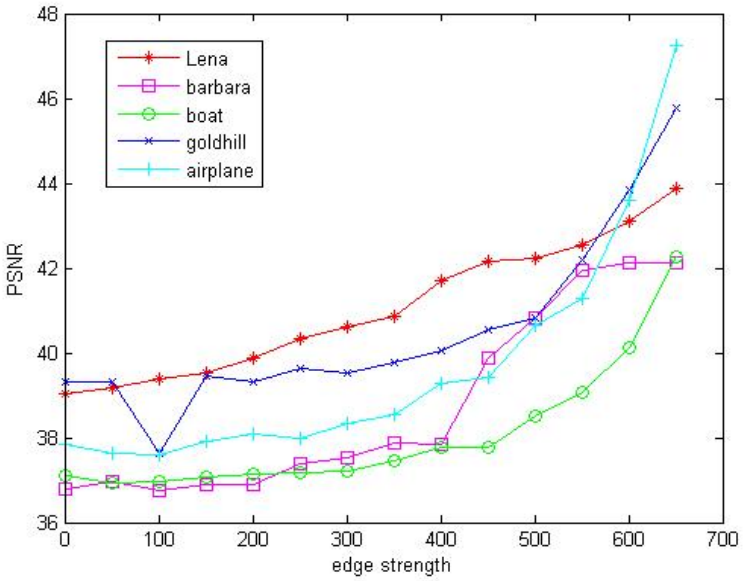


Figure 2.9. Effect of embedding edge strength on PSNR values of marked images

information hiding. The lower the selected embedding strength, the more blocks of coefficients are modified which introduced more distortion, thus the embedded watermark is more robust against malicious attacks. Figure 2.9 shows that the PSNR values of Lena, Barbara, Boat, Goldhill and Airplane are generally varying upward along with increasing edge strength threshold.

The robustness of the watermarking scheme is evaluated in terms of the ratio ρ_w/ρ_{maxs} (detector response to embedded watermark/max detector response of 999 fake watermarks) and Z-value illustrated in Figure 2.10 and 2.11. The ratio ρ_w/ρ_{maxs} and Z-value are generally varying downward along with increasing edge strength threshold. It should be noted that the performance of watermarked "Lena"s with the embedding edge strength 50-100 are better than that of 0 (all blocks are chosen) even though fewer blocks are modified and less distortion is introduced into the cover image. The optimal embedding edge strength appears at the embedding edge strength 100 for "Lena".

Observations show the optimal embedding edge strength is found at 500-550 for "Boat", 50 for "Airplane", 200 for "Goldhill" and 100 or 250 for "Barbara".

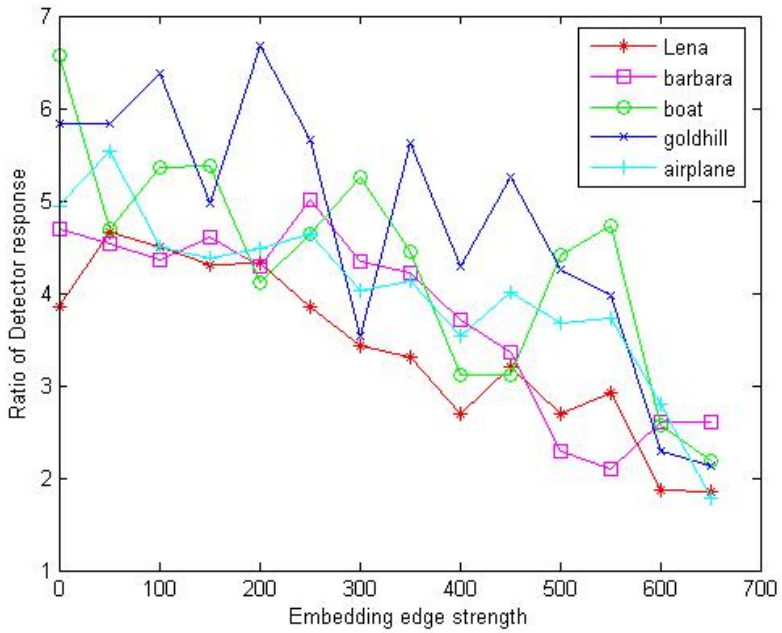


Figure 2.10. Effect of embedding edge strength on watermark robustness measured in term of the ratio ρ_w/ρ_{maxs} (detector response to embedded watermark/max detector response of 999 fake watermarks)

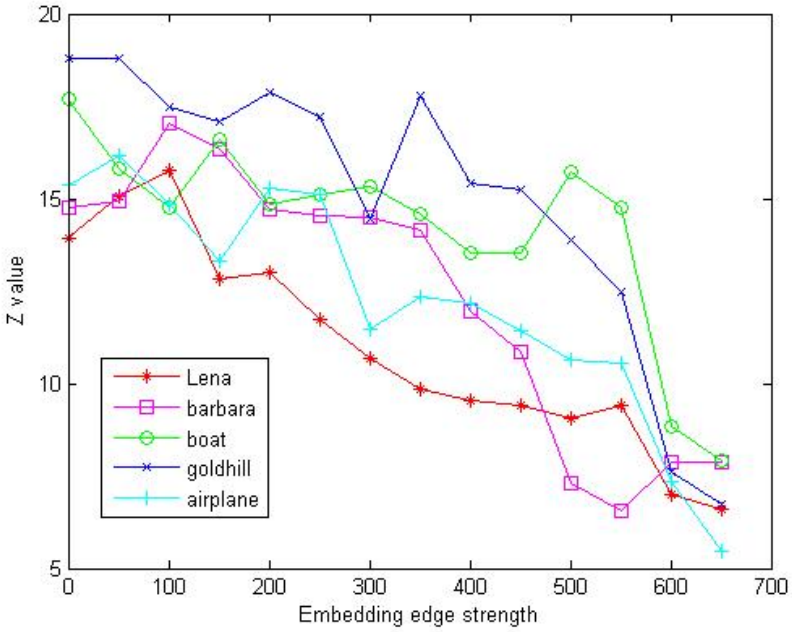


Figure 2.11. Effect of embedding edge strength on watermark robustness measured in terms of Z-value

Therefore, it is essential to take an investigation over cover images before the embedding process and then optimally choose the embedding edge strength so as to maximize the performance of detector response.

2.3.2 Robustness of Watermarking System against Image Attacks

The curvelet transform ensures that most of the energy of the object is localized in just a few coefficients, so it is possible to recover watermarks even from severely degraded images. The proposed curvelet based watermarking system has been tested against a wide range of image attacks. Here we only give the experimental results for standard image "Lena". We choose the block-size 64, the threshold of the embedding edge strength 100, scaling factor 0.4, and then applied the curvelet based algorithm to the 512x512 "Lena". The watermarked "Lena" is shown in Figure 2.12 with PSNR=39.40. Figure 2.12 shows the watermarked images under severe image attacks such as JPEG Compression with quality factor 5, adding Gaussian noise with zero mean and variance=0.1, and the cropping attack in which 75% of the cover image is lost. The watermark detector is applied to those severely corrupted images. The detector response of the embedded watermark against 999 fake watermarks for each case is plotted in Figure 2.13, Figure 2.14 and Figure 2.15 respectively. All watermarks are presented even when the cover image is severely degraded. In Figure 2.13, the detector response of the embedded watermark under JPEG compression is 0.0628 much higher than other responses all less than 0.039 and the threshold 0.0520. In Figure 2.14, the detector response of the embedded watermark attacked by Gaussian noise is 0.2426, far higher than the threshold

PSNR=39.40
(a)

PSNR=21.29
(b)

PSNR=5.33
(c)

PSNR = 1.09
(d)

Figure 2.12. Watermarked "Lena" and distorted "Lena" (a) The watermarked Lena with embedding edge strength 100 (b) The watermarked Lena under JPEG compression with quality factor 5(c) The watermarked "Lena" with Gaussian noise (m=0 and var=0.1) (d) Cropped watermarked "Lena" by 75%

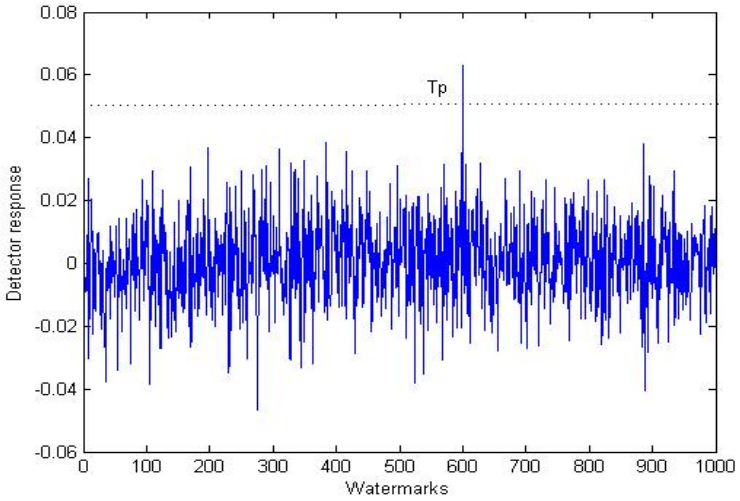


Figure 2.13. The detector responses of the watermarked "Lena" under JPEG Compression with quality factor 5.

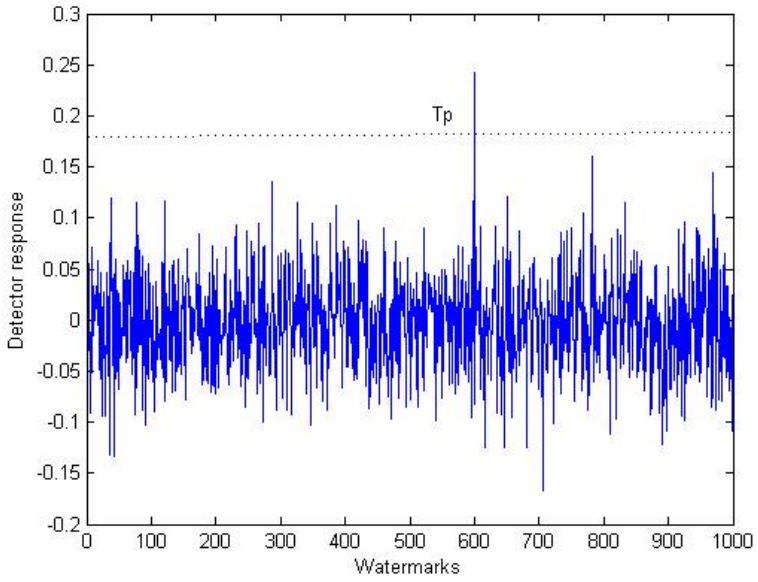


Figure 2.14. The detector responses of the watermarked "Lena" with Gaussian noise (mean=0, variance=0.1) .

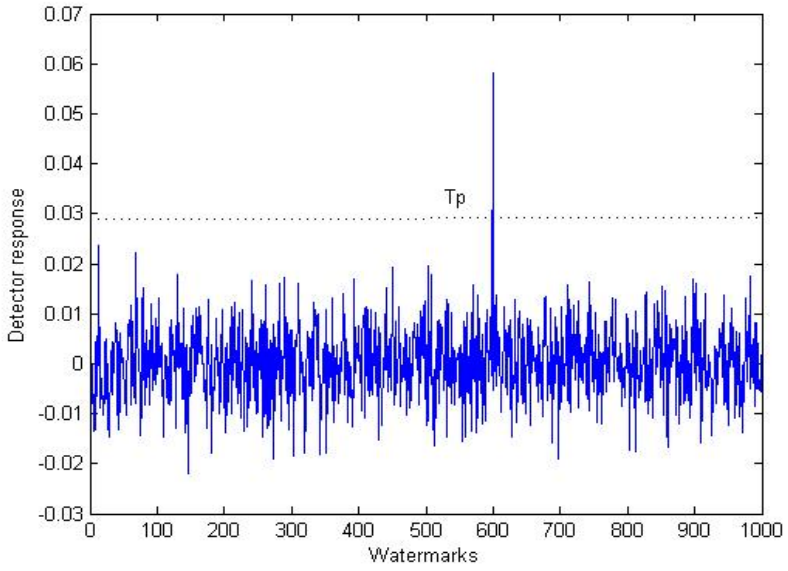


Figure 2.15. The detector responses of the watermarked "Lena" cropped by 75% .

0.1816 and all other responses are less than 0.16. In Figure 2.15, the detector response of the embedded watermark under 75% cropping attack is 0.0582, much higher than 0.0284 (threshold) and 0.0238 (the max response among fake watermarks). All thresholds are computed according to Equation 2.17 in which k =4.

Table 2.2 reports the performance of the robustness of our watermarking system against a wide range of image attacks including JPEG compression, Gaussian noise addition, cropping, histogram equalization, contrast adjustment, low pass filtering, Gaussian blur, Gamma correction, sharpening and rotation. The performance of robustness is evaluated using the ratio of ρ_w/ρ_{maxs} and Z-value defined in Equation 2.18. Values of $\rho_w/\rho_{maxs}>1.5$ and Z-value>4 indicate that the embedded watermark is present. Figure 2.16 has result for collusion attack. Experimental results show the proposed watermarking system is robust against most of image attacks, however the watermark disappeared under one group of attacks—geometric distortions. Our proposed watermarking system is sensitive

Table2.2. Robustness of curvelet-based watermarking system for "Lena" against a wide range of attacks

Attack	PSNR	Detector response of the real watermark	Max response of among 999 fake watermarks	Ratio of ρ_w/ρ_{maxs}	Z-value
No attack	39.40	0.2180	0.0487	4.4763	15.7674
JPEG QF=50	29.35	0.1692	0.0448	3.7768	12.1748
JPEG QF=20	26.77	0.1252	0.0440	2.8455	9.1226
JPEG QF=5	21.29	0.0628	0.0385	1.6312	4.6140
Gaussian noise v=0.05	7.63	0.2364	0.1315	1.7977	6.4098
Gaussian noise v=0.1	5.33	0.2426	0.1604	1.5125	5.2315
Cropped 50%	2.81	0.1442	0.0384	3.7552	12.1334
Cropped 75%	1.09	0.0582	0.0238	2.4453	8.0329
Gaussian Blur (5x5)	34.97	0.1783	0.0423	4.2151	14.1536
Low pass filtering (3x3)	25.86	0.1073	0.0345	3.1101	10.0508
Histogram Equalization	13.09	0.3186	0.0727	4.3824	14.7111
Gamma correction	11.73	0.2115	0.0493	4.2901	15.1408
Contrast adjustment	11.84	0.2555	0.0597	4.2797	14.8322
Sharpening	17.45	0.5343	0.1066	5.0122	19.5525
Rotation by 30º	4.65	0.0140	0.0554	0.2527	0.9896

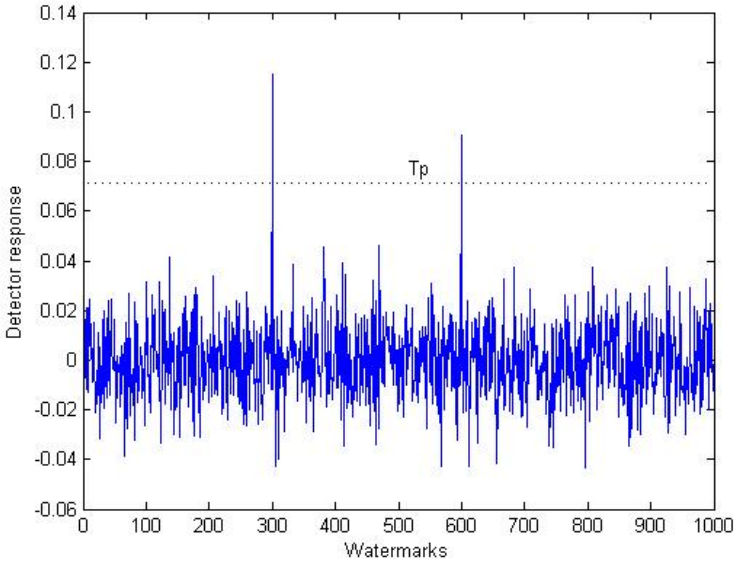


Figure 2.16. The detector responses for collusion attack

to geometric distortions, such as rotation, scaling, translation and shearing. This is because these distortions disturb the correspondence between the blocks of the original image and the blocks of the distorted image. Thus the detection of the watermark requires a synchronization step to locate the embedded watermark in the content. In Chapter 4, we will introduce a scheme for detecting and recovering from geometric attacks based on the radon transform. By utilizing this scheme before applying the watermark detector, we synchronize the embedding location which allows the system provide high robustness against geometric attacks as well.

2.3.3 Comparison of Robustness of Curvelet and Wavelet Domain Algorithms

The wavelet transform has been extensively studied in the last decade. Many wavelet based watermarking algorithm show strong robustness against various image attacks. We developed a wavelet-based version of our curvelet-based scheme to compare the robustness of the same algorithm applied to different domain. Table 2.3 displays the

comparison of robustness of the curvelet-based and the wavelet-based algorithm against severe image attacks. Observation shows watermarks are well detected in both domains without attacks. The ρ_w/ρ_{maxs} and Z-value of wavelet algorithm is slightly better than that of curvelet algorithm when no attack is applied to cover image. However, the watermarks cast into curvelet domain provide high tolerance to severe image quality degradation where the watermarks in wavelet domain completely diminished under the 75% cropping attack and were undetectable in severe JPEG compression and Gaussian noise addition.

We also compare the performance of our algorithm in wavelet and curvelet domain when the cover image is subjected to less severe image attacks, for instance, JPEG compression with relatively high quality factor. In Figure 2.17., the evaluated ρ_w/ρ_{maxs} is plotted against JPEG compression with increasing quality factor of the image (from 5% to 90%). It turns out that the wavelet based algorithm performed better than curvelet based algorithm when the quality factor is greater than 70; the performance of curvelet based algorithm is better when the quality factor is smaller than 70. Figure 2.18 compares the performance of both algorithms in terms of Z-value. The curvelet based algorithm always performs better than

Table 2.3. Comparison of robustness of curvelet and wavelet domain algorithm wi/wo attacks

	Curvelet domain			Wavelet domain		
	PSNR	ρ_w/ρ_{maxs}	Z-value	PSNR	ρ_w/ρ_{maxs}	Z-value
No attack	39.40	4.4763	15.7674	38.28	4.9427	16.5237
JPEG QF=5	21.29	1.6312	4.6140	21.26	1.0666	3.9833
Gaussian noise v=0.1	5.33	1.5125	5.2315	5.31	1.0271	3.961678
Cropped 75%	1.09	2.4453	8.0329	1.09	-0.20022	-0.648328

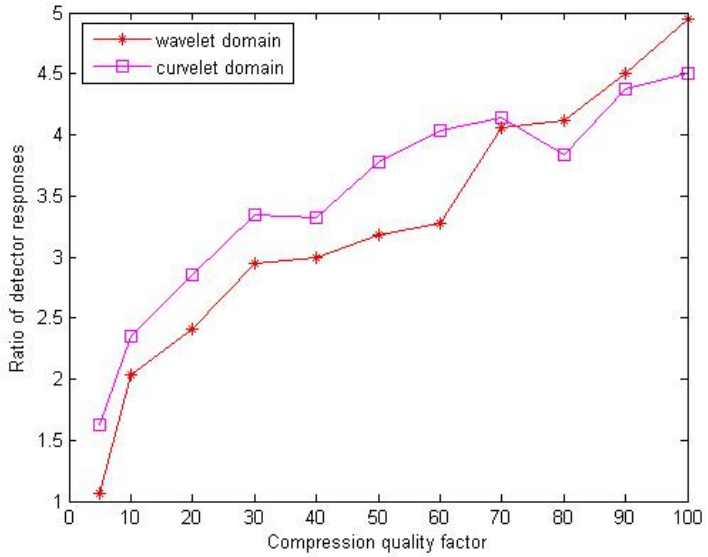


Figure 2.17. Comparison of the ratio ρ_w/ρ_{maxs} in curvelet and wavelet based algorithm against JPEG compression with increasing quality factor.

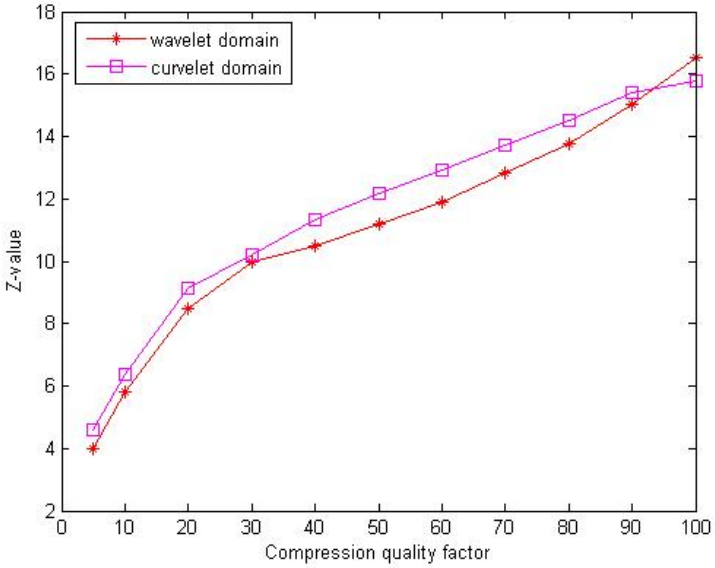


Figure 2.18. Comparison of Z-values in curvelet and wavelet based algorithm against JPEG compression with increasing quality factor..

the wavelet based algorithm. Therefore, we conclude the block-based curvelet domain algorithm shows advantages over the same approach operating in the wavelet domain.

In the literature, a huge number of papers proposed watermarking applications operating in the wavelet transform [15,16,17,28]. We developed several watermarking algorithm in

wavelet domain according to their approach. Table 2.4 shows a comparison of robustness of our curvelet algorithm and a typical second level wavelet domain algorithm [15, 17] against various image attacks. When no attack is applied, the ratio of ρ_w/ρ_{maxs} and Z-value of DWT algorithm are slightly higher than that of our curvelet algorithm. However the measured ratio of ρ_w/ρ_{maxs} and Z-value of DWT algorithm are no longer higher than that of curvelet based algorithm when attacks are applied to the cover image. Observation shows the curvelet algorithm has better performance in JPEG compression, Gaussian noise addition, cropping, low pass filtering, Gaussian blur; the difference becomes more

Table 2.4. Comparison of robustness of our curvelet domain algorithm and a typical wavelet domain algorithm

Attack	Our Curvelet Algorithm			Typical DWT Algorithm		
	PSNR	Ratio of ρ_w/ρ_{maxs}	Z-value	PSNR	Ratio of ρ_w/ρ_{maxs}	Z-value
No attack	39.40	4.4763	15.7674	39.34	4.8822	15.8362
JPEG QF=50	29.35	3.7768	12.1748	29.39	3.3817	11.0594
JPEG QF=20	26.77	2.8455	9.1226	26.75	2.7688	8.6296
JPEG QF=5	21.29	1.6312	4.6140	21.28	1.2794	4.2457
Gaussian noise v=0.05	7.63	1.7977	6.4098	7.60	1.6256	5.2614
Gaussian noise v=0.1	5.33	1.5125	5.2315	5.30	1.1329	3.7659
Cropped 50%	2.81	3.7552	12.1334	2.81	4.3852	12.0520
Cropped 75%	1.09	2.4453	8.0329	1.09	-0.2844	-0.7253
Gaussian Blur (5x5)	34.97	4.2151	14.1536	34.99	3.9306	12.4469
Low pass filtering (3x3)	25.86	3.1101	10.0508	25.86	2.5103	7.3608
Histogram Equalization	13.09	4.3824	14.7111	13.07	4.2276	14.1094
Gamma correction	11.73	4.2901	15.1408	11.73	4.4607	14.5168
Contrast adjustment	11.84	4.2797	14.8322	11.84	4.4505	14.2472
Sharpening	17.45	5.0122	19.5525	17.45	7.2633	22.9741
Rotation by 30º	4.65	0.2527	0.9896	4.65	-0.8020	-2.4110

distinct under severe image attacks. The two algorithms show almost the same robustness against those attacks such as histogram equalization, gamma correction and contrast adjustment. Both algorithms collapse if rotation is applied to the cover image. The experimental results confirmed the robustness of our curvelet domain based algorithm.

2.4 Conclusion

Curve edges are essentially the frame part of an image which survives severer image degradation and most intentional attacks. In this chapter, a robust watermarking scheme in curvelet domain is proposed. We embedded the watermark into the significant curvelet coefficients in those blocks with strong edges according to edge map. The fidelity of the protected image is well maintained and the watermark embedded into curvelet coefficients provides high tolerance to severe image quality degradation and showing advantages over watermarking algorithm in the wavelet domain.

Chapter 3

Perceptual Data Hiding in the Curvelet Domain

One important issue in watermarking consideration is how to hide the highest possible amount of information without affecting the visual quality of the host data. We will describe a novel perceptual data hiding method in still images based on Barten's (1990) contrast sensitivity model [56]. The cover image is transformed into curvelet domain and a Just Noticeable Distortion (JND) is computed for each curvelet coefficient, taking into account frequency sensitivity and masking effects. The watermark consists of a pseudo random sequence which is adaptively added to the significant curvelet coefficients, with embedding strength adjusted by the JND. Experiments show the transparency of watermarking system is improved while the robustness against various image attacks is maintained as well.

3.1 Human Visual System

Many approaches have been proposed so far to model the characteristics of the Human Visual System (HVS) [46,47,48,49,50]. A perceptual model generally attempts to account for measuring perceptual variations. It is well known that the response of the HVS varies with the spatial frequency and brightness of its input. Generally, noise is less visible in highly textured regions, edges, dark and bright areas. Most models are concerned with two main concepts: frequency sensitivity and masking.

The first concept is concerned with the sensitivity of the human eye to a sine grating stimulus; as the sensitivity of the eye depends strongly on display background luminance and spatial frequency of the stimulus [88]. Spatial frequencies are perceived as patterns or textures. The spatial frequency response is usually described by the sensitivity to luminance contrast (i.e., changes in luminance) as a function of spatial frequency: this is called the contrast sensitivity function (CSF). Human eyes are most sensitive to luminance differences at mid-range frequencies, our sensitivity decreases at lower and higher frequencies. Two-dimensional spatial frequency patterns can be represented by their magnitude and orientation. It has been shown that the sensitivity of the eye is not only dependent on frequencies of different patterns but on their orientations [51,52,53]. In particular, the eye is most sensitive to vertical and horizontal lines and edges in an image and is least sensitive to lines and edges with a 45-degree orientation.

The second concept refers to any destructive interaction and interference among stimuli that are closely coupled, or the visibility reduction of one image component due to the presence of other components. Context affects perception: a texture that is easy to see in isolation might be difficult to see when added to a highly textured image. That is, the presence of one signal can hide or mask the presence of another signal. Masking is a measure of an observer's response to one stimulus when a second "masking" stimulus is also present. In vision, two principal cases are frequency masking, in which the presence of one frequency masks the perception of another, and brightness masking, in which local brightness masks contrast changes.

3.2 Barten's Model for Contrast Sensitivity

The dynamic range of luminance in a region of a picture is represented by contrast. In particular, by letting $L(x, y)$ be the luminance of a pixel at position (x, y) and L_o the local mean background luminance, we can define local contrast as [88]

$$C = \frac{L(x, y) - L_0}{L_0} \tag{3.1}$$

If an image is obtained by adding a sinusoidal stimulus to a uniform background, the spatial luminance of the image is given by [88]:

$$L(x, y) = L_0 + \Delta L \cos(2\pi f (x \cos\theta + y \sin\theta)) \tag{3.2}$$

where f, θ and ΔL are, respectively, the frequency, orientation and amplitude of the superimposed stimulus. The frequency f, measured in *cycles/ degree,* is implemented as a function of the frequency υ measured in *cycles/m* and the viewing distance D between the observer and the monitor measured in meters [88]:

$$f = (\pi D / 180) v \tag{3.3}$$

ΔL is increased until the observer perceives it. Such a threshold value of ΔL will be referred to as ΔL_{jn}, the minimum contrast necessary to just detect a sine wave of a given frequency f and orientation θ superimposed to a background L_0. Thus, the concept of just noticeable contrast JNC is expressed as [54]:

$$JNC = \frac{\Delta L_{jn}}{L_0} \tag{3.4}$$

The inverse of *JNC* is commonly referred to as the *contrast sensitivity function* (CSF) [55] that measures the capability of the human eye to notice a sinusoidal stimulus on a uniform background:

$$S_c = \frac{1}{JNC} = \frac{L_0}{\Delta L_{jn}} \tag{3.5}$$

It has been found that, for different values of f and θ, the major factors *JNC* (or equivalently S_c) depends upon are: (1) the frequency of the stimulus f, (2) the orientation of the stimulus θ, (3) background luminance L_o, and (4) the viewing angle w.

Many analytical expressions of *CSF* can be found in the literature. We consider the one obtained by Barten [56] by fitting data of psychophysical experiments. According to Barten's model, the factors influencing human vision are taken into account by the following expression:

$$S_c(f,\theta,w,L_0) = a(f,w,L_0) f \exp\left(-\Gamma(\theta) b(L_0) f\right) \cdot \sqrt{1 + c \cdot \exp\left(b(L_0) f\right)} \tag{3.6}$$

with:

$$a(f,w,L_0) = \frac{540\left(1 + 0.7/L_0\right)^{-0.2}}{1 + \frac{12}{w \cdot \left(1 + f/3\right)^2}},$$
$$b(L_0) = 0.3\left(1 + 100/L_0\right)^{0.15},$$
$$c = 0.06,$$
$$\Gamma(\theta) = 1.08 - 0.08\cos(4\theta)$$

where the frequency of the stimulus f is measured in *cycles/degree;* the orientation of the stimulus θ in *degrees;* the observer viewing angle w in *degrees,* and the mean local background luminance L_0 in *candelas/m*^{2}. In particular, the term $\Gamma(\theta)$ takes into account that the eye sensitivity is not isotropic [56].

Plots of S_c against luminance, frequency and orientation of stimulus illustrate the behavior of Barten's model. In particular, the plots of *CSF* with respect to frequency are reported for several values of background luminance with a constant orientation θ and viewing angle w (e.g. $\theta = 0$ and $w = 180/\pi\sqrt{12}$). All the curves have the same trend for all values of background luminance: the maximum sensitivity is obtained in the middle range of frequencies, the sensitivity decreases in the low and high part of the frequency range. Plot of the just noticeable stimulus ΔL_{jn} against luminance L for a frequency (e.g.15 *cycles/degree*) indicates the distortion is less visible in dark and bright regions. Observations also show horizontal (or vertical) stimuli are more visible than those oriented at 45º. The behavior of Barten's model is consistent with the results achieved by psychophysical experiments.

3.3 Perceptual Model in Curvelet Domain

3.3.1 Computing JND Profile in Curvelet Domain

In this section, we will construct a perceptual model for computing the Just Noticeable Distortion (JND) for each curvelet coefficient. Figure 3.1 shows the procedures for the computing JND thresholds. The proposed JND profile of a still image depends on the spatial frequency sensitivity, the sensitivity to local gray contrast and texture masking.

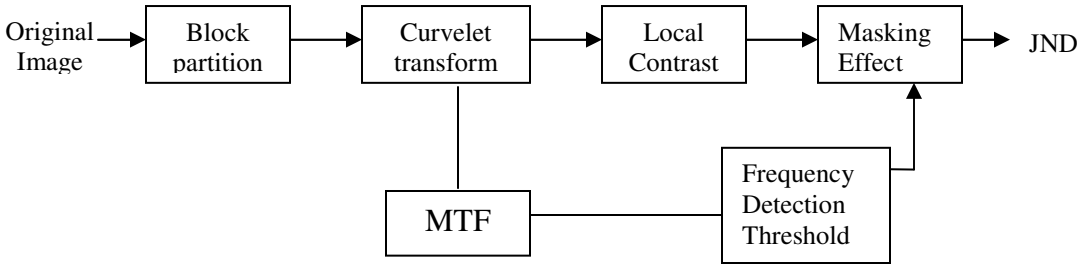


Figure 3.1. Proposed JND system in curvelet domain

The spatial frequency sensitivity is the modulation transfer function (MTF), which provides relative tolerance of the HVS to noise at different spatial frequencies and different orientations.

3.3.1.1 Curvelet Subbands Division and Coefficients Grouping

Recall from Chapter 2 that the idea of the curvelet transform is to compute the inner product between the signal and the curvelet function to realize the sparse representation of the signal expressed as:

$$c(j,\ell,k) := \left\langle f, \varphi_{j,\ell,k} \right\rangle \tag{3.7}$$

where j, ℓ, k are the scale, direction and translation parameters respectively.

Curvelets at scale 2^{-j}, orientation θ_ℓ and position $x_k^{(j,\ell)} = R_{\theta_\ell}^{-1}\left(k_1 \cdot 2^{-j}, k_2 \cdot 2^{-j/2}\right)$ can be expressed as

$$\varphi_{j,l,k}(x) = \varphi\left(R_{\theta_\ell}\left(x - x_k^{(j,l)}\right)\right) \tag{3.8}$$

where $\theta_\ell = 2\pi \cdot 2^{-\lfloor j/2 \rfloor} \cdot \ell$, l=0,1,…, $0 \le \theta_\ell < 2\pi$, R_θ is the rotation by θ radians.

Since the FDCT-USFFT[44] uses some concentric squares to separate scales, there is one such grid per scale and angle. With the increase of the scales, rectangular grids expand along directions and frequency plane. We let $I^l_{s,t}$ be a grid of curvelet coefficients, where *s* denotes the scale parameter while *t* denotes the angular parameter and *l* denotes the block index. Coefficients with the same spectral indices can be grouped together to form a grid of dimension *mxn*. Then the curvelet coefficients grouped together is denoted by $I^l_{\omega,\theta} = I^l_{s,t}(k_1, k_2)$, where roughly

$$\omega = s + \frac{k_1}{m} 2^{-j} \quad \theta = t + \frac{k_2}{n} 2^{-j/2} \tag{3.9}$$

and k_1=0,1,…m-1, k_2=0,1,…,n-1. For each block B_l in cover image, FDCT-USFFT is applied then the curvelet subband coefficients can be denoted by

$$\left\{I^l(\omega,\theta)\right\} = \left\{I^l_{s,t}(k_1,k_2)\right\} = FDCT - USFFT\left\{B_l(x,y)\right\} \tag{3.10}$$

3.3.1.2 JND Profile in Curvelet Domain

The JND profile of each subband curvelet coefficient denoted by $I^l_{\omega,\theta}$ is a function of local image properties, such as local contrast, contrast sensitivity, the frequency detection threshold of subband image and local texture properties. The JND of $I^l_{\omega,\theta}$ is established in following steps.

Step 1: Firstly, since human visual sensitivity to luminance patterns is reduced as the local luminance is increased; the just noticeable distortion (JND) is related to the local contrast. In our model, we define the local contrast in curvelet domain by

$$C_{s,t}^{l}(k_1,k_2)=\frac{|I(\omega,\theta)|}{\max(I_{s,t}^{l}(k_1,k_2))^{\lambda}}, \tag{3.11}$$

Step 2: Secondly, we define a modulation transfer function based on Barten's contrast sensitivity model. Barten's contrast sensitivity function (CSF) which offers an indication of the capability of human eyes to notice a sinusoidal stimulus on a uniform background is generally a function of the frequency of the stimulus *f* in *cycles/degree,* the orientation of the stimulus θ in *degrees,* the observer viewing angle *w* in *degrees* and the mean local background luminance L_0 in *candelas/m*^{2}. If we assume the viewing angle *w* and the mean local background luminance L_0 to be constants then S_c (see Equation 3.6) becomes a function of two parameters: radial spatial frequency ω, orientation in degree θ. We define the modulation transfer function as

$$MTF(\omega,\theta)=\min[S_c(\omega,\theta),S_c(\omega)] \tag{3.12}$$

Here

$$S_c(\omega,\theta)=a(\omega)\cdot\omega\cdot\exp(-\Gamma(\theta)\cdot 0.33\cdot\omega)\cdot\sqrt{1+0.06\cdot\exp(0.33\omega)} \tag{3.13}$$

$$S_c(\omega)=1.08\cdot a(\omega)\cdot\omega\cdot\exp(0.33\cdot\omega)\cdot\sqrt{1+0.06\cdot\exp(0.33\omega)} \tag{3.14}$$

where

$$a(w) = \frac{0.48}{1 + \frac{12}{16.54 \cdot (1 + \omega / 3)^2}},$$
$$\Gamma(\theta) = 1.08 - 0.08\cos(4\theta)$$

Step 3: Then frequency detection threshold of subband image $I_{s,t}^{l}$ is determined by

$$T_{s,t}^{l} = \frac{\iint_{I_{s,t}^{l}} \omega \, d\omega \, d\theta}{\iint_{I_{s,t}^{l}} MTF(\omega, \theta) \omega \, d\omega \, d\theta} \tag{3.15}$$

Since the above formula is independent of the image content, the frequency detection threshold can be calculated in advance to reduce the computation complexity.

Step 4: Finally, based on the masking function model developed by Scott Daly [90], the JND profile of $I_{s,t}^{l}(k_1, k_2)$ is given by

$$JND\left(I_{s,t}(k_1, k_2)\right) = T_{s,t} \cdot \left(1 + \left(0.0153\left(392.498 \cdot \left|\frac{C_{s,t}(k_1, k_2)}{T_{s,t}}\right|\right)^{\gamma}\right)^4\right)^{1/4} \tag{3.16}$$

where γ is a function of texture properties of $I_{s,t}^{l}(k_1, k_2)$ in the block B_l given by

$$\gamma\left(I_{s,t}^{l}(k_1, k_2)\right) = \begin{cases} 1, & if \quad B_l \in S_1 \\ 0.7 & if \quad B_l \in S_2 \end{cases} \tag{3.17}$$

S_1 and S_2 are referring to the blocks with weak edges and strong edges respectively. We use Canny edge function to detect edges in cover image. The blocks with strong edge are

defined in Equation 2.13 with edge strength greater than a threshold. Those blocks whose edge strength is less than the threshold are considered with weak edges. Thus, $JND\left(I_{s,t}\left(k_1,k_2\right)\right)$ is expected to have lower values in flat areas, whereas textured areas should have higher values.

3.3.2 Watermarking Embedding with JND Adjustment

According to our proposed watermarking algorithm (see Figure 2.3), the blocks with strongest edges are always selected for embedding watermark. If a hacker is aware of the underlying algorithm, he will deliberately modify the area of strong textures so as to destroy the inserted watermark. We may want to include as many as possible blocks for watermark embedding so the embedded watermark is more robust against malicious attacks. The proposed JND model presented in this chapter is designed for the purpose of hiding more information into host signals yet reducing the effect on the visual quality of the host image. The JND model determines the embedding strength of watermark element so that it can be inserted with less visual distortion. The enhanced framework of the proposed watermarking system with JND adjustment is depicted in Figure 3.2.

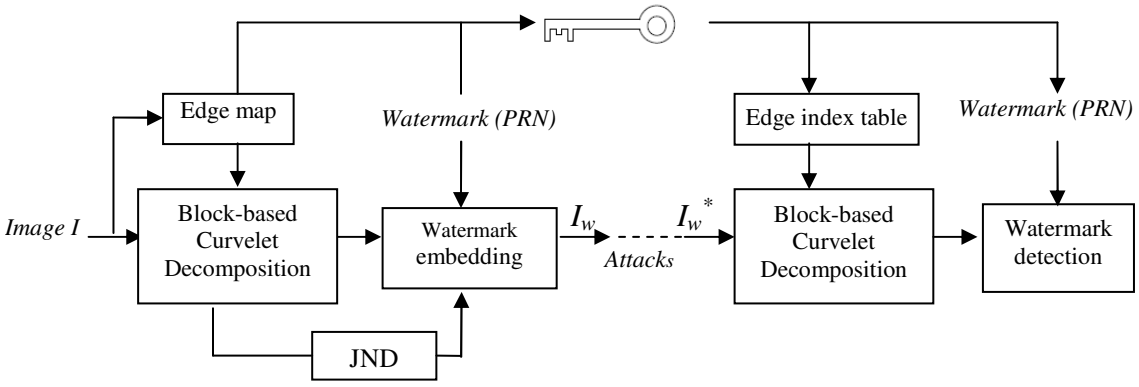


Figure 3.2. Framework of proposed watermarking system with JND

Given the image partitioned into blocks B_l, the FDCT-USFFT is applied and the curvelet coefficients $I^l_{s,t}(k_1,k_2)$ are obtained and the $JND(I^l_{s,t}(k_1,k_2))$ is calculated. The watermark sequence $W=\{w_1,w_2,\ldots w_m\}$ generated from a normal distribution of zero mean and unit variance is superimposed into curvelet coefficients according to:

$$c^{l\ W}_{s,t}(k) = g^l_{s,t}(k)\cdot c^l_{s,t}(k) + a\cdot f(JND(I^l_{s,t}(k))\cdot |\, c^l_{s,t}(k)\,|\cdot w_i \tag{3.18}$$

$$g^l_{s,t}(k) = \begin{cases} 1 & if\ JND(I^l_{s,t}(k))\cdot\left|c^l_{s,t}(k)\right| > \left|a\cdot c^l_{s,t}.w_i + c^l_{s,t}\right| \\ JND(I^l_{s,t}(k)), & otherwise \end{cases}$$

$$f(JND(I^l_{s,t}(k)) = \begin{cases} 1, & if \quad JND(I^l_{s,t}(k))\cdot\left|c^l_{s,t}(k)\right| > \left|a\cdot c^l_{s,t}.w_i + c^l_{s,t}\right| \\ 0 & otherwise \end{cases}$$

Hence the embedding strength of watermark is limited by $JND(I^l_{s,t}(k_1,k_2))$. The effect of JND modeling is to adjust the watermark embedding strength when it is added to perceptually sensitive regions. When the noise introduced by the watermark is larger than the JND threshold, then the embedding strength is tuned so that the hidden signal is just below the perceptual threshold. If the introduced noise is fairly small and all distortions are below JND thresholds, then Equation 3.18 is reduced to the straight embedding formula defined in Equation 2.15.

3.4 Experimental Results

We have tested our proposed watermarking scheme on several images (Lena, Barbara, Boat, Airplane, Goldhill, Peppers, Baboon) of dimension of 512 ×512. They have been

partitioned into blocks of $n \times n$ pixels with n=64, thus obtaining 64 blocks. The selected blocks are decomposed into curvelet domain via FDCT-USFFT and we obtained curvelet coefficients along with the JND profile for each single coefficient. The watermark embedding strength is adjusted according to Equation 3.18 so that the watermark noise does not exceed the JND.

Figure 3.3 displays the original Lena together with the watermarked Lena without/with JND adjustment. We choose those blocks with edge strength greater than 400 for this particular example. We evaluated the watermarked image using three distortion measures: mean squared error (MSE), peak-signal-to noise ratio (PSNR) and Universal Image Quality Index (UIQI). Observation shows the watermarked "Lena" under JND adjustment is of the quality PSNR =42.42, UIQI =0.999797 and MSE=0.9308. Compared

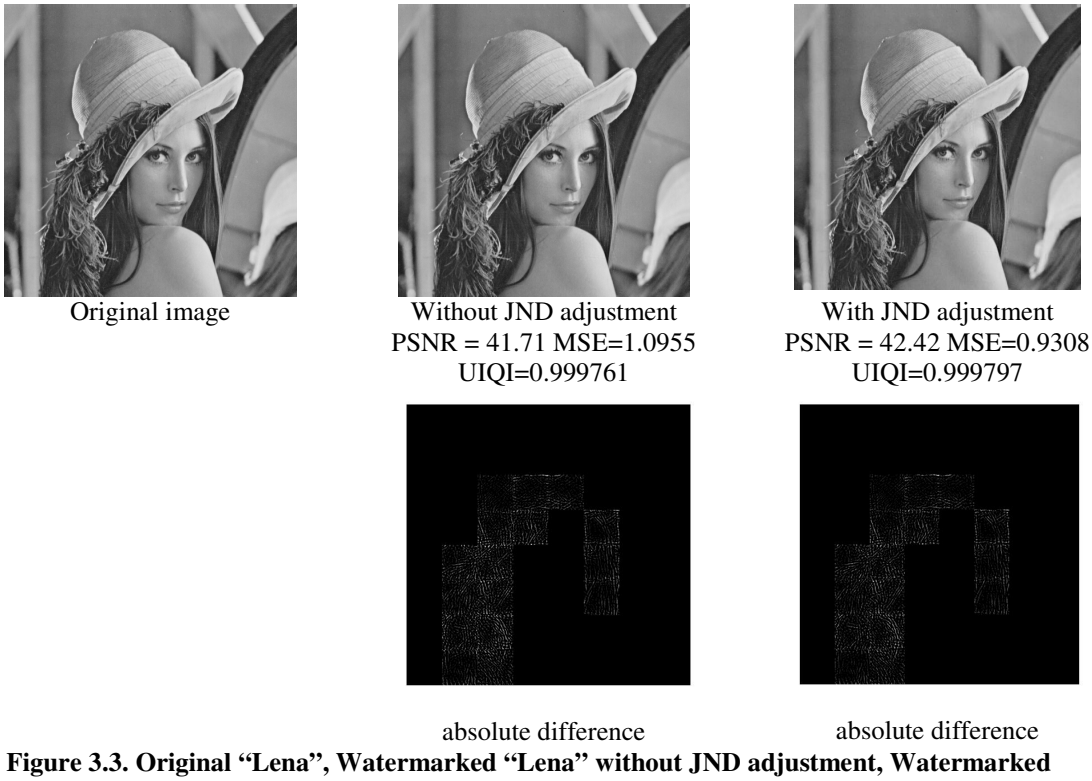


Figure 3.3. Original "Lena", Watermarked "Lena" without JND adjustment, Watermarked "Lena" with JND adjustment. The distance between the original and the watermarked image is measured using PSNR, MSE and UIQI.

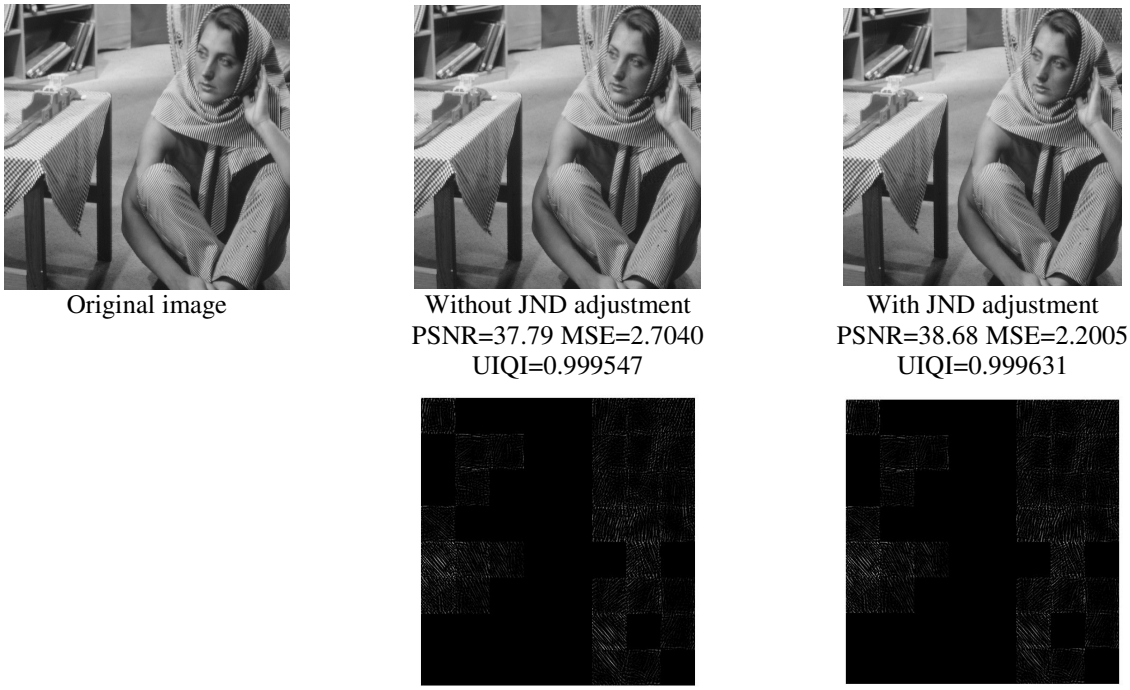


Figure 3.4. Original "Barbara", watermarked "Barbara" without JND adjustment, watermarked "Barbara" with JND adjustment. The distance between the original and the watermarked image is measured using PSNR, MSE and UIQI.

with the watermarked "Lena" without JND adjustment of the quality PSNR =41.71, UIQI =0.999761 and MSE=1.0955, the one with JND modeling has smaller distance to the original "Lena", and the distortion in perceivable regions is reduced. Therefore better image fidelity is obtained through JND adjustment. Figure 3.4 gives the evaluation for another image, Barbara, showing the same results.

We have run the test with increasing embedding edge strength. Figure 3.5, 3.6 and 3.7 illustrate the measuring results for the standard image Lena using PSNR, MSE and UIQI respectively. In Figure 3.5, 3.6 and 3.7, the line (asterisk marker) represents the distortion of the watermarked image controlled by JND model. It can be seen that JND always improves the quality of watermarked images.

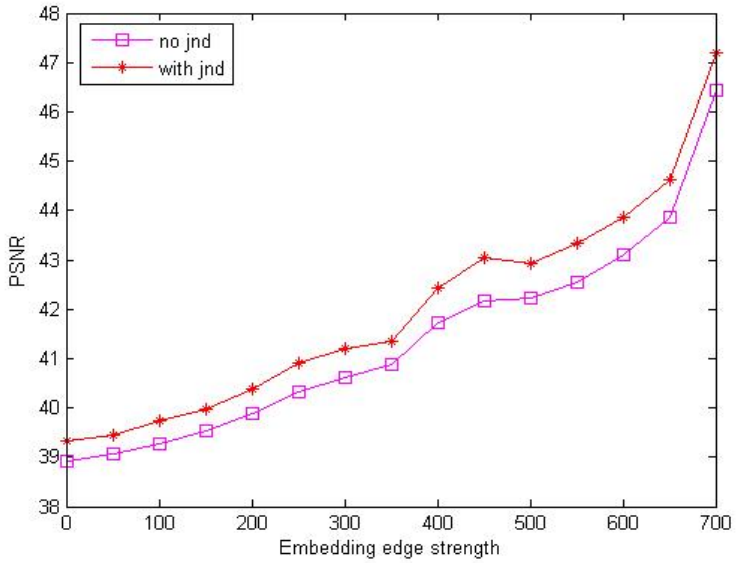


Figure 3.5. Effect of JND modeling on PSNR of watermarked Lena

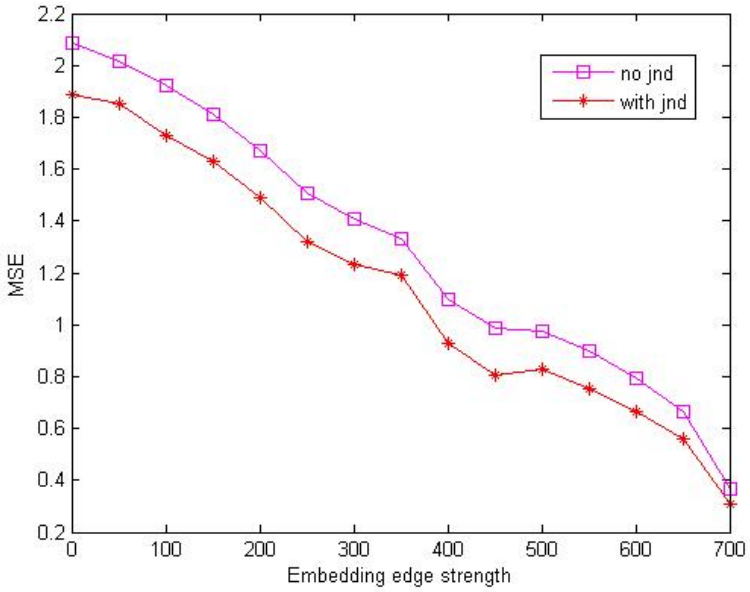


Figure 3.6. Effect of JND modeling on MSE of watermarked Lena

We measured the response of the detector correlation on watermarked "Lena" for both cases along with increasing embedding edge strength. We report the experimental results in Table 3.1. ρ_w denotes the detector response of the embedded watermark; ρ_{maxs} denotes the max response among 999 fake randomly generated watermarks. Z-value is a

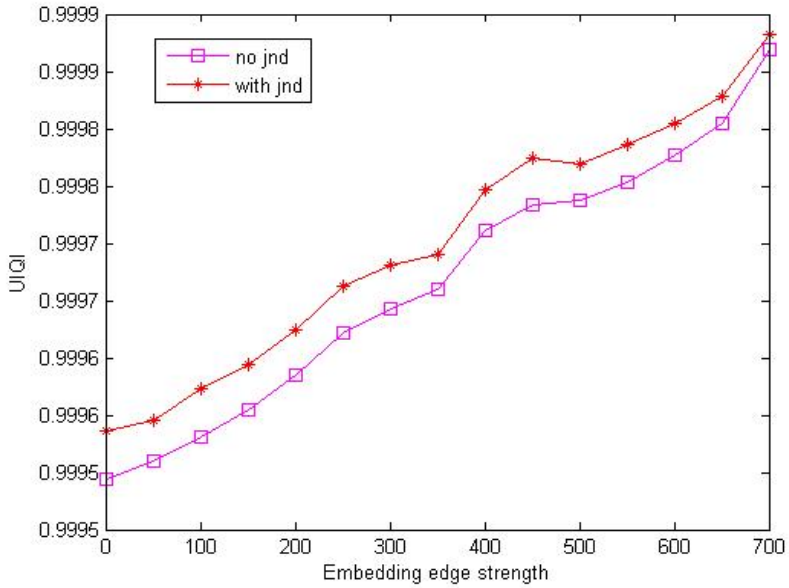


Figure 3.7. Effect of JND modeling on UIQI of watermarked Lena.

statistical measure defined in Equation 2.18. If Z-value is higher than 4, that means the detector response of embedded watermark is higher than the responses of $\{- 4\sigma_{s,} +4\sigma_s\}$ 99.99% fake watermarks. It can be seen the watermarks are well detected in both cases. We have applied our JND model to a set of test images such as Barbara, Boat, Goldhill, Airplane, Peppers and Baboon (displayed in Figure 3.8). The experimental results are given in Table 3.2. In particular, Baboon is a test image with high texture, so we need to

Table 3.1. Performance of watermarked "Lena" wi/wo JND against increasing edge strength

Edge Strength	Lena without JND adjustment			Lena with JND adjustment		
	PSNR	ρ_w / ρ_{maxs}	Z-value	PSNR	ρ_w / ρ_{maxs}	Z-value
0	38.91	3.73	13.42	39.33	3.41	12.35
100	39.27	4.40	15.51	39.73	4.13	14.39
200	39.88	4.33	13.00	40.37	3.95	11.97
300	40.63	3.44	10.69	41.21	3.10	9.61
400	41.71	2.69	9.55	42.42	2.41	8.54
500	42.22	2.65	9.04	42.94	2.39	8.01
600	43.11	1.87	6.99	43.87	1.62	6.10
700	46.44	1.55	4.87	47.20	1.36	4.27

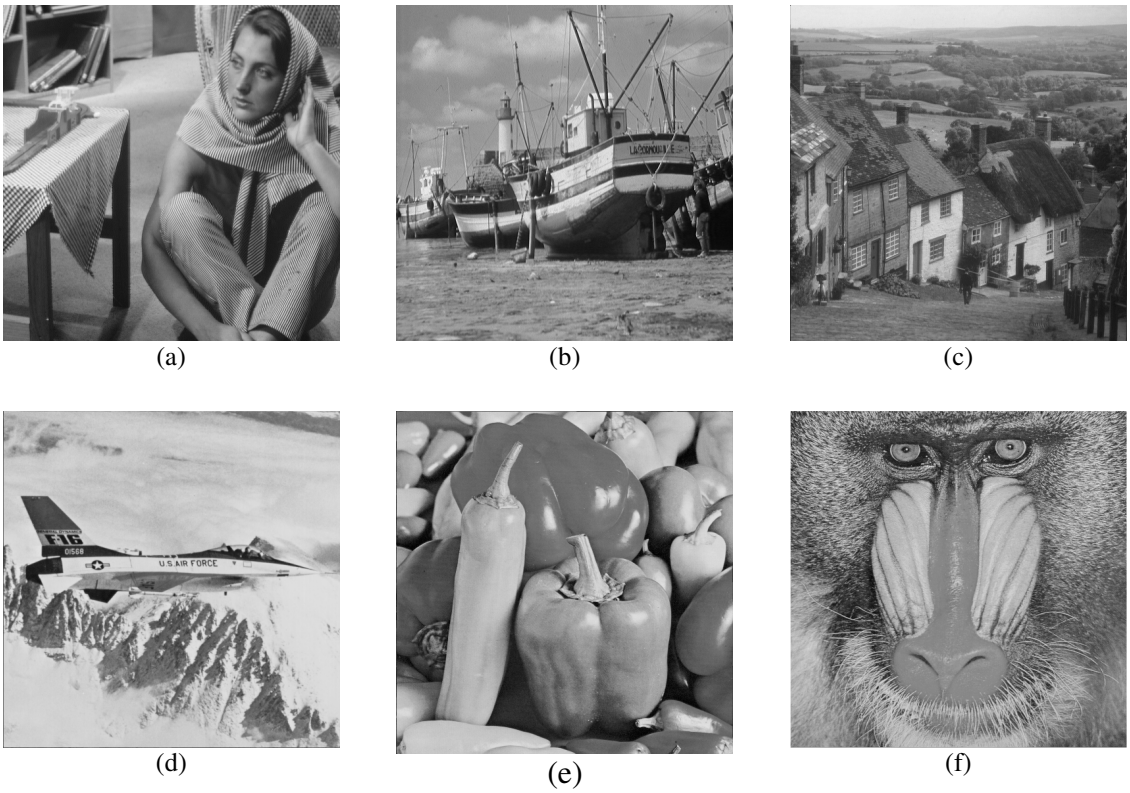


Figure 3.8. Test images (a) Barbara (b) Boat (c) Goldhill (d) Airplane (e) Peppers (f) Baboon

choose a relatively high edge strength threshold in order to limit the number of embedding blocks. Therefore, only a small number of blocks with very strong edges are selected and distorted to accommodate watermark energy. Although the value of PSNR and UIQI is apparently low for "Baboon", the distortion is barely perceivable because eyes are not sensitive to the noise in high texture area. On the other hand, the "Peppers"

Table 3.2. Performance of watermarked images wi/wo JND adjustment

Image ID	Edge strength	without JND adjustment				with JND adjustment			
		PSNR	UIQI	ρ_w / ρ_{maxs}	Z-value	PSNR	UIQI	ρ_w / ρ_{maxs}	Z-value
Barbara	450	39.88	0.999720	3.35	10.83	40.67	0.999767	3.05	9.82
Boat	550	39.06	0.999538	4.72	14.77	39.88	0.999616	4.35	13.46
Goldhill	500	40.81	0.999722	4.25	13.90	41.52	0.999763	3.83	12.44
Airplane	350	38.46	0.999463	4.17	12.35	39.09	0.999535	3.75	11.22
Pepper	100	39.08	0.999627	4.61	14.53	39.28	0.999644	4.40	13.84
Baboon	800	36.61	0.998984	3.96	14.20	37.40	0.999152	3.61	12.91

Original “Lena” | Watermarked “Lena” PSNR=39.73 | Absolute difference

Figure 3.9. Original “Lena”, watermarked “Lena” and absolute difference

has relatively weak texture. We choose low edge strength in order to include more curvelet coefficients for watermark embedding. In general, the selected threshold for edge strength is depending on the requirement of application. The JND is effective for unperceivable data hiding when the amount of hidden data is large.

We also investigated the robustness of the watermarking system controlled by JND model. Using standard image Lena, we chose 100 as the embedding edge strength threshold. Original “Lena”, watermarked “Lena” and absolute difference are displayed in Figure 3.9. Figure 3.10 and 3.11 show the detector response to the embedded watermark and the maximum among fake watermarks when the watermarked image is subjected to JPEG compression with increasing quality factor. It can be observed that the watermark embedded with JND adjustment survives even severe JPEG compression with quality factor 5 as well. Figure 3.12, 3.13 and 3.14 show the corrupted watermarked “Lena” with JND modeling under other severe image attacks, along with the corresponding detector response against 999 fake watermarks. It can be seen the embedded watermark yields a response higher than the threshold while fake ones are all below it.

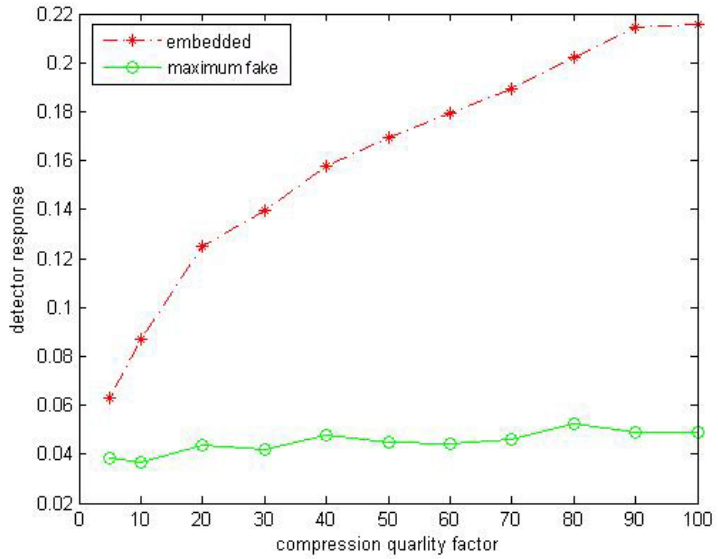


Figure 3.10. The detector response to the embedded watermark and the maximum among 999 fake watermarks against JPEG compression. The embedding strength is not limited by JND.

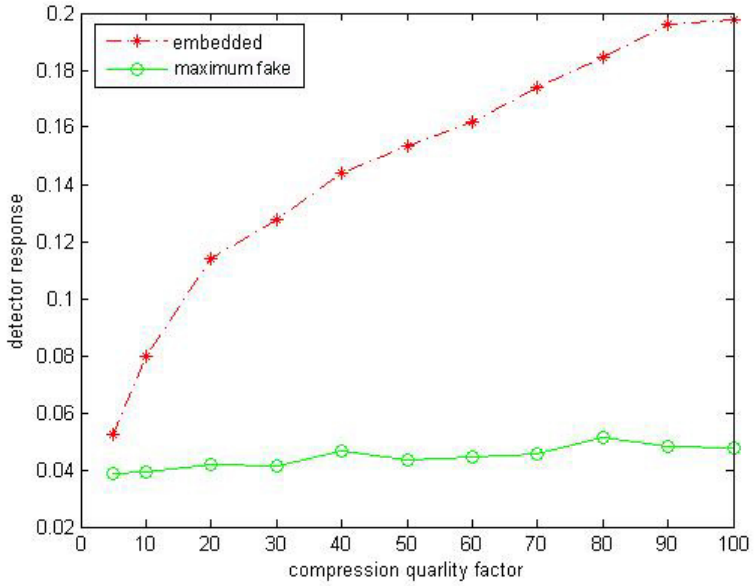


Figure 3.11. The detector response to the embedded watermark and the maximum among 999 fake watermarks against JPEG compression. The embedding strength is limited by JND.

Finally, Table 3.3 gives the comparison of the robustness of our curvelet algorithm with and without JND adjustment against a wide range of attacks. Experimental results

PSNR= 5.33, UIQI=0.447103

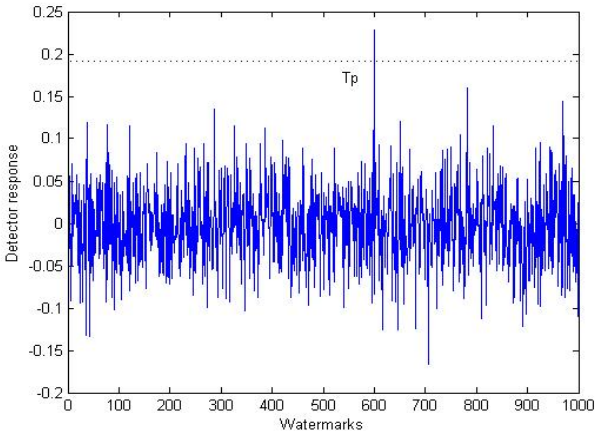


Figure 3.12. Watermarked "Lena" with Gaussian noise (m=0 and var=0.1) and corresponding detector response of the embedded watermark against 999 fake watermarks

PSNR= 1.09, UIQI= 0.140117

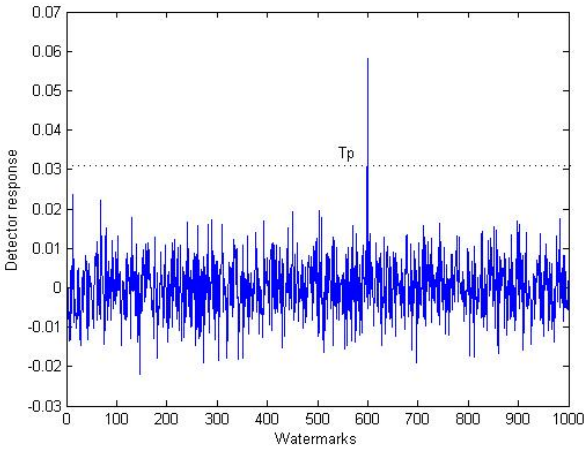


Figure 3.13. Cropped watermarked "Lena" by 75% and corresponding detector response of the embedded watermark against 999 fake watermarks

PSNR= 21.29, UIQI=0.973641

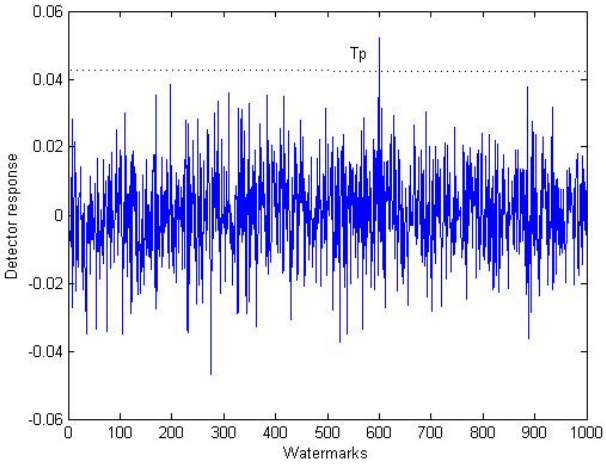


Figure 3.14. Watermarked "Lena" under JPEG compression with quality factor 5 and corresponding detector response of the embedded watermark against 999 fake watermarks

show JND modeling does not affect the robustness of the embedded watermark. Both robustness and imperceptibility are likely to be got by the employment of the model that characterizes the phenomena regulating human vision.

Table 3.3. Comparison of robustness of our curvelet algorithm wi/wo JND adjustment

Attack	without JND adjustment				With JND adjustment			
	PSNR	UIQI	ρ_w/ρ_{maxs}	Z-value	PSNR	UIQI	ρ_w/ρ_{maxs}	Z-value
No attack	39.27	0.999581	4.40	15.51	39.73	0.999623	4.13	14.39
JPEG QF=50	29.35	0.995880	3.78	12.17	29.40	0.995924	3.54	11.17
JPEG QF=20	26.77	0.992511	2.84	9.12	26.78	0.992537	2.72	8.42
JPEG QF=5	21.29	0.973615	1.63	4.61	21.29	0.973641	1.35	3.88
Gaussian noise v=0.05	7.63	0.603491	1.79	6.41	7.60	0.603390	1.68	5.99
Gaussian noise v=0.1	5.33	0.447215	1.51	5.23	5.30	0.447103	1.43	4.92
Cropped 50%	2.81	0.288188	3.76	12.13	2.81	0.288101	3.52	11.23
Cropped 75%	1.09	0.140117	2.45	8.03	1.09	0.140117	2.45	8.07
Gaussian Blur(5x5)	34.97	0.998861	4.22	14.15	34.92	0.998848	3.95	13.11
Low pass filtering (3x3)	25.85	0.990641	3.11	10.05	25.83	0.990576	2.88	9.36
Histogram Equalization	13.09	0.899886	4.38	14.71	13.08	0.899856	4.09	13.71
Gamma correction	11.73	0.948968	4.28	15.14	11.73	0.948973	4.04	14.09
Contrast adjustment	11.84	0.954234	4.28	14.83	11.84	0.954363	4.01	13.75
Sharpening	17.45	0.943965	5.01	19.55	17.53	0.944855	4.71	18.15
Rotation by 30º	4.65	0.154410	0.25	0.99	4.65	0.154433	0.25	0.98

3.5 Conclusion

In this chapter, we propose a novel perceptual data hiding method in still images based on Barten's contrast sensitivity model [56]. The cover image is partitioned into blocks and curvelet decomposition is applied to those blocks whose edge strength is higher than a threshold. The proposed Just Noticeable Distortion (JND) model, which takes into account frequency sensitivity and masking effects, is computed for each curvelet coefficient. The watermark is adaptively added to the significant curvelet coefficients under the control of JND model. Experiments show the transparency of watermarking system is improved while the robustness against various image attacks is also maintained. In particular, we may want to include as many as possible blocks for watermark embedding thus the system payload is maximized and the embedded watermark is more robust against malicious attacks such as centered cropping. The JND model imposing constraints over the embedding strength of watermark enables highest possible amount of information hiding without compromising the quality of the data to be protected.

Chapter 4

Detecting and Recovering From Geometric Attacks

An important problem constraining the practical exploitation of robust watermarking technologies is the low robustness of the existing algorithms against geometrical distortions such as rotation, scaling, translation and shearing. All these attacks can be uniquely described by general affine transforms. In this chapter, we propose a robust estimation method based on edge detection and the radon transform. A heuristic search algorithm is developed in searching for the right grid of edge map for estimating affine matrix coefficients. The method is efficient even when severe degradations have occurred, including JPEG compression with a quality factor of 10%. Results with the Stirmark benchmark confirm the high robustness of the proposed method.

4.1 Introduction

Many proposed image watermarking techniques are sensitive to geometric distortions, such as rotation, scaling, translation, cropping, shearing and change of aspect ratio. It is well known that a small amount of rotation and/or scaling can dramatically disable the receiver from detecting the watermark. For instance, it is evident that rotation by a certain angle will substantially lower the performances of watermarking applications for many block-based embedding algorithms. This is because the rotation breaks the correspondence between the blocks of the original image and the blocks of the rotated

image. Thus the detection of the watermark requires a synchronization step to locate the embedded watermark in the content.

4.1.1 General Affine Transforms

All geometric attacks can be uniquely described by general affine transforms. An affine transformation is an important class of linear two-dimensional geometric transformations, which maps variables (e.g., pixel intensity values located at position (x_1, y_1) in an input image) into new variables (e.g., x_2, y_2) in an output image) by applying a linear combination of rotation, scaling, shearing and all other linear geometric transform. Affine transformation can be written as follows:

$$\begin{vmatrix} x_2 \\ y_2 \end{vmatrix} = A \times \begin{vmatrix} x_1 \\ y_1 \end{vmatrix} + B \ , \ A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{ and } \quad B = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{4.1}$$

The matrix A, known as the linear part, contains four coefficients a, b, c and d. The vector B, known as the translation part, has two coefficients t_x, t_y. We describe the types of affine transformations in Table 4.1. If we can identify the type of affine transform, and further find out the values of coefficients in matrix A and/or B, then we are able to detect the geometric attack and restore the cover image to its original state using the inversed version of Equation (4.1) before watermark detector is applied. Therefore the problem of detecting and recovering from geometric attack is converted to solve the coefficients in affine transform matrixes. Not consider the translation, the inverse transform solving for the coefficients in A, we need at least two set of corresponding pixel positions ((x_1, y_1), (x_2, y_2)) in the cover image before and after an image attack given by:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \left[\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} x_1' & y_1' \\ x_2' & y_2' \end{bmatrix}^{-1} \right]^T \tag{4.2}$$

where T is the transpose of the matrix, (x_i, y_i) and $(x_i' y_i')$ are the positions before and after an attack respectively.

Table 4.1. Types of affine transformations in common geometric distortions

Type	Description	Affine matrix	Coordinate equations
Identity	No change applied	$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$x_2 = x_1$ $y_2 = y_1$
Rotation	Rotate an image thorough a specified angle θ	$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$	$x_2 = x_1 \cos\theta - y_1 \sin\theta$ $y_2 = x_1 \sin\theta + y_1 \cos\theta$
Scaling	Zoomed or shrink the size of an image to a specified scale	$A = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$	$x_2 = s_x x_1$ $y_2 = s_y y_1$
Shear (horizontal)	Slides one edge of an image along the horizontal X axis,	$A = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix}$	$x_2 = x_1 + \alpha y_1$ $y_2 = y_1$
Shear(vertical)	Slide one edge of an image along the vertical Y axis.	$A = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix}$	$x_2 = x_1$ $y_2 = \alpha x_1 + y_1$
Translation	Shift the position of the element in an image into a new position in an output image	$B = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$	$x_2 = x_1 + t_x$ $y_2 = y_1 + t_y$

4.1.2 Existing Watermarking Resynchronization Solutions

Many efforts have been made to estimate and compensate for geometric distortion in order to synchronize the location of embedded watermark.

One solution is to insert a watermark in a way that intrinsically resists this sort of manipulation and thereby avoids the need for a synchronization pattern [32,35,57]. Shelby Pereira and Thierry Pun presented [32] an approach for embedding a digital watermark into an image based on geometric invariant properties in Fourier transform domain. They proved that the rotation, scale and translation (RST) invariant is sufficient for dealing with any combination or permutation of rotation, scale and translation in any order. However, the resulting watermarked image quality is not good due to interpolation errors, and the watermarking system has found to be weakly resistant to lossy compression and cropping.

The second solution is to embed the watermark with image normalization. The key idea is to geometrically transform the image into a standard form. The parameters of the normalized image are computed from the geometric moments of the image [58,59]. The advantage of using geometric moments for normalization parameters computations is that it is more image dependent, allowing the decoder to estimate them without the need for the original image. The disadvantage of image normalization is that before/after watermark embedding, the original/inverted image must be normalized. These transforms introduce distortion and the computation is intensive.

The third solution is to identify what the distortions are and then invert them before applying the detector. One commonly used strategy is embedding a template [35,36] as a reference used in the synchronization step during watermarking detection process. Introducing a template is a good strategy to detect geometric distortion since it is robust in terms of concentrating a significant amount of energy into a few points in the

frequency components. However, this approach in general requires exhaustive search for the template, bringing a significant computational burden to the watermarking detection. It should be noted that a combination of geometric attacks may remove a template, usually represented by peaks in a transform domain, so that it is impossible to detect the image distortion as expected.

In a recent paper, Deguillaume et al demonstrated a method [60] that relies on the determination of the regular grid of points for estimating the affine matrix coefficients, and is based on the computation of the hough transform or radon transform, which are known to be very robust in detecting alignments. They embed a periodical structure with many repetitions in order to get a high number of peaks. This requires a significant amount of signal energy for this structure, which might interfere with the real watermark in case of a logo that owner chooses to identify the ownership, hence decreasing the capacity of the watermarking application.

4.1.3 Properties of Radon Transform in Detecting Geometric Distortion

The properties of radon transformations provide accurate estimation of scaling and/or rotation transforms which perfectly match the requirement of watermarking application [61,62]. The radon transform for a set of parameters (ρ,θ) is the line integral through the image $f(x,y)$, where the line is positioned corresponding to the value of (ρ,θ). If a line is represented by $x\cos\theta + y\sin\theta = \rho$, ρ is shortest distance from the line to the origin and θ is the angle the line makes with the horizontal-axis, then the projection function is written as:

$$R_f(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy \tag{4.3}$$

The δ() is the Dirac delta function. If *f(x,y)* is an image and *g(x,y) = f((x/s),(y/s))* is the image scaled by s (s>0) in both directions, then the RT of image *g(x,y)* is easily found to be

$$R_g(\rho,\theta) = sR_f\left(\frac{1}{s}\rho,\theta\right). \tag{4.4}$$

in other words, the RT amplitude of the scaled image is only multiplied by the scale factor *s*. If $f(r,\phi)$ is an image represented in polar form and $g(r,\phi) = f(r,\phi-\phi')$ is the image rotated by ϕ' around the (r,ϕ) coordinate system's origin, then the RT of image $g(r,\phi)$ is easily determined to be

$$R_g(\rho,\theta) = R_f(\rho,\theta-\phi') \tag{4.5}$$

i.e., the RT of the rotated image is rotated by ϕ' .

The properties of radon transform given in Equation 4.4 and 4.5 are very desirable in watermarking applications in which resistance to geometric attacks is required. It is well known that rotation and scaling are the most common geometric attacks which will dramatically disable the receiver from detecting the watermark. The properties of radon transform provide an efficient and accurate estimation to detect such attacks in watermarking applications.

4.2 Proposed Approach

Inspired by Deguillaume's [60] method, we proposed a method in this chapter to estimate geometric attacks based on edge detection, radon transform and watermark detection. We detect edges in selected part of a cover image (see detail next section), resulting in an edge map. The generation of the edge map is an essential step and lays a foundation for further estimation. Three important components are considered to estimate geometric attacks. First, we apply radon transform over the edge map, store the location of main axis θ and the max ρ on the main axis as reference parameters. In later detection stage, the reference parameters are compared with the corresponding ones to detect rotation and/or scaling attacks. Secondly, a grid of edge strength is constructed based on the edge map, from which a reference sequence is generated and normalized. Geometric distortion changes the orientation and the shape of the grid but does not change the underlying regularity of the grid. Thus, in the detection stage, we search for a grid from which a sequence is generated and normalized that has maximum autocorrelation with the reference sequence. A breadth-first iterative-deepening A* search (BF-ID) algorithm is applied to find the accurate solution with less cost than exhaustive search. A found max correlation coefficient that is higher than a threshold helps to resolve the coefficients in an affine transform matrix or confirms the estimation from other components. Third, the normalized reference sequence described above is embedded as watermarks into the spread spectrum signals in the geometric feature blocks (i.e. four corner and center blocks) of the grid. An exhaustive full correlation search is applied if necessary to find the embedding blocks. Presence of such watermarks in those blocks is used to detect translation and cropping attack.

Transform parameters such as the rotation angle, scaling factor or shear parameter of distorted images are accurately estimated based on the combined results of the above three estimating components. Once the geometric distortion is found, we can then invert the distorted watermarked image back to its original state then apply the watermarking detector.

There are a number of advantages for this approach. First, edges are essentially the frame part of an image which survives image degradation and intentional attacks. Estimating the distortion based on edge detection is more reliable than using embedded structures, patterns or local peaks. Second, a very small amount of information is embedded in the geometric feature blocks in the grid in order to identify the desired shape, orientation and position of the grid. It hardly decreases the capacity of the watermarking system. Third, the robust estimation method is combined with radon transform which is known to be very robust in detecting alignments, even when noise is introduced. Fourth, a heuristic search algorithm is developed to reduce the amount of computation, permitting an efficient estimation.

4.2.1 Watermarking Embedding

The embedding procedure of the proposed approach is described as follows:

Step 1: Analyze the structure of the image based on edge and line detection. In our work, we use the optimal Canny edge detector [89]. Canny method differs from the other edge-detection methods in that it uses two different thresholds to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This

method is therefore less likely than others to be fooled by noise, and more likely to detect true weak edges. We only detect the edges within a centered circle, so the edge output will not be affected by rotation or small cropping. Figure 4.1 shows the detected edges in Lena. The edge map is a binary image $B=[b_{ij}],\ i=1,2\ldots N,\ j=1,2\ldots N$ contains 1 where the edges are detected and 0 elsewhere.

Step 2: A grid of edge strength is produced based on the edge map shown in Figure 4.2. We divide the edge map into small blocks and compute the edge strength of each block as:

$$e_k = \sum_{i,j}^{n} b_{ij} \tag{4.6}$$

where n is the block size. The data collected from grid is constructed into a sequence $E(e_1,e_2,\ldots e_k)$ and normalized with zero mean and unit variance.

Step 3: Embed the normalized sequence as a watermark into the geometric feature blocks (i.e. the four corners and the center) on the grid. The reference sequence $E(e_1,e_2,\ldots e_k)$ obtained in step 2 is divided into five parts, and each is additively embedded into the spread spectrum signals on the geometrical feature blocks as displayed in Figure 4.3. The spread spectrum signals may be computed in the DCT, DWT or DFT domain. We suggest using DWT coefficient because of its robustness against common image processing. We do not suggest use curvelet decomposition because it doesn't show advantages over wavelet decomposition in edge weak area, and the computation for wavelet is more efficient. The watermark embedding process is given by

$$f'(m,n) = f(m,n) + \alpha \cdot \left| f(m,n) \right| \cdot w_i \tag{4.7}$$

Figure 4.1. Detected edges of Lena in selected circle area

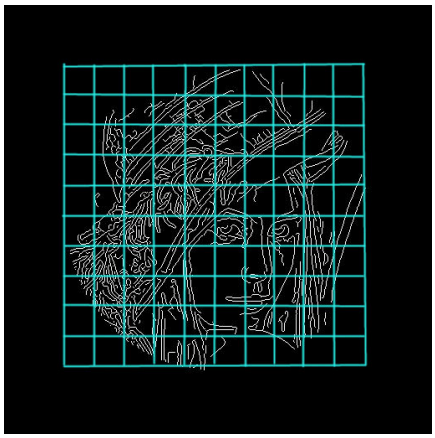


Figure 4.2. The grid of edge strength based on edge map

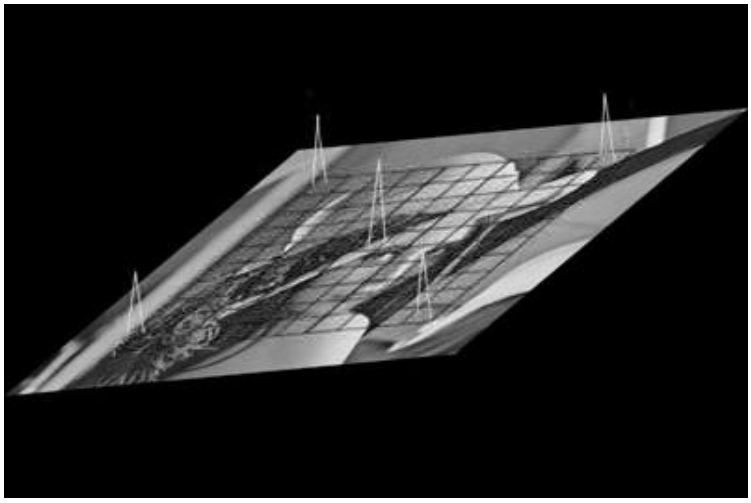


Figure 4.3. Peaks indicate the embedding area of watermarks in Lena

α is local weighting factor that provides control over the watermarking strength and adapts to minimize the artifacts caused by watermark.

Step 4: We apply radon transform over the edge map, which converts the *x,y*-representation into a ρ,θ -representation, ρ being a distance of projection from the origin and θ is the angle the line makes with the horizonal-axis. The location of main axis θ_{max} and the max ρ_{max} on the main axis are stored as reference parameters needed in watermark detection.

4.2.2 Estimation of Affine Transform in Watermarking Detection

We first compute the edges of the cover image. The radon transform (RT) is applied to the edge map of the cover image. Let us first consider the case of a scaling and/or a rotation attack on the watermarked image by s_a and/or θ_a, respectively. Based on the RT properties, it is obvious that after the attack, the location of the new maximum will be $s_a\rho_r$ and $\theta_r+\theta_a$, where ρ_r and θ_r are the reference parameters stored during the embedding process. Subsequently, the watermarked image is scaled by the scaling factor

$$s_a = \frac{\rho_{\max}}{\rho_r} \tag{4.8}$$

and rotated by the angle

$$\theta_a = \theta_{\max} - \theta_r \tag{4.9}$$

The radon transform over the edge map of the watermarked image reveals the scaling factor and the rotation angle which will permit a successful detection.

However, the radon transform is not sufficient to determine arbitrarily combined geometrical attacks such as rotation, shearing and other attacks. The determination of the general affine transform applied to an image relies on the determination of the regular grid of edge strength described in the previous section. Our approach relies on the regularity of the grid being unchanged although the grid shape, orientation and position may be altered by attacks. Once the distorted image has been corrected based on RT properties, the points extracted from a corrected image are auto-correlated with a known reference sequence:

$$\delta = \frac{s^* \cdot s}{\|s^*\| \cdot \|s\|} \tag{4.10}$$

where s denotes the reference sequence and s^* denotes the sequence obtained from the grid of edge map in distorted image. For a single rotation and/or scaling attack, if the output δ is far greater than threshold τ (e.g. 0.85), it confirms the radon transform for estimating the affine matrix coefficients. If the δ is far less than threshold τ, we conclude the cover image has gone through some other attacks in addition to rotation and scaling, most commonly a rotation combined with shearing. If the δ is in the range close to the threshold (e.g. 0.83-0.88), a tuning process (small rotation in both directions or small up-down scaling) is launched with the purpose searching for the max δ, then we can make a decision according to the value of δ_{max} by comparing with the threshold. The tuning process is used to reduce the probabilities of false positive/negative. If the detection is confirmed, the distorted image is inverted accordingly.

Detection of translation can be achieved by performing an exhaustive full correlation search for the watermarks embedded in the geometric feature blocks of the grid:

$$\sigma = \max_{t_x, t_y}[\frac{1}{p}\sum_{i=1}^{p} s_i \cdot \phi(x, y)]_{I(x+t_x, y+t_y)} \tag{4.11}$$

where t_x and t_y define the translation, $\phi(x, y)$ are selected coefficients from spread spectrum signals, s_i denotes the mark, p is the size of s. If the output σ is greater than the threshold T_σ given by:

$$T_\sigma = \frac{\alpha}{3M}\sum_{i=1}^{M}\left|\phi^*(x, y)\right|, \tag{4.12}$$

then the watermarks are detected.

A cropping attack can be detected if one or more geometric feature block(s) on the grid are cropped, because the watermark is embedded to the geometric feature blocks on the grid. Thus if the max correlation is found between the watermarks and the coefficients of only some geometric feature blocks, cropping attack is detected.

The detailed steps we used to estimate general affine transforms are depicted in the flow chart in Figure 4.4. θ_r (known main axis), ρ_r (known max distance on main axis) and E_r (known sequence) are given as reference parameters. Three branches are extended in the flow chart: the left one is to determine whether or not scaling attack is applied to cover image, the right one is to determine whether or not flip attack is applied to cover image, the middle one is the major branch that determine all the other geometric attacks. The middle branch begins with a detection and estimation of rotation attack. Then the

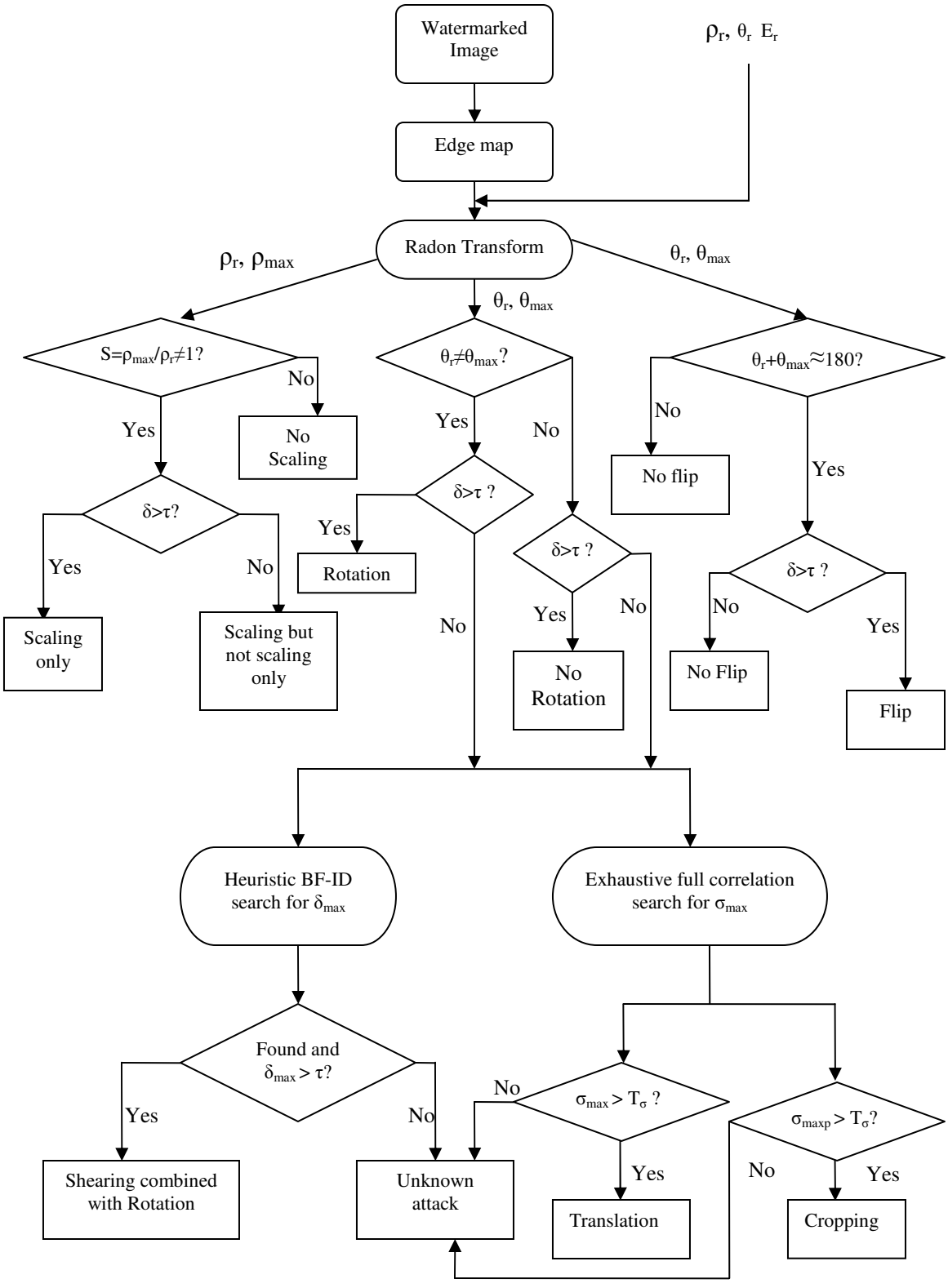


Figure 4.4. Flow chart of detail steps used to determine single and/or combined attacks

problem is reduced to searching for the peak in the correlation defined in Equation 4.10 and 4.11 before and after affine transform. In order to search for peak δ_{max} in Equation 4.10, a breath-first heuristic search and iterative-deepening A* (BF-ID) algorithm is performed on possible affine transform parameters. The proposed search algorithm is illustrated in the next section. If a value for δ_{max} is found that is greater than threshold, it is used to estimate an affine transform which is a combination of rotation and shearing. At the same time, an exhaustive search is performed to find the peak σ_{max} defined in Equation 4.11; if a σ_{max} found to have full correlation with all the coefficients in the geometric feature blocks on the grid, it is used to estimate a translation attack. A σ_{max} found to have correlation with only some partial coefficients in the geometric feature blocks helps to detect cropping attack (see detail in next section).

4.2.3 Breath-First Iterative-Deepening A* Search for Affine Transform Coefficients

Most geometric attacks can be uniquely described using general affine transforms illustrated in section 4.1. A is used to represent a succession of n arbitrary linear transforms A_i, $i=1\ldots n$ yields another linear transform, which can be expressed as $A= An\cdots A_1$ We will let A represents any rotation, shearing, scaling or any combination of them. Let's consider a typical case, rotation combined with shearing; the parameters for the geometric transform are:

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \tag{4.13}$$

We use the breadth-first iterative-deepening A* algorithm to search for the optimized coefficients for *A* matrix. The search tree shown in Figure 4.5 represents the possible paths in which one leads to the goal node.

Let's consider a typical attack in the case of rotation combined with shearing, each node in the tree represents a rotation angle. The root represents the rotation angle that is estimated by the radon transform. If the correlation δ after inverting rotation is lower than threshold, then the cover image has undergone combined geometrical attacks, the estimation of rotation angle solely based on RT properties is inaccurate. Then we must select a certain set of rotation angles. For each particular angle, we randomly selected a number of shearing factors in the range of [-1,1]. The number of shearing factors tested for each particular angel is increased along with the increasing tree layers. We computed the correlations defined in Equation 4.10 for each pair (θ, S), i.e., rotation angle and shearing factor. Then the node will receive a score according to the correlation output and the score assigned to its ancestor. The nodes with scores higher than a lower bound

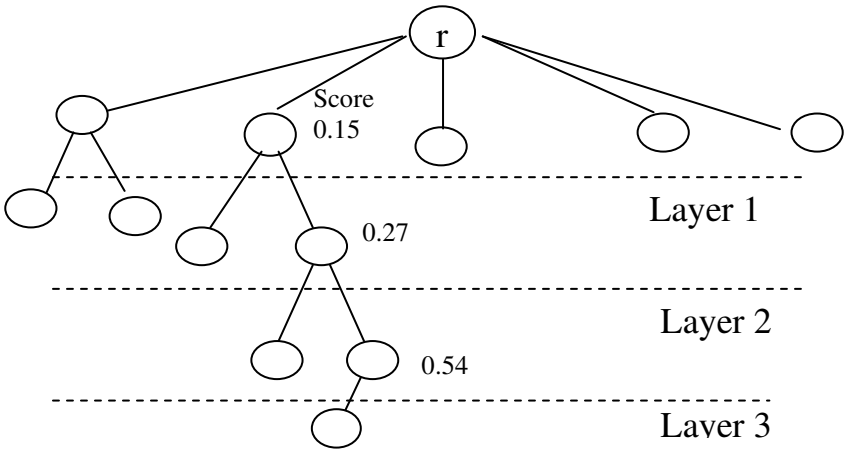


Figure 4.5. Breath first search tree for estimation of coefficients for *A*

will open to expand. Our breadth-first algorithm uses a lower bound on the cost of an optimal solution to prune the search space, and the quality of the lower bound has a significant effect on the efficiency of the algorithm. The better the lower bound selected, the fewer nodes are expanded and stored. (In fact, given an optimal lower bound, the algorithm does not expand any more nodes than A*) If no solution is found, it decreases the lower bound and repeats the search. A lower bound can be obtained by finding an approximate solution to the search problem. There are many possible ways to quickly determine a lower bound. An obvious method is to choose the average correlation computed for the nodes stored in the queue. The goal for our search must satisfy the following requirement: the correlation coefficient δ defined in Equation 4.10 must be greater that a threshold τ. Thus, the pseudocode of the breadth-first algorithm is given below:

```
Procedure ExpandNode( Node n)
        Successors(n) ←Neighbors of n under certain condition
        For each n' ∈ Successors(n) do
                Layer(n')=Layer(n)+1
                Ancestor(n') ←n
                CC(n')=compute( δ )
                Score(n')=CC(n')-0.01*Layer(n')+ 0.01*Score(n)

Algorithm BFHS (Node root_node, goal)
        queue = [ ];
        node = root_node;
        add_to_back_of_queue (Successors (node));
        loop:
                if is_empty (queue)  then report FAIL
                else:
                        new_node = remove_from_queue (queue)
                        if achieves_goal (new_node) SUCCEED
                        else
                                if  Score(new_node)>lower bound(queue)
                                        ExpandNode(new_node)
                                        add_to_back_of_queue (Successors(new_node))
```

4.3 Experimental Results

We have tested our proposed approach on gray images (Lena, Baboon, Boat, Airplane, Goldhill, etc.) of dimension of 512 ×512. Here we give the results for standard image "Lena". The original "Lena", the watermarked "Lena" with PSNR=40.72, and the edge map of Lena are displayed in Figure 4.6. The radon transform of the edge map is shown in Figure 4.7. The main axis is located at θ=122º and ρ_{max}=181; they are reference parameters required in later detection process. In this section, we will illustrate how to estimate geometric distortion in detail. We first demonstrate the estimation of single

Figure 4.6. Original "Lena", watermarked "Lena" and the edge map of "Lena" (left to right)

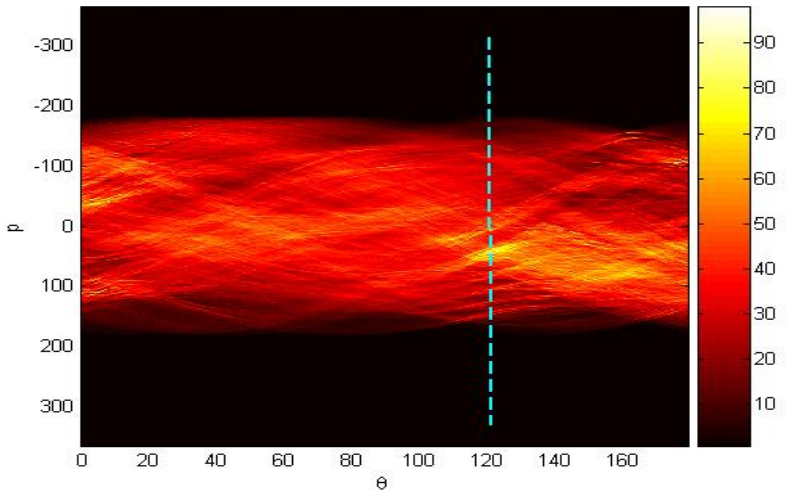


Figure 4.7. Radon transform of "Lena" edge map with main axis θ=122º and ρ_{max}=181

affine transformation parameter, and then we explore the methodology that solves the combined affine transformations.

4.3.1 Rotation

The RT property given in Equation 4.9 provides an accurate estimation for the rotated angle in a single rotation attack: The rotation angle can be estimated in terms of the shift of the principal directions of the main axes based on the aligned points. An example illustrating the performed experiments is given in Figure 4.8. The cover image Lena has

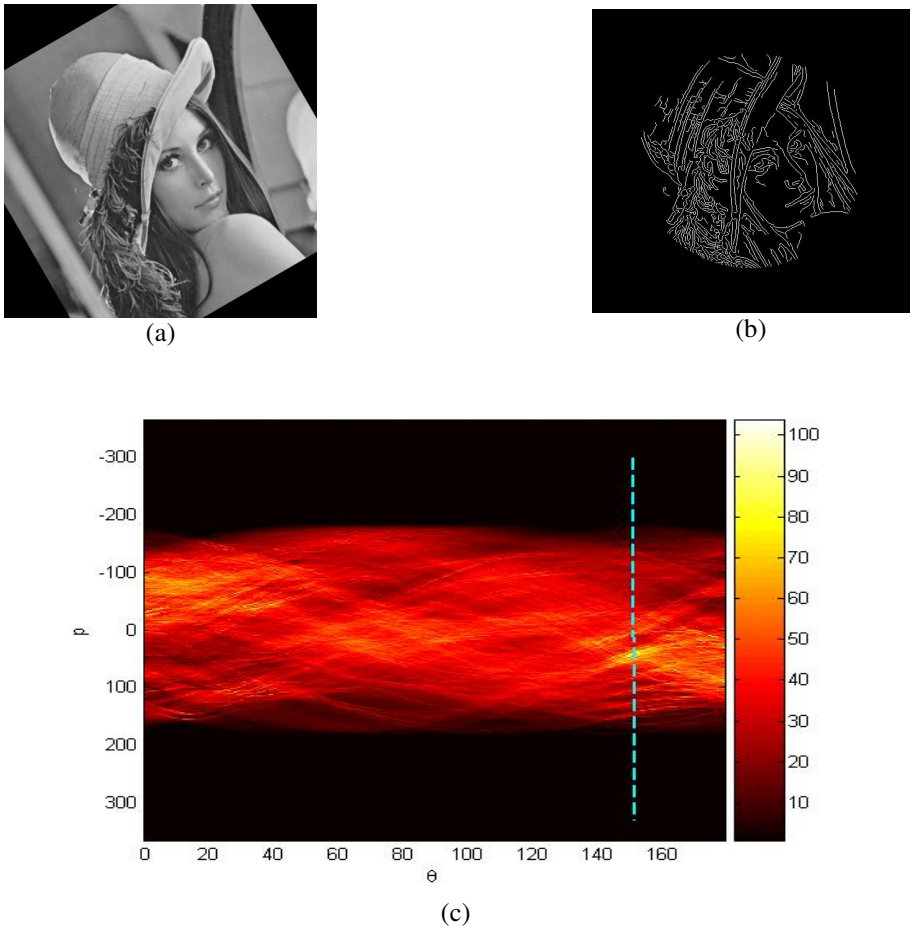


Figure 4.8. Detect rotation in radon transform (a) Rotated "Lena" with angle 30º (b) Edge map of rotated "Lena" (c)Radon transform of edge map in rotated Lena with main axis at θ=152º, ρ_{max}=181.

been rotated by 30°. Referring to parameter θ=122º, the detected main axis based on the strongest peak in radon transform is moved to θ=152º, i.e., shifted by 30°. It should be.noted that if the image has gone through other geometric distortion in addition to rotation, then such distortion will interfere with the direction of the main axis in radon transform. That means we cannot estimate the angle solely depending on the main axis shift. This issue will be discussed in detail later. Therefore, we need to confirm the estimation using the correlation defined in Equation 4.10 between a known reference sequence and the one obtained from the corrected image. The output δ is 0.999956 far higher than the chosen threshold τ=0.85, which confirms the estimation for the rotation angle θ. If the correlation δ is less than the chosen threshold, we consider the cover image has gone through more than one geometric attack

4.3.2 Scale

The scale operator performs a geometric transformation which can be used to shrink or zoom the size of an image. The scaling factor can be estimated in terms of the ratio given by Equation 4.8. Figure 4.9 shows the radon transforms applied to the edge map of the scaled Lena by factor 0.5. It can be seen ρ_{max}= 90 which is half of the original size. We also noticed the main axis is slightly shifted, although the image is not rotated. Such error is introduced by sub-sampling or pixel replication/interpolation due to image reduction or image zooming. If we strictly follow the estimation and correct the image accordingly, then we compute the correlation δ in order to proof check the correctness of the estimation. A value of δ =0.871 will launch a tuning process with the purpose to search for the max δ, the correlation δ =0.999 is found at θ=122º. Thus, the sub-sampling error

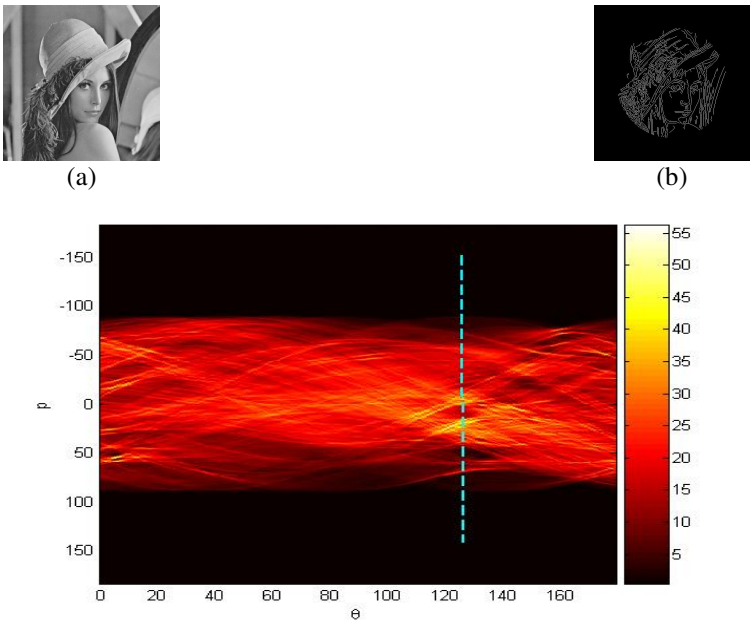


Figure 4.9. Detect scaling in radon transform (a) Scaled "Lena" with scaling factor 0.5 (b) Edge map of scaled "Lena" (c) Radon transform of edge map of scaled Lena with main axis at θ=124º, ρ_{max}=90.

due to image reduction is corrected by tuning.

4.3.3 Flip

The flipping operator flips the image from left to right along the vertical axis or in the up-down direction along the horizontal axis. Figure 4.10 demonstrates the detection of a flipped "Lena" in the left-right direction, when the position (x_1, y_1) of an input image maps to the position of (x_1, -y_1) of an output image. We performed radon transform on the edge maps in both images. The main axes θ_1 =122º and θ_2=60º obtained from input and output images approximately follow the rule of $\theta_1 + \theta_2 = 180^\circ$. When we observed the main axis is shifted that distance, there are multiple possibilities for such change. In addition to left-right flip, a rotation attack or up-down flip is possibly applied to the cover image. The estimation of correlation output δ helps to distinguish the possible cases, and

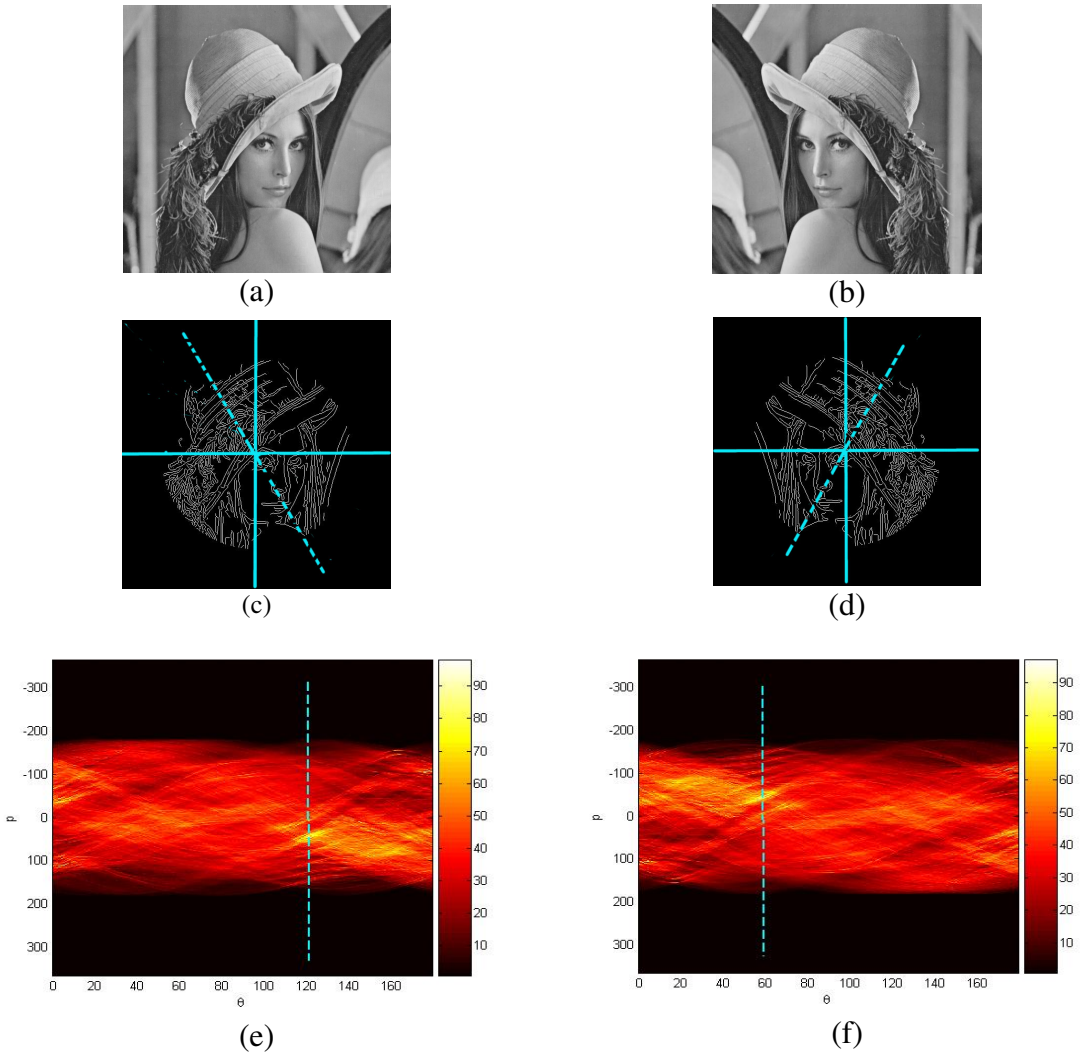


Figure 4.10. Detect flip attack in radon transform (a) original Lena (b) Flipped Lena (c) edge map of the original Lena (d) edge map of the flipped Lena (e) Radon transform of (c) with the strongest peak appears at θ_1=122° ρ_1=181 (f) Radon transform of the flipped Lena with the strongest peak appears at θ_2 = 60° and ρ_2=181

a tuning process is also evolved to minimize the error. Thus left-right flip is confirmed with the correlation δ= 0.999 and the watermark embedded in geometrical blocks is also detected.

4.3.4 Translation and cropping

The translation operator maps the position of each picture element (x_1, y_1) in an input image into a new position (x_2, y_2) in an output image where $x_2 = x_1 + t_x$ and $y_2 = y_1 + t_y$. The

dimension of the image is fixed. If the new coordinates (x_2, y_2) are outside the image, the translation operator will either ignore them or it may link the higher coordinate points with the lower ones so as to wrap the result around back onto the visible space of the image (so-called *circular* translations). Cropping is another geometric attack in which part of the image information is lost. Both attacks can be detected through an exhaustive search of the watermarks embedded in the geometrical feature blocks on the edge map grid. If the positions of the embedding areas - the center and the corners of the grid are shifted by specified translation (t_x, t_y), we will conclude a translation attack was applied. Failing to find the watermarks in geometric feature blocks indicates part of the grid is

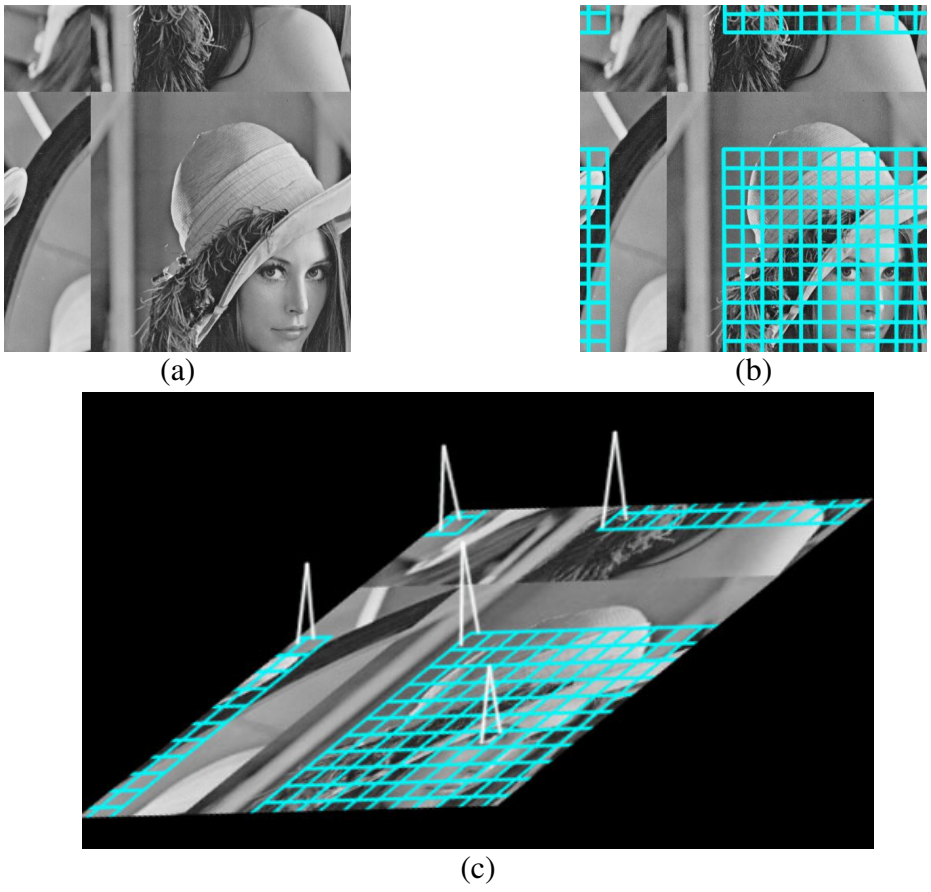


Figure 4.11. Detection of translation (a) "Lena" under circular translation (b) "Lena" under circular translation with the grid of edge strength (c) Detected watermarks in the geometrical feature blocks on the grid.

cropped. Figure 4.11 and 4.12 demonstrate the detection of translation and cropping in term of watermark search. It should be noted that if a small cropping is applied to the cover image which keeps the grid intact, then the cropping cannot be detected using this approach.

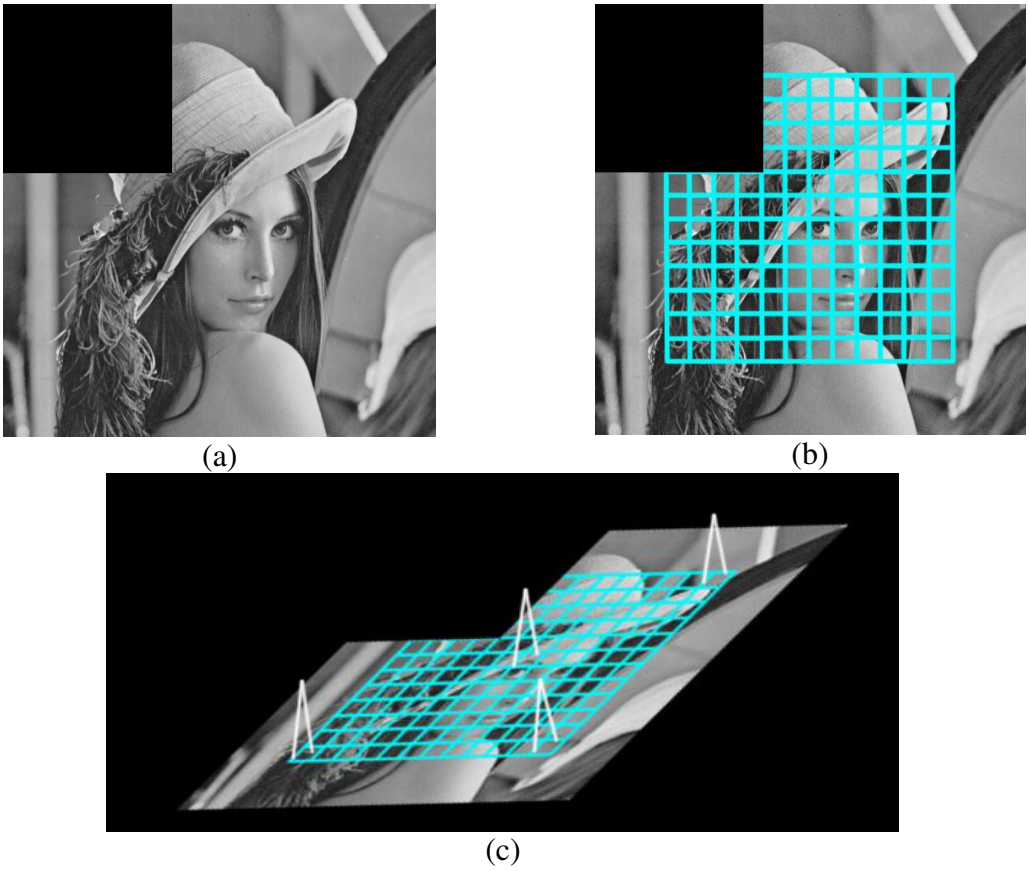


Figure 4.12. Detection of cropping (a) "Lena" with top-left corner cropped (b) Cropped grid inside Lena (c) Detected watermarks in the geometrical feature blocks of the grid; the one in top left corner is lost

4.3.5 Shearing combined with rotation

The shear operator maps the position (x_1, y_1) of an input image to the new position in an output image. We are looking for solve the coefficients in affine transform described in Table 4.1. We consider the shearing attack is combined with rotation, because the

estimation procedure is always starting from an assumed rotation angle. A single shearing attack is simply considered as one combined with rotation of degree 0. When an image is distorted by shearing, the shape and the orientation of the grid based on edge map are also changed, however the underlying regularity of the grid is kept. An example is illustrated in Figure 4.13, where the cover image "Lena" is rotated by 17º and horizontally sheared with factor 0.3. We are looking for the normalized sequence obtained from the grid that is maximally correlated with a known sequence. We follow the breadth-first iterative-deepening A* algorithm as described before. We first picked -150, -120, -90, -60, -30, 0, 30, 60, 90, 120, 150, 180 for rotation angle as expanding nodes. For each particular angle, we randomly selected 10-15 shear factors in the range [-1,1]. We computed the correlation with a known sequence for rotation-shearing pair: when the tested rotation-shearing pair is closer to the right one, the output δ is also closer to 1. The correlation output is contributed the node's score. The layer number of the node and the score of node's ancestor both affect node score. The nodes receiving a high score are allowed for further expanding as illustrated in Figure 4.14 .

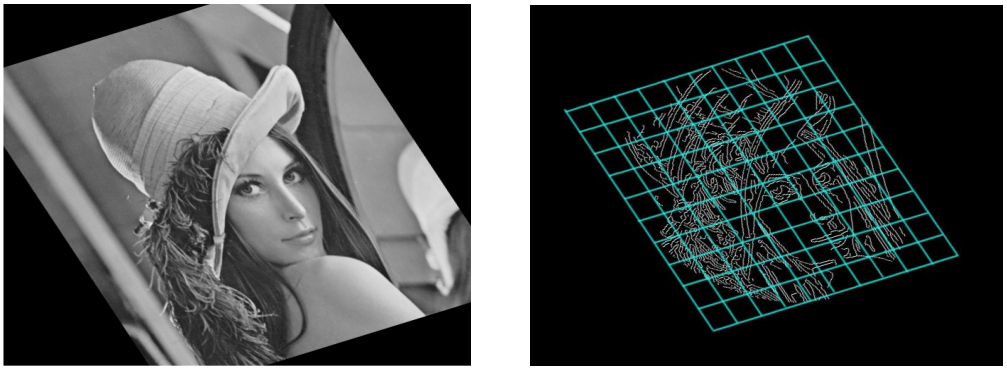


Figure 4.13. Lena and corresponding grid of edge map after rotation combined with shearing attack where the shear factor is 0.3 and the rotation angle is 17°.

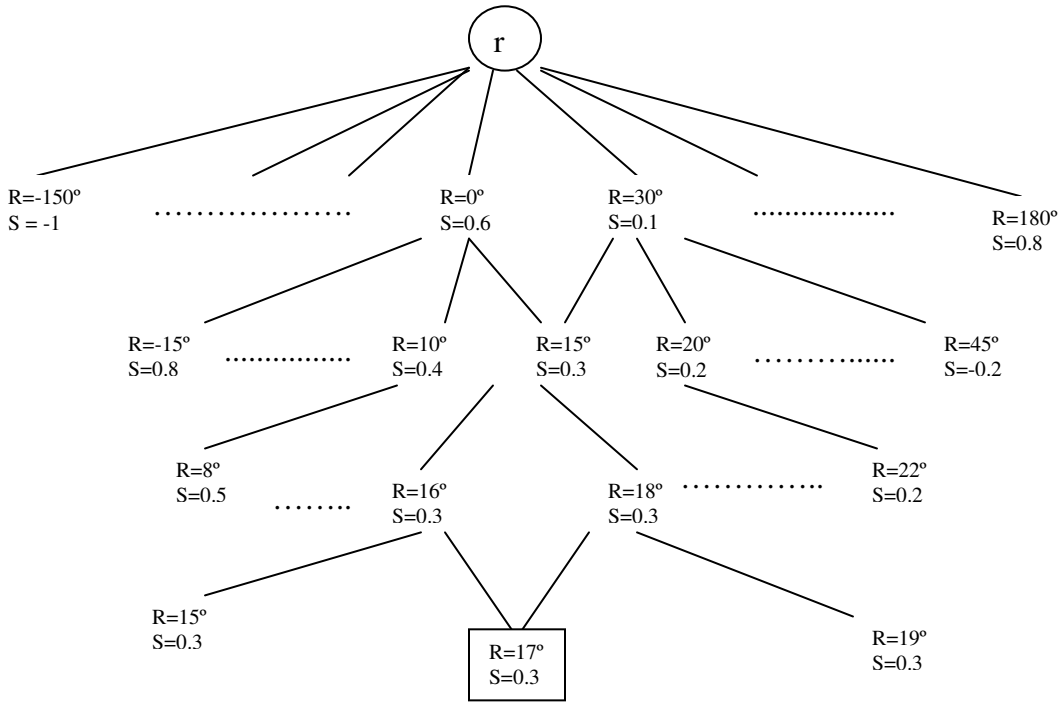


Figure 4.14. An example of breath first search tree for estimation of coefficients for affine transform

As the search tree grows, we can quickly determine the right angle is in the interval [0, 30]. The search tree grows down and eventually we find the peak δ appears at the R=17º and S=0.3, here, R denotes the rotation degree and S denotes the shearing factor. A plot of correlation δ against rotation degree in the range [-150, 180] is given in Figure 4.15.

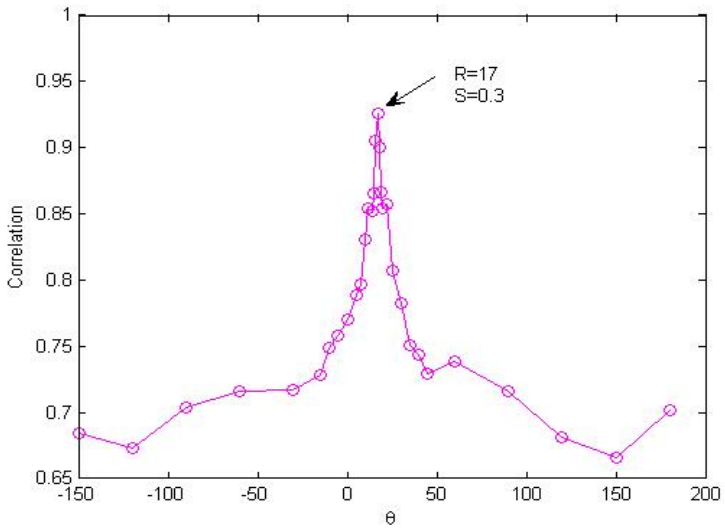


Figure 4.15. Plot of correlation δ against rotation degree in the range [-150, 180]. The peak δ = 0.93 is obtained at R=17º and Shearing factor 0.3.

Figure 4.16. Revert the image back to original state

The peak is obtained at R=17º and S=0.3. Once the coefficients for distortion are solved, the cover image is restored shown in Figure 4.16. The total search time use matlab software on a computer with Intel Pentium M processor 1.86 GHz takes a few minutes.

4.3.6 Geometric attack combined with JPEG compression

We tested our system with the image having undergone JPEG compression with a quality factor (QF) of 10% after rotation and shearing distortion. We successfully estimate the geometric distortion even though the image was significantly compressed. Figure 4.17 shows the corrupted image after a combination of rotation, shearing and JPEG compression with QF=10%; the quality of the image is severely degraded. We followed the search algorithm and found the peak of correlation $\delta = 0.87$ also appears at R=17º, S= 0.3 shown in Figure 4.18. Therefore, we conclude the search scheme is robust against image degradation such as JPEG compression. Thus, we can invert the geometric transform prior to applying the watermarking detector.

Figure 4.17. Corrupted Lena rotated by 17°, sheared with factor 0.3 and compressed with QF=10%

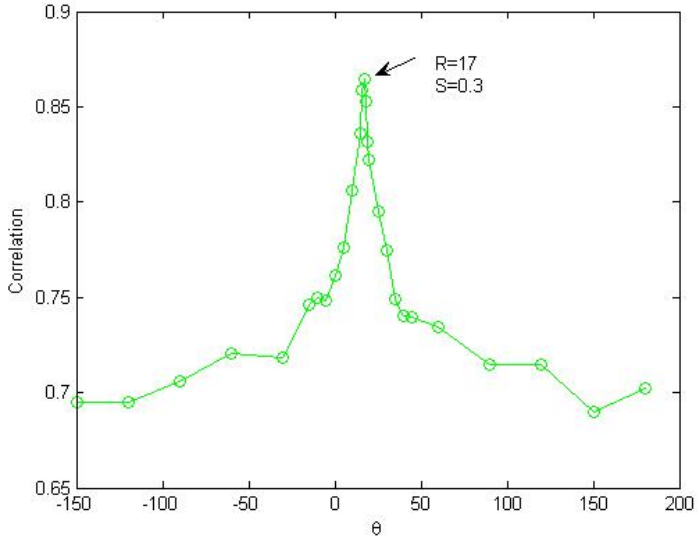


Figure 4.18. Plot of correlation δ against rotation degree in the range [-150, 180]. The cover image is under JPEG compression with quality factor 10%. The peak δ=0.87 is also obtained at R=17º and Shearing factor 0.3.

Finally, our proposed approach has been tested with Stirmark benchmark [37] with respect to the geometrical attacks. Five proposed images were marked with a PRN and the required PSNR of about 38 dB. Each test image has gone through rotation (up to 180º), Cropping (up to 20%), Shearing, Flip, Scaling attacks (0.5 -1.5). The performance of our approach is reported in Table 4.2

Table 4.2. Results of Stirmark geometrical attacks

Geometrical attack	Stirmark score
Scaling	1.00
Cropping	0.99
Shearing	1.00
Rotation	1.00
Flip	1.00

4.4 Conclusion and Discussion

In this chapter, we presented a scheme for detection and recovering geometric attacks in image watermarking. The proposed method provides an accurate estimation of single and combined geometrical distortions based on edge detection and radon transform which is known to be very robust in detecting alignments. A breath-first heuristic search and iterative-deepening A* algorithm is applied so as to reduce the cost for detecting distortion considering possible affine transform parameters. The high efficiency of the method is demonstrated even when image degradations have occurred, including JPEG compression with a quality factor of 10%. And the experimental results show that this method requires fair computation and achieved high precision.

It should be noted the current design of system does not apply to image aspect ratio change. The images used in experiments are uniformly scaled. We consider the following approach to compensate for this limitation. We may require the aspect ratio of the original image as side information which may be needed in detection stage. An aspect ratio change is detected by referring to this information. Then the image is recovered to original aspect ratio and we proceed as described above.

Chapter 5

Statistical Analysis for Watermarking Detection

The choice of threshold in watermarking detection has great impact on the validity of watermarking system. For a valid system, it is necessary for the probability of detection to be very high, and the probability of false alarm should remain as low as possible. A technique for validly and effectively selecting the threshold is proposed based on statistical analysis over the host signals and embedding schemes. The technique is also used for the marking algorithm adjusted by JND model. Experiments show the scheme keeps both the probability of false positive and the probability of false negative low and is generally robust against a wide range of image attacks.

5.1 Introduction

The detection of the watermark needs to be very reliable in real-life applications. The probability of detection needs to be very high, and the probability of false alarm should remain as low as possible for a real and efficient watermarking scheme. There is a trade-off between the imperceptibility and the detectability of the watermark: the system designer aims to embed the strongest possible signal, to ensure its reliable detection, but at the same time limit its strength to keep it imperceptible.

Most watermarking systems aim to extract or detect a watermark without the use of the original. That is, they are blind methods. In practice, additive embedding strategies and a "correlation detector" [26,28] are often used, such detection often takes the form

$$z = \frac{\Sigma^{N} c^{*}_{m,n} \cdot w_i}{N} \tag{5.1}$$

z is retrieved by correlating the watermark sequence $W\{w_1, w_2, \ldots, w_N\}$ directly with all N coefficients of the received image signal $c^{*}_{m,n}$. z values range over some interval, but only the sequence that was originally embedded yields a high correlation output, in which case we can conclude that the image has been marked with W, as show in Figure 5.1.

Often the detection decision is made with respect to a threshold T_z, such that the detector reports the presence of a watermark if and only if $z>T_z$,. The detection threshold can be derived either experimentally or analytically. One common way is to determine the

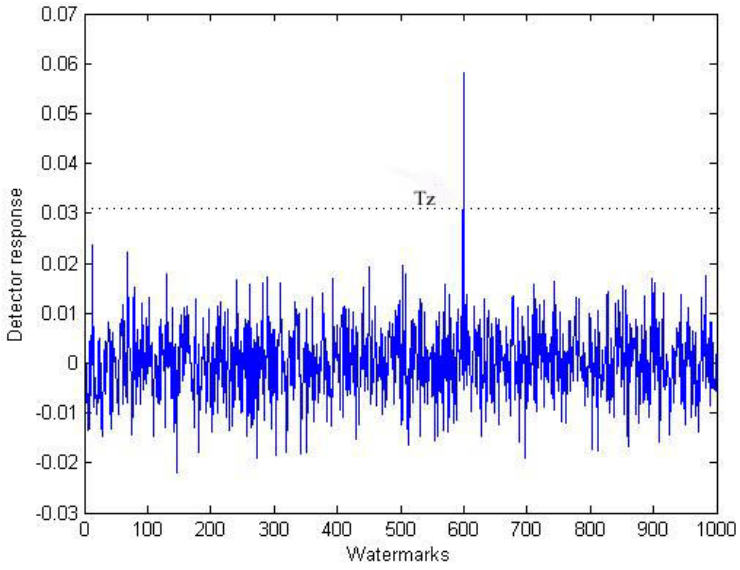


Figure 5.1. The detector response of embedded watermark against random sequences: A large number of random sequences tested, only the sequence that was originally embedded yields a high correlation output.

threshold as [28]:

$$T_z = \frac{\alpha}{2MN} \sum_{n=1}^{N} \sum_{m=1}^{M} \left| c_{m,n}^{*} \right| \tag{5.2}$$

However, the authors [28] did not provide theoretical proof and statistical analysis of false alarms for the threshold selection. In the following section, we will propose a technique for validly and effectively selecting the threshold based on statistical analysis over the host signals and embedding schemes.

5.2 **Statistical Analysis for Proposed Threshold Selection**

As in the typical watermarking scenarios, we will begin with the cover image *I*, which is decomposed into a transform domain, say, curvelet. A vector C of coefficients is selected and the watermark $W\{w_1, w_2, \ldots, w_N\}$ as a pseudo random sequence is additively added to the elements in C, yielding:

$$c_i^{*} = c_i + \alpha \cdot \left| c_i \right| \cdot w_i \tag{5.3}$$

In watermark detection, let I^{*} be the watermarked and possibly corrupted image, and a vector C^{*} is extracted from curvelet domain of I^{*}. We suppose that the watermarked image has not been corrupted. Therefore, the detector response as the correlation between C^{*} and the testing watermark $\hat{W}$ is:

$$z = \frac{1}{N} \sum_{i=1}^{N} \left(c_i \cdot \hat{w}_i + \alpha \cdot \left| c_i \right| \cdot w_i \cdot \hat{w}_i \right) \tag{5.4}$$

In particular, if the testing watermark $\hat{W}$ matches the watermark W embedded in the image, z becomes:

$$z = \frac{1}{N}\sum_{i=1}^{N}(c_i \cdot w_i + \alpha \cdot |c_i| \cdot w_i^2) \tag{5.5}$$

The correlation value z is a random variable, whose probability density function can be assumed to be Gaussian, in accordance with the central limit theorem; its parameters have been studied with the following hypotheses: c_i are equally distributed random variables, having symmetrical probability density function and zero mean. If N is large enough, the watermark here as weight factors $w_i \in W\{w_1, w_2, \ldots, w_N\}$ has the property that different vectors W and $\hat{W}$ follows:

$$\frac{1}{N}\sum_{i=1}^{N} w_i \hat{w}_i = \begin{cases} 1 & if \quad W = \hat{W} \\ 0 & if \quad W \neq \hat{W} \end{cases} \tag{5.6}$$

According to these assumptions, mean and variance of z can be estimated:

$$\mu_z = \begin{cases} \alpha\mu_{|c|} & if \quad W = \hat{W} \\ 0 & if \quad W \neq \hat{W} \end{cases} \tag{5.7}$$

and

$$\sigma_z^2 = \begin{cases} \frac{1}{N}\left(\left(1 + 2\alpha^2\right)\sigma_c^2 + \alpha^2\sigma_{|c|}^2\right) & if \quad W = \hat{W} \\ \frac{1}{N}\left(1 + \alpha^2\right)\sigma_c^2 & if \quad W \neq \hat{W} \end{cases} \tag{5.8}$$

where $\mu_{|c|} = E[|c|]$, $\sigma_c^2 = \text{var}[c]$, and $\sigma_{|c|}^2 = \text{var}[|c|]$, since α^2 is far smaller than 1, we approximate:

$$\sigma_z^2 \approx \frac{1}{N}\sigma_c^2 \tag{5.9}$$

for both cases. Thus, corresponding to the two cases, $W = \hat{W}$ and $W \neq \hat{W}$, two random Gaussian variables z_1 and z_2 are obtained, having the approximate variance $\sigma_z = \frac{\sigma_c}{\sqrt{N}}$ and means respectively $\mu_1 = 0$ and $\mu_2 = \alpha \cdot \mu_{|c|}$. In order to distinguish the two cases with minimum probabilities of false results, we need:

$$k_1 = \frac{T_z}{\sigma_z} > 3 \quad \text{and} \quad k_2 = \frac{\mu_2}{\sigma_z} > 6 \tag{5.10}$$

since in the interval $\{-3\sigma,+3\sigma\}$ 99.73% of values are included. Figure 5.2 shows the probability density function of two random variables z_1 in case of $W \neq \hat{W}$ and z_2 in case of $W = \hat{W}$, having the same variance σ_z and mean respectively μ_1, μ_2. A proper threshold T_z needs to be at least three times the value of σ_z, thus exclude 99.73% of the values in z_1, leading to 0.07% chance of false alarm. Hence, such considerations are useful to properly choose the decision threshold as:

$$T_z = k.\frac{1}{N}.\sqrt{\sum_{i=1}^{N} c_i^2} \tag{5.11}$$

where k is greater than 3.

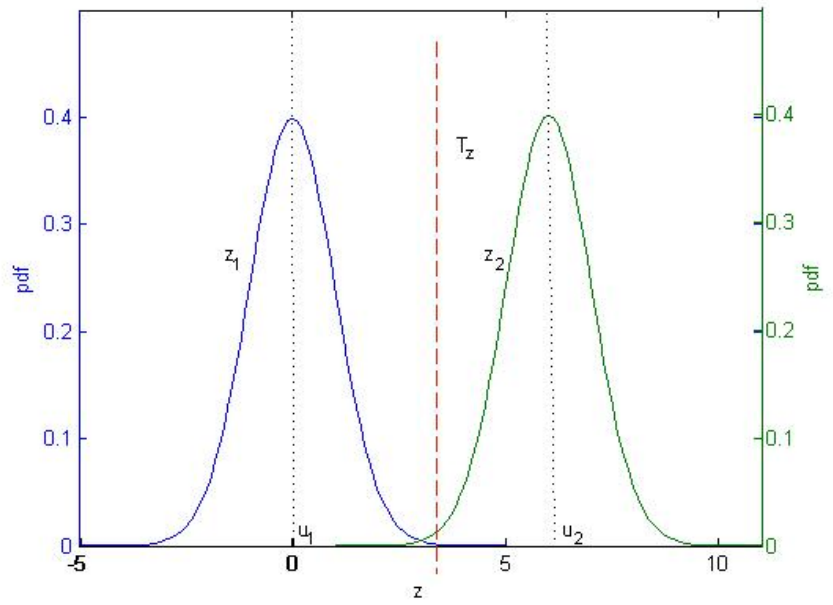


Figure 5.2. Probability density function of two random variables z_1 and z_2, representing detector responses from unwatermarked images and from watermarked images, having the same variance σ and mean respectively μ_1, μ_2. Such considerations are useful to properly choose the decision threshold T_z.

5.3 Detection Error Analysis

For correct behavior of the watermark detection system, it is important to properly choose the decision threshold. In the practical watermark decoder, only one of the following situations is possible:

H0: the image is not marked with *W*;
H1: the image is marked with *W*.

To discriminate between H0 and H1, the detector computes z and compares it with a threshold T_z. If z is lower than T_z then the detector decides the image is not marked with *W*, whereas if z is higher than the threshold, the decoder assumes the image is marked with *W*. The best value T_z is the one that minimizing detector error, or the probability of deciding for the wrong hypothesis:

$$P_E = P(1|0)\cdot P(0) + P(0|1)\cdot P(1)$$
$$= P(z > T_z|0)P(0) + P(z < T_z|1)P(1) \quad (5.12)$$

where $P(0)$ and $P(1)$ are the prior probabilities of H0 and H1, $P(z<T_z|1)$ is the probability of missing the presence of the mark (false negative) and $P(z>T_z|0)$ the probability of asserting the presence of W when W is not actually present (false positive).

5.3.1 False Positive Error

A false positive occurs when a watermark detector indicates the presence of a watermark in an unwatermarked image [4]. Figure 5.3 illustrates how false positive errors can occur. The marked part represents the frequency of occurrence of each possible value that can be output from the watermark detector when no watermark is actually present.

The false positive model depends on the watermark detection algorithm and the manner

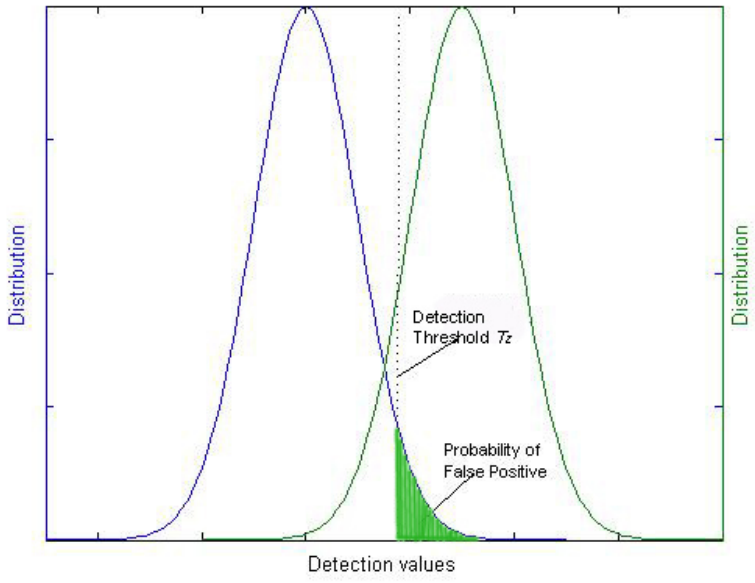


Figure 5.3. Example detector output distributions and a detection threshold. The area under the left-hand curve to the right of the threshold represents the probability of a false positive.

in which the detector is used. The watermark is chosen to have a mean of zero; the detector output, z as defined in Equation 5.5; will also have a mean value of zero. The probability that the detector will output a value of x is then given by [4]:

$$P_z(x) = \frac{1}{\sqrt{2\pi}\sigma_z} \exp\left(\frac{-x^2}{2\sigma_z^2}\right) \tag{5.13}$$

and the probability of a false positive is

$$P_{fp} = \int_{T_z}^{\infty} P(x)dx = \int_{T_z}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_z} \exp\left(\frac{-x^2}{2\sigma_z^2}\right)dx = erfc\left(\frac{T_z}{\sigma_z}\right) \tag{5.14}$$

Figure 5.4 shows an example of the resulting false positive rates as a function of the detection threshold T_z. We follow the watermarking embedding and detection algorithm presented in Chapter 2 and applied to host image "Lena", we set the edge strength

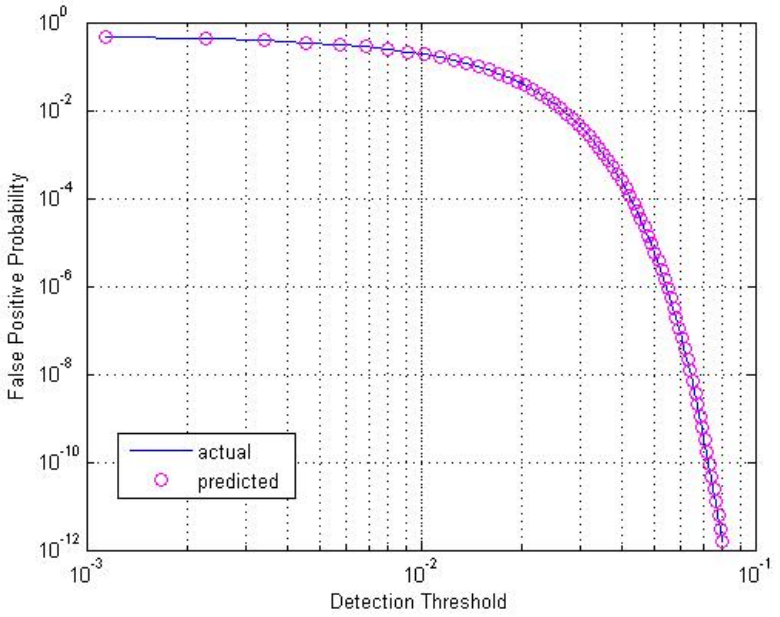


Figure 5.4. Watermark false positive probabilities for "Lena" as a function of detection thresholds. The curve measured with predicted σ_z and u_z is well matched to the curve measured with actual experimental σ_z and u_z.

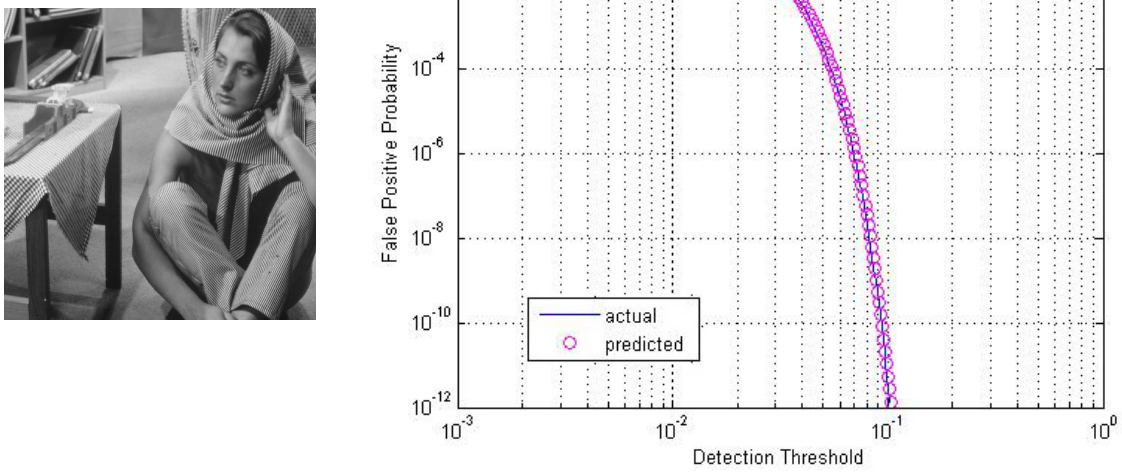


Figure 5.5. Watermark false positive probabilities for "Barbara" as a function of detection thresholds. The curve measured with predicted σ_z and u_z and the curve measured with actual experimental σ_z and u_z are well matched.

threshold as 0, so all blocks are chosen. The measured rates with actual σ_z and u_z are shown with solid line in the graph. The rates measured with predicted σ_z and u_z in Equation 5.7 and 5.9 are indicated by circle markers. As we can see from these curves, the false positive estimation is quite accurate with respect to the predicted parameters. Figure 5.5 shows another example of the resulting false positive rates as a function of the detection threshold T_z for host image Barbara, showing the similar results.

5.3.2 False Negative Error

A false negative occurs when a watermark detector fails to detect a watermark that is present [4]. Figure 5.6 indicates that a false negative occurs because the detector output distribution, represented by the right-hand curve, intersects the threshold T. An analysis of the false negative probability follows the same lines as that for the false positive probability. The false negative model also depends on the watermark detection algorithm

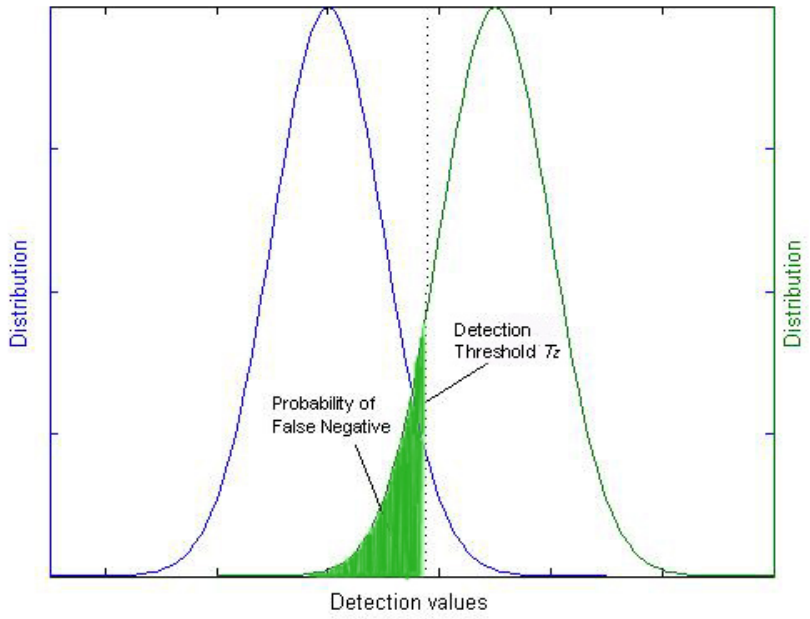


Figure 5.6. Example detector output distributions and a detection threshold. The shaded area under the right-hand curve to the left of the threshold represents the probability of a false negative.

and the manner in which the detector is used in Equation 5.5. The detector output, z defined in Equation 5.5, will have a mean value of $\mu = \alpha \cdot \mu_{|c|}$ and approximate variance $\sigma_z = \frac{\sigma_c}{\sqrt{N}}$. Thus, the probability of a false negative is given by:

$$\begin{aligned} P_{fn} &= \int_{-\infty}^{T_z} P(x)dx = \int_{-\infty}^{T_z} \frac{1}{\sqrt{2\pi}\sigma_z} \exp\left(\frac{-(x-u)^2}{2\sigma_z^2}\right)dx \\ &= \int_{2\mu-T_z}^{\infty} P(x)dx = \int_{2\mu-T_z}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_z} \exp\left(\frac{-(x-u)^2}{2\sigma_z^2}\right)dx \end{aligned} \quad . \tag{5.15}$$

However, unlike the case of false positive probabilities, there are many more variables to consider before analyzing the probability of a false negative. This is because false negative probabilities are highly dependent on both the watermark detector and the embedder, as well as what happens to an image between the time a watermark is embedded and the time it is detected. A watermark might be severely distorted by

low/high pass filtering, lossy compression, or any of a wide variety of processes, thus increasing the probability of a false negative. Such attacks by some adversary will significantly affect the variance and, especially, the mean of the detector response Experiments show the estimated mean and variance of the detector response given by Equation 5.7 and 5.9, especially the estimated mean, are different from the actual experimental values due to the embedding algorithm or the distortion introduced to the cover image. Figure 5.7 shows the watermark false negative probabilities for "Lena" as a function of detection thresholds computed by predicted variance. Observation shows the false negative probability based on the predicted variance is not quite accurate, it is getting closer to the experimental results when the false negative probability is in between 10^{-10} and 10^{-25}. Similarly, an example of false negative probabilities measurement for host image "Barbara" is given in Figure 5.8.

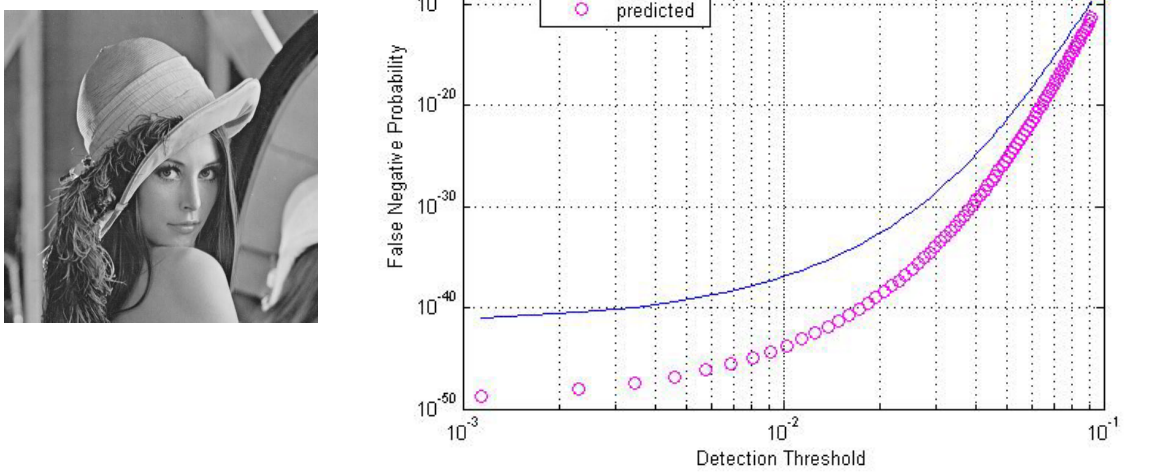


Figure 5.7. Watermark false negative probabilities for "Lena" as a function of detection thresholds. The curve (by circle marker) computed from predicted σ_z is alienated from the curve (in solid line) measured with actual experimental σ_z especially for low threshold values.

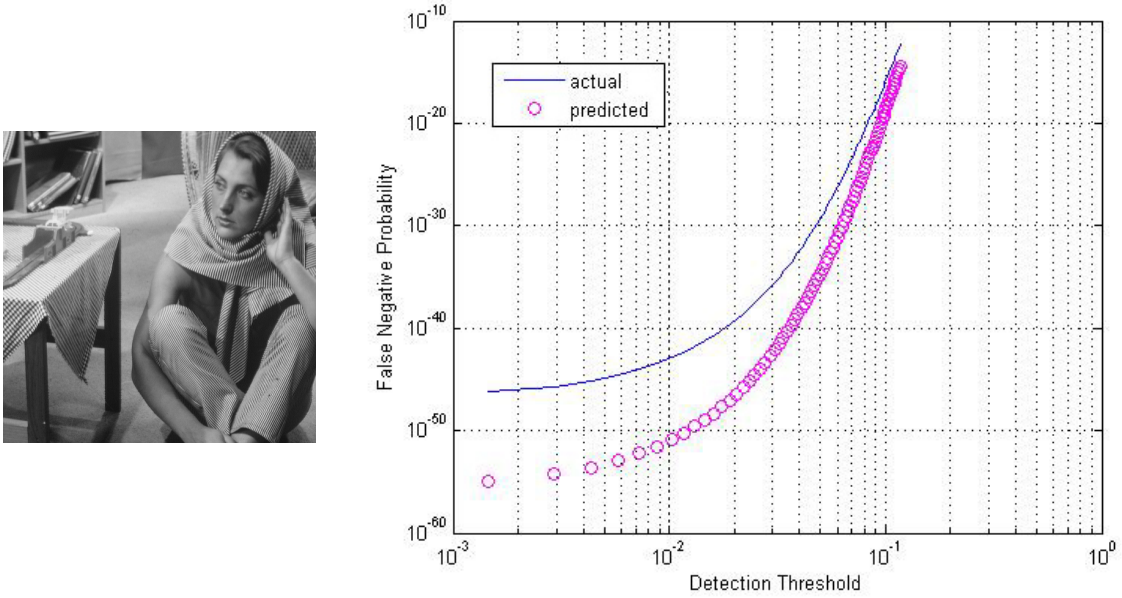


Figure 5.8. Watermark false negative probabilities for "Barbara" as a function of detection thresholds. Both curves are depending on the experimental mean value of detector responses. The curve (by circle marker) measured with predicted σ_z is alienated from the curve (in solid line) measured with experimental σ_z especially for low threshold values.

We can see there is a trade-off between the probabilities of false positive and the probabilities of false negative. As the threshold increases, the probabilities of false positive decrease and the probabilities of false negative rise. The performance of the system can be interpreted by considering both probabilities at once using a *receiver operating characteristic (ROC) curve*, plotting the false positive probability (the *x*-axis) against the false negative probability (the *y*-axis) as a function of threshold (see Figure 5.9 for an ROC curve for "Lena"). The shape of the curve produced with predicted σ_z is closely matched the one produced by actual σ_z in experiments. We obtained the ROCs for other cover images using the same approach; the behavior of the curves varies slightly from one image to the next. The ROC curve of curvelet based watermarking system applied to "Barbara" is demonstrated in Figure 5.10.

In a real watermarking system, we wish to reduce both the probability of false positive

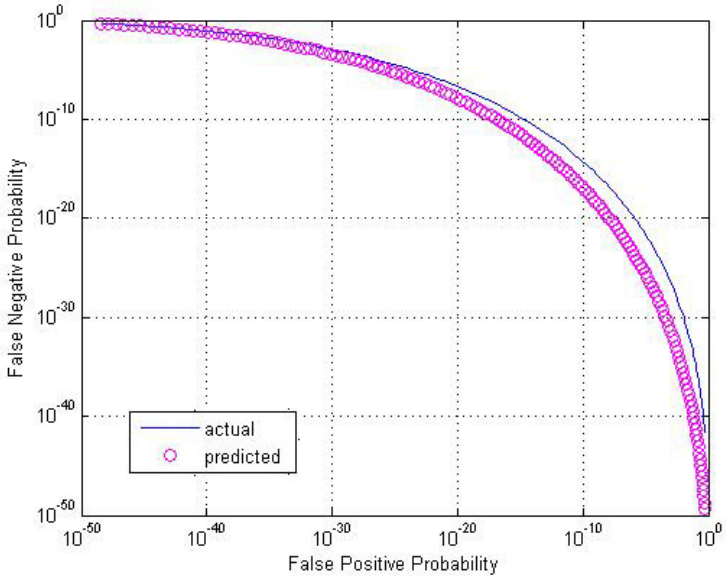


Figure 5.9. ROC curve of curvelet based watermarking system applied to "Lena"

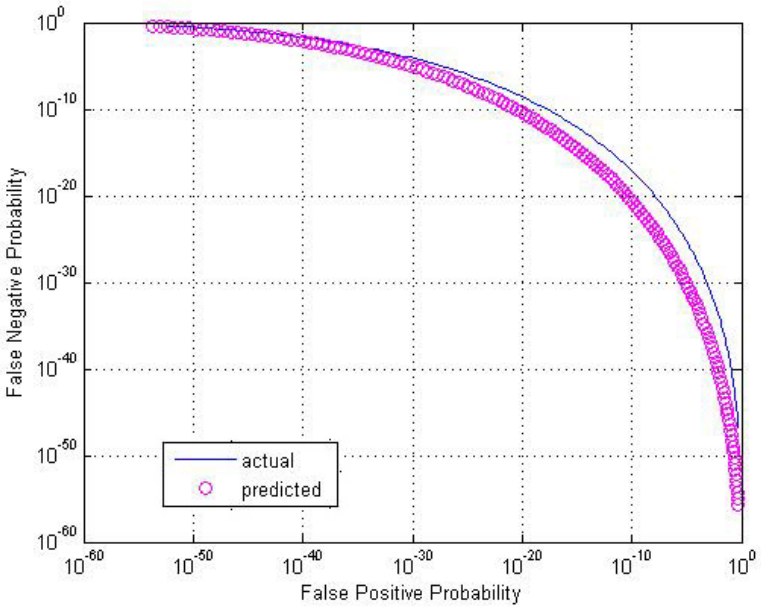


Figure 5.10. ROC curve of curvelet based watermarking system applied to "Barbara"

and the probability of false negative. ROC curve provides a graphical representation for the performance of marking system, to which we can refer for choosing an appropriate mid point between two probabilities. After examining the ROC curves and the curve of

the false positive probabilities as a function of the detection thresholds for all tested images, we determined that selecting a false positive rate between 10^{-7} and 10^{-10} makes the false negative rate under no attacks fall between 10^{-10} and 10^{-30} which is pretty safe for our watermarking system.

5.4 Experimental results

We applied the watermarking embedding and detection algorithm presented in Chapter 2 to the standard image "Lena". In order to verify the validity of the threshold, we computed the variance of z according to Equation 5.9. The estimated value is comparable to the actual one determined experimentally. The experiment was performed against 1000 watermarks; each outputs a correlation value with the signals from host data. The actual variance is computed among the 1000 values. As Table 5.1 shows, the theoretically estimated values are very close to the experimental ones when $W \neq \hat{W}$. N is the number of the selected curvelet coefficients. Table 5.2 is comparing the theoretically estimated σ_z and the actual experimental σ_z when the cover image Lena is subjected to a variety of attacks. Observations show the estimation for the variance of z is accurate. Figure 5.11 illustrates the effect of JND modeling on the false positive probability and corresponding ROC.

The prediction of the false positive error is accurate while the estimation of the false negative error is hardly true because a lot more variables are involved and affect the actual values of u and σ_z. In Figure 5.11(b), although the curve generated with estimated σ_z is not totally overlapped with the curve produced by actual experimental σ_z, the shapes

Table 5.1. A list of estimated variance of z in comparison with experimental ones.

Data	N	Estimated σ_z	Experimented σ_z
Lena	393216	0.011466	0.011506
Lena	301056	0.014594	0.013839
Lena	202752	0.019915	0.020120
Lena	61440	0.045906	0.046651
Barbara	374784	0.015338	0.015499
Barbara	251904	0.020934	0.02075
Barbara	73728	0.04733	0.047625
Boat	301056	0.018585	0.018555
Boat	196608	0.024053	0.024579
Boat	116736	0.032565	0.033405
Goldhill	356352	0.012386	0.012428
Goldhill	258048	0.015618	0.015465
Goldhill	184320	0.019674	0.019723
Airplane	270336	0.019746	0.019934
Airplane	215040	0.023953	0.023722
Airplane	104448	0.036544	0.036019

Table 5.2. A list of estimated variance and corresponding experimental ones when cover image Lena is tested and subjected to a variety of attacks (N=301056).

Data	Estimated σ_z	Experimented σ_z
JPEG QF=90	0.016837	0.016680
JPEG QF=50	0.016759	0.016506
JPEG QF=20	0.016684	0.016380
JPEG QF=5	0.016433	0.016646
Gaussian noise v=0.05	0.034814	0.035906
Gaussian noise v=0.1	0.043411	0.044830
Cropped 50%	0.013438	0.013137
Cropped 75%	0.008693	0.008388
Gaussian Blur (5x5)	0.015684	0.015549
Low pass filtering (3x3)	0.013794	0.013775
Histogram Equalization	0.018274	0.018109
Gamma correction	0.018190	0.017986
Contrast adjustment	0.018449	0.018370
Sharpening	0.025433	0.024838
Rotation by 30º	0.015313	0.015247
Horizontally Sheared 0.3	0.015912	0.015898

of the two curves are quite similar, thus to which we will refer and find a mid point that reduce both probabilities as possible. Observation from Figure 5.11(c)(d) shows the

graphs generated with JND modeling is almost identical to those graphs without JND modeling which indicates our proposed watermarking under JND modeling does not affect the correctness of estimation of false positive probability and ROC.

We performed the statistical analysis over detector responses and selected the detection threshold according to analysis results. According to ROC curve, we selected the false

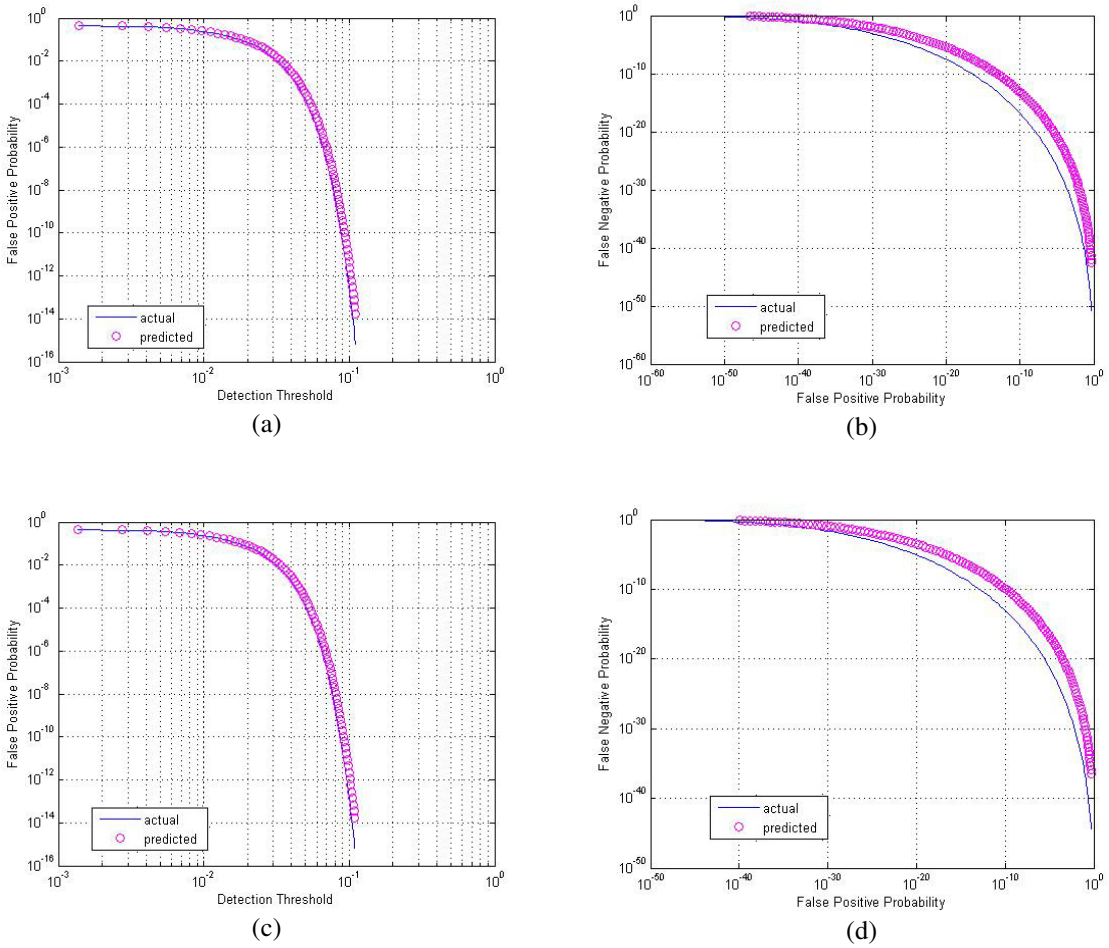


Figure 5.11. Effect of JND modeling on watermark false positive probabilities and ROC (a) Watermark false positive probabilities of detection thresholds when the curvelet based algorithm is applied to "Lena" without JND adjustment, edge strength is determined as 100. (b) Corresponding ROC curve of (a). (c) Watermark false positive probabilities of varying detection thresholds when the embedding algorithm is applied to "Lena" with JND adjustment, edge strength is also 100. (d) Corresponding ROC curve of (c)

positive $P(z>T_z) \leq 10^{-7}$ where the false negative $P(z<T_z)$ is between 10^{-20} and 10^{-30}. The detector responses of fake watermarks is Gaussian distributed, the variance of the distribution can be estimated according to Equation 5.9. Thus, the parameter k in Equation 5.11 is determined to be 5.2 to satisfy the requirement of false positive.

5.5 Detector Performance under Attacks

We performed experiments with a variety of attacks. In Figure 5.12, the response of the detector of the embedded watermark is plotted against JPEG compression quality factor from 5 to 100, along with the detection threshold and the highest detector response when the 999 fake watermarks are tested. Experiments results show the detector response with quality factor $\geq$ 10% are all higher than the threshold. Thus the embedded watermark survives the severe image compression when the quality factor is 10% or up and the false

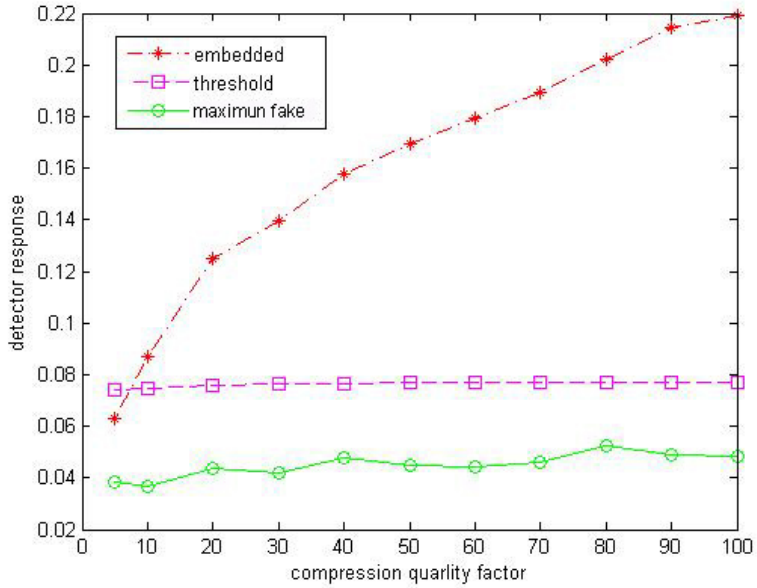


Figure 5.12. Detector response of the embedded watermark is plotted against JPEG compression with increasing quality factor (from 5% to 100%), along with the detection threshold and the maximum response among 999 fake watermarks

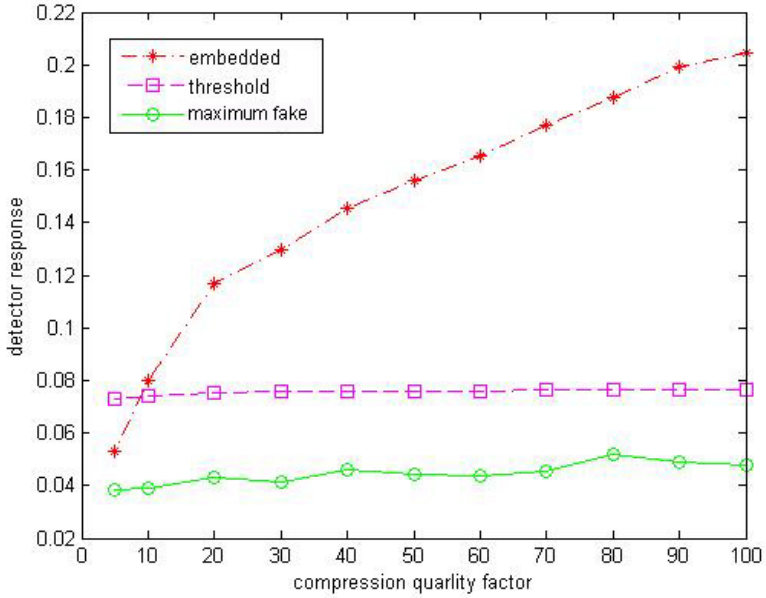


Figure 5.13. Detector response of the watermark embedding with JND adjustment is plotted against JPEG compression with increasing quality factor, along with the detection threshold and the maximum response among 999 fake watermarks

positive probability $\leq 10^{-7}$. The detector response of the embedded watermark under JPEG compression with the quality factor 5% is higher than the max detector response of 999 fake watermarks but lower than the threshold. The detector responses to fake watermarks are all lower than the threshold. Figure 5.13 shows that the embedded watermark adjusted with JND modeling provides high tolerance to compression attack as well.

We also investigated the robustness of the detector against cropping attacks. Figure 5.14 shows the cropped "Lena" with increasing cropping percentage. Figure 5.15 and Figure 5.16 plot the detector response of the embedded watermark with and without JND adjustment respectively against cropping (from 0% to 75%), along with the detection threshold and maximum response among 999 fake watermarks. It can be seen that the detector response of embedded watermark is higher than the threshold corresponding to

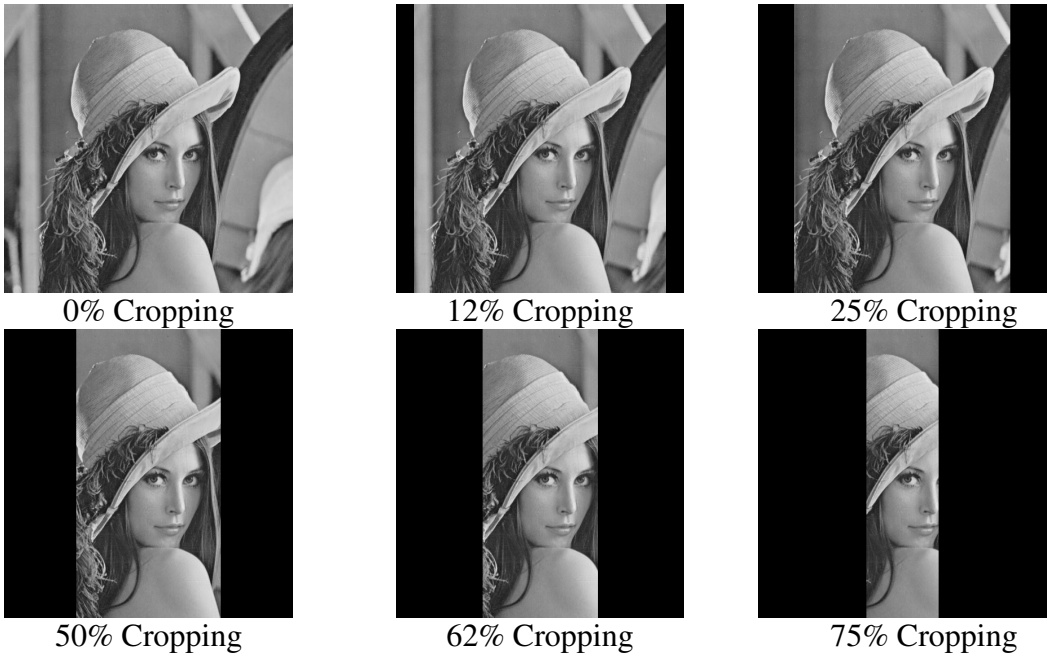


Figure 5.14. Cropped "Lena" on both sides with increasing cropping percentage

$P(z>T_z) \leq 10^{-7}$ when the cover image is less than 75% cropped. The watermarking system with JND modeling provides almost identical robustness against cropping attack in comparison with the system without JND modeling.

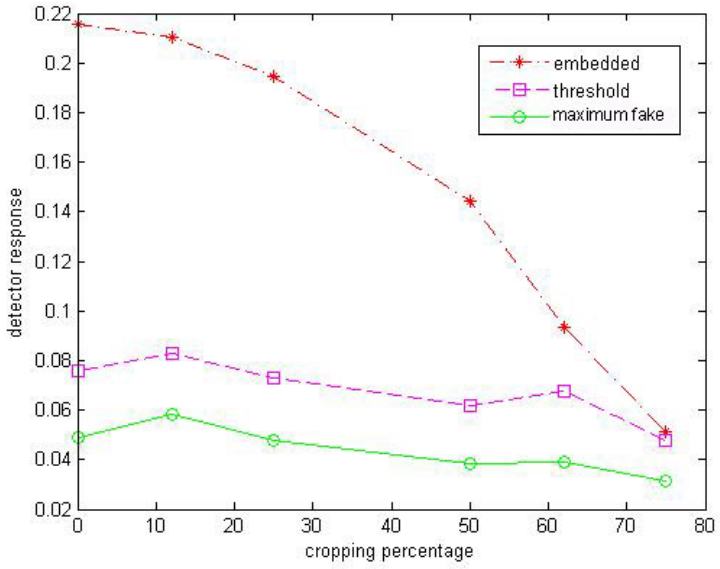


Figure 5.15. Detector response of embedded watermark is plotted against cropping attack (from 0% to 75%) along with the detection threshold and the second highest response.

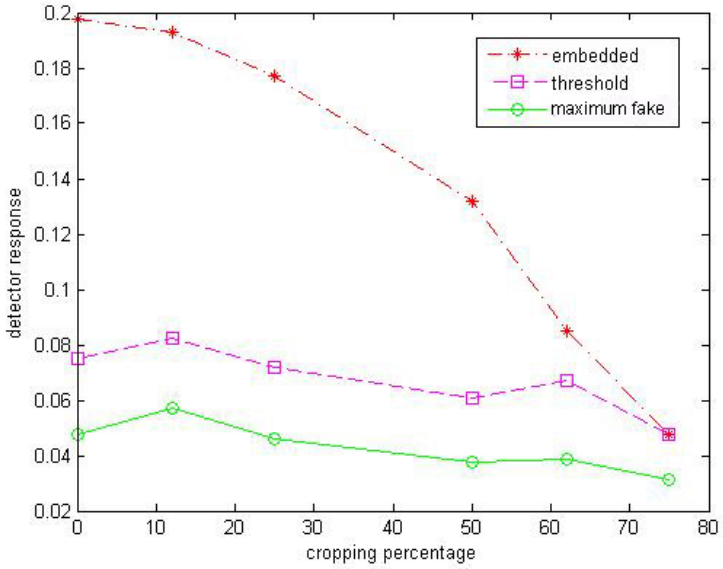


Figure 5.16. Detector responses of embedded watermark with JND adjustment is plotted against cropping attack (0% to 75%) along with the detection threshold and the second highest response.

Finally, in Table 5.2 the detection thresholds are computed for a wide range of attacks. We distinguish between the cases whether or not JND modeling is applied. It should be noted that geometrical distortions are detected and corrected using the scheme proposed in Chapter 4, and the detector response is computed after the geometric distorted image is restored to its original state. All detection thresholds listed in Table 5.2 are computed based on false positive probability $P(z>T_z) \leq 10^{-7}$. We can see all detector responses to fake watermarks are lower than the computed thresholds in both cases. The detector responses of embedded watermarks wi/wo JND modeling are higher than the thresholds when the following attacks are applied: JPEG(QF=50, QF=20), adding Gaussian noise (m=0, v=0.05), cropping (50%, 75%), Gaussian blur (window 5x5), low pass filtering (window 3x3), histogram equalization, Gamma correction, contrast adjustment, sharpening, rotation, shearing and translation. The detector responses of embedded watermarks are below the thresholds in both cases when JEPG compression (QF=5) is

Table 5.3. The detection threshold is computed for a wide range of attacks

Attack	Without JND			With JND		
	Response of embedded watermark	Maximum fake response	Threshold	Response of embedded watermark	Maximum fake response	Threshold
No attack	0.2156	0.0489	0.0759	0.1976	0.0478	0.0749
JPEG QF=50	0.1692	0.0447	0.0757	0.1536	0.0434	0.0748
JPEG QF=20	0.1252	0.0440	0.0750	0.1140	0.0418	0.0740
JPEG QF=5	0.0628	0.0385	0.0730	0.0521	0.0385	0.0720
Gaussian noise v=0.05	0.2364	0.1315	0.1885	0.2204	0.1309	0.1883
Gaussian noise v=0.1	0.2426	0.1604	0.2360	0.2279	0.1598	0.2358
Cropped 50%	0.1442	0.0384	0.0618	0.1318	0.0375	0.0610
Cropped 75%	0.0511	0.0317	0.0478	0.0479	0.0314	0.0478
Gaussian Blur (5x5)	0.178330	0.0423	0.0689	0.1635	0.0413	0.0680
Low pass filtering (3x3)	0.1073	0.0344	0.0583	0.0991	0.0343	0.0578
Histogram Equalization	0.3186	0.0727	0.1161	0.2935	0.0718	0.1149
Gamma correction	0.2115	0.0493	0.0756	0.1949	0.0482	0.0748
Contrast adjustment	0.2555	0.0597	0.0937	0.2340	0.0584	0.0926
Sharpening	0.5343	0.1066	0.1466	0.4891	0.1039	0.1446
Rotation by 30º	0.1396	0.0438	0.0715	0.1280	0.0436	0.0709
Horizontally Sheared 0.3	0.1384	0.0410	0.0689	0.1274	0.0407	0.0682
Translation	0.2156	0.0489	0.0759	0.1976	0.0478	0.0749

applied to cover image. When Gaussian noise (v=0.1) is added to the cover image, the detector response of embedded watermark without JND modeling is slightly higher than

the threshold while the one with JND modeling is slightly lower than the threshold. Therefore, we conclude our curvelet-based watermarking algorithm is robust against a wide range of attacks with respect to false positive rate 10^{-7} and the proposed JND modeling with the purpose to improve the quality of cover image by controlling embedding strength does not affect the robustness of our watermarking algorithm.

5.6 Conclusion

In this chapter, a statistical technique is proposed to select the threshold for watermark detection. This technique is based on the statistical analysis over the host signals and embedding schemes. The technique is also used for the marking algorithm adjusted by JND modeling. Experiments show the scheme is able to keep both probability of false positive and the probability of false negative as low as possible and is generally robust against a wide range of image attacks.

Chapter 6

Image/Video Quality Assessment Using M-SVD

Objective image/video quality measurement is a challenging problem in a variety of image/video processing applications ranging from lossy compression to printing. There is an increasing need to develop an objective quality measure that may predict the perceived image/video quality automatically. Moreover, an ideal image/video quality measure should be able to describe the amount of distortion as well as the distribution of error. Undoubtedly, there is a need for an objective measure that provides more information than a single numerical value. Very few multi-dimensional measures exist in the relevant literature today. In this chapter, we present a new quality measure, M-SVD, which can express the quality of distorted images either numerically or graphically. Based on the Singular Value Decomposition (SVD), it consistently measures the distortion both across different distortion types and within a given distortion type at different distortion levels. This approach first is applied to grayscale image and then extended it to color image and video quality measure as well. Our experiments show the graphical measure displays the amount of distortion as well as the distribution of error in all images or in all the frames of video sequence, while the numerical measure has a good correlation with perceived image or video quality, outperforming PSNR and other objective measures. Finally, M-SVD is applied to the watermarked images generated by our block-based watermarking algorithm in curvelet domain. The graphical measure describes the distribution of modified blocks and the amount of error introduced to such

blocks. The numerical measure, expressed as a single value, indicates the overall visual quality of watermarked images. Evaluation results show the quality of the watermarked images is improved when JND adjustment is used for embedding.

6.1 Introduction

Quality measures can be classified into two categories: subjective and objective [64]. Subjective evaluation is cumbersome, as human observers can be influenced by several critical factors such as the environmental conditions, motivation and mood. Objective evaluation is considerably more stable but may not correlate well with the Human Visual System [65,66,67,68,69,70]. Objective measures in the literature can be classified into three types according to the type of information needed during quality assessment: Measures that require both the original image/video and the distorted image/video are called “full-reference” or “non-blind” methods [71,72], measures that do not require the original image/video are called “no-reference” or “blind” methods [73,74,75,76,77], and measures that require both the distorted image and partial information about the original image/video are called “reduced-reference” methods [78]. Currently, the most commonly used full-reference objective evaluation tools, the Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR), are very unreliable, with poor correlation with the HVS. Many efforts have been made to design image/video quality assessment models that incorporate perceptual quality measures by considering the characteristics of HVS. In spite of their complicated algorithms, the more recent HVS-based objective measures do not appear to be superior to the simple pixel-based MSE and PSNR.

Wang and Bovik [34] presents a new full-reference numerical measure for comparing grayscale images, called the Universal Image Quality Index (UIQI), which is described in Equation 1.27. The authors claim that it performs significantly better than the widely used MSE distortion metric in evaluation of perceived distortion. The dynamic range of UIQI is [-1,1], with the best value achieved when the two images in comparison are identical (UIQI=1). As described in the paper [34], this index models any distortion as a combination of three different factors: loss of correlation, mean distortion and variance distortion. The index is computed using a sliding window approach with a window size of 8x8, leading to a quality map of the image. The overall quality index is the average of all the Q values in the quality map. Q produces unstable results when either $\left(\mu_x^2 + \mu_y^2\right)$ or $\sigma_x^2 + \sigma_y^2$ (see Equation 1.27) is very close to zero. To avoid this problem, the measure has been generalized to the Structural Similarity Index (SSIM) [79]:

$$SSIM = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{6.1}$$

Q is a special case of SSIM that can be derived by setting C_1 and C_2 to 0. As in the case of Q, the overall image quality MSSIM is obtained by computing the average of SSIM values over all windows.

6.2 Quality Measures Using M-SVD

Every real matrix A can be decomposed into a product of 3 matrices $A = USV^T$, where U and V are orthogonal matrices, $U^TU = I$, $V^TV = I$, and $S = diag\ (s1, s2, ...)$. The diagonal entries of S are called the singular values of A, the columns of U are called the left

singular vectors of *A*, and the columns of *V* are called the right singular vectors of *A*. The Singular Value Decomposition (SVD) [80] has a variety of applications in scientific computing, signal processing, automatic control, and many other areas. SVD is one of the most useful tools of linear algebra with several applications to multimedia including image compression [81] and watermarking [82,83,84].

6.2.1 M-SVD for Gray Scale Images

The M-SVD image quality measure we propose is a bivariate measure that computes the distance between the singular values of an $n \times n$ block of the original image and the singular values of the corresponding block in distorted image:

$$D_k = Sqrt\left[\sum_{i=1}^{n}\left(s_i - \hat{s}_i\right)^2\right] \tag{6.2}$$

where s_i and $\hat{s}_i$ are the singular values of the original block and distorted block. If the image size is $r \times c$, we have $(r/n) \times (c/n)$ blocks. The set of distances, when displayed in a graph, represents a "distortion map".

The numerical measure is derived from the graphical measure. It computes the global error expressed as a single numerical value depending on the distortion map:

$$M\text{-}SVD = \frac{\sum_{k=1}^{(r/n)\times(c/n)} \left|D_k - D_{mid}\right|}{(r/n)\times(c/n)} \tag{6.3}$$

where D_{mid} represents the mid point of the sorted D_i, $r \times c$ is the image size, and n is the block size.

In experiments, the measure is applied to distorted version of 512x512 grayscale Lena image. The six distortion types and parameters corresponding to five distortion levels are listed in Table 6.1. Figure 6.1 presents the distortion maps, which illustrate the amount of distortion, the type of distortion, and the distribution of error. These are obtained as grayscale images by mapping the D_k values to the range [0,255]. We choose 8 as block size so the size of the distortion map is 64×64. The darker pixel values indicate small

Table 6.1. Distortion types and levels applied to tested image.

Type \ Level	Level 1	Level 2	Level 3	Level 4	Level 5
JPEG	10:1	20:1	30:1	40:1	50:1
JPEG2000	10:1	20:1	30:1	40:1	50:1
Gaussian blur	1	2	3	4	5
Gaussian noise	3	6	9	12	15
Sharpening	10	20	30	40	50
DC-shifting	4	8	12	16	20

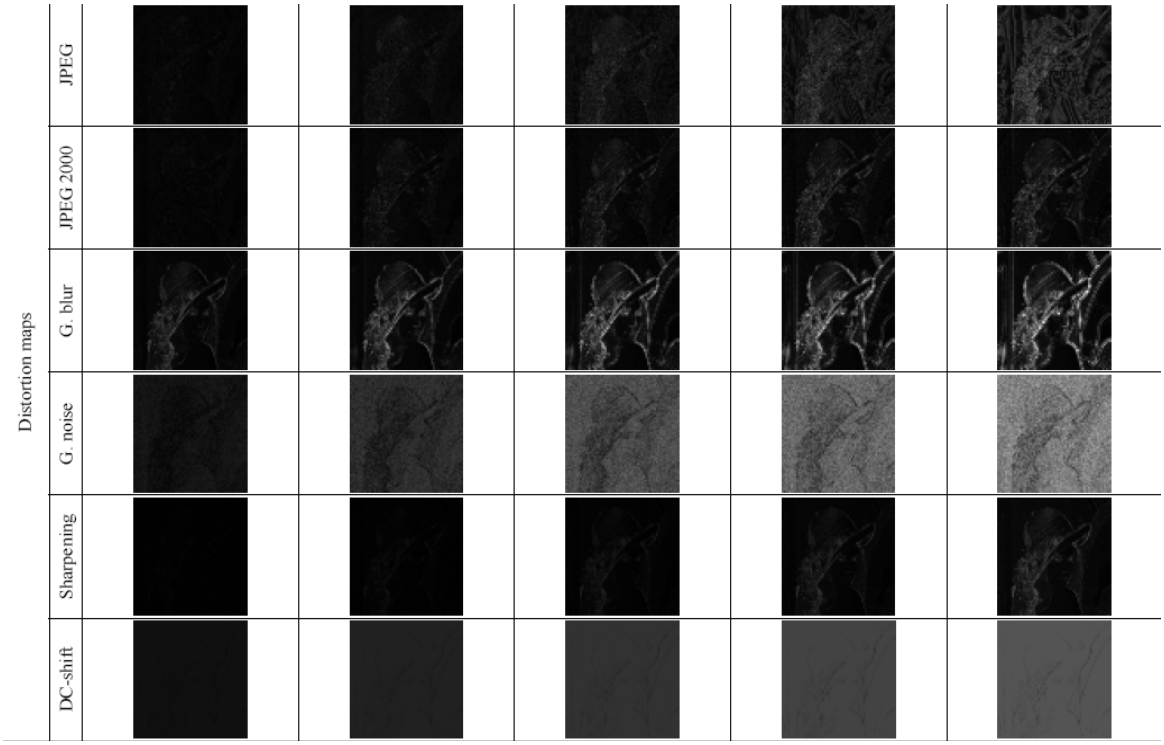


Figure 6.1. The corresponding distortion maps of the distorted images defined in Table 6.1.

distortions, and the lighter pixel values indicate larger distortions. The global error expressed as a single numerical value is presented in Equation 6.3. High quality printouts of each set of distorted images were subjectively evaluated and a score representing the quality was assigned to each image. Pearson correlation between the single numerical value with subjective evaluation is computed. Analysis shows the performance of M-SVD is much better than PSNR, UIQI and MSSIM. Table 6.2 and 6.3 display the correlation coefficients between subjective evaluation and M-SVD in comparison with other objective models across different distortion types in different distortion levels.

Table. 6.2. Correlation coefficients between subjective evaluation and M-SVD in comparison with other objective models across each distortion type.

Distortion Type\Measure	PSNR	UIQI	MSSIM	M-SVD
JPEG	0.974	0.904	0.928	0.977
JPEG2000	0.949	0.688	0.801	0.952
Gaussian blur	0.816	0.917	0.906	0.929
Gaussian noise	0.901	0.984	0.987	0.975
Sharpening	0.955	0.908	0.947	0.937
DC-shifting	0.914	0.637	0.643	0.718

Table. 6.3. Correlation coefficients between subjective evaluation and M-SVD in comparison with other objective models across each distortion level.

Distortion Level\Measure	PSNR	UIQI	MSSIM	M-SVD
1	0.808	0.744	0.781	0.890
2	0.751	0.808	0.853	0.954
3	0.529	0.885	0.910	0.962
4	0.369	0.914	0.929	0.958
5	0.439	0.940	0.947	0.924

Table 6.4 Overall correlation of four objective measures with subjective evaluation

PSNR	UIQI	MSSIM	M-SVD
0.697	0.839	0.833	0.928

Table 6.4 displays the overall performance of the measures using the correlation between subjective evaluation using DMOS (Difference Mean Opinion Score) and objective prediction. M-SVD outperforms all three measures; in particular, the correlation is improved by approximately 10% relative to the state-of-art metrics UIQI and MSSIM.

6.2.2 M-SVD Extended for Color Images and Video Quality Assessment

We extended this approach to evaluate the quality of color images. First, the original and distorted images are resampled to 4:4:4, Y, Cb, Cr format. The block-based SVD is computed on each Y, Cb, Cr layer respectively and combined to obtain the global error measure using a weighted summation as follows. Let $M\text{-}SVD^{Y}$, $M\text{-}SVD^{Cb}$ and $M\text{-}SVD^{Cr}$ be the error measures of *Y*, *Cb*, *Cr* layers. The combined quality error index is:

$$M\text{-}SVD = 0.8 \cdot M\text{-}SVD^{Y} + 0.1 \cdot M\text{-}SVD^{Cb} + 0.1 \cdot M\text{-}SVD^{Cr} \quad (6.4)$$

where the weights *0.8*, *0.1* and *0.1* are obtained experimentally where the output M-SVD has the best correlation with subjective evaluation (DMOS). Therefore, the luminance component Y makes the major contribution.

An improved version of M-SVD taking in account human visual system is developed for full-reference (FR) video quality assessment. The original and processed video sequences are resample to 4:4:4, *Y, Cb, Cr* format. A spatial-temporal-luminance alignment is included into the system to normalize the input sequences. The block-based SVD is computed on the layer *Y, Cb, Cr* in both the original frames and the corresponding processed frames. Then, the local error measure as Equation 6.2 computes the distance between the singular values of the original frame block and the singular values of the

distorted frame block. It is well known that human eyes are highly-sensitive to high contrast areas, especially around edges in video. We use Sobel edge function [94] to compute edges in the luminance layer. The resulting edge map is a binary image containing 1 where edges are detected and 0 elsewhere. This output binary image allows us to assign an edge index to each block within a frame. Those blocks in a frame with rich edges receive higher edge index as follows:

$$B_k = \frac{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}}{\frac{1}{r \times c}\sum_{i=1}^{r}\sum_{j=1}^{c} b_{ij}} \tag{6.5}$$

and

$$B_k = \begin{cases} B_k & if \quad B_k \geq \tau \\ 1 & if \quad B_k < \tau \end{cases} \tag{6.6}$$

where B_k denotes the edge index of k^{th} block, *rxc* denotes the frame dimension, *n* is the block size, and τ is a threshold. The local error measure with edge detection is given by:

$$D_k = B_k \cdot Sqrt\left[\sum_{i=1}^{n}\left(s_i - \hat{s}_i\right)^2\right] \tag{6.7}$$

The frame-level error measure M-SVD$_j$ of jth frame is obtained as a weighted summation of $M\text{-}SVD_j^{Y}$, $M\text{-}SVD_j^{Cb}$ and $M\text{-}SVD_j^{Cr}$ defined in Equation 6.3 and 6.4. The frame-level error is expressed as a single numerical value based on local error measures. Finally the overall quality of the entire sequence is defined as:

$$M\text{-}SVD = \frac{\sum_{1}^{n} M - SVD_j}{n} \tag{6.8}$$

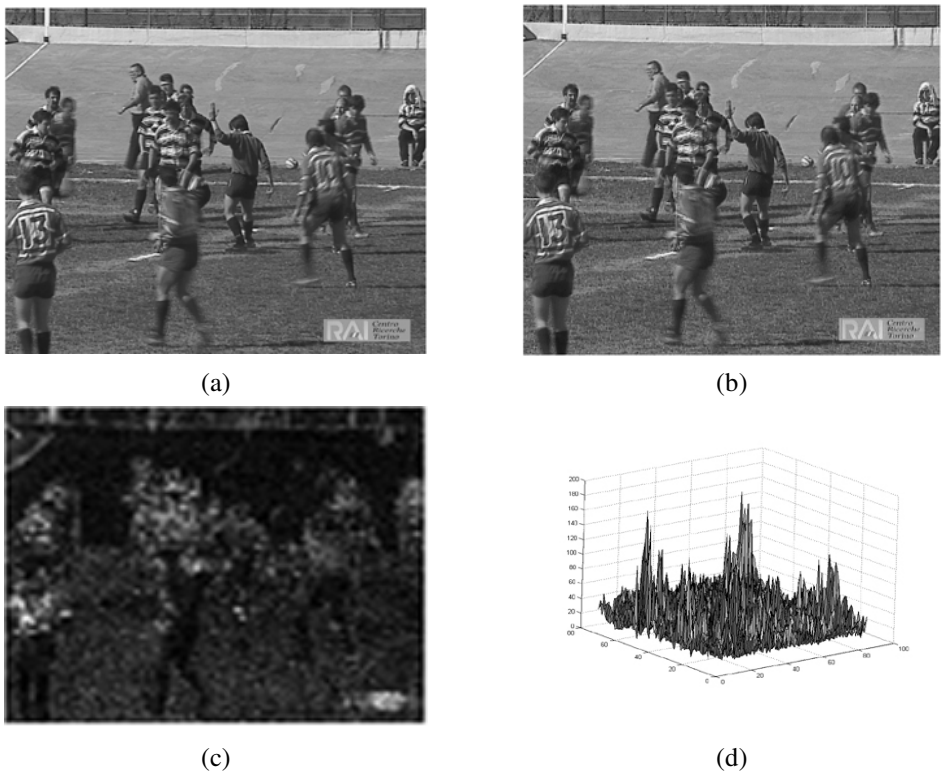
where *n* is the number of frames in a video sequence. This leads to a quality measure that is equal to the average *M-SVD* error measure of all frames.

This measure is tested on the Video Quality Experts Group (VQEG) phase I FR-TV test data set [86] for evaluating the correlation between a candidate objective measurement and the subjective results (DMOS). Four metrics are used in the evaluation of objective results. Metric 1 is the correlation coefficient between objective and subjective scores after variance-weighted regression analysis, including a test of significance of the difference. Metric 2 is the correlation coefficient between objective and subjective scores after non-linear regression analysis. The first two metrics assess the accuracy of an objective model. Metric 3 is the Spearman rank-order correlation coefficient between the objective and subjective scores. This correlation method only assumes a monotonic relationship between the two scores. A higher correlation coefficient output obtained use above three metrics means the model's predictions are more consistent. Metric 4 is the ratio of "outlier-points" to total number of points. The model's prediction consistency can be measured by the number of outlier points (defined as having an error greater than some threshold) as a fraction of the total number of points. A smaller outlier fraction means the model's predictions are more consistent.

Our experiments show the graphical measure displays the amount of distortion as well as the distribution of error in all frames of the video sequence while the numerical measure has a good correlation with perceived video quality that outperforms PSNR and other objective measures by a clear margin. Figure 6.2 shows the distortion maps as 2-dimensional and 3-dimensional graphs that provide the amount of the error as well as its

distribution in a frame. Figure 6.3 shows a plot of the error series of all frames contained in a video sequence. Table 6.5 presents the performance comparison of video quality assessment models on VQEG Phase I FR-TV Test Data Set (all test video sequences included). P1-P9 [86] are nine different proponent models submitted to VQEG for evaluation. P0 (PSNR) is included by VQEG as a reference objective model. SSIM (Structural Similarity Index) is a recent presented model in [79]. It can be observed that the proposed model M-SVD with edge detection outperforms all other measures by a clear margin.

Figure 6.4 gives the non-linear regression analysis (metric 2) of the subjective/objective scores on all video sequences in the VQEG Phase I test given by PSNR and M-SVD.



(a) (b)

(c) (d)

Figure 6.2. The distortion maps as a 2 and 3-dimensional graphs for one frame in luminous layer. (a) original frame size of 568x680 (b) processed frame size of 568x680 (c) 2-dimensional distortion map size of 71x85 (d) 3-dimensional distortion map.

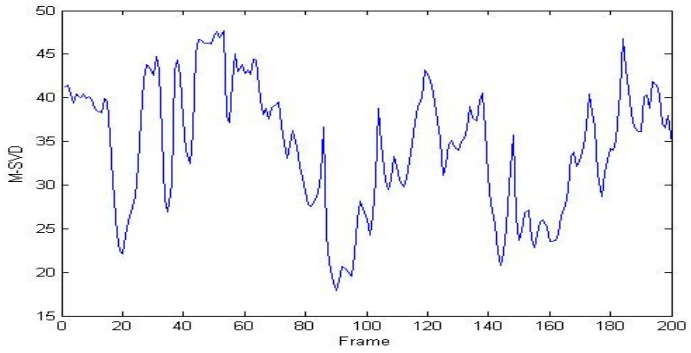


Figure 6.3. The error series of all frames in one distorted video sequence.

Table. 6.5. Performance comparison of video quality assessment models on VQEG Phase I Test Data Set (all test video sequences included).

Model	Metric 1	Metric 2	Metric 3	Metric 4
P0 (PSNR)	0.804	0.779	0.786	0.678
P1 (CPqD)	0.777	0.794	0.781	0.650
P2 (T/S)	0.792	0.805	0.792	0.656
P3 (NHK)	0.726	0.751	0.718	0.725
P4 (KDD)	0.622	0.624	0.645	0.703
P5 (EPFL)	0.778	0.777	0.784	0.611
P6 (TAPESTRIES)	0.277	0.310	0.248	0.844
P7 (NASA)	0.792	0.770	0.786	0.636
P8 (KPN)	0.845	0.827	0.803	0.578
P9 (NTIA)	0.781	0.782	0.775	0.711
SSIM	0.864	0.849	0.812	0.578
M-SVD/Edge Detection	0.893	0.877	0.799	0.486

Each shows the scatter plot of subjective and objective scores and the fitted curve. 160 video sequences are tested and each is represented as a point in the graph. The vertical axis indicates the subject measurement denoted by DMOS (subjective evaluation) while the horizontal axis is the objective measure output. Non-linear regression analysis results indicate the performance of M-SVD/Edge detection is better than that of PSNR.

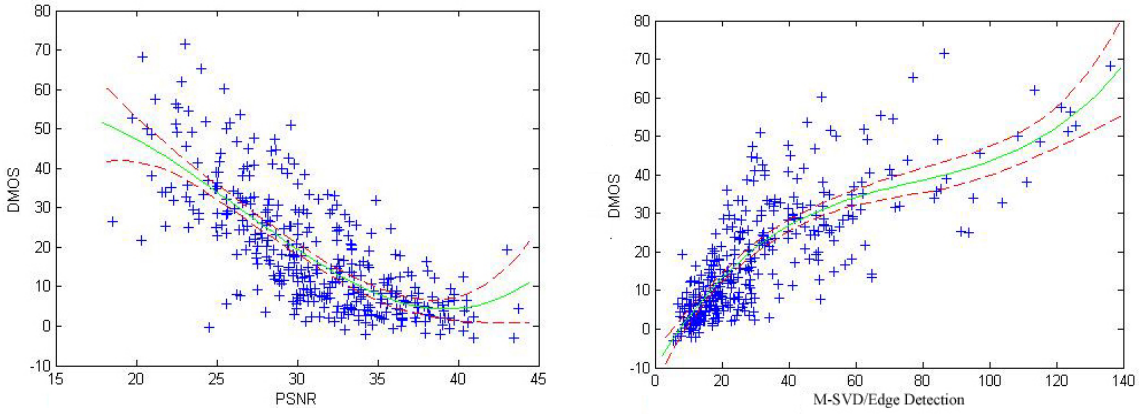


Figure. 6.4. The scatter plot comparison of objective models on all video sequences in the VQEG Phase I test dataset given by PSNR and M-SVD with Edge Detection

6.3 Evaluating the Visual Quality of Watermarked Images

We apply M-SVD defined in Equation 6.3 and Equation 6.4 to seven gray scale 512×512 images (Lena, Barbara, Boat, Goldhill, Airplane, Peppers and Baboon). These images are used with the curvelet based watermarking algorithms presented in Chapter 2. Each test image is partitioned into 64×64 blocks and curvelet transform is computed for each block. We embedded the watermark into selected scale and curvelet coefficients, all blocks are modified. Table 6.6 demonstrated the watermarked images together with evaluated 2D and 3D distortion maps. The overall quality of the image is estimated using both PSNR and M-SVD. The graphical measure is represented by distortion map showing the amount of the distortion as well as the distribution of the error.

Table 6.7 shows the results of applying M-SVD to Lena that has been watermarked with JND adjustment, and the edge strength threshold is varying between 0 and 600. Observations show the estimated value of M-SVD is reduced when the threshold is increased where fewer blocks are modified. All M-SVD global errors become smaller

when JND modeling is applied and the distortion peaks appeared in the 3-D distortion map are also reduced. The overall quality of watermarked image is measured by both PSNR and M-SVD. Evaluation results show the quality of the watermarked images is improved when JND adjustment is used for watermark embedding.

Table 6.6. Watermarked images together with evaluated 2D and 3D distortion maps

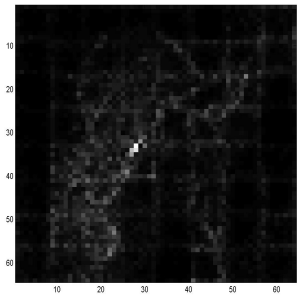
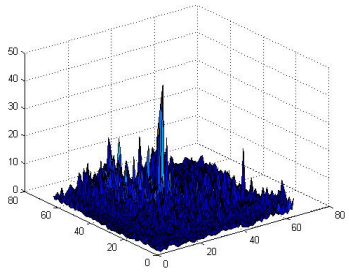
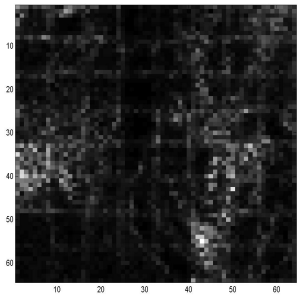
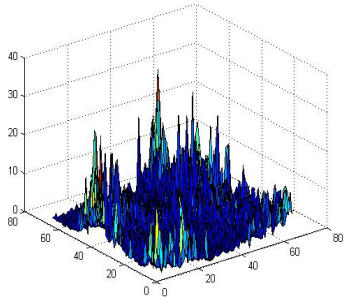
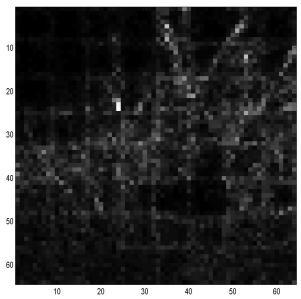
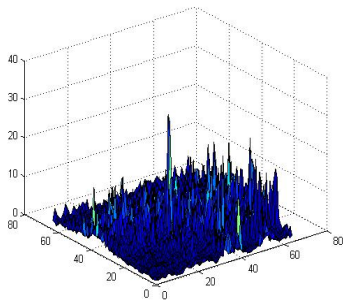
Watermarked Image	2D distortion map	3D distortion map
 PSNR=38.91 M-SVD=1.91		
 PSNR=36.67 M-SVD=2.44		
 PSNR= 37.01 M-SVD= 2.03		

Table 6.6. (Continue)

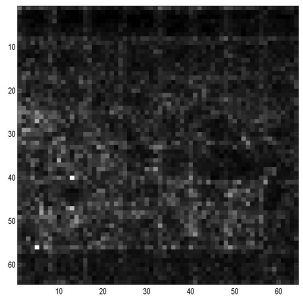
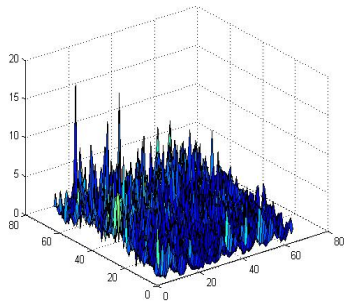
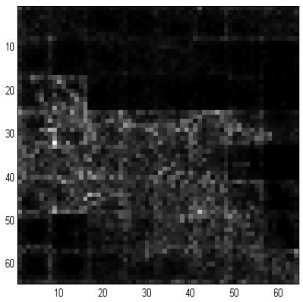
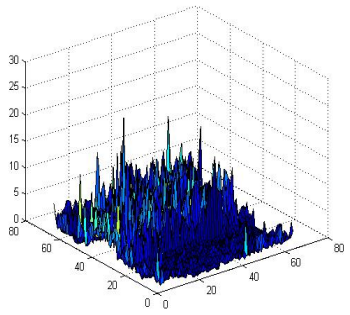
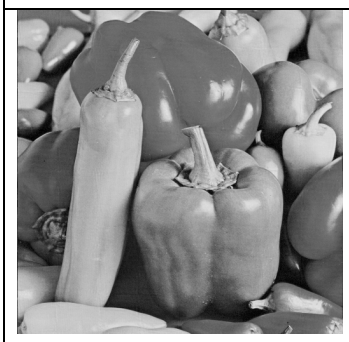
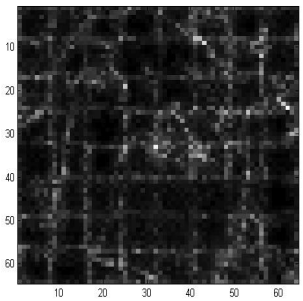
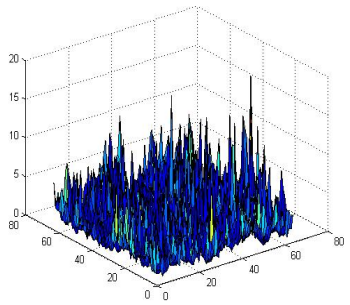
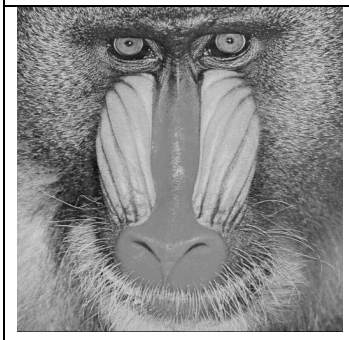
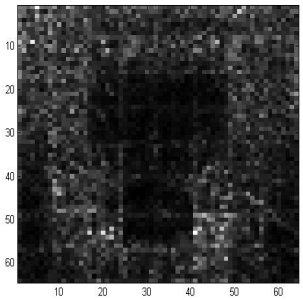
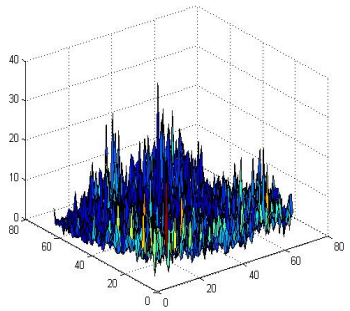
 PSNR= 39.16 M-SVD= 1.27		
 PSNR= 37.76 M-SVD= 2.17		
 PSNR= 38.98 M-SVD= 2.05		
 PSNR= 34.00 M-SVD= 2.88		

Table 6.7. Graphical measure and the numerical measure in watermarked images wi/wo JND adjustment along with increasing edge strength threshold

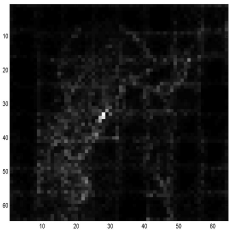
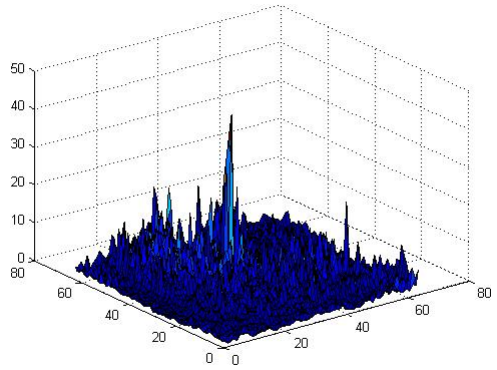
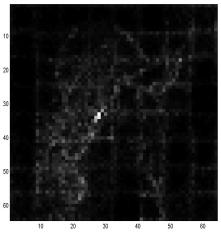
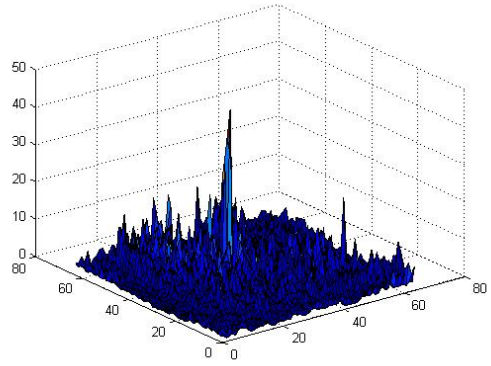
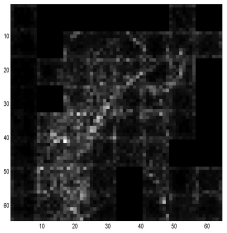
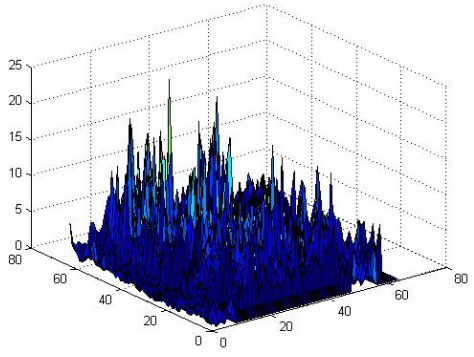
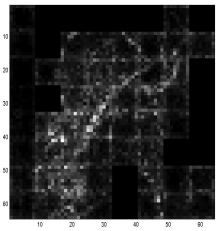
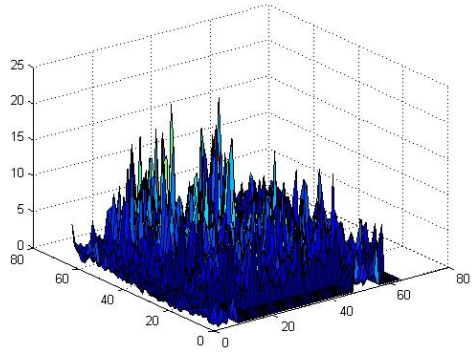
Watermarked Lena without JND adjustment along with 2-D and 3-Distortion maps	Watermarked Lena with JND adjustment along with 2-D and 3-D Distortion map
THRESHOLD =0 PSNR=38.91 M-SVD=1.91	THRESHOLD =0 PSNR= 39.33 M-SVD= 1.79
  	  
THRESHOLD =100 PSNR=39.27 M-SVD=2.06	THRESHOLD =100 PSNR= 39.73 M-SVD= 1.93
  	  

Table 6.7. (Continue)

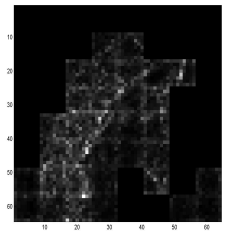
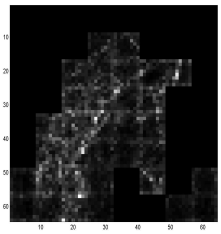
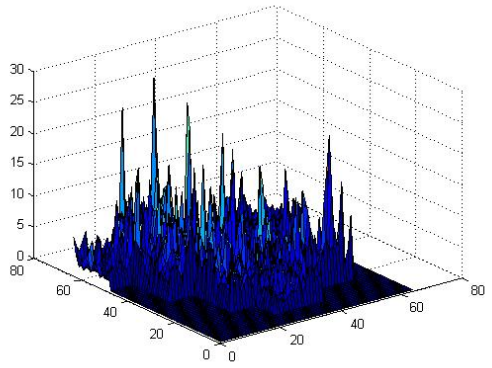
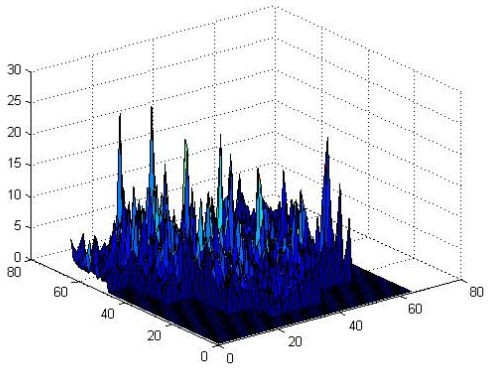
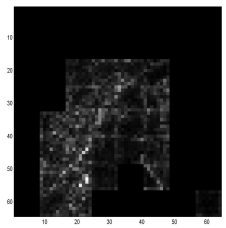
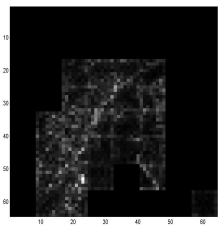
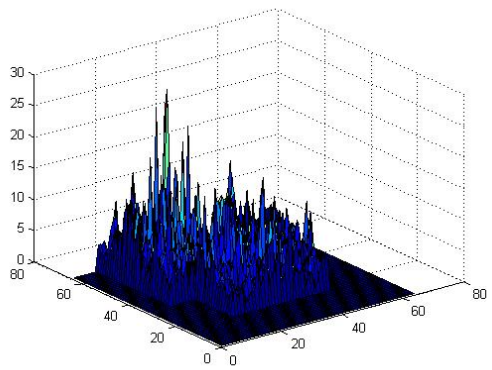
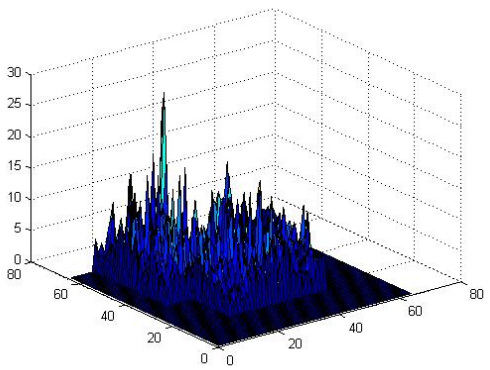
THRESHOLD=200 PSNR= 39.87 M-SVD= 2.06	THRESHOLD=200 PSNR= 40.37 M-SVD= 1.94
 	 
	
THRESHOLD=300 PSNR= 40.62 M-SVD= 1.72	THRESHOLD=200 PSNR= 41.20 M-SVD= 1.60
 	 
	

Table 6.7. (Continue)

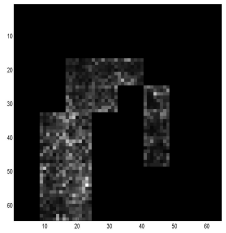
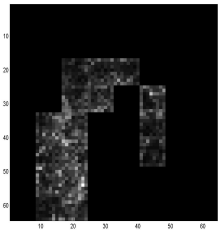
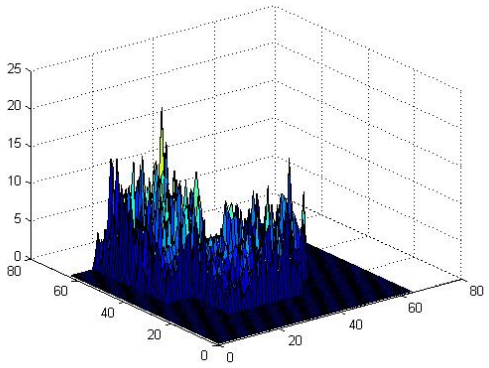
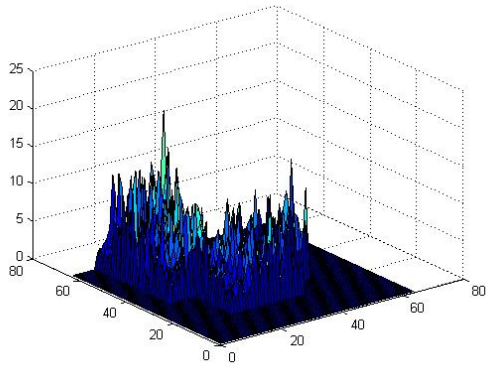
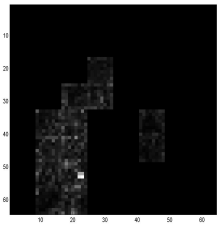
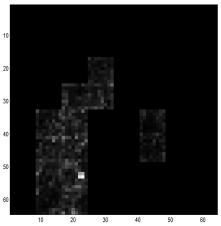
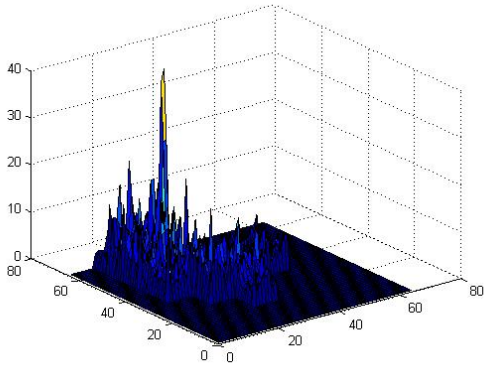
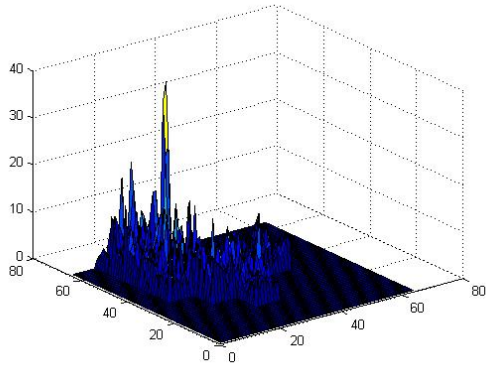
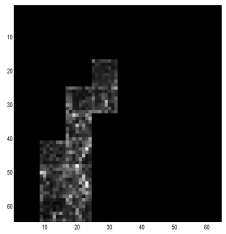
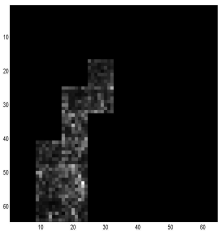
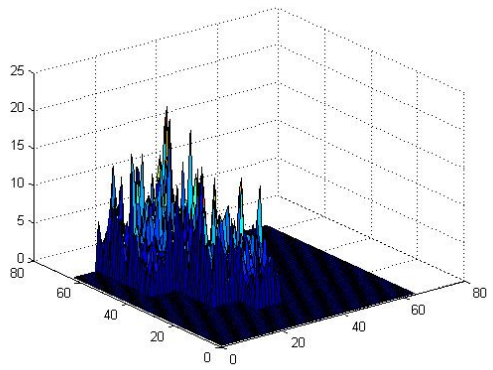
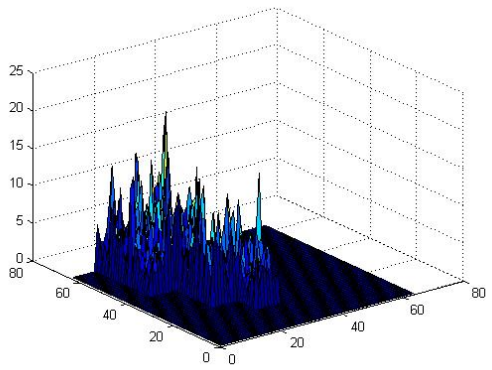
THRESHOLD=400 PSNR= 41.71 M-SVD= 1.25	THRESHOLD=400 PSNR= 42.42 M-SVD= 1.14
 	 
	
THRESHOLD=500 PSNR= 42.22 M-SVD= 1.08	THRESHOLD=500 PSNR= 42.93 M-SVD= 0.99
 	 
	

Table 6.7. (Continue)

THRESHOLD=600 PSNR= 43.11 M-SVD= 0.88		THRESHOLD=600 PSNR= 43.87 M-SVD= 0.81	
			
			

6.4 Conclusion and Future Work

The proposed M-SVD measure can express the quality of distorted images/videos either numerically or graphically. The graphical measure consistently displays the type and amount of distortion as well as the distribution of error in all the images or in all the frames of a distorted video sequence. The numerical measure is a derivation from the graphical measure which is well correlated with subjective evaluation. The quality of watermarked images produced by our curvelet based algorithm is estimated using M-SVD. The performance of our algorithm is evaluated using graphical and numerical measures along with various parameters for our watermarking algorithm. The performance of JND modeling (presented in Chapter 3) with the purpose to improve the quality of marked image is confirmed with M-SVD evaluation. In the future, we are

thinking of applying the M-SVD error measurement as evaluation step when we choose the desired parameter for watermark embedding so as to optimize the performance of our curvelet based watermarking system.

References

[1] A. M. Eskicioglu and E. J. Delp, "Overview of Multimedia Content Protection in Consumer Electronics Devices," *Signal Processing: Image Communication*, Vol. 16, No. 7, pp. 681-699, April 2001.

[2] A. M. Eskicioglu, J. Town and E. J. Delp, "Security of Digital Entertainment Content from Creation to Consumption," *Signal Processing: Image Communication, Special Issue on Image Security*, Vol. 18, Issue 4, pp. 237-262, April 2003.

[3] E. T. Lin, A. M. Eskicioglu, R. L. Lagendijk, and E. J. Delp, "Advances in Digital Video Content Protection," *Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery*, 2004.

[4] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, 2002.

[5] M. Swanson, B. Zhu and A. Tewfik, "Transparent robust image watermarking," International Conference on Image Processing Proceedings, ICIP 1996, pp. 211-214.

[6] R. G. Schyndel, A. Tirkel, and C. Osborne, "A Digital Watermark," *Proceedings of 1994 International Conference on Image Processing (ICIP 1994)*, Austin, Texas, November 13-16, 1994, pp. 86-90.

[7] S. D. Lin and C.-F. Chen, "A Robust DCT-Based Watermarking for Copyright Protection," *IEEE Transactions on Consumer Electronics*, Volume 46, No. 3, August 2000, pp. 415-421.

[8] W. Bender, D. Gruhl, and N. Morimoto, "Techniques for Data Hiding," *IBM Systems Journal*, Vol. 35, Nos. 3-4, 1996, pp. 313-336.

[9] I. Pitas, "A Method for Signature Casting on Digital Images," *Proceedings of 1996 International Conference on Image Processing (ICIP 1996)*, Vol. 3, Lausanne, Switzerland, September 16-19, 1996, pp. 215-218.

[10] R. B. Wolfgang and E. J. Delp, "A Watermark for Digital Images," *Proceedings of 1996 International Conference on Image Processing (ICIP 1996)*, Vol. 3, Lausanne, Switzerland, September 16-19, 1996, pp. 219-222.

[11] N. Nikolaidis and I. Pitas, "Robust Image Watermarking in the Spatial Domain," Signal Processing, Vol. 66, 1998, pp. 385-403.

[12] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Transactions on Image Processing*, 6(12), December 1997, pp. 1673-1687.

[13] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent Robust Image Watermarking," *Proceedings of 1996 International Conference on Image Processing (ICIP 1996)*, Vol. 3, Lausanne, Switzerland, September 16-19, 1996, pp. 211-214.

[14] J.J.K. Ó Ruanaidh, W.J. Dowling, and F.M. Boland, "Watermarking Digital Images for Copyright Protection," *IEE Proceedings on Vision, Signal and Image Processing*, 143(4), August 1996, pp. 250-256.

[15] D. Kundur and D. Hatzinakos, "Digital Watermarking Using Multiresolution Wavelet Decomposition," *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1998)*, Vol. 5, Seattle, Washington, USA, May 12-15, 1998, pp. 2969-2972.

[16] A. Lumini and D. Maio, "A Wavelet-Based Image Watermarking Scheme," *The International Conference on Information Technology: Coding and Computing (ITCC'00)*, Las Vegas, NV, March 27 - 29, 2000, pp. 122-127.

[17] W. Zhu, Z. Xiong and Y.-Q. Zhang, "Multiresolution Watermarking for Images and Video," *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4), June 1999, pp. 545-550.

[18] R. Caldelli, M. Barni, F. Bartolini and A. Piva, "Geometric-Invariant Robust Watermarking through Constellation Matching in the Frequency Domain," *Proceedings of the 2000 International Conference on Image Processing (ICIP 2000)*, Vancouver, BC, Canada, September 10-13, 2000, Vol. II, Vancouver, Canada, September 10-13, 2000, pp. 65-68.

[19] S. Pereira and T. Pun, "Robust Template Matching for Affine Resistant Image Watermarks," *IEEE Transactions on Image Processing,* 2000, pp. 1123-1129.

[20] G. C. Langelaar and R. L. Lagendijk, "Optimal Differential Energy Watermarking of DCT Encoded Images and Video," *IEEE Transactions on Image Processing*, Vol. 10, No. 1, January 2001, pp. 148-158.

[21] P. H. W. Wong, O. C. Au, and Y. M. Yeung, "A Novel Blind Multiple Watermarking Technique for Images," *IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on Authentication, Copyright Protection and Information Hiding*, Vol. 13. No. 8, August 2003, pp. 813-830.

[22] X. G. Xia, C. G. Boncelet and G. R. Arce, "A Multiresolution Watermark for Digital Images," *Proceedings of the 1997 International Conference on Image Processing(ICIP 1997)*, Washington, DC, October 26-29, 1997.

[23] J. Fridrich, M. Goljan, and R. Du, "Lossless Data Embedding - New Paradigm in Digital Watermarking," *EURASIP Journal on Applied Signal Processing, Special Issue on Emerging Applications of Multimedia Data Hiding*, Volume 2002, Issue 2 (February 2002), pp. 185-196.

[24] C..J.H. Chu and A.W.Wiltz, "Luminance channel modulated watermarking of digital images," *Proceedings of the SPIE Wavelet. Applications Conference*, pp. 437–445, Orlando, FL, USA, April 1999.

[25] J. J. Chae, D. Mukherjee, and B.S. Manjunath, "A robust embedded data from wavelet coefficients," *Proceeding of SPIE, Electronic Imaging, Storage and Retrieval for Image and Video Database*, Volume 3312, pp. 308 -317, San Jose, CA, USA, January 1998.

[26] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "DCT-Domain System for Robust Image Watermarking," *Signal Processing, Special Issue on Copyright Protection and Control*, 66(3), 1998, pp. 357-372.

[27] E. Koch and J. Zhao, "Towards robust and hidden image copyright labeling," *Proceeding of the IEEE International Workshop on Nonlinear Signal and Image Processing*, pp. 452-255, Marmaras, Greece, June 1995.

[28] R. Dugad, K. Ratakonda, and N. Ahuja, "A New Wavelet-Based Scheme for Watermarking Images," *Proceedings of 1998 International Conference on Image Processing (ICIP 1998)*, Vol. 2, Chicago, IL, October 4-7, 1998, pp. 419-423.

[29] P. Tao and A. M. Eskicioglu, "A Robust Multiple Watermarking Scheme in the DWT Domain," *Optics East 2004 Symposium, Internet Multimedia Management Systems V Conference*, Philadelphia, PA, October 25-28, 2004, pp. 133-144.

[30] J. J. K. O'Ruanaidh, W. J. Dowling, and F.M. Boland, "Phase Watermarking of Digital Images," *Proceedings of International Conference on Image Processing*, 1996, Vol. 3, pp. 239–242.

[31] M. Ramkumar, A.N. Akansu, and A.A. Alatan, "A Robust Data Hiding Scheme for Images using DFT, " *Proceedings of 1999 International Conference on Image Processing*, 1999, Vol. 2, pp. 211–215.

[32] J. J. K. O'Ruanaidh, and T. Pun, "Rotation, Scale and Translation Invariant Digital Image Watermarking," *Proceedings of International Conference on Image Processing*, 1997, Vol. 1, pp. 536–539.

[33] B. Chen and G. W. Wornell. “Quantization index modulation: A class of provably good methods for digital watermarking and information embedding,” *IEEE Transactions on Information Theory*, 1999.

[34] Z. Wang, and A.C. Bovik, “A universal image quality index,” *IEEE Signal Process. Lett*., 9(3), 81–84, 2002.

[35] S. Pereira, J. J. K. O'Ruanaidh, F. Deguillaume, G. Csurka and T. Pun, "Template Based Recovery of Fourier-Based Watermarks Using Log-polar and Log-log Maps", *Int. Conference on Multimedia Computing and Systems*, June 1999.

[36] S. Pereira and T. Pun, “Fast Robust Template Matching for Affine Resistant Watermarks”, Lecture Notes in Computer Science: *Third International Workshop on Information Hiding*, Springer, vol. 1768, pp. 199-210, 1999.

[37] M. Kutter, and F.A.P. Petitcolas, "A fair benchmark for image watermarking systems,” *SPIE, Electronic Imaging 99, Security and watermarking of multimedia contents*, vol. 3657, pp. 219-239, January 1999.

[38] http://www.watermarkingworld.org.

[39] http://watermarking.unige.ch/Checkmark.

[40] Y. Meyer, “*Wavelets: Algorithms and Applications*,” *SIAM*, Philadelphia, 1993.

[41] E. J. Candès and D. L. Donoho. “Curvelets – a surprisingly effective nonadaptive representation for objects with edges,” In C. Rabut A. Cohen and L. L. Schumaker, editors, *Curves and Surfaces*, pages 105–120, Vanderbilt University Press, 2000. Nashville, TN.

[42] E. J. Candès and D. L. Donoho., “New tight frames of curvelets and optimal representations of objects with piecewise-C2 singularities,” *Comm. on Pure and Appl. Math*. 57 (2004), 219–266.

[43] E. J. Candès and F. Guo, “New multiscale transforms, minimum total variation synthesis: application to edge-preserving image reconstruction,” *Sig. Process., special issue on Image and Video Coding Beyond Standards* 82 (2002), 1519–1543.

[44] E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying, "Fast Discrete Curvelet Transforms*," Society for Industrial and Applied Mathematics:Multiscale Modeling & Simulation*, 2006, Volume 5 Issue 3, Pages 861-899.

[45] E. J. Candès and D. L. Donoho, “Ridgelets: the key to higher-dimensional intermittency?” *Phil. Trans. R. Soc. Lond*. A. 357 (1999), 2495–2509.

[46] A. B. Watson, editor. *Digital Images and Human Vision.* Cambridge, MA: MIT Press, 1993.

[47] S. Daly, "The Visible Difference Predictor: An Algorithm for the Assessment of Image Fidelity," in A. B. Watson, editor, *Digital Images and Human Vision,* Chapter 14, pp. 179–206. Cambridge, MA: MIT Press, 1993.

[48] J. Lubin. "The Use of Psychophysical Data and Models in the Analysis of Display System Performance," in A. B. Watson, editor, *Digital Images and Human Vision,* pp. 163–178. Cambridge, MA: MIT Press, 1993.

[49] C. J. van den Branden Lambrecht and J. E. Farrell. "Perceptual Quality Metric for Digitally Coded Color Images," *Proceedings of EUSIPCO,* pp. 1175–1178, 1996.

[50] S. Voloshynovskiy, A. Herrigel, N. Baumgaetner, and T. Pun, "A Stochastic Approach to Content-adaptive Digital Image Watermarking," in *Third International Workshop on Information Hiding,* 1999.

[51] F. W. Campbell and J. J. Kulikowski, "Orientation Selectivity of the Human Visual System," *Journal of Physiology,* 187:437–445, 1966.

[52] F. W. Campbell, J. J. Kulikowski, and J. Levinson. "The Effect of Orientation on the Visual Resolution of Gratings," *Journal of Physiology*, 187:427–436, 1966.

[53] M. M. Taylor, "Visual Discrimination and Orientation," *Journal of the Optical Society of America A*, 53:763–765, 1963.

[54] M.P. Eckert and A.P. Bradley, “Perceptual quality metrics applied to still image compression.,” *Signal Processing*, 70, 177-200, 1998.

[55] N. Damera-Venkata, T.D. Kite, W.S. Geisler, B.L. Evans and A.C. Bovik, “Image quality assessment based on a degradation model,” *IEEE Transactions on Image Processing*, 9(4), 636-650, April, 2000.

[56] P.G. Barten, “Evaluation of subjective image quality with the square-root integral method,” *Journal of Optical Society of America*, 7(10), 2024-2031, 1990, October.

[57] C.-Y. Lin, M. Wu, J.A. Bloom, M.L. Miller, I. Cox and Y.M. Lui, “Rotation, scale and translation resilient public watermarking for images,” *ISPIE Security and Watermarking of Multimedia Contents II*, San Jose, CA, 2000.

[58] M. Alghoniemy, and A.H. Tewfik, “Image watermarking by moment invariants,” *Proceedings of the International Conference on Image Processing*, 2000.

[59] L. Zhang, S. Kwong, and G. Wei, "Geometric moment in image watermarking," *Proceedings of the International Symposium of Circuits and Systems*, Bangkok, Thailand, May, 2003.

[60] F. Deguillaume, S. Voloshynovskiy and T. Pun, "A method for the estimation and recovering from general affine transforms in digital watermarking applications," *Proc. SPIE* of *Security and Watermarking of Multimedia Contents IV*, Vol. 4675, p. 313-322, 2002.

[61] P. Toft, "The Radon Transform - Theory and Implementation", Ph.D. thesis, Department of Mathematical Modelling, Technical University of Denmark, June 1996.

[62] C. Lian and D. Sidan, "Rotation, scale and translation invariant image watermarking using Radon transform and Fourier transform," *Emerging Technologies: Frontiers of Mobile and Wireless Communication*, 2004.

[64] A. M. Eskicioglu and P. S. Fisher, "A survey of image quality measures for gray scale image compression," *Proceedings of 1993 Space and Earth Science Data Compression Workshop*, pp. 49-61, Snowbird, UT, April 2, 1993.

[65] J. O. Limb, "Distortion Criteria of the Human Viewer," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 12, pp. 778-793, December 1979.

[66] J. L. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," *IEEE Transactions on Information Theory*, Vol. 20, No. 4, pp. 525-536, July 1974.

[67] D. J. Sakrison, "On the role of the observer and a distortion measure in image transmission," *IEEE Transactions on Communications*, Vol. 25, No. 11, November 1977.

[68] C. F. Hall, "Subjective evaluation of a perceptual quality metric," *Proceedings of SPIE*, Vol. 310, pp. 200-204, 1981.

[69] D. J. Granrath, "The role of human visual models in image processing," *Proceedings of the IEEE*, Vol. 69, No. 5, May 1981.

[70] A. M. Eskicioglu and P. S. Fisher, "Image Quality Measures and Their Performance," *IEEE Transactions on Communications*, Vol. 43, pp. 2959-2965, December 1995.

[71] D. Van der Weken, M. Nachtegael and E. E. Kerre, "A new similarity measure for image processing," *Journal of Computational Methods in Sciences and Engineering*, Vol. 3, No. 2, pp. 209-222, 2003.

[72] A. Beghdadi and B. Pesquet-Popescu, "A new image distortion measure based on wavelet decomposition," *7th International Symposium on Signal Processing and Its Applications*, Paris, France, July 1-4, 2003.

[73] A. C. Bovik and S. Liu, "DCT-domain blind measurement of blocking artifacts in DCT-coded images," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 7-11, 2001.

[74] Z. Wang, A. C. Bovik and B. L. Evans, "Blind measurement of blocking artifacts in images," *Proceedings of IEEE 2000 International Conferencing on Image Processing*, Vancouver, BC, Canada, September 10-13, 2000.

[75] Z. Wang, H. R. Sheikh and A. C. Bovik, "No-reference perceptual quality assessment of JPEG compressed images," *Proceedings of IEEE 2002 International Conferencing on Image Processing*, Rochester, NY, September 22-25, 2002.

[76] P. Marziliano, F. Dufaux, S. Winkler and T. Ebrahimi, "A no-reference perceptual blur metric," *IEEE 2002 International Conference on Image Processing*, Rochester, NY, September 22-25, 2002.

[77] E.-P. Ong, W. Lin, Z. Yang, S. Yao, F. Pan, L. Jiang and F. Moschetti, "A no-reference quality metric for measuring image blur," *7th International Symposium on Signal Processing and Its Applications*, Paris, France, July 1-4, 2003.

[78] M. Carnec, P. Le Callet and D. Barba, "An image quality assessment method based on perception of structural information," *2003 International Conference on Image Processing*, Barcelona, Spain, September 14-17, 2003.

[79] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, “Image quality assessment: from error measurement to structural similarity,” *IEEE Transactions on Image Processing*, Vol. 13, No. 4, April 2004.

[80] D. Kahaner, C. Moler and S. Nash, *Numerical Methods and Software*, Prentice-Hall, Inc, 1989.

[81] S. O. Aase, J. H. Husoy and P. Waldemar, “A critique of SVD-based image coding systems,” *1999 IEEE International Symposium on Circuits and Systems VLSI*, Vol. 4, pp. 13-16, Orlando, FL, May 1999.

[82] V. I. Gorodetski, L. J. Popyack, V. Samoilov and V. A. Skormin, “SVD-based approach to transparent embedding data into digital images,” *International Workshop on Mathematical Methods, Models and Architectures for Computer Network Security (MMM-ACNS 2001)*, St. Petersbug, Russia, May 21-23, 2001.

[83] R. Liu and T. Tan, “A SVD-based watermarking scheme for protecting rightful ownership,” *IEEE Transactions on Multimedia*, Vol. 4, No. 1, pp.121-128, March 2002.

[84] D. V. S. Chandra, “Digital image watermarking using singular value decomposition,” *Proceedings of 45th IEEE Midwest Symposium on Circuits and Systems*, pp. 264-267, Tulsa, OK, August 2002.

[86] A. M. Rohaly, J. Libert, P. Corriveau, and A. Webster, Editors, "Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment," March 2000. Available at http://www.vqeg.org/.

[87] B. Furht and D. Kirovski, *Multimedia Watermarking Techniques and Applications*, Auerbach Publications © 2006.

[88] Chun-Shien Lu, *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*, IGI Publishing © 2005.

[89] J. Canny, ``A computational approach to edge detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6), pp679-698 (1986).

[90] S. Daly, “The visible differences predictor: An algorithm for the assessment of image fidelity”, *SPIE, Human Visual Processing, and Digital Display*, vol.1666,No.3, pp.2- 15, 1992.

[91] P. W. Wong and N. Memon, “Secret and public key image watermarking schemes for image authentication and ownership verification,” *IEEE Transactions on Image Processing*, 10(10), October 2001.

[92] J. A. Bloom, I. J. Cox, T. Kalker, J-P. Linnartz, M. L. Miller, and B. Traw. "Copy Protection for DVD Video," *Proceedings of the IEEE*, 87(7):1267–1276, 1999.

[93] A. Shnayderman, A. Gusev, and A. M. Eskicioglu, “A multidimensional image quality measure using singular value decomposition,” *Proceedings of SPIE Image Quality and System Performance*, Vol. 5294, pp. 82-92, San Jose, CA, January 19-20, 2004.

[94] I. Sobel, “Camera Models and Perception,” Ph.D. thesis, Standford University, Standford, CA, 1970.

Bibliography

Peining Tao was born in Shanghai, China. She received the bachelor degree in clinical medicine from the Medical Center of Fudan University (Former Shanghai Medical University) in 1998, the master degree in computer information science from Brooklyn College, The City University of New York in 2001. She is currently a PhD candidate in computer science at The Graduate Center, the City University of New York and expected to graduate in Feb, 2008.

Her research interests include digital watermarking, image/video quality measurement and image recovery.