This is to certify that the
dissertation entitled

IMAGE WATERMARKING IN THE TIME-FREQUENCY DOMAIN

presented by

MAHMOOD ALAYA AL-KHASSAWENEH

has been accepted towards fulfillment
of the requirements for the

Doctoral degree in Electrical and Computer Engineering

Major Professor's Signature

05/11/07

Date

IMAGE WATERMARKING IN THE TIME-FREQUENCY DOMAIN

By

Mahmood Alaya Al-khassaweneh

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Electrical and Computer Engineering

2007

ABSTRACT

IMAGE WATERMARKING IN THE TIME-FREQUENCY DOMAIN

By

Mahmood Alaya Al-khassaweneh

With the fast development of the internet and multimedia tools in the past decade, the access and the unauthorized reproduction of digital data has become easier and widespread. The ease of access to digital data brings with itself the challenge of content protection. One way to address this problem is through digital watermarking, which has become an important tool in copyright protection applications. The watermarking algorithms proposed so far, focus on time or frequency domain representations of the image. There have been only a few attempts to utilize the joint time-frequency (spatial-spectral) characteristics of an image for watermarking. These time-frequency domain watermarking attempts were mainly focused on detecting the watermark rather than extracting it and did not provide a theoretical framework for the performance analysis of the watermarking algorithms.

In this dissertation, we introduce three new image watermarking schemes in the joint time-frequency domain to address these issues in image watermarking. The first two methods embed the watermark in the time-frequency domain of the image using Wigner distribution. Two different methods for embedding the watermark in the Wigner distribution are introduced; the **Time-Wigner** method where the watermark is embedded directly into the Wigner distribution of the image, and the **Wigner-Wigner** method where the Wigner distribution of the watermark is embedded in the Wigner distribution of the image. The performance of the embedding algorithms and the corresponding watermark detectors are analyzed. It is shown that embedding in the time-frequency domain is equivalent to a non-linear embedding function in the

spatial domain. The third watermarking approach in the time-frequency domain uses the local autocorrelation function of the image. The local autocorrelation function for a subset of pixels chosen from the image is computed and the watermark is embedded in the selected locations of the autocorrelation function. A blind detection algorithm is derived and its performance is quantified by deriving the probability of error. The proposed algorithm is shown to be transparent and robust under attacks. A comparison of the proposed methods with a discrete wavelet transform (DWT) domain based or/and spread spectrum (SS) methods is illustrated through simulations. The detailed analysis of the proposed time-frequency watermarking algorithms shows that looking at this joint domain improves watermarking capacity and robustness compared to existing methods.

To my wife and beloved parents

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere thankfulness to my parents Alaya and Jamilah Al-khassaweneh for their unconditional support, love, and caring which helped me with the journey of life and graduate studies with humility and honesty. I am also grateful to my wife Esraa Al-Sharoa for her encouragement, advice and patience. Her support kept me working hard to achieve this. I would also like to thank my little son Ahmad, who kept me awake all night studying. I would also like to express my gratitude to all my brothers, sisters, and my family in-law for their sincere wishes and encouragement. Especial thanks to my brother Mazin khasawneh, his wife Manar, and their daughter Leen for the help, guidance, and being the best friends.

This dissertation would not have been possible without the able guidance and valuable comments and inputs from my advisor Dr. Selin Aviyente. Dr. Selin Aviyente gave me a break from my mundane life as a programmer and software engineer into the challenging and interesting world of science and engineering. I am grateful for her intellectual and financial support, inspiration, freedom of work, invaluable advice and the trust that gave me the confidence to make the right decisions throughout my graduate studies. My advisor, Dr. Selin Aviyente is not only great visionary scientist with renowned reputation, but she is also very kind and gentle individual I have known. In addition, I am very grateful to my committee members: Dr. Huyder Radha, Dr. John Deller, and Dr. Anil Jain for their guidance and for teaching me many useful courses in electrical and computer engineering.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The digital information revolution has brought profound changes in our lives. Along the many advantages, this revolution has also generated new challenges and new opportunities for innovation [1, 2]. The availability of powerful software and new devices, such as digital camera and camcorder, high quality scanners and printers, digital voice recorder, MP3 player and PDA, have reached consumers worldwide and enable them to create, manipulate, and enjoy the multimedia data.

Internet and wireless network offer an easy way to deliver and exchange information. The security and fair use of the multimedia data, as well as the fast delivery of multimedia content to a variety of end users/devices are important, yet challenging issues. This ease of access to digital data brings with itself the challenge of content protection. Some common attacks on digital data include illegal access to the transmitted data, data content modification, and production and re-transmission of illegal copies. The solutions to these problems will not only contribute to our understanding of this fast moving complex technology, but will also offer new economic opportunities to be explored. Watermarking and encryption techniques were developed in order to provide copyright protection for digital data. Encryption protects the data from piracy attacks during transmission and once the data is received and decrypted, it is no longer protected. On the other hand, watermarking embeds a secret watermark into the original data in a way such that it is always present [3].

Although the concept of watermarking (information hiding) has been mentioned thousands years ago [4], it did not see the light and the attention from researchers except in the past two decades. Research on watermarking has made considerable progress in recent years and attracted attention from both academia and industry.

Techniques have been proposed for a variety of applications, including ownership protection, authentication, access control, and annotation [5]. Watermarking is also found useful as a general tool to send side information in multimedia communication for achieving additional functionalities or enhancing performance. Imperceptibility, robustness against moderate processing such as compression, and the ability to hide many bits are the basic but rather conflicting requirements for many data hiding applications. Several watermarking techniques have been proposed for different multimedia data, like image, video and audio signals [6, 7, 8, 9, 10, 11]. Each data type has it is own characteristics and is treated uniquely when it is watermarked [12, 13, 14, 15, 16]. The focus of this dissertation is on image watermarking.

This chapter is organized as follows. In Section 1.1, the different stages in a general watermarking scheme are discussed. Types, applications, and requirements for the watermark are discussed in Sections 1.2 through 1.4. While Section 1.5 talks about the domains used for watermarking, Section 1.6 summarizes the major measures used to evaluate the performance of a given watermarking algorithm.

1.1 General Scheme

A general watermarking scheme is illustrated in Figure 1.1. The main components in any watermarking scheme are the encoder and the decoder. The encoder embeds the watermark, w, inside the original image, I, using an embedding function, E, to produce the watermarked image, $\hat{I}$. Mathematically this can be represented by,

$$\hat{I} = E(I, w), \tag{1.1}$$

where the embedding function, E, can operate in the spatial domain or some transform domain and can have additive or multiplicative form. On the other hand, the decoder, D, tries to extract or detect the original watermark from the watermarked

image, which is possibly corrupted by attacks,

$$\hat{w} = D(\hat{I}, I), \tag{1.2}$$

where the decoder may or may not use the original image for detecting or extracting the watermark. Depending on the nature of the embedder and the way the watermark is inserted, the watermark may be extracted in the exact form or may be detected. Detecting the watermark can verify the ownership, while extracting it can prove the ownership.

1.2 Visible and Invisible Watermarks

Watermarking techniques can be classified in terms of perceptibility into two groups, perceptible and imperceptible hiding. In perceptible watermarks, a visually meaningful message, such as a logo, is embedded inside the image, which is essentially an image editing or synthesis problem. The visible watermarks explicitly exhibit the copyright, ownership information, or access control policies so as to discourage the misuse of the watermarked images [17, 18]. For example, semitransparent logos are commonly added to the preview images accessible via World Wide Web by copyright holders. In [17], a visible watermarking technique is proposed by modifying the luminance of the original image according to a binary or ternary watermark pattern. The same amount of modification is applied to the local luminance to give a consistent perceptual contrast [19]. In addition, the modification is modulated by a random generated sequence to make it difficult to systematically remove the visible marks via an automated algorithm.

In most copyright and digital rights managements applications invisible watermarks are preferred [20, 21, 22]. Invisible watermarks are used for content and author identification in order to be able to determine the origin of an image. They can also

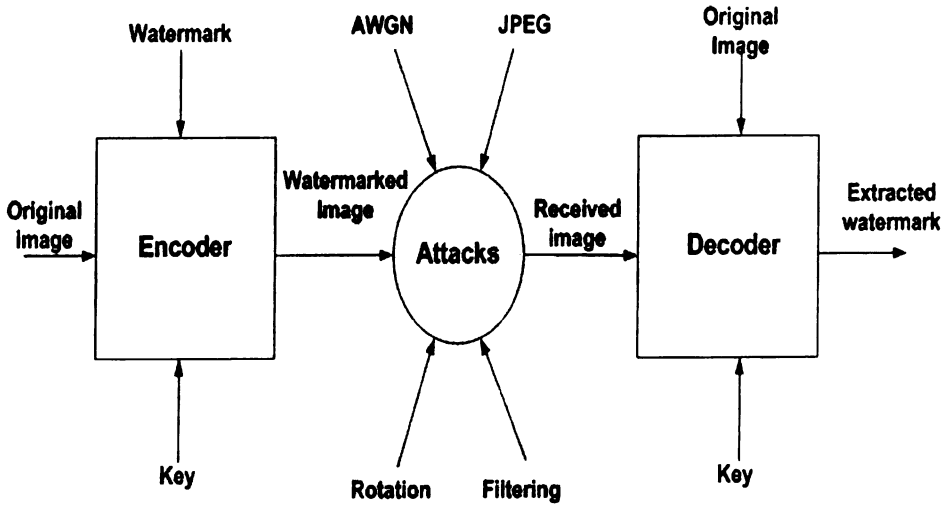

Figure 1.1. A general watermarking scheme.

be used in unauthorized copies detection either to prove ownership or to identify a customer. The invisible scheme does not intend to forbid any access to an image but its purpose is to be able to tell if a specified image has been used without the owner's formal consent or if the image has been altered in any way. This approach is the one that has received the most attention in the past couple of years and it is also the focus of this dissertation [23, 24, 25].

1.3 Applications

Although the original motivation for watermarking was copyright protection, it has since then been used in different applications. Some common applications of watermarking include [26, 27]:

1. Copyright protection: For copyright protection, a watermark indicating ownership is embedded in the original image. The watermark, which is known only to the copyright holder/owner, should survive common processing and intentional attacks so that the owner can show the presence of this watermark in case of dispute to demonstrate the ownership of that particular image. In this application, the goal is watermark detection rather than extraction. The probability of detection should be high and the algorithm should have a low false alarm rate. The total the number of bits that can be embedded are not necessarily high [28].

2. Fingerprinting: In fingerprinting, the hidden data (watermark) is used to trace the originator or the recipients of the image. For example, different watermarks are embedded in different copies of the image before distributing to a number of recipients. The robustness against obliterating and the ability to convey a non-trivial number of bits are required [29].

3. Authentication: A watermark is embedded in the image, and is used later to determine whether the original image is tampered or not. The robustness against removing the watermark or making it undetectable is not a concern as there is no such incentive from the attacker's point of view. However, forging a valid authentication watermark in an unauthorized or tampered image must be prevented [30].

4. Annotation: In annotation, the goal is to embed a large number of bits inside the original image. Although the robustness against intentional attack is not required, some degree of robustness against common processing attacks are desirable. The original host image, preferably, should not be used to extract the watermark in this application [31].

1.4 Requirements

Most watermarking algorithms try to satisfy the following main requirements [32, 33, 34]:

1. Perceptual Transparency: The characteristics of the Human Visual System (HVS) are used to assure that the watermark is not visible. Basically, perceptual transparency means that a watermarked image should look identical to the original one, which means one should not notice any degradation in the perceived quality. Transparency is a basic requirement of digital watermarking.

2. Robustness: The watermark should be detected by an authorized user after the image has undergone attacks such as additive white Gaussian noise (AWGN), compression, filtering, etc. Ideally, the amount of image distortion necessary to remove the watermark should degrade the desired image quality to the point of becoming commercially valueless.

3. Capacity: A watermarking system must allow for a useful amount of information to be embedded into the image. Depending on the application, the amount of data can vary from a single bit to multiple bits [35].

4. Computational complexity: The watermark system should not be computationally complex especially for applications where real-time embedding is desired. Moreover, reducing the number of computations means low cost in designing the hardware for the watermarking algorithm.

Therefore, the general goal of watermarking is to produce a modified data that looks exactly the same as the original data but still contains the watermark that could be used for copyright authentication.

1.5 Domains used for Image Watermarking

The two most common methods used for watermarking digital images are the spatial and the spectral domain methods [36, 37, 38, 39]. The spatial domain methods choose regions of the image according to texture, edges or a random partitioning and embed the watermark in the selected regions [40, 41, 42, 43, 44, 45, 46]. Although the watermarked image is identical to the original one, it is not in general robust to the basic image processing attacks [47, 48, 49]. The spectral domain methods transform the image into the spectral domain using transform methods such as Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), Fourier Mellin Transform and then mark it in the transform domain [50, 51, 52, 53, 54]. In this case, the watermark is inserted in the perceptually significant parts of the image so that it is robust.

Transform domain methods have several advantages over the spatial domain methods [54]. First, they are more robust, since the watermark is inserted in the perceptually significant parts of the image, which corresponds to the mid-frequency range. This range can be easily found in the transform domain [55, 56]. Second, they resist the compression attacks. Since most compression techniques operate in the frequency domain, it is easier to develop a watermarking scheme in the transform domain that overcomes possible compression. Third, some transform domain algorithms are robust against specific geometric transformations such as DFT which is robust to most affine transformations [57].

Spread spectrum is a common technique used for watermarking in both the spatial and the transform domains [58, 59, 60, 61, 62]. The idea is to spread the watermark

over certain pixels of the image. As an example, in [63], where the watermark is embedded in the DCT domain of the image, the authors assume the watermark to be an independent and identically distributed Gaussian random vector that is imperceptibly inserted in a spread-spectrum-like fashion into the perceptually most significant spectral components of the data. It has been shown that the watermark is robust to signal processing operations such as lossy compression, filtering, digital-analog and analog-digital conversion, requantization, and common geometric transformations such as cropping, scaling, translation, and rotation. Since the introduction of the original spread spectrum approach, many developments have been made on the method described in [63] such as using different transform domains to embed the watermark, and the introduction of blind watermark detection algorithms [64, 65, 66]. In [65], the authors propose a multi-bit watermarking algorithm which is based on the idea of spreading the watermark bit over many pixels of the original image using code-division multiplexing. The proposed method has a deterministic watermark embedding scheme that assures total embedding efficiency. Unlike [63], where the watermark is spread out in the DCT domain, the watermark in [65] is embedded in the spatial image domain. Many spread spectrum algorithms have the following limitations [67]:

1. Spread spectrum, in general, allows the detection of the watermark rather than extraction.

2. If the energy of the watermark is reduced because of fading-like distortions, it will lead to unreliable detection of the watermark [69].

3. Most of the spread spectrum techniques do not take into account the non-stationarity of the original image or the attack interference.

An example for using the DWT domain for watermarking is in [68]. In [68], the authors embed the watermark by quantizing certain DWT coefficients in different

subbands. The watermark is extracted without the need of the original image. Another well-known watermarking method in the transform domain is [52]. In [52], the authors present a watermarking algorithm in the wavelet domain. The watermark is masked according to the characteristics of the human visual system (HVS), where masking is accomplished pixel by pixel by taking into account the texture and the luminance content of all the image subbands. The watermark consists of a pseudorandom sequence which is adaptively added to the largest detail bands. The watermark is detected by computing the correlation between the watermarked coefficients and the watermarking sequence without refereing to the original image.

Although transform domain algorithms have more advantages in providing robustness, sometimes it is difficult to satisfy imperceptibility constraints in the spatial domain simultaneously with the spectral domain constraints. In order to take full advantage of both the spatial and the spectral domains, researchers have started looking at the joint time-frequency representation of the image, which gives a more comprehensive representation of the image compared to looking at each domain individually [70, 71, 72, 73, 74]. This approach also provides flexibility in the amount of data that can be hidden inside an image. The use of joint time-frequency domain is the focus of this dissertation. Table 1.1 summarizes some well-known watermarking algorithms in literature.

Table 1.1. Summary of some well-known watermarking algorithms

Method	Domain	Multi-bit	Blind
Barni, etc. [52]	Transform (DWT)	Yes	Yes
Cox, etc. [63]	Transform (DCT)	Yes	No
Mayer, etc. [65]	Spatial	Yes	No
Kundur, etc. [68]	Transform (DWT)	Yes	Yes
Stankovic, etc. [70]	Time-frequency (Wigner)	No	No

1.6 Performance Measures

In order to evaluate a watermarking algorithm, certain sets of measures should be met. Although, there are no agreed-upon sets, most work in the watermarking literature use the following measures for performance evaluation [75]:

1. Imperceptibility: For invisible watermarking method, the watermark should be imperceptible and the human eye should not be able to distinguish between the watermarked and the original images. This measure is subjective, and thus is not always a reliable way of evaluating the watermarking algorithm.

2. Peak Signal to Noise Ratio (PSNR): This measure is related to imperceptibility, where having higher PSNR means higher imperceptibility. This quantitative measure is given for an $N \times N$ image by,

$$\text{PSNR(dB)} = 10 \log_{10} \left(\frac{255^2}{\frac{1}{N^2} \sum_{x,y} \left(\hat{I}(x,y) - I(x,y) \right)^2} \right), \tag{1.3}$$

   where $\hat{I}(x,y)$ and $I(x,y)$ are the watermarked and the original images, respectively. For a good watermarking algorithm, the PSNR value should be above 30dB.

3. Correlation Coefficient: This is a measure between the extracted and the original watermark, where higher correlation value means the extracted watermark is the one of interest. This measure is mathematically given by,

$$\langle w, \hat{w} \rangle = \frac{\sum_y w(y)\hat{w}(y)}{\sqrt{\sum_y w^2(y) \sum_y \hat{w}^2(y)}}. \tag{1.4}$$

where, w and $\hat{w}$ are the original and extracted watermarks, respectively.

4. Probability of Error: This measure is used to study the probability of detecting a false watermark and assume that it is the correct one, i.e., probability of false alarm, P_{FA}, and detecting the correct watermark and assume that it is the false one, i.e., probability of miss, P_M. The probability of error in terms of P_{FA} and P_M is given by,

$$P_e = p_0 P_{FA} + p_1(P_M). \tag{1.5}$$

where, p_0 and p_1 are the a priori probabilities.

5. Other Performance Measures: There are other performance measures which are application dependent. For example, complexity is an issue if the watermarking is done in real time, while it is not an issue for applications where the embedding can take place offline. Alternatively, the time needed for the watermarking algorithm is another issue, especially if the algorithm is to be implemented by hardware.

The rest of this dissertation is organized as follows. Chapter 2 gives some background on time-frequency distributions and summarizes previous watermarking methods in the time-frequency domain. Chapters 3 through 5, introduce the Time-Wigner method, the Wigner-Wigner method, and the autocorrelation method, respectively. The derivation and the analysis for the embedding and the detection/extraction algorithms are given for each method. Moreover, simulation results to evaluate the performance of the proposed algorithms and comparisons with existing methods are provided. Chapter 6 gives a detailed comparison of the three methods proposed in this dissertation and offers techniques for performance improvement. Finally, Chapter 7 concludes this dissertation with a summary of contributions and future work.

CHAPTER 2

WATERMARKING IN THE TIME-FREQUENCY DOMAIN

Time-frequency distributions are well-known signal representation tools that have been used in a variety of applications including signal detection, classification, and analysis [76]. Despite their widespread use in analyzing non-stationary signals, their application to the signal watermarking problem has been limited until recently. Although time-frequency analysis identifies the time at which various signal frequencies are present, the difficulty of implementing and understanding these distributions, limited their usage especially in the case of image watermarking, for which the time-frequency distribution is a four dimensional distribution. Despite these challenges, the representation of the energy of the signal simultaneously in time and frequency makes time-frequency distributions a strong candidate for the watermarking problem.

In this chapter, we give a brief background on time-frequency distributions in general and on Wigner distribution in particular. The main properties of Wigner distribution are discussed in detail. After a brief introduction to time-frequency distributions provided in Section 2.1, we summarize some of the recent work in the area of watermarking in the joint time-frequency domain and discuss the major properties of these methods in Section 2.2. Finally, Section 2.3 summarizes the contributions of this dissertation to watermarking in the joint time-frequency domain literature.

2.1 Background On Time-Frequency Distributions

Time-frequency distributions are bilinear transforms of a signal that represent the energy distribution over time and frequency [77, 78]. The need for a combined time-frequency representation stemmed from the inadequacy of the individual time domain and frequency domain analysis to fully describe the nature of non-stationary signals. A time-frequency distribution of a signal provides information about how the spectral content of the signal evolves with time, thus providing an ideal tool to dissect, analyze, and interpret non-stationary signals. This is performed by mapping a one dimensional signal in the time domain, into a two dimensional time-frequency representation of the signal. A variety of methods for obtaining the energy density of a function, simultaneously in the time and the frequency have been devised, most notably the short time Fourier transform and the Wigner distribution. The general class of bilinear time-frequency distributions, named Cohen's class of distributions [76], are defined as,

$$C(t,\omega) = \frac{1}{4\pi^2}\int\int\int s^*(t-\frac{1}{2}\tau)s(t+\frac{1}{2}\tau)\phi(\theta,\tau)e^{j(-\theta t-\tau\omega+\omega t)}dtd\tau d\theta, \qquad (2.1)$$

where $\phi(\theta,\tau)$ is a two dimensional function called the kernel. The kernel determines the distribution and its properities.

Among many time-frequency distributions, Wigner distribution has received the most attention in the watermarking literature. Wigner distribution is a well-known member of the Cohen's class of distributions, for which $\phi(\theta,\tau) = 1$. For a one-dimensional continuous time signal, $s(t)$, Wigner distribution is defined as,

$$W(t,\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} s\left(t+\frac{\tau}{2}\right)s^*\left(t-\frac{\tau}{2}\right)e^{-j\omega\tau}d\tau, \qquad (2.2)$$

where τ is the time lag variable. Equation (2.2) suggests that to find the Wigner distribution at a particular time, we sum the sequence obtained from the product of the signal at a past time, τ, with the signal at a future time, τ. This product, $s\left(t+\frac{\tau}{2}\right)s^*\left(t-\frac{\tau}{2}\right)$, is called the autocorrelation function and it will be discussed in detail in Chapter 5.

This definition can be extended to the discrete-time domain. For a one-dimensional discrete time signal, $s(n)$, of length N, the Wigner distribution is,

$$WD(n,\omega) = 2\sum_{m=-\infty}^{\infty} s(n+m)s^*(n-m)e^{-j2m\omega}, \tag{2.3}$$

where n and $\omega = 2\pi k/N$ are the time and the frequency variables respectively.

Wigner distribution has many properties that make it a good choice for water-marking applications. First, it satisfies the frequency and the time marginals which makes it a valid energy distribution. The time and frequency marginals are given by,

$$\sum_{\omega} WD(n,\omega) = |s(n)|^2, \tag{2.4}$$

and

$$\sum_{n} WD(n,\omega) = |S(\omega)|^2, \tag{2.5}$$

respectively. Second, it is invertible, i.e. the signal can be retrieved from its Wigner distribution up to a phase constant as:

$$s(t) = \frac{1}{2\pi s^*(0)}\int_{-\infty}^{\infty} W\left(\frac{t}{2},\omega\right)e^{jt\omega}d\omega. \tag{2.6}$$

For real and positive valued discrete time signals, the signal can be retrieved from its

Wigner distribution as,

$$s(n) = \sqrt{\sum_{\omega} WD(n, \omega)}. \tag{2.7}$$

Equation (2.7) implies that for a positive real-valued signal, the original signal can be retrieved from its Wigner distribution by taking the square root of the inverse Fourier transform of the Wigner distribution evaluated at $m = 0$. Finally, the Wigner distribution of a real signal is even symmetric. These properties will simplify the embedding and detection algorithms in image watermarking.

The invertibility property in equation (2.7), which is valid for images, is not valid for general signals, where the signal is not necessarily positive or real-valued. Many algorithms have been proposed for synthesizing a time signal from a given Wigner distribution. In [79, 80], the signal is computed for even and odd indices separately through performing the eigen-decomposition of the autocorrelation matrix. In [81, 82], the signal is synthesized using a set of basis functions. The authors in [81, 82], formulate the synthesis problem as approximating a two-dimensional function. This two-dimensional function is formulated as a product of two one-dimensional functions using two least square procedures. The first procedure involves expressing a time-frequency function as a bilinear combination of the basis auto and cross-Wigner functions. The least squares approximation leads to an eigenvalue-eigenvector decomposition of a symmetric matrix. The second procedure involves the approximation of a pre-computed matrix as an outer product of two vectors. In [83], the synthesis is accomplished by using a reference signal known a priori or found iteratively. The most recent method for synthesizing the signal given its Wigner distribution was developed in [84] by finding the discrete-time signal whose Wigner distribution best matches a specified time-frequency distribution in the sense of the least mean squared error.

The effectiveness of Wigner distribution in signal analysis has inspired researchers to adapt this distribution to image processing. Wigner distribution has been extended

to two-dimensional signals such as images in [85] as,

$$WD(n_1, n_2, k_1, k_2) =$$

$$\sum_{m_1=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{m_2=-\frac{N}{2}}^{\frac{N}{2}-1} s(n_1 + m_1, n_2 + m_2) s^*(n_1 - m_1, n_2 - m_2) e^{-j\frac{4\pi}{N}(m_1 k_1 + m_2 k_2)}. \tag{2.8}$$

This extension yields a four-dimensional representation which makes it difficult to interpret the resulting distribution and increases the computational complexity. For an $N \times N$ image this creates N^4 watermarkable points. However, due the computational complexity and the difficulty of interpreting these points, in this dissertation, we find the Wigner distribution for a subset of pixels of the image using equation (2.3). For an $N \times N$ image, the one-dimensional Wigner distribution creates N^3 watermarkable points. Although there is a reduction in the number of watermarkable cells by a factor of N, the distribution in equation (2.3) is easier to implement and visualize, less computationally complex, and still provides N^3 watermarkable cells which is higher than the N^2 cells, which are available in the individual time or frequency domains.

2.2 Previous Work on Image Watermarking in the Time-Frequency Domain

The idea of watermarking in the joint time-frequency domain has attracted some attention in recent years. Most of the recent work has concentrated on using the Wigner distribution as the signal transform before embedding the watermark. Many researchers use equation (2.3) to find the Wigner distribution of an image by scanning it row by row or choosing a subset of pixels from the original image. For example, in [70], the authors used a two-dimensional chirp signal with a variable spatial frequency

as the watermark. The watermark is characterized by a linear frequency change and can be detected by using two-dimensional (2-D) time-frequency distributions. The projections of the 2-D Wigner distribution and the 2-D RadonWigner distribution are used in the watermark detection process. Although the authors were able to detect the watermark efficiently under most attacks, there was no discussion on the extraction of the watermark and there was no theoretical analysis of the performance of the method. Moreover, the algorithm did not discuss the potential for multi-bit watermarking.

In [71], the Wigner distribution is used for watermark embedding. The watermark is embedded in a subset of the transformed cells in the Wigner domain. These cells are selected such that the watermark will survive the JPEG compression. Since the resultant watermarked distribution is not a valid Wigner distribution, the time signal that has the closest distribution in the mean square error sense is found [82]. Although this algorithm detects the presence of the watermark under JPEG attacks, no experimental results for other types of attacks are reported. Moreover, the error in detecting the watermark was not studied and the extraction of the watermark was not discussed.

In [73], a fragile image watermarking method using Wigner distribution is presented. The watermark is an FM modulated signal which is embedded in the diagonal elements of the image. The particular features of this signal in the time-frequency domain are used to identify the watermark. The Wigner distribution is used to extract the watermark. Since the focus of this method was fragile watermarking, no study on the robustness of the proposed algorithm has been done.

Time-frequency distributions have also been used for audio watermarking. The authors in [74], present a non-blind, robust watermarking scheme for audio signals. The watermarking algorithm is based on the Singular Value Decomposition (SVD) of the spectrogram of the signal. The SVD of the spectrogram is modified adaptively

according to the watermark message.

In this dissertation, we introduce three new methods for embedding the watermark into the image using the Wigner distribution. For simplicity, we assume that we have an $N \times N$ image and a watermark sequence of length N. The first method, Time-Wigner method, embeds the watermark directly into the Wigner distribution of the image, while the second one, Wigner-Wigner method, embeds the Wigner distribution of the watermark into the Wigner distribution of the image. The third method makes use of the autocorrelation domain, which is related to the Wigner distribution through a Fourier transform, and uses it for watermark embedding.

2.3 Contributions of this Dissertation

In this dissertation, we introduce three new image watermarking methods in the joint time-frequency domain. Unlike the previous work in the time-frequency domain, a complete mathematical analysis is provided for both embedding and detection/extraction stages. The proposed methods put no constraints on the characteristics of the watermark such as parameterizing it as a linear chirp as in previous work. Moreover, we introduce a multi-bit watermarking algorithm which is suitable for hiding larger amounts of data. We also compare the proposed joint time-frequency watermarking algorithms with the current time and frequency domain methods.

The first class of algorithms will develop two new watermarking methods that combine the spatial and the spectral domains, for both embedding and detection. The first method consists of embedding the watermark directly in the Wigner distribution of the image, the Time-Wigner method, while the second method consists of transforming the watermark into the Wigner domain and then embedding it into the Wigner distribution of the image, the Wigner-Wigner method. The corresponding detection algorithms are also derived [86, 87].

We also introduce a new image watermarking method that is equivalent to wa-

termarking in the Wigner domain. The watermark is embedded in the local autocorrelation domain. The autocorrelation function is related to the Wigner distribution through a Fourier transform and has no aliasing and inversion problems. The time-varying autocorrelation function for randomly chosen pixels is found and the watermark is embedded such that the modified autocorrelation is still a valid autocorrelation function. This will ensure the invertibility of the autocorrelation function and will enable us to extract the embedded watermark bits [88].

In the following chapters, we discuss each method in detail. The embedding and detection/extraction algorithms are derived, the simulation results are provided, and a comparison with existing time and/or frequency based watermarking methods are carried out for each proposed method.

CHAPTER 3

THE TIME-WIGNER WATERMARKING METHOD

In this chapter, the Time-Wigner method which embeds the watermark sequence directly into the Wigner distribution of the image is introduced. The idea of embedding the watermark in the transform domain is a common idea used with Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT) domains [55, 68]. However, spreading the watermark in the joint time-frequency domain is a very recent idea and not much work has been done in this area. Similar to embedding the watermark in the transform domain, i.e. DCT, DFT, and DWT, the proposed method can be thought of as spreading the sequence in the joint-time frequency domain.

In the proposed method, the Wigner distribution for a subset of pixels chosen from the original image is computed. The watermark, which could be either a Gaussian distributed sequence or a binary sequence, is embedded inside the Wigner distribution of the chosen pixels such that the watermarked distribution is as close as possible to a valid Wigner distribution. The embedding algorithm is simplified to a non-linear function in time which makes the embedding less computationally complex.

Two detection algorithms for the Gaussian and the binary watermark cases are derived and their performances are quantified through an analysis of the probability of error. In addition, the performance of the Time-Wigner method is compared with the well-known spread spectrum method [63] to demonstrate the robustness and the potential of the proposed method.

This chapter is organized as follows. Section 3.1 gives a detailed analysis of the watermarking embedding algorithm in the Wigner domain. It shows that watermarking in the Wigner domain is equivalent to a non-linear embedding function in the time

domain. Section 3.2 studies the error introduced in the inversion of the watermarked distribution from the time-frequency domain to the time domain and gives more insight about the choice of the weighting matrix. In Section 3.3, the performance of the proposed Time-Wigner method for the Gaussian distributed watermark case is analyzed. Both the probability of detection and the probability of miss are derived for detecting the watermark. On the other hand, Section 3.4 deals with the binary watermark sequence case. The probability of error in detecting the watermark is derived. Section 3.5 provides simulation results to demonstrate the performance of the proposed method under attacks. A comparison between the Time-Wigner method and spread spectrum watermarking method is given. Discussion about the proposed Time-Wigner method is given in Section 3.6. Finally, Section 3.7 summarizes the major contributions of this chapter.

3.1 Watermark Embedding

In the Time-Wigner method, the watermark is embedded directly into the Wigner distribution of the image. Figure 3.1 shows a block diagram for the proposed watermark embedding algorithm. The embedding algorithm has three main stages. The first stage transforms a subset of pixels of length L chosen randomly from the host image into the Wigner domain to produce L^2 watermarkable cells. In the second stage, the watermark is embedded inside the resulting Wigner distribution. The cells in the joint time-frequency domain, where the watermark is embedded are chosen such that the resultant watermarked distribution is close to a valid Wigner distribution. The third stage involves computing the inverse Wigner transform for the watermarked distribution.

In the proposed method, we assume the size of the host image to be $N \times N$ and the watermark to be $L \leq N$. Moreover, for simplicity and other reasons discussed in the embedding algorithm, we choose $L = N$ unless otherwise stated.

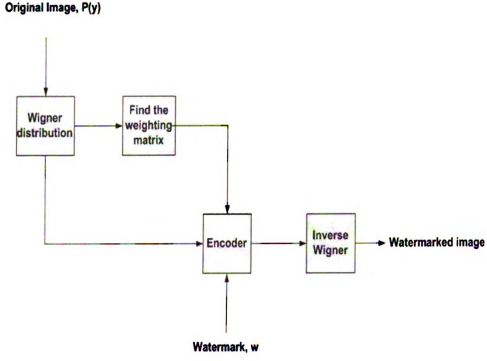

Figure 3.1. The block diagram for the watermark embedding in the Time-Wigner method.

The corresponding embedding algorithm can be summarized as follows.

1. Transform a subset of pixels of length N at least, $P(y)$, chosen randomly from the image, $I(x, y)$, to the time-frequency domain using the Wigner distribution,

$$WD_P(y, \omega_y) = 2 \sum_m P(y+m) P(y-m) e^{-j2\omega_y m}, \tag{3.1}$$

where ω_y is the vertical frequency variable and $WD_P(y, \omega_y)$ is the Wigner distribution of the subset of pixels $P(y)$.

The pixels, $P(y)$, can be chosen randomly to provide more security to the algorithm, or by edge detection algorithms to improve the robustness. In this section, we choose $P(y)$ randomly and the key that contains the locations of the chosen cells is sent as a side information to be used in the decoding stage.

2. Embed the watermark w inside the Wigner distribution $WD_P(y, \omega_y)$,

$$\hat{WD}_P(y, \omega_y) = WD_P(y, \omega_y) + A_P(y, \omega_y) w(y), \tag{3.2}$$

where $A_P(y, \omega_y)$ is a time-frequency dependent weighting matrix that is related to $WD_P(y, \omega_y)$, and and $A_P(y, \omega_y) w(y)$ is an element by element multiplication for every column of $A_P(y, \omega_y)$ and $w(y)$.

The length of $P(y)$ is set to N, which will produce $A_P(y, \omega_y)$ of size $N \times N$. The multiplication in $A_P(y, \omega_y) w(y)$ means that every frequency in $A_P(y, \omega_y)$ is multiplied by same weight determined by the watermark, $w(y)$. This explains why we choose the watermark length to be N. In the case where the watermark length is less than N, we can append zeros to the watermark to get a watermark sequence of length N.

The weighting matrix, $A_P(y, \omega_y)$, is chosen such the the watermarked distribution is very close to a valid Wigner distribution. The specifics of how the weighting matrix is chosen will be explained in detail in Section 3.2.

3. Find the watermarked image by taking the inverse transform assuming equation (3.2) corresponds to a valid Wigner distribution,

$$\hat{P}(y) = \sqrt{\sum_{\omega_y} \hat{WD}_P(y, \omega_y)}. \tag{3.3}$$

Equation (3.3) can be simplified as follows,

$$\begin{aligned}
\hat{P}(y) &= \sqrt{\sum_{\omega_y} \hat{WD}_P(y,\omega_y)}, \\
&= \sqrt{\sum_{\omega_y} \left(WD_P(y,\omega_y) + A_P(y,\omega_y)w(y)\right)}, \\
&= \sqrt{\sum_{\omega_y} \left(2\sum_m P(n+m)P(n-m)e^{-j2\omega_y m} + A_P(y,\omega_y)w(y)\right)}, \\
&= \sqrt{2\sum_m P(n+m)P(n-m)\sum_{\omega_y} e^{-j2\omega_y m} + \sum_{\omega_y} A_P(y,\omega_y)w(y)}, \\
&= \sqrt{2\sum_m P(n+m)P(n-m)\delta(2m) + \sum_{\omega_y} A_P(y,\omega_y)w(y)}, \\
\hat{P}(y) &= \sqrt{P^2(y) + \left(\sum_{\omega_y} A_P(y,\omega_y)\right) w(y)}. \qquad (3.4)
\end{aligned}$$

This simplification reduces the embedding function to a non-linear function in the spatial domain, which is dependent on the weighting matrix, $A_P(y,\omega_y)$. Equation (3.4) was derived with the assumption that the watermarked distribution is a valid Wigner distribution. However there is an error introduced in the inversion process,

$$\mathrm{E} = \overline{WD}_P(y,\omega_y) - \hat{WD}_P(y,\omega_y), \qquad (3.5)$$

where, $\overline{WD}_P(y,\omega_y)$ is the Wigner distribution of $\hat{P}(y)$ and $\hat{WD}_P(y,\omega_y)$ is the watermarked Wigner distribution. This error is saved as a key and sent to the receiver for more accurate watermark extraction.

In the next section, we study this error in more detail and look into its role in determining the weighting matrix.

3.2 Error Introduced in the Inversion Process

The simplification of the embedding function in Section 3.1 was carried out with the assumption that the watermarked distribution is a valid Wigner distribution. However, this assumption is hard to satisfy and an error is introduced in the inversion process. This section gives some insight about this error and its effect on the choice of the weighting matrix, $A_P(y, \omega_y)$.

To study the effect of the approximation in equation (3.4), we look at how different the Wigner distributions of the signal in equation (3.3) is from the Wigner distribution in equation (3.2). Let $A_P(y, \omega_y) = C \cdot WD_P(y, \omega_y)$, where C is a constant and let the Wigner distribution of $\hat{P}(y)$ be $\overline{WD}_P(y, \omega_y)$. Ideally, $\overline{WD}_P(y, \omega_y)$ and $\hat{WD}_P(y, \omega_y)$ should be identical. However, an error E, is introduced by equation (3.3) in the inversion process,

$$\mathrm{E} = \overline{WD}_P(y, \omega_y) - \hat{WD}_P(y, \omega_y). \tag{3.6}$$

In the proposed embedding method, the error E is saved as a key and used for watermark extraction.

To study this error, we compute the Normalized Mean Square Error (NMSE) between $\overline{WD}_P(y, \omega_y)$ and $\hat{WD}_P(y, \omega_y)$,

$$NMSE = \frac{\frac{1}{N^2} \sum_{i}^{N} \sum_{j}^{N} \left(\overline{WD}_P(i,j) - \hat{WD}_P(i,j) \right)^2}{\sum_{i}^{N} \sum_{j}^{N} \hat{WD}_P^2(i,j)}. \tag{3.7}$$

Table 3.1 shows the average NMSE for different images over all time-frequency points. The NMSE is computed from the error introduced in the inversion of the Wigner distribution for different images. The results suggest that the approximation

used for the inversion of the Wigner distribution is valid and introduces a small amount of error.

Table 3.1. The average Normalized Mean Square Error introduced by the approximation of the Wigner distribution in Time-Wigner method.

Image	NMSE	Standard deviation (sd)
Lena	3.21×10^{-8}	4.92×10^{-10}
Barbara	3.07×10^{-8}	4.98×10^{-10}
Camera Man	2.93×10^{-8}	7.64×10^{-11}
Peppers	3.11×10^{-8}	4.78×10^{-10}

Further, we can study the time-frequency locations where the error is concentrated by finding the difference between the two Wigner distributions,

$$WD_D = \overline{WD}_P(y, \omega_y) - \hat{WD}_P(y, \omega_y). \tag{3.8}$$

At each time point, i.e. for every column in $WD_D(y, \omega_y)$, we find the histogram of the maximum differences in equation (3.8) over frequency. Figure 3.2 shows that the maximum error is concentrated around the low frequencies. Since the error is concentrated in the low frequencies, the weighting matrix, $A_P(y, \omega_y)$, can be chosen such that the watermark is embedded in the middle frequency range, which is less affected by this approximation error. The corresponding weighting matrix is,

$$A_P(y, \omega_y) \propto \begin{cases} \frac{WD_P(y,\omega_y)}{\max(WD_P(y,\omega_y))}, & \omega_1 \leq |\omega_y| \leq \omega_2 \\ 0, & \text{elsewhere} \end{cases}, \tag{3.9}$$

where ω_1 and ω_2 are the normalized frequencies that can be determined empirically with typical values of $\omega_1 = \frac{1}{6}$ and $\omega_2 = \frac{1}{3}$. However, since the error is available as side information at the receiver, the condition in equation (3.9) can be relaxed and the the

time-frequency points with the largest values are chosen for watermark embedding.

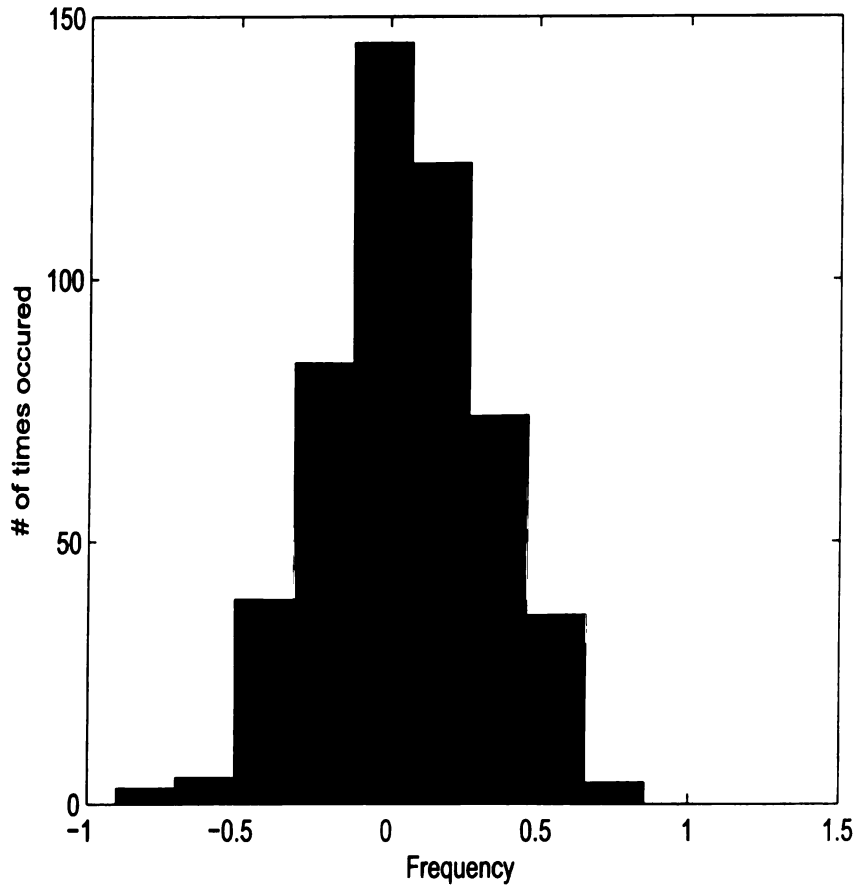

Figure 3.2. The average histogram for the difference of the two Wigner distributions in the Time-Wigner method.

In the following two sections, we derive the probability of error in detecting the watermark for two different cases. In Section 3.3, the performance of the detector for the Gaussian watermark case is given, whereas in Section 3.4, the performance of the detector for the binary watermark case is discussed.

3.3 Watermark Detection for the Gaussian Case

For copyright protection applications, it is important to detect the existence of the watermark, and not necessarily extract the actual watermark, even after the watermarked image is attacked [75, 89]. For these applications, the probability of error in detecting the correct watermark is used as a measure to study the performance of the detection algorithm. Moreover, copyright applications usually have access to the original image and blind watermarking is not that crucial. Thus, in this dissertation, we assume that we have access to the original image, or at least to the pixels used for watermarking.

In this section, we will study the performance of the Time-Wigner watermarking method for a Gaussian distributed watermark sequence. We define two hypotheses: H_1, the hypothesis that the embedded watermark exists and H_0, the hypothesis that there is no watermark embedded or that the embedded watermark is not the one that the detector is testing for. Since we have access to the original image, we can extract a function that depends on the watermark by squaring equation (3.4) and subtracting the square of the image from it,

$$\left(\sum_{\omega_y} A_P(y,\omega_y)\right) w(y) = \hat{P}^2(y) - P^2(y). \tag{3.10}$$

The extracted function is compared with a series of possible watermarks to determine which watermark has been embedded,

$$\left\langle \left(\sum_{\omega_y} A_P(y,\omega_y)\right) w(y), \hat{w}(y) \right\rangle \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array} \eta, \tag{3.11}$$

where $< x_1, x_2 >$ is the inner product of x_1 and x_2, $w(y)$ is the embedded watermark

sequence with variance σ_1^2, $\hat{w}(y)$ is any other watermark sequence with variance σ_2^2 and η is the threshold used to detect the watermark.

By defining the probability of false alarm as P_{FA} and the probability of detection as P_D, the probability of error P_e is written as,

$$P_e = p_0 P_{FA} + p_1(1 - P_D), \tag{3.12}$$

where p_0 and p_1 are the a priori probabilities for H_0 and H_1, respectively.

For the case that the a priori probabilities of H_0 and H_1 are $\frac{1}{2}$,

$$P_e = \frac{1}{2}P\left(\sum_y A_P(y)w(y)\hat{w}(y) > \eta\right) + \frac{1}{2}P\left(\sum_y A_P(y)w^2(y) < \eta\right), \tag{3.13}$$

where $A_P(y) = \sum_{\omega_y} A_P(y, \omega_y)$.

In order to find the threshold η that minimizes P_e, the distribution of $\sum_y A_P(y)w^2(y)$ and the distribution of $\sum_y A_P(y)w(y)\hat{w}(y)$ should be derived. The full derivation is given in the appendix.

Let,

$$z_1 = \sum_y A_P(y)w^2(y). \tag{3.14}$$

The mean and the variance of this random variable are given by,

$$\mu_{z_1} = \sigma_1^2 \sum_y A_P(y), \tag{3.15}$$

$$\sigma_{z_1}^2 = 2\sigma_1^4 \sum_y A_P^2(y). \tag{3.16}$$

Let,

$$z_2 = \sum_y A_P(y)w(y)\hat{w}(y). \tag{3.17}$$

The mean and the variance of z_2, assuming w and $\hat{w}$ are independent, are given by,

$$\mu_{z_2} = 0, \tag{3.18}$$

$$\sigma_{z_2}^2 = \sigma_1^2 \sigma_2^2 \sum_y A_P^2(y). \tag{3.19}$$

For large N, the probability density functions (pdfs) of z_1 and z_2, using the central limit theorem [90], are assumed to be Gaussian,

$$f_{z_1}(z) \sim N(\mu_{z_1}, \sigma_{z_1}), \tag{3.20}$$

$$f_{z_2}(z) \sim N(\mu_{z_2}, \sigma_{z_2}). \tag{3.21}$$

In order to find the minimum probability of error detector, we differentiate P_e with respect to η,

$$\frac{\partial P_e}{\partial \eta} = 0, \tag{3.22}$$

which yields,

$$f_{z_1}(\eta) - f_{z_2}(\eta) = 0. \tag{3.23}$$

Substituting the pdfs of z_1 and z_2 in equation (3.22) and taking the natural log yields,

$$\left[\frac{\sigma_2^2 - 2\sigma_1^2}{4\sigma_1^4\sigma_2^2}\right]\eta^2 - \left[\frac{\sum_y A_P(y)}{2\sigma_1^2}\right]\eta + \frac{\left(\sum_y A_P(y)\right)^2}{4} - \left[\sum_y A_P^2(y)\right]\ln\left(\frac{\sigma_2}{\sqrt{2}\sigma_1}\right) = 0. \tag{3.24}$$

The threshold, that minimizes the probability of error, is given by,

$$\eta = \frac{\sigma_1^2\sigma_2^2\sum_y A_P(y)\left[-1 \pm \sqrt{1+\frac{2\sigma_1^2-\sigma_2^2}{\sigma_2^2}\left[1-4\ln(\frac{\sigma_2}{\sqrt{2}\sigma_1})\frac{\sum_y A_P^2(y)}{(\sum_y A_P(y))^2}\right]}\right]}{2\sigma_1^2-\sigma_2^2}. \tag{3.25}$$

The threshold derived in equation (3.25) is image dependent. This dependency on the image is reflected through the time-frequency weighting matrix, $A_P(y,\omega_y)$. Therefore, the time-frequency distribution of the image is taken into account when choosing the appropriate threshold.

3.4 Watermark Detection for the Binary Watermark Sequence Case

The watermarking algorithm that embeds a Gaussian watermark detects the existence of a specific identification watermark in the multimedia content. It usually serves as an evidence of ownership. On the other hand, the multi-bit watermarking system extracts the embedded watermark and it is usually used for data hiding or ownership declaration. Thus, binary watermarks are preferable over the Gaussian ones in these applications.

In this section, we assume the watermark to be a binary randomly generated sequence of length N. For the Time-Wigner method, we can extract the watermark from equation (3.2) as,

$$\hat{w}(y) = \left(\frac{\hat{WD}_P(y,\omega_y) - WD_P(y,\omega_y) - E(y,\omega_y)}{A_P(y,\omega_y)}\right), \tag{3.26}$$

which can be written as,

$$\hat{w}(y) = \left(\frac{\hat{P}^2(y) - P^2(y) - \sum_{\omega_y} E(y, \omega_y)}{\sum_{\omega_y} A_P(y, \omega_y)} \right), \tag{3.27}$$

where $\hat{w}(y)$ is the extracted watermark after possible attacks and $E(y, \omega_y)$ is the error key.

We assume the attack, $n(y)$, to be Gaussian and independent from the watermark and $\hat{w}(y)$ consists of $\{-1, 1\}$. We can write the extracted watermark as,

$$\hat{w}(y) = w(y) + n(y), \tag{3.28}$$

The detection rule will be,

$$\langle w(y), \hat{w}(y) \rangle \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{3.29}$$

where, $\hat{w}(y) = w(y) + n(y)$ if the watermark is embedded, and $\hat{w}(y) = n(y)$ if no watermark is embedded.

The probability of false alarm can be written as,

$$P_{FA} = P\left(z(y) > \eta\right). \tag{3.30}$$

where,

$$z(y) = \frac{\sum_y w(y)n(y)}{\sqrt{\sum_y w^2(y) \sum_y n^2(y)}}. \tag{3.31}$$

For simplicity, we assume the mean of $n(y)$ to be zero. Thus, the expected value for $n^2(y)$ is σ_n^2. Assuming the length of the watermark is large and using the weak law of large numbers (WLLN), we can use the following approximations,

$$\sum_y n^2(y) = \sigma_n^2 N. \tag{3.32}$$

Moreover, since $w^2(y) = 1$, we can write $\sum_y w^2(y) = N$. Thus, we can rewrite $z(y)$,

$$z(y) \approx \frac{\sum_y w(y)n(y)}{\sigma_n N}. \tag{3.33}$$

For large N we can assume $z(y)$ to have a Gaussian distribution with mean and variance given by,

$$\mu_{z(y)} = 0, \tag{3.34}$$

and

$$\sigma_{z(y)}^2 = \frac{1}{N}, \tag{3.35}$$

Thus, the probability of false alarm can be written as,

$$P_{FA} = Q\left(\eta\sqrt{N}\right), \tag{3.36}$$

where $Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^{+\infty} \exp\left(-\left(\frac{t^2}{2}\right)\right) dt$.

Similarly, to find the probability of miss, we consider,

$$z(y) = \frac{\sum_y w(y)\hat{w}(y)}{\sqrt{\sum_y w^2(y) \sum_y \hat{w}^2(y)}}, \tag{3.37}$$

By noting that $\sqrt{\sum_y \hat{w}^2(y)} = N(1+\sigma_n^2)$ using the WLLN, we can find the mean and the variance for $z(y)$ to be,

$$\mu_{z(y)} = N, \tag{3.38}$$

and

$$\sigma^2_{z(y)} = N\sigma_n^2, \tag{3.39}$$

Thus, the probability of miss is given by,

$$P_M = 1 - Q\left(\frac{\eta - N}{\sqrt{N}\sigma_n}\right). \tag{3.40}$$

The probability of error for correct extraction can then be written as,

$$P_e = \frac{1}{2}\left(Q\left(\eta\sqrt{N}\right) + 1 - Q\left(\frac{\eta - N}{\sqrt{N}\sigma_n}\right)\right), \tag{3.41}$$

where, p_0 and p_1 are assumed to be equal. The probability of error in detecting the correct watermark depends on the watermark length, N, the attack variance, σ_n^2, and the threshold, η. The parameters N and η are user defined, and should be chosen in a way that reduces the probability of error.

3.5 Simulation Results and Comparison

This section provides simulation results to demonstrate the performance of the proposed method for the binary watermark case. Similar simulations can be carried out for the Gaussian watermark case. Although the Time-Wigner method has been applied to a large number of natural gray-scale images [91], in this section we give a detailed performance analysis for the 512×512 Lena image and a randomly generated binary watermark of length 256. The performance measures discussed in Section 1.6 are used to evaluate the performance of the proposed method.

The capacity of the embedding algorithm has been studied, where different watermarks with different lengths have been embedded and the corresponding PSNRs between the watermarked and the original image are computed. The probability of false alarm derived in Section 3.4 is compared with the experimental results for different attacks. The proposed method is tested under different attacks including Additive White Gaussian Noise (AWGN), median filtering, rotation, and JPEG compression with different compression ratios (CRs). The well-known Discrete Cosine Transform (DCT) method by Cox et al. [63] has been implemented for comparison.

The proposed watermark embedding algorithm has been applied to a large number of images. Table 3.2 shows the average bit error rates (BERs) under different attacks. Since the performance of the algorithm does not vary much with the choice of the image (as can be seen in Table 3.2), in the rest of this section we focus on the performance of the algorithm for the Lena image.

Table 3.2. Average bit error rate in detecting the watermark under different attacks using 100 different images.

Attack	BER
AWGN (PSNR=48.13db)	0.0059±0.0021
AWGN (PSNR=36.0db)	0.0094±0.0045
AWGN(PSNR=14.15db)	0.0432±0.0121
JPEG (CR=1.7)	0.0078±0.0034
JPEG (CR=7.7)	0.0125±0.0085
JPEG (CR=20)	0.0134±0.0087
MF (3×3)	0.0016±0.0013
MF (5×5)	0.0041±0.0022
MF (7×7)	0.0066±0.0024

3.5.1 The Choice of the Watermark Length

In order to determine how many bits to embed in the host image, we calculated the Peak Signal to Noise Ratio (PSNR) between the host and the watermarked images for different watermark lengths. Figure 3.3 shows that even when we embed a large number of bits, such as 2048 bits, the PSNR values remain in the 50dB range. The PSNR values vary from 100dB for the case of 16 bits to 54dB for the 2048 bits case.

Moreover, to study the performance of the detector derived in Section 3.4 for different watermark lengths, we find the ROC curves for different values of N. For different watermark lengths, N, the probability of false detection and the probability of false alarm in the presence of no attack are found for different thresholds. Figure 3.4 shows that larger watermark lengths provide better ROC curves. In fact, for watermarks of length greater than 128, the ROC curve starts to approach the ideal.By looking back to equations (3.36) and (3.40), we can see that increasing N will increase the probability of detection, $1 - P_M$, and decrease the probability of false alarm, P_{FA}, for a given threshold, η and a fixed σ_n. Thus, increasing N will produce less probability of error, which agrees with the experimental results in Figure 3.4.

In the following simulations, we select the length of the watermark to be 256 for a targeted PSNR of 62dB. Figure 3.5 shows the original Lena image, while Figure 3.6 shows the watermarked Lena image. It is clear that there is no visual difference between the original and the watermarked images which satisfies the invisibility of the watermark. As mentioned earlier, the proposed Time-Wigner method has been tested under different types of attacks including additive white Gaussian noise (AWGN), median filtering, rotation, and JPEG compression. In the following subsections, we discuss each attack in detail.

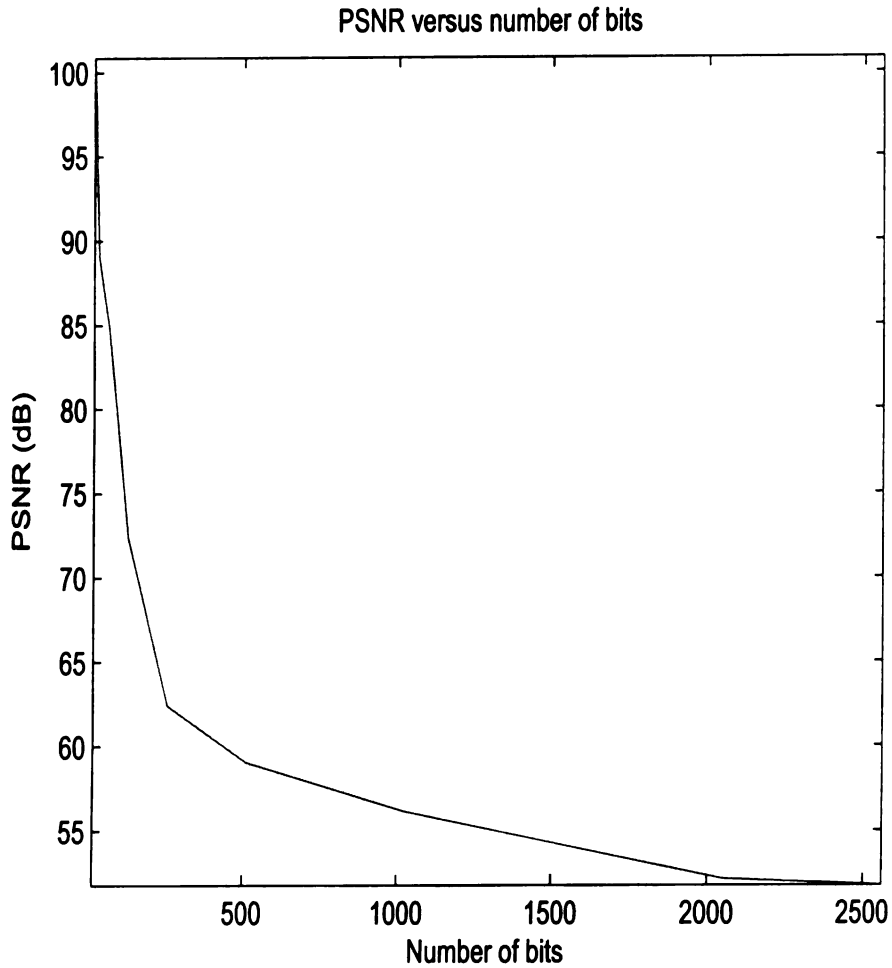

Figure 3.3. PSNR versus number of bits.

3.5.2 Additive White Gaussian Noise (AWGN)

An additive white Gaussian noise with different noise levels was added into the watermarked image. The extracted watermark was correlated with 100 randomly generated watermarks at the receiver. The correlation detector has the highest value at the watermark number 50 which corresponds to the actual embedded watermark as shown in Figure 3.7. Figure 3.8 shows the attacked watermarked image under AWGN with PSNR=14.15dB. The extracted watermark produces the highest correlation even under this amount of distortion.

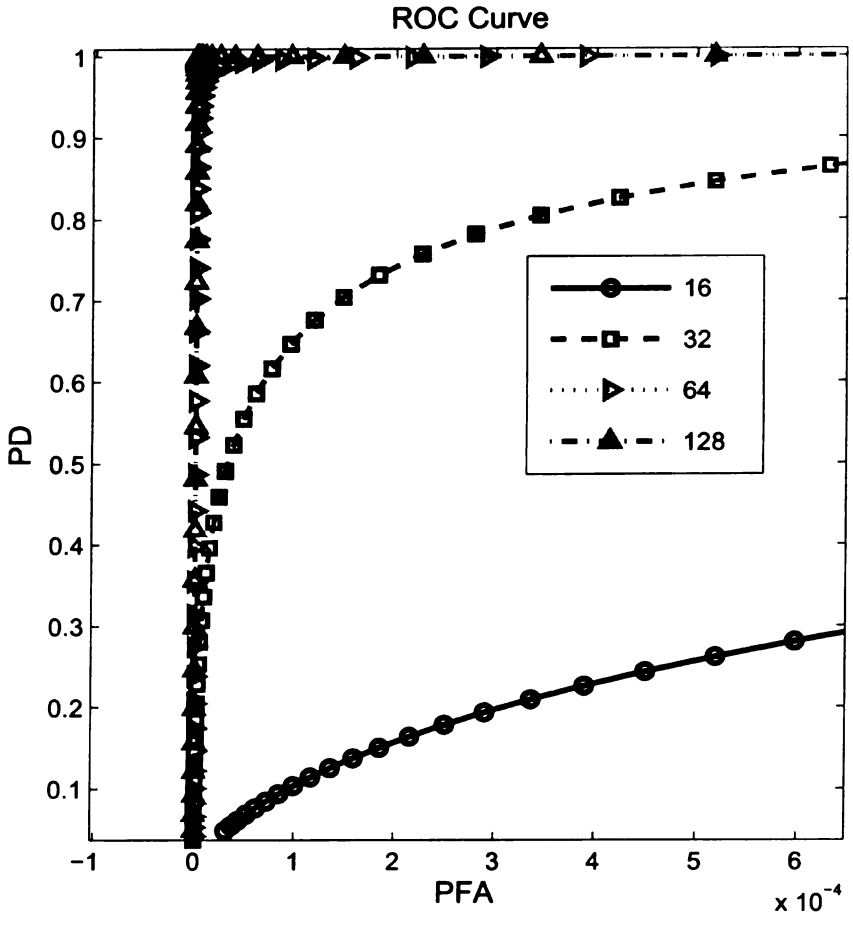

Figure 3.4. The ROC curves for different watermark lengths.

Moreover, the correctness of equation (3.36), $P_{FA} = Q\left(\eta\sqrt{N}\right)$, was validated through simulation. Figure 3.9 compares the probability of false alarm computed from the analytic expression and the simulation results under AWGN with PSNR=28.18dB. The graph shows that the analytical and the experimental curves are very close to each other, which validates the assumptions in Section 3.4.

3.5.3 Median Filtering

The second type of attack applied to the watermarked image is the median filtering. A median filter of size $F \times F$ is applied to the watermarked image. The correlation

Figure 3.5. The original Lena512 image.

detector results for median filtering attack are shown in Figure 3.10. The desired watermark is detected even when the watermarked image is degraded significantly with median filter of size 16×16. Choosing the pixels to be watermarked randomly improve the robustness of the proposed method under median filtering, since the median filtering attack with small filter size will most likely produce pixels with values close to the watermarked pixels before attack. Thus, the Wigner distribution of the attacked pixels and the pixels before the attack will be very close to each other and this will lead to an accurate extraction of the watermark. This robustness stays

Figure 3.6. The watermarked Lena512 image with PSNR=62dB.

till the median filtering attack start changing the selected pixels by big values. Figure 3.11 shows the degraded watermarked image for 9×9 median filtering, while Figure 3.12 validates again the correctness of equation (3.36) for median filtering.

3.5.4 Rotation

Rotation attack with different rotation angles is also applied to the watermarked image. The watermarked image is rotated counter clock-wise using the bilinear interpolation method. The proposed Time-Wigner method successfully detects the embedded watermark even under large rotation angles, i.e 7°, as seen in Figure 3.13.

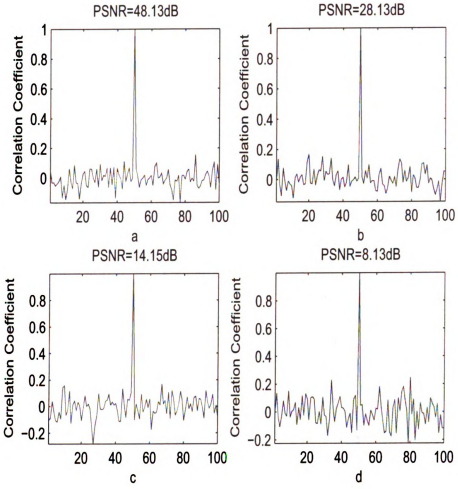

Figure 3.7. The normalized correlation detector response for the Time-Wigner method applied to Lena512 image under AWGN with different PSNRs, a. PSNR=48.13dB,b. PSNR=28.13dB, c. PSNR=14.15dB, d. PSNR=8.13dB.

Equation (3.36) was verified for rotation attack as shown in Figure 3.15.

3.5.5 JPEG Compression

One of the most important attacks that an image watermarking algorithm should survive is the JPEG compression. The watermarked image was compressed with JPEG at different compression ratios. Similar to other attacks, the detection of the watermark was accurate even at high compression ratios, as shown in Figure 3.16.

Figure 3.8. The watermarked image degraded by AWGN (PSNR=14.15dB).

Figure 3.17 shows the attacked image having visible blocking artifacts and yet the algorithm is still able to detect the watermark. Again, equation (3.36) was verified for JPEG attack as shown in Figure 3.18

3.5.6 Comparison between the Time-Wigner and the Spread Spectrum Methods

For the comparison with the DCT method proposed by Cox [63], a watermark sequence of length 256 is embedded in the 256 highest magnitude coefficients in the (DCT) of the Lena image. Figure 3.19 through Figure 3.21 show the correlation

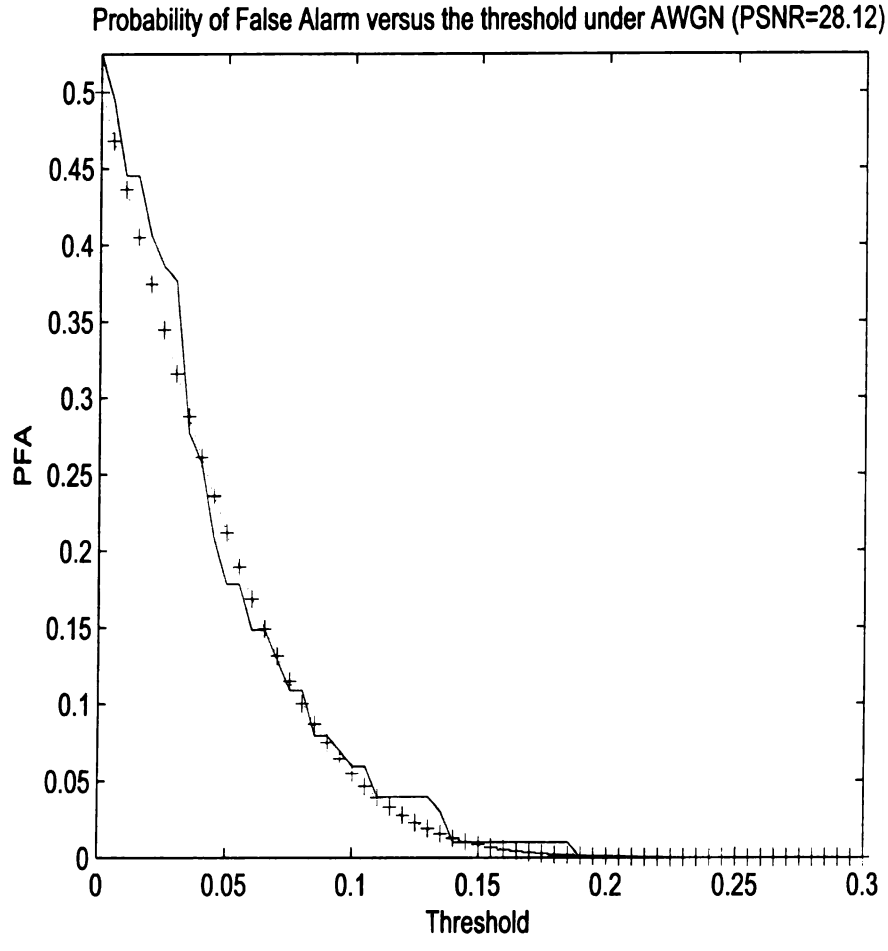

Figure 3.9. The probability of false alarm versus the threshold under AWGN with PSNR=28.18dB.

detector response for both methods under AWGN, median filtering and JPEG compression respectively. The proposed Time-Wigner method performs better than the DCT method under all attacks. Both, the Time-Wigner and DCT methods perform well under AWGN. In the median filtering case, the proposed method has higher correlation coefficients under all filter sizes. The DCT method embeds the watermark in the largest magnitude DCT coefficients of the host image, which requires the use of small weighting coefficients to provide an invisible watermark. Thus, the strength

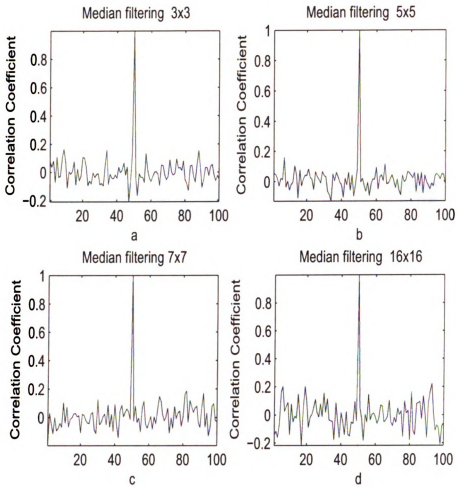

Figure 3.10. The normalized correlation detector response for the Time-Wigner method applied to Lena512 image under median filtering with different filter sizes, a. size=3 × 3,b. size=5 × 5, c. size=7 × 7, d. size=16 × 16.

of the watermark will be low and any distortion in the image will affect the detection of the watermark. The choice of the weighting matrix along with the use of the error key improve the robustness and hence the detection of the watermark in the proposed Time-Wigner method.

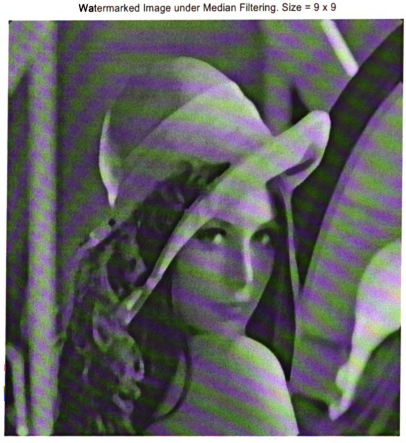

Figure 3.11. The watermarked image degraded by median filter of size 9×9.

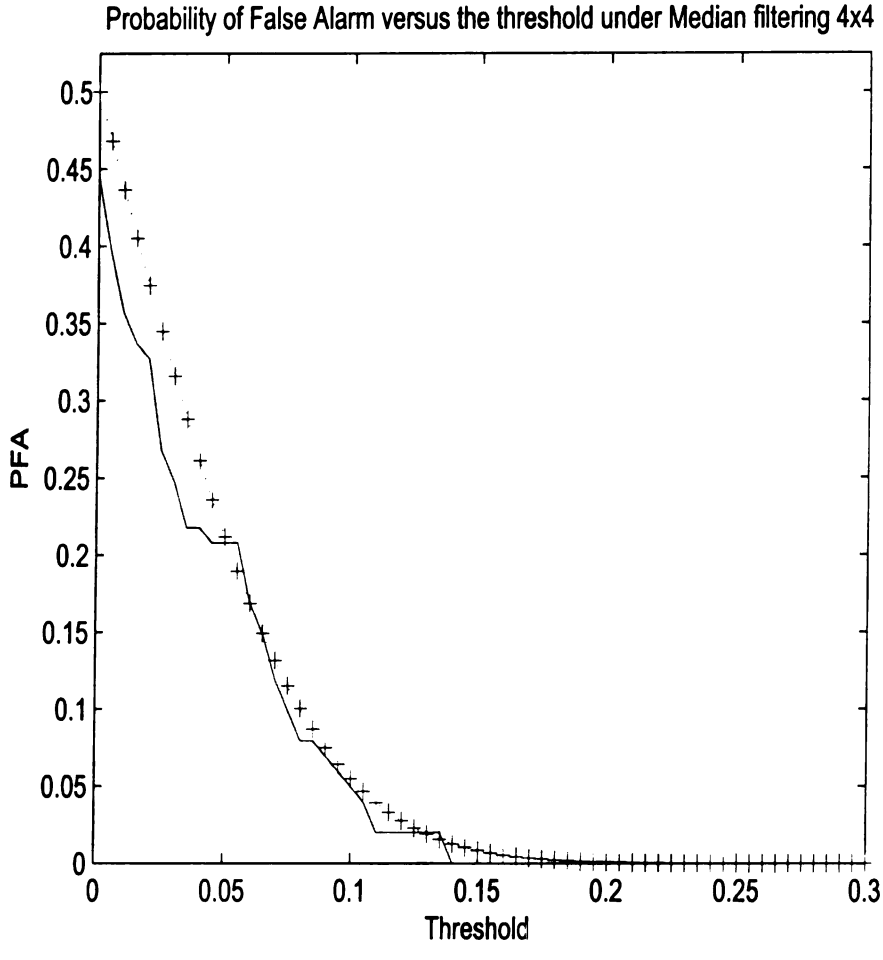

Figure 3.12. The probability of false alarm versus the threshold under median filtering with filter size=4 × 4.

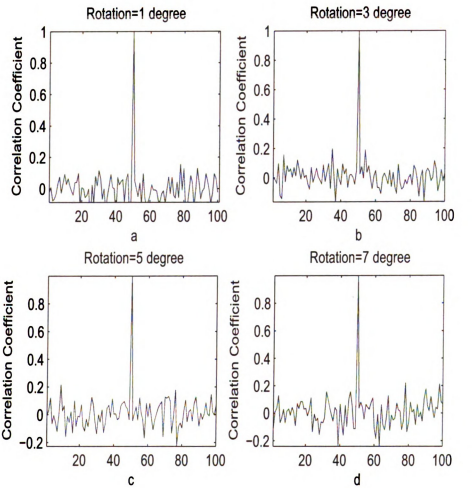

Figure 3.13. The normalized correlation detector response for the Time-Wigner method applied to Lena512 image under rotation with different angles, a. degree=1°,b. degree=3°, c. degree=5°, d. degree=7°.

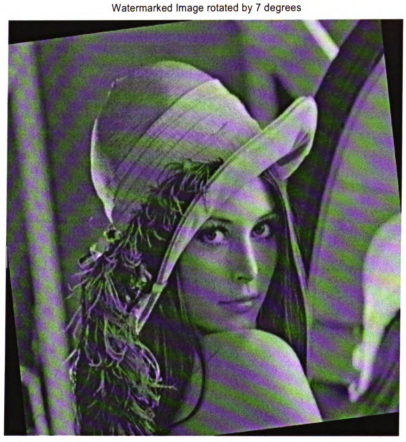

Figure 3.14. The watermarked image degraded by rotation of 7°.

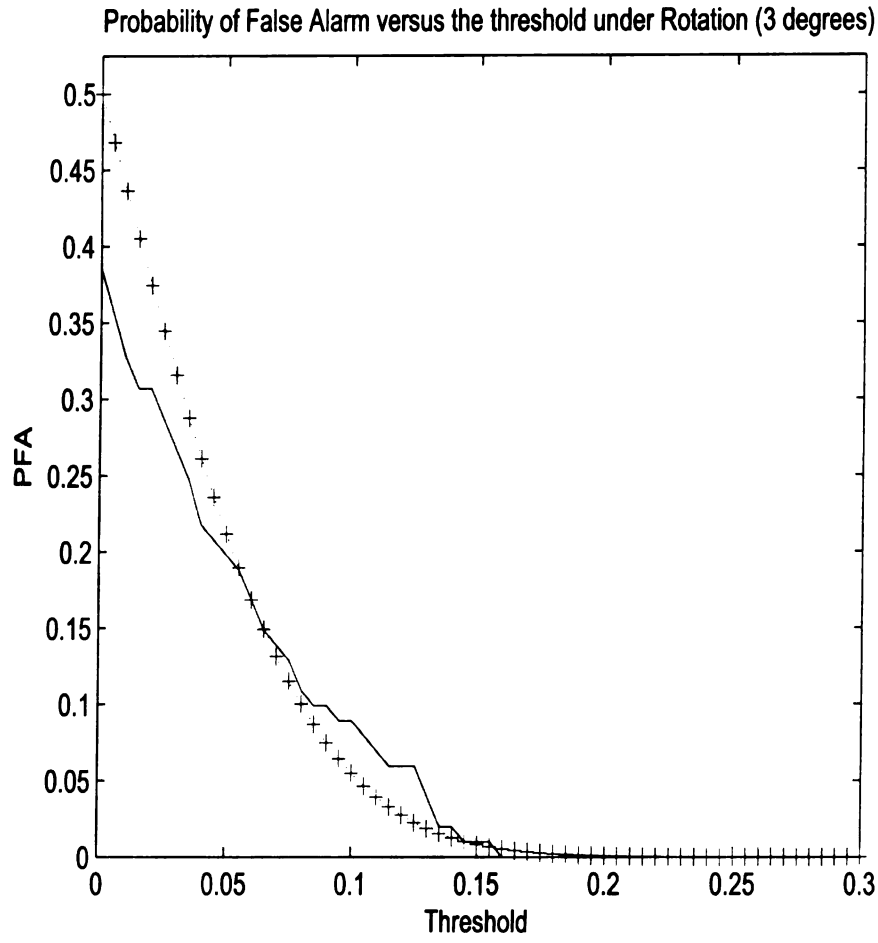

Figure 3.15. The probability of false alarm versus the threshold under rotation of 3°.

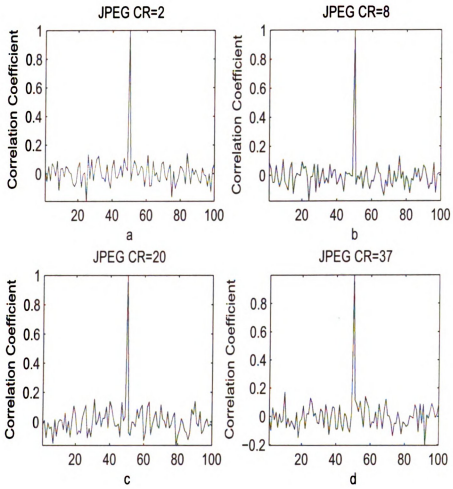

Figure 3.16. The normalized correlation detector response for the Time-Wigner method applied to Lena512 image under JPEG compression with different compression ratios, a. CR=2,b. CR=8, c. CR=20, d. CR=37.

Figure 3.17. The watermarked image degraded by JPEG compression with CR=37.

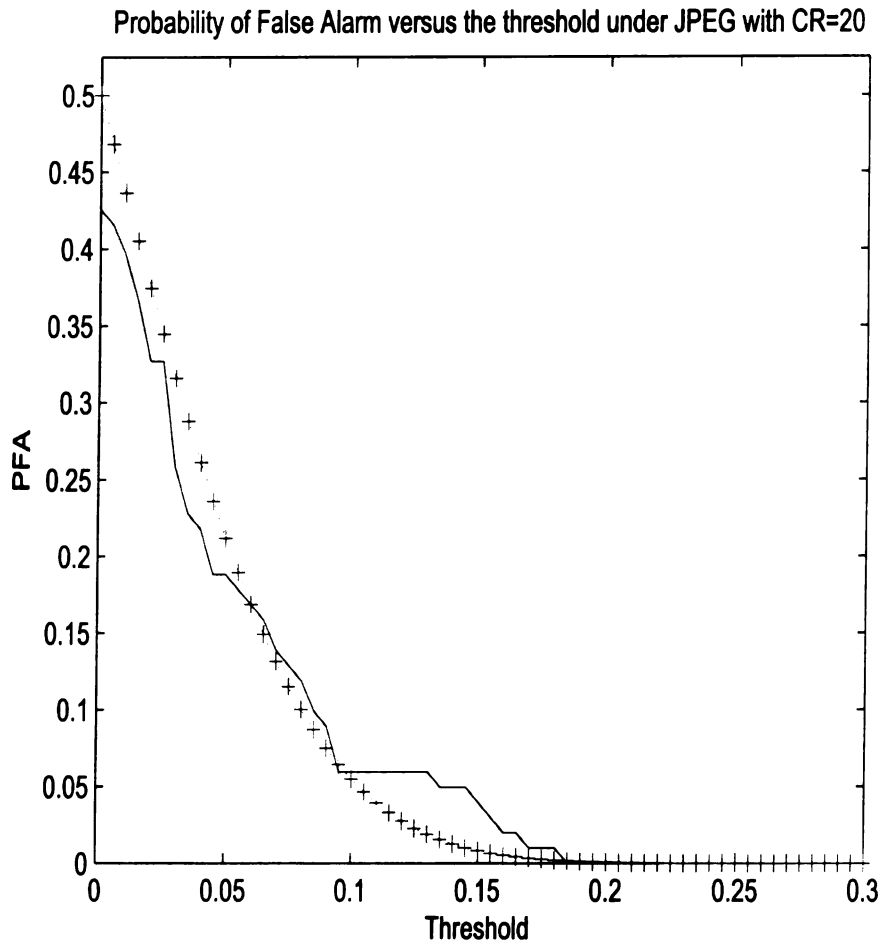

Figure 3.18. The probability of false alarm versus the threshold under JPEG compression with CR=20.

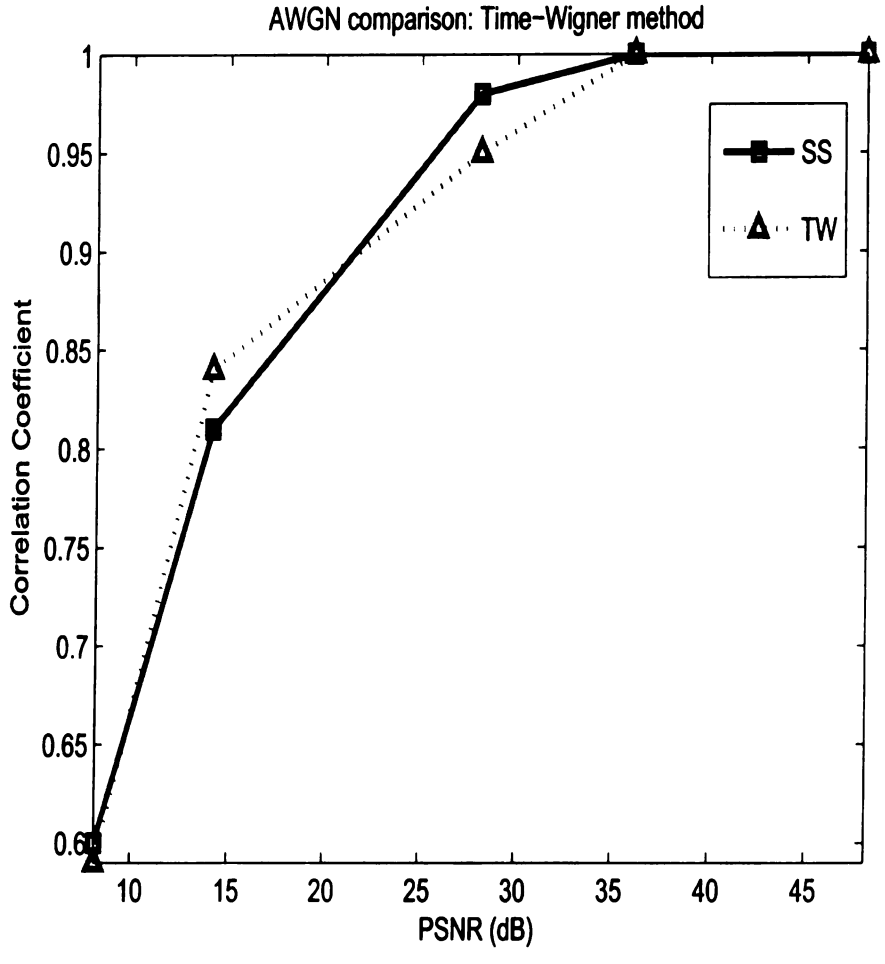

Figure 3.19. Comparison between spread spectrum and Time-Wigner methods under AWGN.

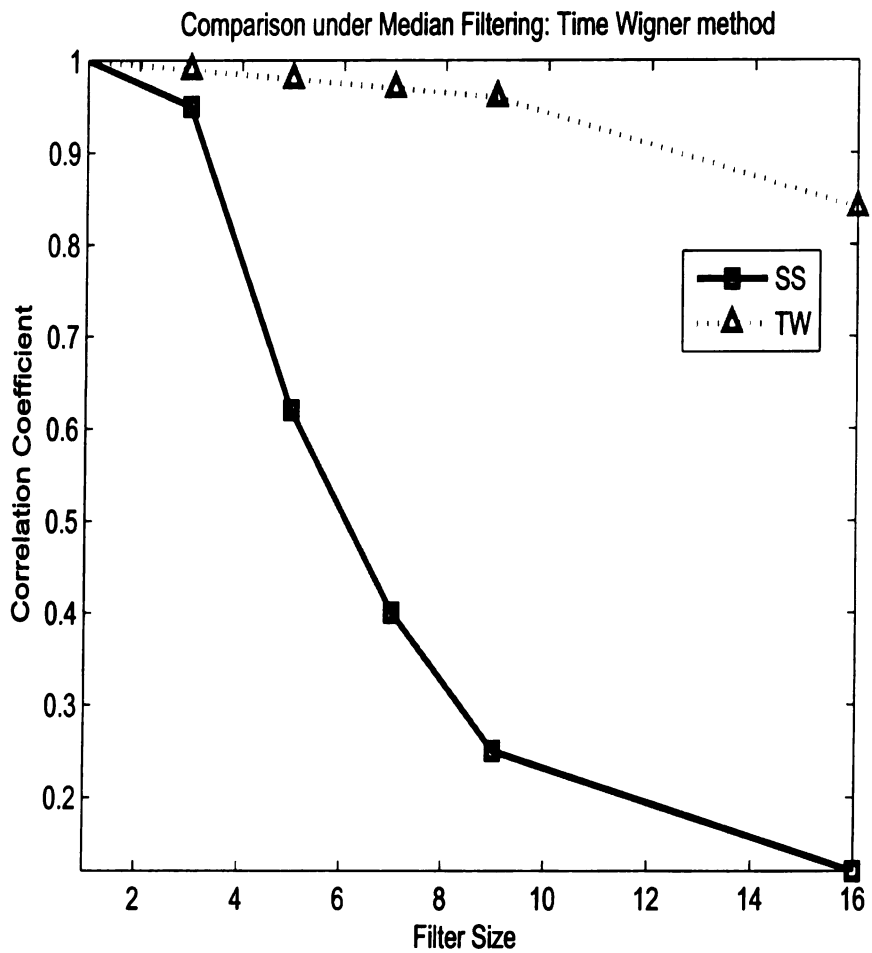

Figure 3.20. Comparison between spread spectrum and Time-Wigner methods under Median Filtering.

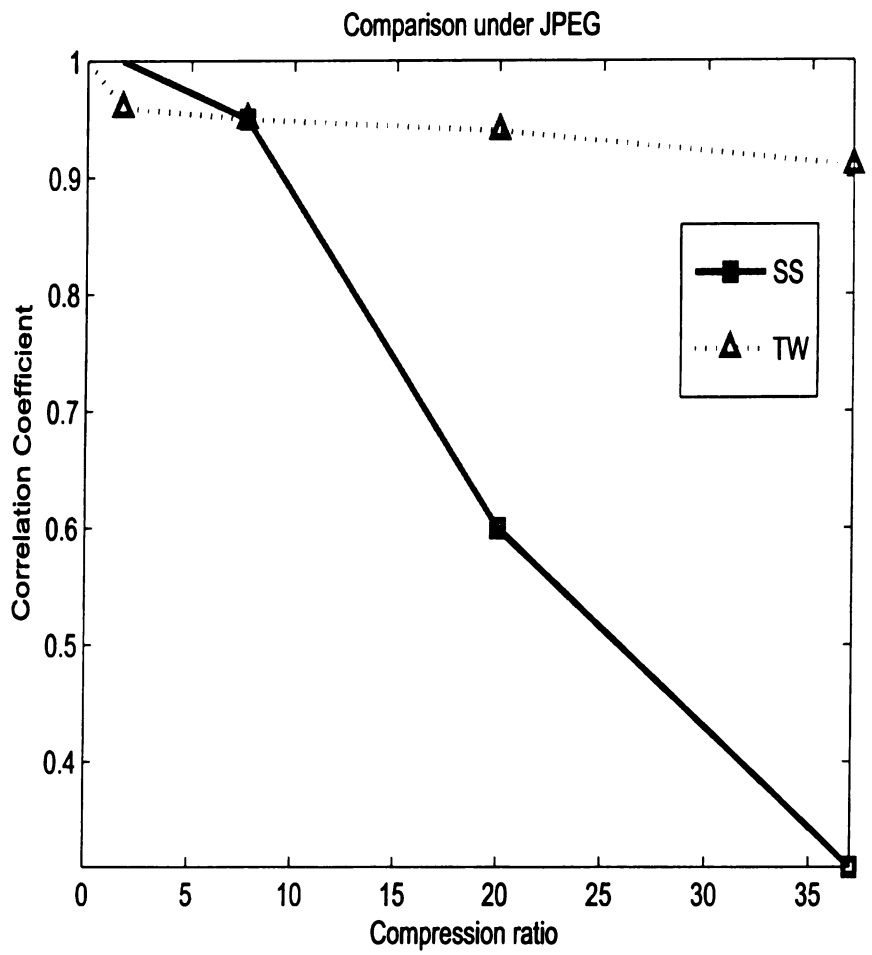

Figure 3.21. Comparison between spread spectrum and Time-Wigner methods under JPEG compression.

3.6 Discussion

The main attribute of the Time-Wigner method, is the fact that it could be implemented in the time domain directly. This is due to the fact that the time-frequency domain embedding function can be simplified to non-linear function in time,

$$\hat{P}(y) = \sqrt{P^2(y) + \left(\sum_{\omega y} A_P(y, \omega_y)\right) w(y)}. \tag{3.42}$$

This simplified function uses the time-frequency characteristics of the image through the weighting matrix $A_P(y, \omega_y)$.

For many watermarking applications, the embedding process can be done off line, so the time required to embed the watermark does not matter. This means that whether we use the non-simplified or the simplified version of the embedding function will not have any effect on the transmitter (encoder). The difference appears when we detect/extract the watermark. If we use the non-simplified function in the time-frequency domain, we need to find the Wigner distribution of the image, or of the subset of pixels from the image used for watermarking, which requires a lot of online computations. On the other hand, if we use the simplified function, the Wigner distribution is not needed to detect the watermark and thus we reduce the number of computations.

The Time-Wigner method has the flexibility of embedding a Gaussian or binary watermark. Depending on the application, one can choose whether to embed a Gaussian or binary watermark sequence. The simulation results provided in Section 3.5 are for the binary sequence case. Similar simulations can be carried out for the Gaussian case. In this case, equation (3.25), which is dependent on η, is used for

watermark detection,

$$\eta = \frac{\sigma_1^2\sigma_2^2\sum_y A_P(y)\left[-1 \pm \sqrt{1+\frac{2\sigma_1^2-\sigma_2^2}{\sigma_2^2}\left[1-4\ln(\frac{\sigma_2}{\sqrt{2}\sigma_1})\frac{\sum_y A_P^2(y)}{(\sum_y A_P(y))^2}\right]}\right]}{2\sigma_1^2-\sigma_2^2}. \tag{3.43}$$

For the special case when $\sigma_1^2 = \sigma_2^2 = \sigma^2$. The threshold, η, reduces to,

$$\eta = \sigma^2\sum_y A_P(y)\left[-1+\sqrt{\left[2-4\ln(\frac{1}{\sqrt{2}})\frac{\sum_y A_P^2(y)}{(\sum_y A_P(y))^2}\right]}\right]. \tag{3.44}$$

The special case for the threshold derived for the Gaussian watermark case, in equation (3.44), assumes that $\sigma_1^2 \neq \sigma_2^2$ is sufficient to distinguish between the true watermark and the false one at the receiver. Since the watermark is a Gaussian random variable with zero mean, it is distinguished through its variance. So, if two Gaussian random variables have two different variances, then these two random variables are not identical. In other words, the detector will not falsely detect the watermark if $\sigma_1^2 \neq \sigma_2^2$, because it will recognize that the detected watermark is the false one and ignore it. The main confusion occurs when $\sigma_1^2 = \sigma_2^2$ and using equation (3.44) will enable the detection of the watermark. The detection of the watermark is dependent on the right choice for η. For example, equation (3.44) determines the value for η that provides the minimum probability of error, P_e.

3.7 Summary

In this chapter, we proposed a new image watermarking method in the Wigner domain. The proposed Time-Wigner method in the time-frequency domain was sim-

plified to a non-linear function in the time domain. The simplified function uses the time-frequency characteristics of the image through the weighting matrix $A_P(y, \omega_y)$. This simplification is based on the assumption that the watermarked distribution is a valid Wigner distribution. However, this is not always true and an error is introduced by the inversion process. The introduced error is analyzed and found to be concentrated in the low frequency range. This error is saved as a key and used for watermark extraction/detection at the receiver.

Two detection algorithms are derived. The first one assumes the watermark to be a Gaussian sequence, while the second one assumes the watermark to be a binary sequence. Depending on the application, one can choose whether to embed a Gaussian or a binary watermark sequence. Gaussian watermarks are suitable for applications where detecting the watermark is the main goal, like broadcast monitoring and owner identification. On the other hand, binary watermarks are used, in addition to the previous applications, in covert communication. In covert communication, the watermark is a secret message which contains some information for a specific usage, i.e military.

To evaluate the performance of the proposed Time-Wigner method, a binary watermark sequence is embedded inside Lena image. The robustness of the proposed algorithm under attacks is shown through extensive simulations. The watermark is successfully detected, even under severe distortions. Moreover, a comparison between the Time-Wigner and the DCT methods is carried out. The DCT method is chosen because of its robustness and the similarities it has with the Time-Wigner method in terms of the way the watermark is spread over the image. In general, the proposed Time-Wigner method performs better than the DCT method under all attacks discussed in this chapter. In addition, Time-Wigner method has more flexibility in choosing the number of bits to be embedded and still retain high PSNR values. Although Time-Wigner and DCT methods use non-blind algorithms for detecting the

watermark, the proposed Time-Wigner method requires only the subset of pixels used for watermark embedding to detect the watermark, $P(y)$, while the DCT requires the whole image at the receiver.

CHAPTER 4

THE WIGNER-WIGNER WATERMARKING METHOD

In the previous chapter, the Time-Wigner method was shown to have high robustness. However, once the Wigner distribution of the image is found, the watermark may be detected/extracted using an estimation attack. In order to provide more security to the Time-Wigner method, we propose and evaluate a novel time-frequency watermarking algorithm using Wigner distribution. Unlike the Time-Wigner method, in this chapter the Wigner distribution of the watermark is embedded inside the Wigner distribution of a subset of pixels, $P(y)$, chosen from the image, $I(x, y)$. This method, the Wigner-Wigner, can be considered as an improved version of the Time-Wigner method to increase the security of the watermark. Embedding the Wigner distribution of the watermark inside the Wigner distribution of the image makes the watermark more secure, since extracting or detecting the watermark involves evaluating the Wigner distribution for both $P(y)$ and the watermark.

Similar to Chapter 3, two detection algorithms for the Gaussian and binary watermark cases are derived. In addition, we compare the performance of the proposed method with a similar watermarking algorithm in the multiresolution domain in [67], to demonstrate the robustness and the potential of the proposed method.

This chapter is organized as follows. Section 4.1 gives a detailed analysis of the watermarking embedding algorithm in the Wigner domain. It shows that the Wigner-Wigner watermarking method in the time-frequency domain is equivalent to a nonlinear embedding function in the time domain. In Section 4.2, the error introduced in the inversion of the watermarked distribution from the time-frequency domain to the time domain, and the choice of the weighting matrix are discussed. Sections 4.3 and 4.4, analyze the performance of the proposed Wigner-Wigner method for the Gaussian

distributed watermark case and the binary watermark case, respectively. Section 4.5 provides simulation results to demonstrate the performance of the proposed method under attacks. A comparison between the Wigner-Wigner method and a DWT-based watermarking method is given. While Section 4.6 discusses some key points in the proposed algorithm, Section 4.7 summarizes the major contributions of this chapter.

4.1 Watermark embedding

In the Wigner-Wigner method, the Wigner distribution of the watermark is embedded into the Wigner distribution of the image. The block diagram for this method is given in Figure 4.1. The embedding algorithm in the Wigner-Wigner method has four main stages. Similar to the Time-Wigner method, the first stage transforms a subset of the pixels of the image to the Wigner domain. In the second stage, the watermark is transformed to the Wigner domain. In the third stage, the Wigner distribution of the watermark is embedded inside the Wigner distribution of the chosen subset. The last stage involves finding the inverse Wigner transform for the watermarked distribution.

Similar to the Time-Wigner method, we assume the size of the host image to be $N \times N$ and the watermark to be $L \leq N$. Moreover, for simplicity, we choose $L = N$ unless otherwise stated. The watermark embedding algorithm can then be summarized as follows:

1. Transform a subset of pixels, $P(y)$, chosen randomly from the image, $I(x, y)$, to the time-frequency domain using Wigner distribution,

$$WD_P(y, \omega_y) = 2 \sum_m P(y+m)P(y-m)e^{-j2\omega_y m}. \tag{4.1}$$

   The pixels, $P(y)$, are chosen randomly and the key that contains the locations of the chosen cells is sent as a side information to be used for watermark detection.

2. Transform the watermark sequence, w, to the time-frequency domain using

Wigner distribution,

$$WD_w(y,\omega_y) = 2\sum_m w(y+m)w^*(y-m)e^{-j2\omega_y m}. \tag{4.2}$$

3. Embed the Wigner distribution of the watermark sequence inside the Wigner distribution of $P(y)$,

$$\hat{WD}_P(y,\omega_y) = WD_P(y,\omega_y) + A_P(y,\omega_y) \odot WD_w(y,\omega_y), \tag{4.3}$$

where $A_P(y,\omega_y) \odot WD_w(y,\omega_y)$ is an element by element multiplication. The weighting matrix, $A_P(y,\omega_y)$, is again chosen such the the watermarked distribution is very close to a valid Wigner distribution. Unlike the Time-Wigner method in this case, the Wigner distribution of the watermark is spread out on the whole time-frequency plane. The specifics of how the weighting matrix is chosen will be explained in detail in Section 4.2.

4. Obtain the watermarked image by taking the inverse transform assuming equation (4.3) is still a valid Wigner distribution,

$$\hat{P}(y) = \sqrt{\sum_{\omega_y} \hat{WD}_P(y,\omega_y)}. \tag{4.4}$$

The embedding algorithm described in equation (4.4) can be simplified as follows,

$$
\begin{aligned}
\hat{P}(y) &= \sqrt{\sum_{\omega_y} \hat{WD}_P(y,\omega_y)}, \\
&= \sqrt{\sum_{\omega_y} \left(WD_P(y,\omega_y) + A_P(y,\omega_y) WD_w(y,\omega_y)\right)}, \\
&= \sqrt{2\sum_{m} P(n+m)P(n-m)\delta(2m) + \sum_{\omega_y} A_P(y,\omega_y) * w^2(y)}, \\
&= \sqrt{P^2(y) + \left(\sum_{\omega_y} A_P(y,\omega_y)\right) * w^2(y)}, \\
\hat{P}(y) &= \sqrt{P^2(y) + \left(\sum_{\omega_y} A_P(y,\omega_y)\right) * w^2(y)}, \qquad (4.5)
\end{aligned}
$$

where $*$ corresponds to convolution. Similar to the Time-Wigner method, the simplification reduces $\hat{P}(y)$ to a non-linear function of the image and the watermark sequence in the spatial domain. The time-frequency dependence of the embedding function is through the time-frequency dependent weighting matrix $A_P(y,\omega_y)$. Equation (4.5) assumes the watermarked distribution is a valid Wigner distribution. However, there is an error introduced in the inversion process,

$$
\mathrm{E} = \overline{WD}_P(y,\omega_y) - \hat{WD}_P(y,\omega_y), \qquad (4.6)
$$

where, $\overline{WD}_P(y,\omega_y)$ is the Wigner distribution of $\hat{P}(y)$ and $\hat{WD}_P(y,\omega_y)$ is the watermarked Wigner distribution. This error is saved as a key and sent to the receiver for more accurate watermark extraction.

4.2 Error Introduced in the Inversion Process

Similar to the Time-Wigner method, the inversion of the watermarked distribution assumes that the distribution in equation (4.3) is a valid Wigner distribution. However, as we have shown in the Time-Wigner method, this is usually not true, and an error is introduced in the inversion process. In this section, we study the effect of this approximation by looking at how different the Wigner distribution of the signal in equation (4.4) is from the Wigner distribution in equation (4.3).

Let $A_P(y,\omega_y) = C \cdot WD_P(y,\omega_y)$, where C is a weighting constant and let the Wigner distribution of $\hat{P}(y)$ be $\overline{WD}_P(y,\omega_y)$. Ideally, $\overline{WD}_P(y,\omega_y)$ and $\hat{WD}_P(y,\omega_y)$ should be identical. However, an error E, which again is kept as a key, is introduced by equation (4.4) in the inversion process. In order to quantify this error, E, we compute the Normalized Mean Square Error (NMSE) between $\overline{WD}_P(y,\omega_y)$ and $\hat{WD}_P(y,\omega_y)$.

Table 4.1 shows the average NMSE for different images over all time-frequency points. The NMSE is computed from the error introduced in the inversion of the Wigner distribution for different images. Similar to the results of the Time-Wigner method, the error introduced in the inversion process in the Wigner-Wigner method is small, which validates the approximation used for the inversion of the Wigner distribution.

Table 4.1. The average Normalized Mean Square Error introduced by the approximation of the Wigner distribution in Wigner-Wigner method.

Image	NMSE	Standard deviation (sd)
Lena	3.11×10^{-8}	5.02×10^{-10}
Barbara	3.02×10^{-8}	4.92×10^{-10}
Camera Man	2.99×10^{-8}	8.21×10^{-11}
Peppers	2.99×10^{-8}	4.67×10^{-10}

In order to study the time-frequency locations where the error is concentrated, we find the difference between the two Wigner distributions,

$$WD_D = \overline{WD}_P(y, \omega_y) - \hat{WD}_P(y, \omega_y). \tag{4.7}$$

At each time point, i.e. for every column in $WD_D(y, \omega_y)$, we find the histogram of the maximum differences in equation (4.7) over frequency. Figure 4.2 shows that the maximum error is concentrated, once again and similar to the Time-Wigner method, around the low frequencies. It is also clear that since the Wigner-Wigner method spreads the watermark throughout the image, the error is more spread out compared to the Time-Wigner method. Since the error is concentrated in the low frequencies, we will choose the weighting matrix such that the watermark is embedded in the middle frequency range, which is less affected by this approximation error. The corresponding weighting matrix is,

$$A_P(y, \omega_y) \propto \begin{cases} \frac{WD_P(y,\omega_y)}{max(WD_P(y,\omega_y))}, & \omega_1 \leq |\omega_y| \leq \omega_2 \\ 0, & \text{elsewhere} \end{cases}, \tag{4.8}$$

where ω_1 and ω_2 are the normalized frequencies that can be determined empirically with typical values of $\omega_1 = \frac{1}{6}$ and $\omega_2 = \frac{1}{3}$.

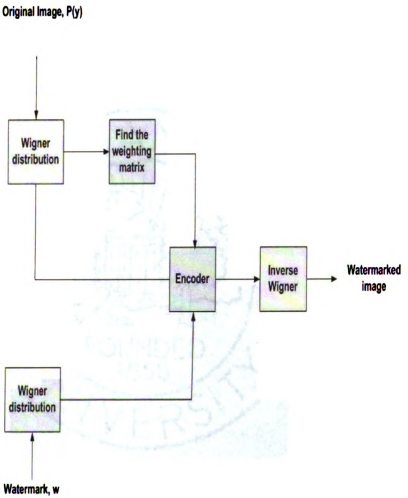

Figure 4.1. The block diagram for the watermark embedding in the Wigner-Wigner method.

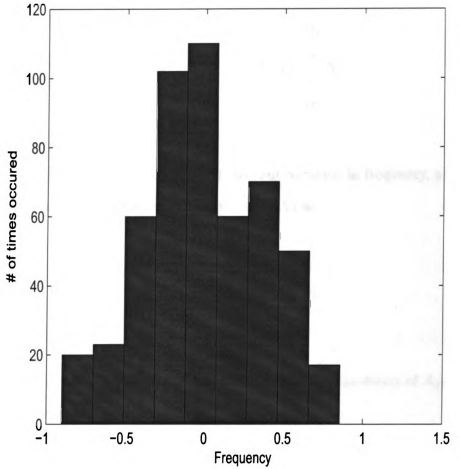

Figure 4.2. The average histogram for the difference of the two Wigner distributions in the Wigner-Wigner method.

4.3 Watermark Detection for the Gaussian Case

In this section, we derive the probability of error in detecting a Gaussian distributed watermark. Similar to Section 3.3, we can detect the watermark in the Wigner-Wigner method using a correlation function derived from equation (4.5) by subtracting the square of the original $P(y)$,

$$\left\langle A_P(y) * w^2(y), \hat{w}^2(y) \right\rangle \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{4.9}$$

where $A_P(y) * w^2(y) = \hat{P}^2(y) - P^2(y)$.

Since convolution in time corresponds to multiplication in frequency, and in order to simplify the derivation we rewrite equation (4.9) as,

$$\langle C(n)Y_1(n), Y_2(n) \rangle \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{4.10}$$

where $C(n)$, $Y_1(n)$ and $Y_2(n)$ correspond to the Fourier transforms of $A_P(y)$, $w^2(y)$ and $\hat{w}^2(y)$, respectively.

To find the minimum probability of error detector. Let,

$$z_1 = \sum_n C(n)Y_1^2(n). \tag{4.11}$$

The mean and the variance of z_1 are given by,

$$\mu_{z_1} = N\sigma_1^4 \left[2\sum_n C(n) + NC(0) \right], \tag{4.12}$$

$$\sigma_{z_1}^2 = 8N^2\sigma_1^8\left[\sum_n C^2(n) + NC^2(0)\right]. \tag{4.13}$$

Readers should refer to the appendix for full derivation. Let,

$$z_2 = \sum_n C(n)Y_1(n)Y_2(n). \tag{4.14}$$

The mean and the variance of this random variable are given by,

$$\mu_{z_2} = N^2\sigma_1^2\sigma_2^2C(0), \tag{4.15}$$

$$\sigma_{z_2}^2 = 4N^2\sigma_1^4\sigma_2^4\left[\sum_n C^2(n) + NC^2(0)\right]. \tag{4.16}$$

Using the central limit theorem, the pdfs of z_1 and z_2 are assumed to be Gaussian,

$$f_{z_1}(z) \sim N(\mu_{z_1}, \sigma_{z_1}). \tag{4.17}$$

$$f_{z_2}(z) \sim N(\mu_{z_2}, \sigma_{z_2}). \tag{4.18}$$

Since $C(n)$ is the Fourier transform of $A_P(y)$, the following relationships hold,

$$\begin{aligned} C(0) &= \sum_y A_P(y), \\ \sum_n C(n) &= NA_P(0), \\ \sum_n C^2(n) &= N\sum_y A_P^2(y). \end{aligned} \tag{4.19}$$

Solving for η using the above facts and the assumption, from equation (4.8), that

$\left(\sum_y A_P(y)\right)^2 >> \sum_y A_P^2(y)$, we get,

$$\eta \approx \frac{N^2 \sum_y A_P(y)\sigma_1^4 \left[\left(\frac{2\sigma_1^2}{\sigma_2^2} - 1\right) \pm \sqrt{2}\left(\frac{\sigma_1^2}{\sigma_2^2} - 1\right)\right]}{\left(\frac{2\sigma_1^4}{\sigma_2^4} - 1\right)}. \tag{4.20}$$

The threshold in equation (4.20), similar to the Time-Wigner method, depends on the weighting matrix $A_P(y, \omega_y)$. Thus, the spatial and the spectral characteristics of the image are taken into account when choosing the appropriate threshold.

4.4 Watermark Detection for the Binary Watermark Sequence Case

The simplified embedding function in equation (4.5) shows that the square of the watermark is used for the embedding, which in the case of a binary sequence of $\{-1, 1\}$ makes the extraction of the watermark impossible because the sign will be lost through the square operation. Therefore, in the Wigner-Wigner method, pre-processing and post-processing steps are introduced to account for this ambiguity. The pre-processing shifts bit -1 to 0, so the embedded watermark is a sequence of $\{0, 1\}$ instead of $\{-1, 1\}$. Equation (4.5) can then be written as,

$$\hat{P}(y) = \sqrt{P^2(y) + \left(\sum_{\omega_y} A_P(y, \omega_y)\right) * w(y)}, \tag{4.21}$$

since $w^2(y) = w(y)$.

At the receiver, the extracted watermark is post-processed by converting every bit 0 to -1. Once the extracted watermark is found with the post-processing step, the same procedure described for Time-Wigner watermark extraction can be applied to

find the probabilities of false alarm and miss, where the extracted watermarked is,

$$\hat{w}(y) = w(y) + n(y). \tag{4.22}$$

Similar to the Time-Wigner method, the probability of error for correct extraction is,

$$P_e = \frac{1}{2}\left(Q\left(\eta\sqrt{N}\right) + 1 - Q\left(\frac{\eta - N}{\sqrt{N}\sigma_n}\right)\right), \tag{4.23}$$

where η is a pre-defined threshold and σ_n^2 is the attack variance.

4.5 Simulation Results and Comparison

In this section, we provide simulation results to demonstrate the performance of the proposed embedding algorithm and the use of reference watermark. The wavelet-based method in [67], has been implemented for performance comparison. In [67], the authors propose a DWT based watermarking algorithm based on quantizing certain DWT coefficients at each level. The DWT method embeds the watermark in the wavelet domain of the image, which reveals the characteristics of the image at different scales. Similarly, the Wigner-Wigner method embeds the watermark in the time-frequency domain, which reveals the characteristics of the image at different frequency components at different times.

In order to determine how many bits we can embed inside an image and keep a high PSNR at the same time, different watermarks with different lengths are embedded inside the host image. Figure 4.3 shows the PSNR values for different watermark lengths. The PSNR values are in 70dB range even for large watermark lengths, i.e. 2048.

The proposed watermark embedding algorithm has been applied to a large number of images [91]. Table 4.2 shows the average bit error rates (BERs) under different

attacks. Since the performance of the algorithm does not vary much with the choice of the image (as can be seen in Table 4.2), in the rest of this section we focus on the performance of the algorithm for the Lena image.

Table 4.2. Average bit error rate in detecting the watermark under different attacks using 100 different images.

Attack	BER
AWGN (PSNR=48.13db)	0.0065±0.0031
AWGN (PSNR=36.0db)	0.0091±0.0056
AWGN(PSNR=14.15db)	0.1821±0.0425
JPEG (CR=1.7)	0.0075±0.0031
JPEG (CR=7.7)	0.0151±0.0061
JPEG (CR=20)	0.0514±0.0134
MF (3 × 3)	0.0182±0.0101
MF (5 × 5)	0.0415±0.0121
MF (7 × 7)	0.0574±0.0210

A binary watermark of length 256 is embedded into the Lena image resulting in a PSNR of 80.2dB. The watermarked image in Figure 4.4 has no visible differences from the original image, which satisfies the imperceptibility condition.

4.5.1 The Performance under AWGN, Median Filtering, Rotation, and JPEG Compression

The performance of the Wigner-Wigner method is demonstrated under similar attacks as in the Time-Wigner case, i.e. AWGN, median filtering, rotation, and JPEG attacks. The extracted watermark was correlated with 100 randomly generated watermarks at the receiver. The performance of the correlation detector is similar to the Time-Wigner method, where the extracted watermark has the highest correlation, when it is correlated with the true watermark, among all possible watermarks at the receiver. The correlation detector response for sample attacks is shown in Figure 4.5.

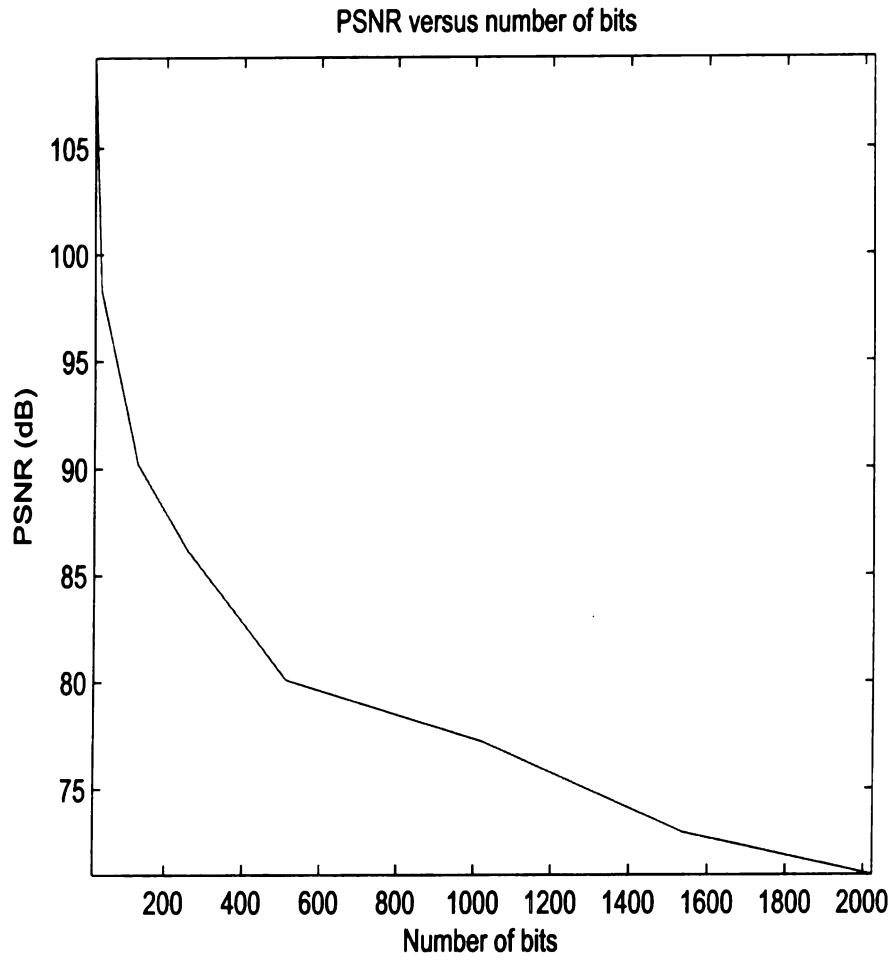

Figure 4.3. PSNR versus number of bits.

4.5.2 Comparison between the Wigner-Wigner and the Wavelet Methods

For the comparison with the DWT method proposed by Kundur [67], a binary watermark sequence of length 256 is embedded. Figure 4.6 through Figure 4.8 show the correlation detector response for both methods under AWGN, median filtering, and JPEG compression respectively. The proposed Wigner-Wigner method performs better than the DWT method under all attacks. The DWT-based method embeds the watermark in every DWT level, where some of these levels are less robust to attacks. Moreover, the Wigner-Wigner method embeds the Wigner of the watermark inside

Figure 4.4. The watermarked Lena512 image with PSNR=80.2dB.

the Wigner distribution of the image, while the DWT embeds the binary watermark itself in the DWT domain, which gives more security to the Wigner-Wigner method. The use of the error key in extracting/detecting the watermark in the Wigner-Wigner method improves the robustness of the proposed algorithm. In addition, the DWT-based method sends more side information in order to extract the watermark. The locations of all modified coefficients in every level at each scale of the DWT should be available at the receiver. This large amount of side information is a drawback for the DWT-based method. Although both methods have the ability of embedding binary

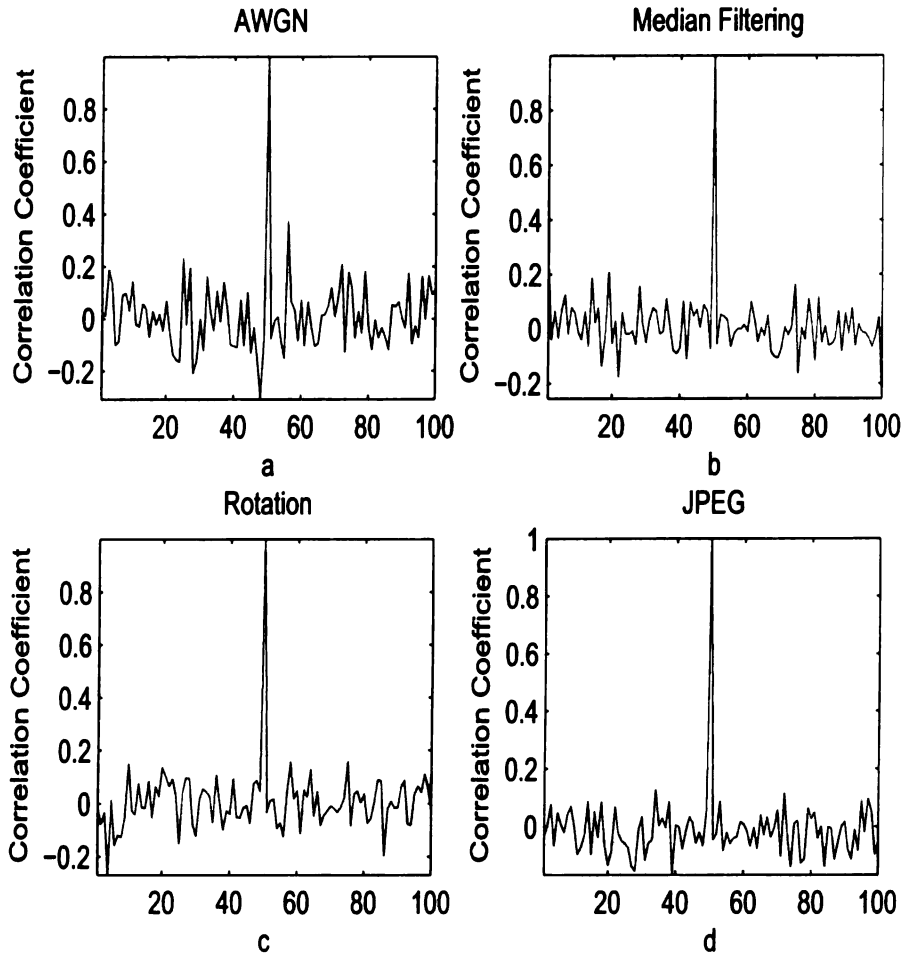

Figure 4.5. The normalized correlation detector response for the Wigner-Wigner method applied to Lena512 image under, a. AWGN=14.5dB,b. Median Filtering size=7 × 7, c. Rotations 1°, d. JPEG CR=20.

watermark sequences, the Wigner-Wigner method has the advantage of embedding Gaussian sequences as well, which makes it suitable for more applications.

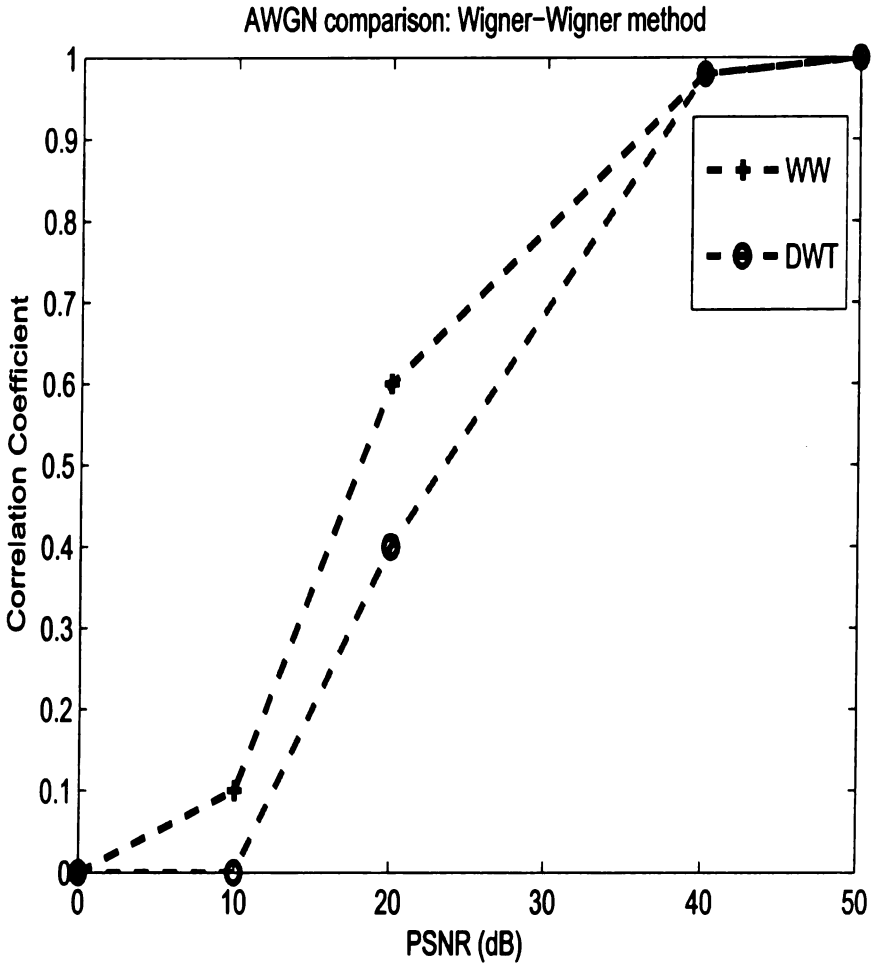

Figure 4.6. Comparison between the DWT and Wigner-Wigner methods under AWGN.

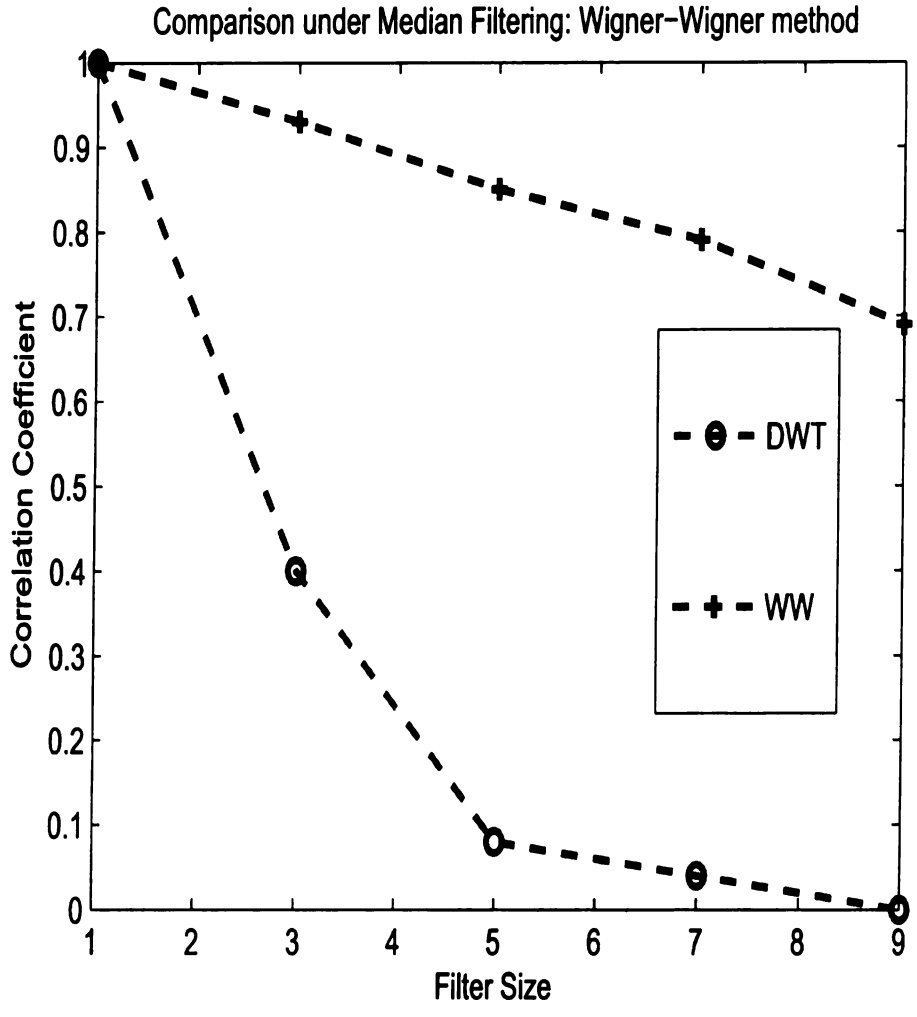

Figure 4.7. Comparison between the DWT and Wigner-Wigner methods under Median Filtering.

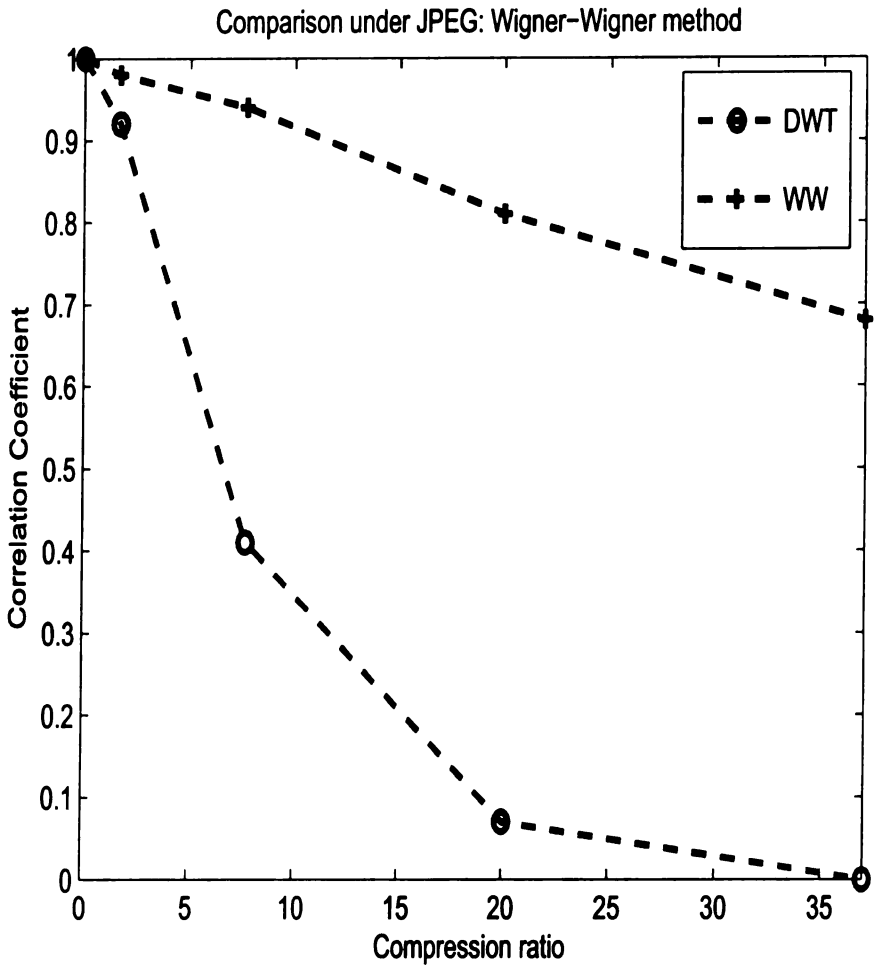

Figure 4.8. Comparison between the DWT and Wigner-Wigner methods under JPEG compression.

4.6 Discussion

The proposed Wigner-Wigner watermarking method, which is an extension of the Time-Wigner method, embeds the Wigner distribution of the watermark inside the image. Therefore, the Wigner-Wigner method uses the time-frequency characteristics for both the image and the watermark, while the Time-Wigner method uses the time-frequency information only for the image. Similar to the Time-Wigner method, the time-frequency domain embedding algorithm in the Wigner-Wigner method is simplified to a non-linear function in time that depends on the square of the watermark,

$$\hat{P}(y) = \sqrt{P^2(y) + \left(\sum_{\omega_y} A_P(y, \omega_y)\right) * w^2(y)}. \tag{4.24}$$

The simplified function can be used for watermark embedding, instead of the time-frequency domain function, to reduce the number of computations.

The Winer-Wigner method has also the ability of embedding both Gaussian and binary watermark sequences. In this case, the threshold that is used to detect the watermark is,

$$\eta \approx \frac{N^2 \sum_y A_P(y)\sigma_1^4 \left[\left(\frac{2\sigma_1^2}{\sigma_2^2} - 1\right) \pm \sqrt{2}\left(\frac{\sigma_1^2}{\sigma_2^2} - 1\right)\right]}{\left(\frac{2\sigma_1^4}{\sigma_2^4} - 1\right)}. \tag{4.25}$$

The threshold in equation (4.25) shows that the Wigner-Wigner method depends on N^2, which makes it more dependent on the length of the subset, $P(y)$, compared to the Time-Wigner method. Moreover, finding the threshold and implementing the detector in the Time-Wigner method is simpler because of the convolution operation in the Wigner-Wigner method, which makes the detection more complicated as we need to find the Fourier transforms for $A_P(y)$, $w^2(y)$ and $\hat{w}^2(y)$.

4.7 Summary

In this chapter, we introduced a novel watermarking algorithm based on embedding the Wigner distribution of the watermark inside the Wigner distribution of the signal. The embedding algorithm in the time-frequency domain is simplified to a non-linear function in time that depends on the square of the watermark,

$$\hat{P}(y) = \sqrt{P^2(y) + \left(\sum_{\omega_y} A_P(y, \omega_y)\right) * w^2(y)}. \tag{4.26}$$

This square operation makes the extraction of the binary watermark impossible, since the sign is lost. Thus, we proposed a pre-processing and post-processing operations, where the binary sequence of $\{-1, 1\}$ is shifted to $\{0, 1\}$ at the embedder, and the received watermark is converted back to $\{-1, 1\}$ at the receiver. This simplification, similar to the Time-Wigner method, is based on the assumption that the watermarked distribution is a valid Wigner distribution. However, since this is hard to satisfy, an error is introduced by the inversion process. The introduced error is analyzed and found to be concentrated in the low frequency range. This error is saved as a key and used for watermark extraction/detection at the receiver.

In this chapter, similar to the previous chapter, two detection algorithms are derived for the Gaussian and the binary watermarks cases, respectively. The performance of the proposed Wigner-Wigner method is evaluated through embedding a binary watermark sequence. The correct detection of the embedded watermark is validated after attacks.

The proposed watermarking method is compared with a well-known DWT-based method. Although the watermark sequence in the DWT is repeatedly embedded in every scale of the DWT of the image, our proposed method, which embeds the watermark just once, outperforms the DWT method in all attacks. The proposed method reduces redundancy in the algorithm and provides higher accuracy and security at

the expense of increased computational complexity.

CHAPTER 5

WATERMARKING IN THE AUTOCORRELATION DOMAIN

The major challenge for watermarking in the Wigner domain discussed in Chapters 3 and 4, is that once the Wigner distribution is watermarked, it is no longer a valid distribution and the time signal, $s(n)$, can be recovered using the approximation in equation (2.6). This approximation introduces some error in detecting or extracting the watermark. This error is sent as a key to add robustness to the detection of the watermark. Therefore, one of the biggest shortcomings of the Wigner-based methods is that the original image and an extra key that contains the error in inverting the Wigner distribution are needed for watermark detection. Another shortcoming of the Wigner-based watermarking methods is that the computation of the weighting matrix in the Wigner-based method makes it unsuitable for real-time applications, where the watermark is required to be embedded in a very short period of time. The non-blind detection algorithm is another disadvantage for the Wigner-based methods. These constraints limit the use of the Wigner-based method in certain applications such as real-time verification systems.

In this chapter, we introduce a new image watermarking method that is equivalent to watermarking in the Wigner domain without the limitations mentioned above. Unlike the Wigner-based methods, the proposed method can embed only a binary watermark sequences. The binary watermark is embedded in the local autocorrelation domain, which is related to the Wigner distribution through a Fourier transform and has no aliasing and invertibility problems. The pixels to be watermarked are chosen randomly from the original image. This ensures the security of the embedded watermark. The time-varying autocorrelation function for the chosen pixels is found and the watermark is embedded such that the modified autocorrelation is still a valid

autocorrelation function. This will ensure the invertibility of the autocorrelation function and will enable us to extract the embedded watermark bits. The robustness of the proposed autocorrelation based watermarking method under different attacks such as rotation, filtering, AWGN, and JPEG compression is evaluated by computing the probability of error.

This chapter is organized as follows. Section 5.1 gives a brief introduction on the autocorrelation function. Section 5.2 introduces the embedding algorithm, whereas Section 5.3 introduces the extraction algorithm. In Section 5.4, the analysis of the proposed method under attacks is provided. Simulation results and comparison with other well-known methods are demonstrated in Section 5.5. Some final remarks and discussion are given in Section 5.6. Finally, Section 5.7, summarizes the main points of this chapter.

5.1 Background

The use of autocorrelation function for watermarking has been previously mentioned in literature [92, 93]. The autocorrelation function used in these papers is the regular autocorrelation function which represents the well-known correlation based detector, that has a peak when the extracted watermark is correlated with the original one. Unlike this autocorrelation function, the autocorrelation function used for watermarking in this chapter represents a time-varying function that is related to the Wigner distribution through an inverse Fourier transform.

It is obvious from equation (2.3),

$$WD(n,\omega) = 2\sum_{m=-\infty}^{\infty} s(n+m)s^*(n-m)e^{-j2m\omega}, \tag{5.1}$$

that the Wigner distribution is the Fourier transform of a time-varying autocorrelation function $r(m,n) = s(n+m)s^*(n-m)$. The autocorrelation function has some

properties that make it a good choice for watermarking applications. First, for real and positive-valued discrete time signals, such as images, the signal can be retrieved from its autocorrelation function as $s(n) = \sqrt{r(n,0)}$. Second, the autocorrelation function of a real signal is symmetric. These properties simplify the embedding and detection algorithms in image watermarking.

Since Wigner distribution is the Fourier transform of every other row of the autocorrelation matrix, embedding the watermark in the Wigner distribution is equivalent to embedding it in the autocorrelation domain. The autocorrelation function for a discrete-time signal of length M can be written as:

$$r(m,n) = s(n+m)s^*(n-m), \tag{5.2}$$

where $m = \lfloor \frac{-M}{2} \rfloor, \frac{-M}{2}+2, ..., \lfloor \frac{M}{2} \rfloor$, $n = 1, ..., M$ and $\lfloor x \rfloor$ is the largest integer less than or equal to x.

The autocorrelation written in matrix form for the signal $s(n) = \{s_1 \ s_2 \ s_3 \ s_4 \ s_5\}$ is:

$$r(m,n) = \begin{bmatrix} 0 & 0 & s_1s_5 & 0 & 0 \\ 0 & s_1s_3 & s_2s_4 & s_3s_5 & 0 \\ s_1^2 & s_2^2 & s_3^2 & s_4^2 & s_5^2 \\ 0 & s_1s_3 & s_2s_4 & s_3s_5 & 0 \\ 0 & 0 & s_1s_5 & 0 & 0 \end{bmatrix}. \tag{5.3}$$

The symmetry of $r(m,n)$ and the invertibility, $s(n) = \sqrt{r(0,n)}$, are clear from the above example. Since $r(m,n)$ is symmetric, we consider only the positive indices,

$$r_+(m,n) = \begin{bmatrix} s_1^2 & s_2^2 & s_3^2 & s_4^2 & s_5^2 \\ 0 & s_1 s_3 & s_2 s_4 & s_3 s_5 & 0 \\ 0 & 0 & s_1 s_5 & 0 & 0 \end{bmatrix}. \tag{5.4}$$

The proposed method modifies the non-zero elements of $r_+(m,n)$ for $m > 0$. $r(0,n)$ is not modified directly to preserve the visual quality of the watermarked image. For a signal of length M, we have $\frac{M^2-2M+1}{4}$ watermarkable points. For the example given in equation (5.3), the number of watermarkable locations in the autocorrelation function will be 4, which correspond to $s_1 s_3$, $s_2 s_4$, $s_3 s_5$ and $s_1 s_5$.

5.2 Watermark Embedding

For simplicity, we consider an $N \times N$ image and a binary watermark sequence w of length L consisting of {-1,1}. The embedding algorithm which is illustrated in Figure 5.1, can be summarized as follows:

1. Choose randomly a subset of pixels from the original image. This subset should have at least $2\sqrt{L}+1$ points from the image to ensure that we have L watermarkable cells by the relationship given in the previous section. In this dissertation, we choose $2L+1$ points, $s(n) = \{s_1, s_2, ..., s_{2L+1}\}$, which give L^2 watermarkable cells. This will provide a degree of freedom in choosing the locations to insert the L watermark bits in the next step. A key K_s containing the locations of the selected pixels is stored.

2. Compute the autocorrelation function $r_+(m,n)$ for $s(n)$ and choose the watermarkable locations according to a randomly generated key K_r. The key, K_r, should choose L distinct locations among the L^2 non-zero points in the autocorrelation function with the exception of the row at $m = 0$.

3. Since every coefficient in $r_+(m,n)$ is a product of two points of the original

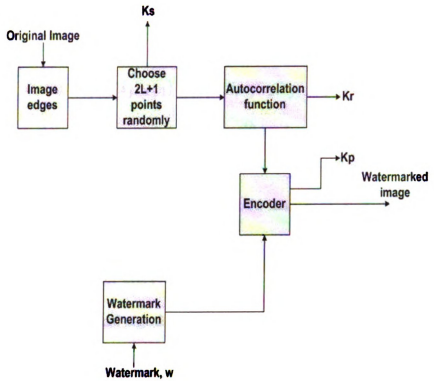

Figure 5.1. The block diagram for the embedding algorithm for the autocorrelation method.

signal $s(n)$, we can write:

$$r_+(i,j) = s_{i+j-1}s_{j-i+1}, \tag{5.5}$$

where $i = 1, 2, ..., L+1$ and $j = 1, 2, ..., 2L+1$.

4. Fix $\min(s_{i+j-1}, s_{j-i+1})$ and modify $\max(s_{i+j-1}, s_{j-i+1})$. We keep the pixel with minimum value unmodified since any small change in a small valued pixel

will cause a perceptible distortion in the watermarked image. Also, modifying large values will provide more robustness against attacks. The embedding process can be written as,

$$\max(\hat{s}_{i+j-1}, \hat{s}_{j-i+1}) = \max(s_{i+j-1}, s_{j-i+1})(1 + \alpha_l w_l), \tag{5.6}$$

where α_l is the weighting coefficient of the l^{th} bit for $l = 1, 2, 3, ..., L$. The value of $\max(s_{i+j-1}, s_{j-i+1})$ before modification is stored in a key, K_p.

5. Modify all locations in $r_+(n, m)$ that contain $\max(s_{i+j-1}, s_{j-i+1})$ with the new value of $\max(\hat{s}_{i+j-1}, \hat{s}_{j-i+1})$.

6. Repeat steps $(3 - 5)$ for every watermark bit.

7. Obtain the watermarked signal $\hat{s}(n)$ by taking the square root of $r_+(0, n)$.

The weighting coefficient α_l is derived such that the $\max(\hat{s}_{i+j-1}, \hat{s}_{j-i+1})$ remains greater than $\min(s_{i+j-1}, s_{j-i+1})$. This ensures that the watermark extraction is possible. The weighting coefficients are derived as follows:

$$\max(s_{i+j-1}, s_{j-i+1})(1 + \alpha_l w_l) \geq \min(s_{i+j-1}, s_{j-i+1}). \tag{5.7}$$

$$\alpha_l w_l \geq \frac{\min(s_{i+j-1}, s_{j-i+1})}{\max(s_{i+j-1}, s_{j-i+1})} - 1. \tag{5.8}$$

If $w_l = 1$ then,

$$\alpha_l \geq \frac{\min(s_{i+j-1}, s_{j-i+1})}{\max(s_{i+j-1}, s_{j-i+1})} - 1. \tag{5.9}$$

If $w_l = -1$ then,

$$\alpha_l \leq 1 - \frac{\min(s_{i+j-1}, s_{j-i+1})}{\max(s_{i+j-1}, s_{j-i+1})}. \tag{5.10}$$

In order to satisfy both equations (5.9) and (5.10), we choose $\alpha_l = c(1 - \frac{\min(s_{i+j-1}, s_{j-i+1})}{\max(s_{i+j-1}, s_{j-i+1})}) > 0$; where $0 \leq c \leq 1$ is a constant. Using this relationship,

the weighting coefficient for each watermark bit is adapted based on the particular pixel value, and the strength of the watermark can be adjusted by choosing c.

As a numerical example to illustrate the embedding process, let $s(n) = \{100, 128, 110, 99, 95\}$, $w = \{1, -1, 1\}$ and $c = 0.2$. The autocorrelation function for $s(n)$ is,

$$r_+(m,n) = \begin{bmatrix} 10000 & 16384 & 12100 & 9801 & 9025 \\ 0 & 11000 & 12672 & 10450 & 0 \\ 0 & 0 & 9500 & 0 & 0 \end{bmatrix}. \tag{5.11}$$

Let K_r contain the locations for s_1s_3, s_2s_4, s_3s_5. For $l = 1$, $w_l = 1$. Since $s_3 = 110 > s_1 = 100$, we embed w_1 into s_3. The results for embedding the first bit are,

$$\begin{aligned} K_r &= s_1 s_3, \\ \alpha_1 &= 0.018, \\ \hat{s_3} &= s_3(1 + \alpha_1 w_1) = 112, \\ K_{p1} &= 110, \\ s(n) &= \{100, 128, 112, 99, 95\}. \end{aligned} \tag{5.12}$$

For the second bit, $l = 2$,

$$\begin{aligned} w_2 &= -1, \\ K_r &= s_2 s_4, \\ \alpha_2 &= 0.045, \\ \hat{s_2} &= s_2(1 + \alpha_2 w_2) = 122.2, \\ K_{p_2} &= 128, \\ s(n) &= \{100, 122.2, 112, 99, 95\}. \end{aligned} \tag{5.13}$$

For the final bit, we get,

$$\begin{aligned} w_3 &= 1, \\ K_r &= s_3 s_5, \\ \alpha_3 &= 0.030, \\ \hat{s_3} &= s_3(1 + \alpha_3 w_3) = 115.4, \\ K_{p_3} &= 112, \\ s(n) &= \{100, 122.2, 115.4, 99, 95\}. \end{aligned} \tag{5.14}$$

5.3 Watermark Extraction

In order to study the performance of the detector, probability of error will be derived. In this dissertation, we assume that we have access to the keys K_s, K_p and K_r at the receiver. The semi-blind extraction algorithm can be implemented by performing the same steps in the embedding algorithm in the reverse order.

The extraction algorithm, as shown in Figure 5.2, can be summarized as follows:

1. Extract the watermarked pixels, $\hat{s}(n)$, using the key K_s.

2. Compute the autocorrelation function $r_+(m,n)$ for $\hat{s}(n)$.

3. Using K_r, determine the modified locations in the autocorrelation function.

4. Find $\max(\hat{s}_{i+j-1},\ \hat{s}_{j-i+1})$ for $r_+(i,j)$ where i and j are determined from K_r and then use the last stored value for $\max(s_{i+j-1},\ s_{j-i+1})$ in K_p to find the sign of $\alpha_l w_l$ and determine the value of $\hat{w}_l$ according to,

$$\hat{w}_l = \text{sgn}(\alpha_l w_l) = \text{sgn}\left(\frac{\max(\hat{s}_{i+j-1}, \hat{s}_{j-i+1})}{\max(s_{i+j-1}, s_{j-i+1})} - 1\right). \tag{5.15}$$

which can be written as,

$$\left(\frac{\max(\hat{s}_{i+j-1}, \hat{s}_{j-i+1})}{\max(s_{i+j-1}, s_{j-i+1})} - 1\right) \begin{array}{c} \hat{w}_l = 1 \\ > \\ < \\ \hat{w}_l = -1 \end{array} 0. \tag{5.16}$$

5. Replace $\max(\hat{s}_{i+j-1},\ \hat{s}_{j-i+1})$ with $\max(s_{i+j-1},\ s_{j-i+1})$.

6. Repeat steps (3-5) for the next watermark bit. The embedded watermark sequence is extracted in the reverse order, $\hat{w}_l$ for $l = L, L-1, ..., 1$.

As a numerical illustration of the extraction algorithm, we continue with the example given in the previous section. As we mentioned earlier, the watermark is extracted in the reverse order. Therefore, in the first run we extract the first bit of the watermark sequence,

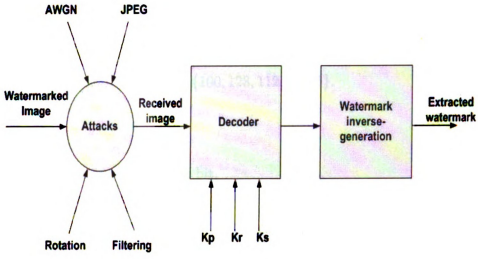

Figure 5.2. The block diagram for the watermark extraction algorithm for the auto-correlation method.

$$
\begin{aligned}
K_r &= s_3 s_5, \\
s(n) &= \{100, 122.2, 115.4, 99, 95\}, \\
\hat{w_3} &= \operatorname{sgn}\left(\frac{115.4}{112} - 1\right) = 1, \\
\hat{s_3} &= K_{p_3} = 112, \\
s(n) &= \{100, 122.2, 112, 99, 95\}.
\end{aligned}
\tag{5.17}
$$

The output for the second run is,

$$\begin{aligned} K_r &= s_2 s_4, \\ s(n) &= \{100, 122.2, 112, 99, 95\}, \\ \hat{w_2} &= \text{sgn}\left(\frac{122.2}{128} - 1\right) = -1, \\ \hat{s_2} &= K_{p_2} = 128, \\ s(n) &= \{100, 128, 112, 99, 95\}. \end{aligned} \tag{5.18}$$

Finally for the first watermark bit,

$$\begin{aligned} K_r &= s_1 s_3, \\ s(n) &= \{100, 128, 112, 99, 95\}, \\ \hat{w_1} &= \text{sgn}\left(\frac{112}{110} - 1\right) = 1, \\ \hat{s_3} &= K_{p_1} = 110, \\ s(n) &= \{100, 128, 110, 99, 95\}. \end{aligned} \tag{5.19}$$

The above example shows that we can extract the watermark and get the original signal without any error assuming that there is no corruption in the received image. The following section analyzes the effect of the attacks the image may undergo through on the recovery of the watermark.

5.4 Analysis of the Algorithm under Attacks

In the Wigner-based methods, the extraction algorithm for the binary watermark case allows us to extract the whole watermark sequence at once. Therefore, we are

able to model any attack on the watermarked image as additive noise. However, in the autocorrelation method, the watermark is extracted bit by bit in reverse order. This method of extraction, allows us to model the attack as additive noise on each watermark bit and thus we can find the probability of error in extracting every watermark bit. As mentioned in [95], we can model the attacks on the watermarked image as additive white gaussian noise n_k, which is uncorrelated with the pixel value. For each pixel in the watermarked sequence $\hat{s}_k$, the pixel value after an attack can be written as,

$$\bar{\hat{s}}_k = \hat{s}_k + n_k. \tag{5.20}$$

Therefore, the detection rule in equation (5.16) is modified as:

$$\left(w_l + \frac{n_l}{\alpha_l \max(s_{i+j-1}, s_{j-i+1})}\right) \begin{array}{c} \hat{w}_l = 1 \\ > \\ < \\ \hat{w}_l = -1 \end{array} 0. \tag{5.21}$$

where $w_l = \frac{1}{\alpha_l}\left(\frac{\max(\hat{s}_{i+j-1}, \hat{s}_{j-i+1})}{\max(s_{i+j-1}, s_{j-i+1})} - 1\right)$.

The probability of error P_{e_l}, assuming equal a prior probabilities for {-1,1}, for the l^{th} watermark bit can be derived as,

$$P_{e_l} = \frac{1}{2}\left[P(n_l > \alpha_l \max(s_{i+j-1}, s_{j-i+1})) + P(n_l < -\alpha_l \max(s_{i+j-1} \cdot s_{j-i+1}))\right] \tag{5.22}$$

The variance of noise is estimated using the robust median estimator in the discrete wavelet domain given by [96]:

$$\hat{\sigma}_{n_l} = \frac{\text{Median}(|Y_{ij}|)}{0.6745}, Y_{ij} \in \text{subband } HH_1. \tag{5.23}$$

The mean $\hat{\mu}_{n_l}$ of n_l can be estimated by subtracting the mean of the original image from the mean of the received image. By plugging in the expression for α_l from Section 5.2, we get

$$P_{e_l} = Q\left(c\frac{\max(s_{i+j-1}, s_{j-i+1}) - \min(s_{i+j-1}, s_{j-i+1}) - \hat{\mu}_{n_l}}{\hat{\sigma}_{n_l}}\right), \tag{5.24}$$

where $Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^{+\infty} \exp\left(-\left(\frac{t^2}{2}\right)\right) dt$.

From equation (5.24) it is seen that as c increases, the watermark strength increases and P_{e_l} decreases as expected. It is also observed that as the difference between $\max(s_{i+j-1}, s_{j-i+1})$ and $\min(s_{i+j-1}, s_{j-i+1})$ increases, the probability of error will decrease. This suggests that if we embed the watermark into elements of the autocorrelation matrix that correspond to the correlation of pixels that have a large absolute difference, the algorithm will be more robust against attacks. Moreover, equation (5.24) gives the error for every watermark bit, so for a watermark of length L we can find the bit error rate (BER).

It should be noted that the model in equation (5.20) is not always realistic. For example, in some cases the attack n_k is actually correlated to the signal and cannot be modeled as additive white Gaussian noise. Thus, equation (5.24) will not be valid for all attacks.

5.5 Simulation Results and Comparison

In this section, we demonstrate the performance of the proposed method through various simulations with different attacks. We also, compare the proposed method with spread spectrum method [63] and the DWT method [68]. The reason to choose these two methods is because of the similarity between the proposed method and the methods in [63, 68]. In [63], the watermark is embedded in the DCT domain and is spread out through the whole image, similarly, the proposed method embeds the wa-

termark in the autocorrelation domain and the watermark is also spread out through the whole image. On the other hand, the method in [68] embeds the watermark in the DWT domain repeatedly in different scales. The proposed method, is similar in the way it embeds every watermark bit repeatedly in all locations of $r_+(n,m)$ that contain $\max(s_{i+j-1}, s_{j-i+1})$. We also show the performance of the proposed method in embedding and extracting logos when the watermarked image undergoes distortion.

The watermark embedding algorithm in the autocorrelation domain has been applied to a large number of images [91] using $c = 0.2$ and $L = 256$. Table 5.1 shows the average bit error rates (BERs) under different attacks. Since the performance of the algorithm does not vary much with the choice of the image (as can be seen in Table 5.1), in the rest of this section we focus on the performance of the algorithm for the Lena image.

Table 5.1. Average bit error rate in detecting the watermark under different attacks using 100 different images.

c	0.2
AWGN (PSNR=48.13db)	0.0075±0.0051
AWGN (PSNR=36.0db)	0.0421±0.0063
AWGN(PSNR=14.15db)	0.1572±0.0098
JPEG (CR=1.7)	0.0215±0.0076
JPEG (CR=7.7)	0.0842±0.0153
JPEG (CR=20)	0.1421±0.0213
MF (3×3)	0.1041±0.0211
MF (5×5)	0.1105±0.0134
MF (7×7)	0.1254±0.0242

The autocorrelation method, unlike the Wigner-based methods, allows us to extract one bit of the watermark at a time. Therefore, we will report the results in

terms of the average bit error rates (BERs).

The watermarked Lena image is similar to the original one with no visible differences with PSNR=44.5dB. Figure 5.3 shows the watermarked image using $c = 0.2$. The algorithm has been tested under different attacks and for different embedding parameters. We ran the algorithm 100 times by generating different watermark sequences. The average bit error rates with their standard deviations are reported. Table 5.2 shows the effect of the choice of c on the robustness of the proposed algorithm under additive white gaussian noise 'AWGN', JPEG compression, and median filtering 'MF'. The algorithm maintains a low (BER) even for low JPEG compression ratio.

Table 5.2. Bit error rate in detecting the watermark under different attacks for different values of c.

c	0.05	0.1	0.2
AWGN (PSNR=48.1dB)	0.03±0.01	0.02±0.01	0.01±0.01
AWGN (PSNR=36.1dB)	0.09±0.01	0.05±0.02	0.04±0.01
AWGN(PSNR=28.1dB)	0.11±0.01	0.09±0.02	0.05±0.01
AWGN(PSNR=14.2dB)	0.14±0.03	0.13±0.03	0.15±0.02
AWGN(PSNR=8.1dB)	0.17±0.03	0.15±0.03	0.15±0.03
JPEG (CR=1.7%)	0.04±0.01	0.03±0.01	0.02±0.01
JPEG (CR=7.7%)	0.12±0.02	0.10±0.02	0.08±0.02
JPEG (CR=20%)	0.16±0.02	0.14±0.03	0.14±0.02
JPEG (CR=37%)	0.18±0.02	0.16±0.01	0.16±0.03
MF (3×3)	0.11±0.01	0.10±0.03	0.10±0.02
MF (5×5)	0.12±0.02	0.12±0.02	0.11±0.01
MF (7×7)	0.13±0.02	0.12±0.01	0.12±0.01

Figure 5.3. The watermarked image with PSNR=44.5dB using $c = 0.2$.

5.5.1 Comparison between Autocorelation, Wavelet, and Spread Spectrum Methods

The proposed algorithm is also compared with a well-known watermarking algorithm based on the Discrete Wavelet Transform (DWT) introduced by Kundur and Hatzinakos [68]. In their work, the authors propose a DWT based watermarking algorithm based on quantizing certain DWT coefficients at each level. For comparison, we embed the same watermark sequences. As another comparison, we compare the proposed method with the well-known spread spectrum watermarking [63]. In this method, the

watermark is embedded into the highest magnitude DCT coefficients of the image. The watermark is extracted by comparing the DCT coefficients of the watermarked image and the original image. In [68, 63], the authors try to detect the watermark, so in this comparison, we use a correlation based detector to detect the watermark,

$$\langle w(y), \hat{w}(y) \rangle \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{5.25}$$

where, w, and, $\hat{w}$, are the original and the extracted watermarks, respectively.

In all three methods, the extracted watermark is correlated with the true watermark and the correlation value is reported for different attacks. For the sake of this comparison, a binary watermark of length 256 is embedded in all methods. The results indicate that our method outperforms the DWT method and has better, but close, performance with the spread spectrum method for the tested attacks as shown in Figure 5.4 through Figure 5.6. The DCT method embeds the watermark in the largest magnitude DCT coefficients of the host image, which requires the use of small weighting coefficients to provide an invisible watermark. Thus, the strength of the watermark will be low and any distortion in the image will affect the detection of the watermark. On the other hand, the DWT method is based on quantizing certain DWT coefficients at each level. This quantization introduces some error in the extraction process. Embedding the watermark in the pixels with the largest values in the autocorrelation domain, explain the close performance of the proposed method with the DCT method.

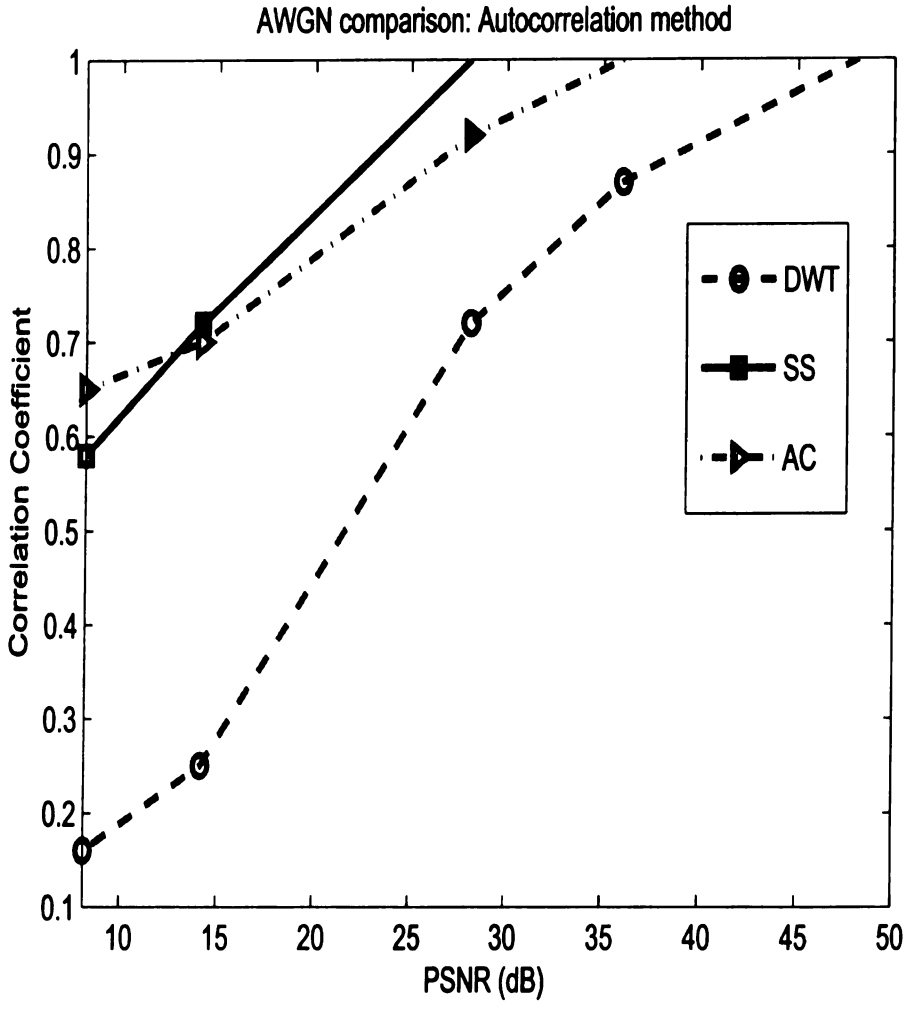

Figure 5.4. Comparison between SS, DWT, and Autocorrelation methods under AWGN.

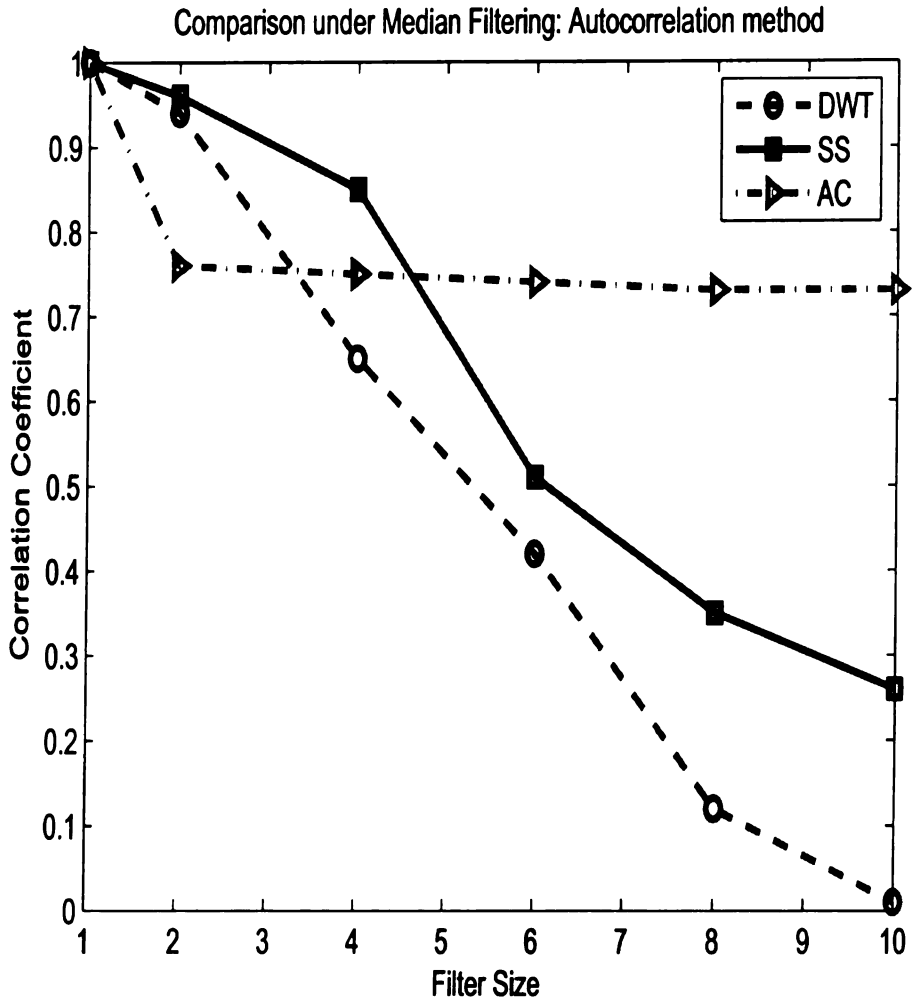

Figure 5.5. Comparison between SS, DWT, and Autocorrelation methods under Median Filtering.

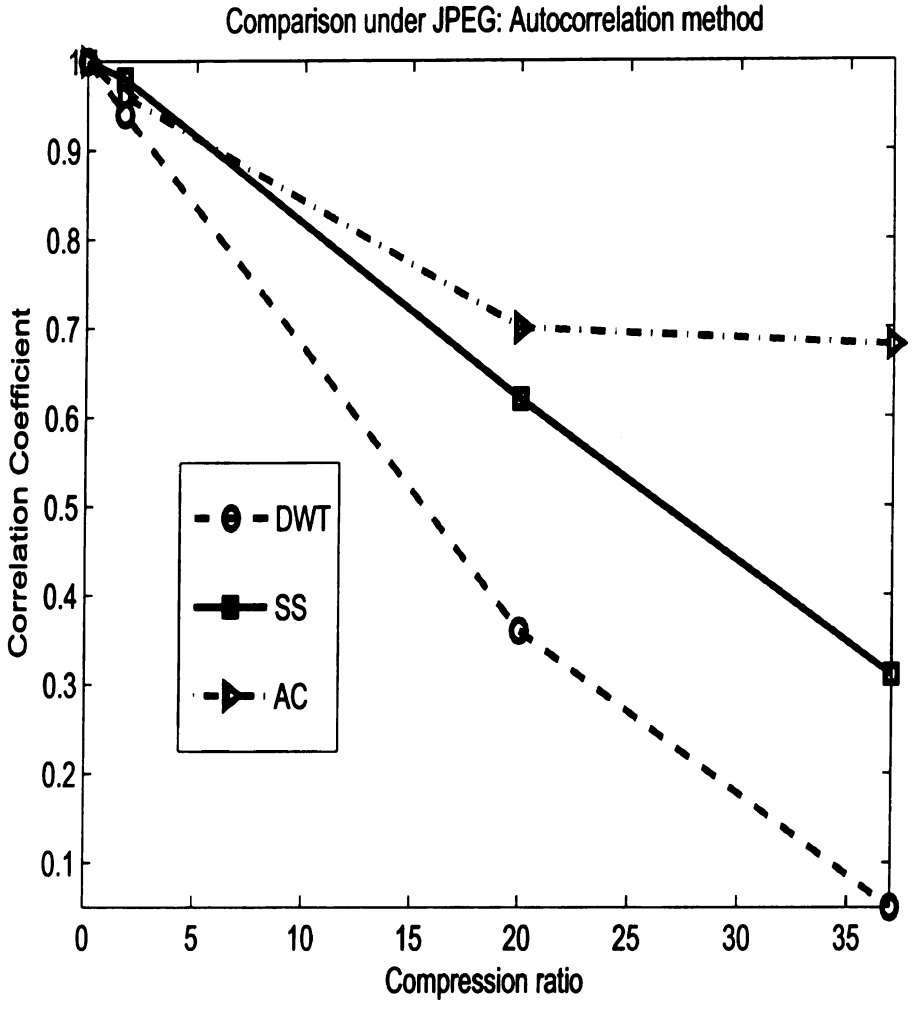

Figure 5.6. Comparison between SS, DWT, and Autocorrelation methods under JPEG compression.

5.5.2 Results for the Binary Logo Case

We have also embedded a logo image as the watermark inside the image. The logos, serve as a quick check for signaling and locating tampering. The decision on whether an image is altered or not, can be made automatically by comparing the extracted pattern with the original one, if available, or by human testing based on visualizing the extracted pattern. The latter case uses a reasonable assumption that the human can distinguish a 'meaningful' pattern from a random one. Figure 5.7 shows the extracted logo under different attacks. It is clear that we are able to extract the watermark even when the image goes through different distortions. Moreover, to test the performance of the proposed method when the watermarked image goes under multiple attacks, we applied the AWGN (PSNR=28.13dB), JPEG (CR=5), rotation (3°), and filtering (3×3) attacks in sequence to the watermarked image. Although the image went through multiple attacks, the extracted logo is still recognizable as shown in Figure 5.8 and has a correlation value of 0.8 with the original logo.

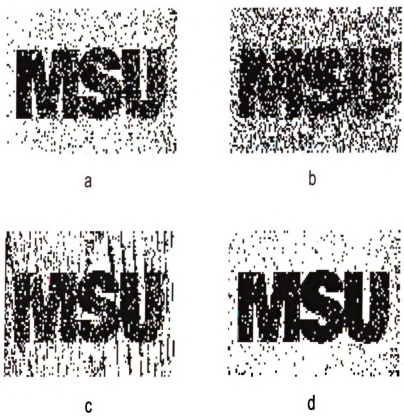

Figure 5.7. The extracted logo under different attacks a- AWGN (PSNR=22.11dB), b- JPEG (CR=7.7), c- Rotation (7 degrees), and d- Median filtering (5×5).

Figure 5.8. The extracted logo after subsequent attacks of 1- AWGN (PSNR=28.13dB), 2- JPEG (CR=5), 3- Rotation (3 degrees), and 4- Median filtering (3×3).

5.6 Discussion

In this chapter, we proposed a watermarking algorithm in the autocorrelation domain. This method, unlike the Wigner-based methods, can embed only multi-bit watermark. The embedding algorithm embeds one bit of the watermark at a time. Therefore, at the receiver, we extract the individual watermark bits. This way of embedding and extraction allows us to model the attack as additive noise on each pixel value of the image,

$$\bar{\hat{s}}_k = \hat{s}_k + n_k. \tag{5.26}$$

Therefore, we derived the probability of error in detecting every watermark bit,

$$P_{e_l} = Q\left(c\frac{\max(s_{i+j-1}, s_{j-i+1}) - \min(s_{i+j-1}, s_{j-i+1}) - \hat{\mu}_{n_l}}{\hat{\sigma}_{n_l}}\right), \tag{5.27}$$

where $Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^{+\infty} \exp\left(-\left(\frac{t^2}{2}\right)\right) dt$.

The probability of error in extracting the watermark in equation (5.27) can be used to improve the performance of the proposed algorithm. If P_{e_l} is greater than a predefined threshold, an error occurred in extracting the l^{th} watermark bit and therefore, the bit should be reversed,

$$\begin{array}{c} w_l = \bar{w}_l \\ P_{e_l} \quad > \quad \eta, \\ < \\ w_l = w_l \end{array} \tag{5.28}$$

where $\bar{w}_l$ is the complement of w_l. However, improving the performance this way depends on the accuracy of the noise model, since not all attacks can be modeled

as additive noise. For example, AWGN can be modeled as an additive noise, while JPEG compression is not well approximated by additive noise. This is illustrated in Figure 5.9.

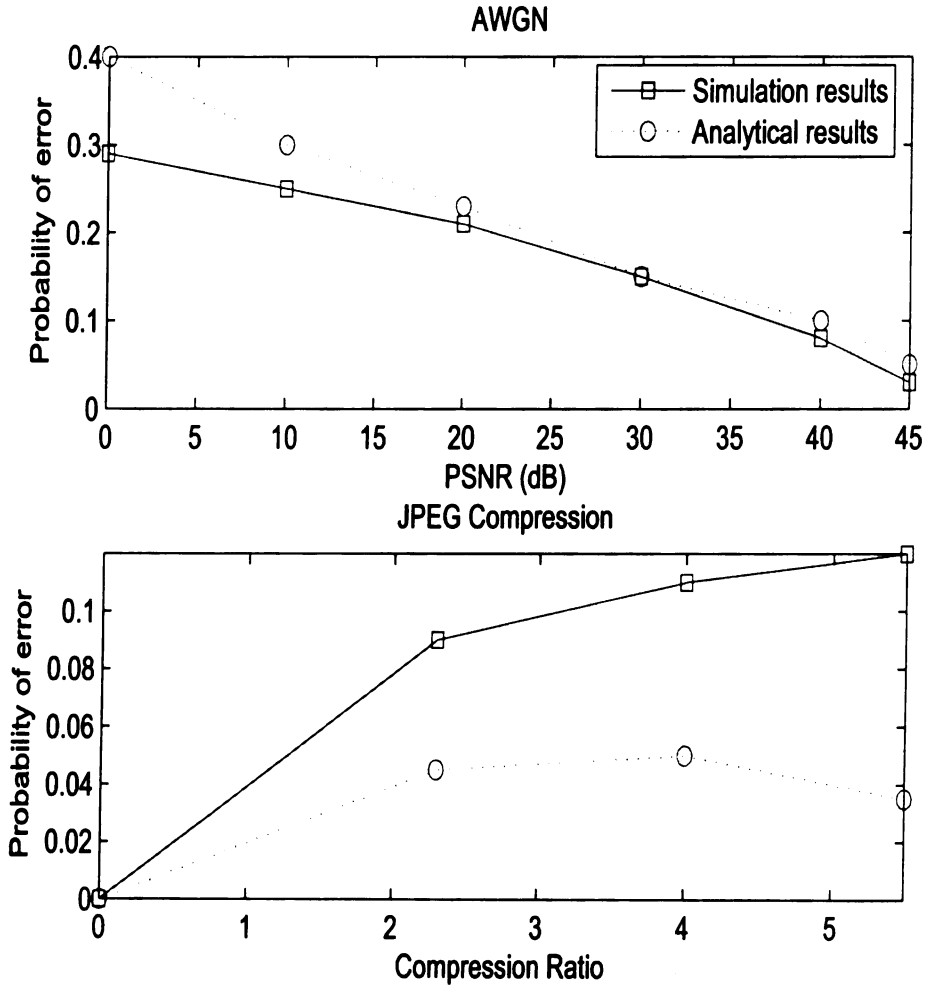

Figure 5.9. Comparison in computing the probability of error from the simulations and the analytical results in equation (5.27).

5.7 Summary

In this chapter, we presented a new robust watermarking algorithm based on the local autocorrelation function. The autocorrelation function is modified such that the watermarked function is still a valid autocorrelation function. A semi-blind detection algorithm is derived and its performance is quantified by deriving the probability of error. The proposed autocorrelation based watermarking algorithm is shown to be robust and provides high capacity. The number of bits that can be embedded for an $M \times M$ image is $\frac{M(M-1)^2}{4}$, which is around 33421488 bits for an image of size 512×512. This does not mean that we can embed this large number of bits without degrading the visual appearance of the watermarked image, but it enables us to embed a large number of bits that can be suitable for any application and at the same time keep a high PSNR for the watermarked image.

In addition to the ability to embed large amount of data, the algorithm is shown to have reasonable computational complexity and high watermark security. The computational complexity of the proposed embedding algorithm is of order $O(M^2)$. Thus, the autocorrelation method can be used in real-time applications. In terms of security, the semi-blind extraction algorithm makes it more secure compared to Wigner-based methods, because some keys and side information, not the original image, are needed for watermark extraction. This side information can be reduced or eliminated which in turn will increase the security of the algorithm. For example, instead of generating K_r randomly, we can choose the first successive L non-zero points in the autocorrelation function $r_+(m,n)$. Moreover, we can choose a row from the image and use it for watermark embedding, to get rid of the K_p key.

The comparison with the well-known spread spectrum and DWT-based methods shows the superior performance of the proposed method. The proposed method uses semi-blind detection algorithm to detect the watermark, while the SS and DWT

methods need the original image for watermark detection. The semi-blind detection algorithm makes the proposed algorithm suitable for a wide class of applications.

CHAPTER 6

A COMPARATIVE STUDY OF THE THREE PROPOSED TIME-FREQUENCY WATERMARKING METHODS

In this chapter, we provide a quantitative comparison of the three time-frequency based image watermarking algorithms proposed in this dissertation. Moreover, we introduce techniques to improve the performance of the proposed methods. In Section 6.1, a comparison between the three proposed methods will be carried out based on computational complexity, watermarking capacity, detection/extraction type, and robustness. In Section 6.2, techniques to improve the performance of the proposed methods are introduced.

6.1 Comparison between the Three Time-Frequency Domain Watermarking Methods

In Chapters 3 through 5, we have introduced three different methods. For each method, we gave a complete mathematical analysis at the encoder and the decoder. Moreover, comparisons with competitive well-known methods in other transform domains have been carried out. In this section, we compare the three proposed methods in terms of computational complexity, capacity, and robustness. Before we proceed with this comparison, we like to emphasize that the Wigner-based methods have the ability to embed Gaussian and binary watermark sequences, unlike the autocorrelation method which embeds a binary watermark. Although the simulation results provided for the Wigner-based methods are for the binary watermark case, similar simulations can be carried out for the Gaussian watermark case. The binary watermarks are preferable over the Gaussian ones in many applications including data hiding and ownership declaration. Thus, we focused on embedding binary watermarks throughout this dissertation. For the rest of this chapter, we assume the watermark

is a binary sequence of length M, and the image is of size $M \times M$ unless otherwise stated.

We also like to emphasize that in some applications, it is desirable to embed more than one watermark, where each watermark has a different purpose. For example, the first watermark may reveal the name of the image owner and the second watermark reveals the date of creating that image. A good watermarking algorithm should be able to detect both watermarks, even when the watermarked image goes under different attacks. As an example on using the proposed algorithms to embed multiple watermarks, we embedded two binary watermarks each of length 512 bits inside the lena image using the autocorrelation method. Figure 6.1 shows the correlation function under AWGN (PSNR=22.1dB). The maximum correlation occurs at sequence 100 and sequence 150, which correspond to the first and the second watermarks, respectively.

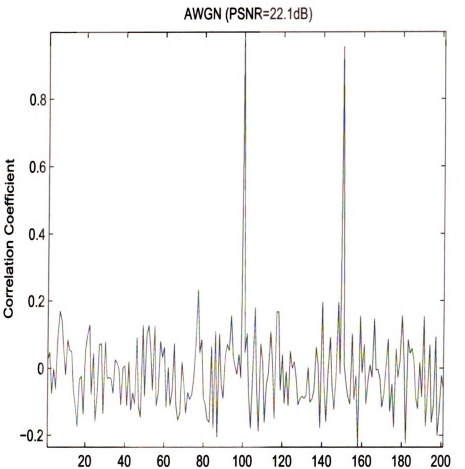

Figure 6.1. The normalized correlation detector response for the autocorrelation method with two embedded watermarks under AWGN with PSNR=22.1dB.

6.1.1 Computational Complexity

We have shown that the Wigner-based methods reduces the embedding algorithms to a non-linear functions in the time domain. However, these non-linear functions are dependent on the weighting matrix which is dependent on the Wigner distribution of the chosen pixels, $P(y)$. Thus, the most computationally complex part in the Wigner-based methods is finding the weighting matrix. To compute the weighting matrix, we need to perform $O(M^2)$ multiplications to find the autocorrelation function for the input pixels. Moreover, we need $M^2 log(M^2)$ multiplications to find the Fourier transform of the resultant autocorrelation function in order to compute the Wigner distribution for $P(y)$. Thus, we need $M^2 + M^2 log(M^2)$ computations to find the weighting matrix for the Wigner-based methods, i.e. $O(M^2 log(M^2))$. On the other hand, the autocorrelation method embeds the watermark in the autocorrelation domain, which has a computational complexity of $O(M^2)$. The computation of the Wigner distribution makes the Wigner-based methods more computationally complex compared to the autocorrelation method. This computational complexity limits the use of the Wigner-based methods to limited applications.

6.1.2 Capacity

In terms of the capacity, the simplified embedding functions for the Time-Wigner and the Wigner-Wigner methods are,

$$\hat{P}(y) = \sqrt{P^2(y) + \left(\sum_{\omega_y} A_P(y, \omega_y)\right) w(y)}, \tag{6.1}$$

and

$$\hat{P}(y) = \sqrt{P^2(y) + \left(\sum_{\omega_y} A_P(y, \omega_y)\right) * w^2(y)}, \tag{6.2}$$

respectively. From the simplified functions, the maximum number of bits that can be embedded inside $P(y)$ are M, if the length of $P(y)$ is M. Therefore, for an image of size $M \times M$, we have the capacity to embed up to M^2 bits, since we can choose $P(y)$ to be the whole image, i.e. of length M^2. The autocorrelation method, on the other hand, has a capacity of embedding $\frac{M(M-1)^2}{4}$ bits. Although the number of watermarkable cells is high in all of the three methods, not all of the watermarkable cells are used for watermarking. The constraint of having a high PSNR value limits the number of embedded bits. Figure 6.2 shows the PSNR values for different watermark lengths using the three proposed method. The Wigner-Wigner method provides higher PSNR values, since embedding one bit of the watermark sequence in the autocorrelation method will result in changing many pixels form the original sequence, $P(y)$. Moreover, the Time-Wigner method requires the same watermark to be embedded into every column of the Wigner distribution of $P(y)$, which lowers the PSNR value.

6.1.3 Non-Blind and Blind Detection

As discussed in Chapters 3 and 4, both Wigner-based methods require the original image or at least the original chosen pixels, $P(y)$, for watermark detection/extraction. On the other hand, the autocorrelation method does not need the original data for watermark extraction. The semi-blind detection algorithm for the autocorrelation method makes it more useful for a wide range of practical applications.

6.1.4 Robustness

In terms of robustness, the proposed methods are shown to be robust against different types of image processing attacks and perform better than some of the most well-known watermarking methods. In this subsection, we compare the performance of the proposed time-frequency methods with each other under attacks. In order to provide a fair comparison, we do not use the reference watermark technique in the Wigner-

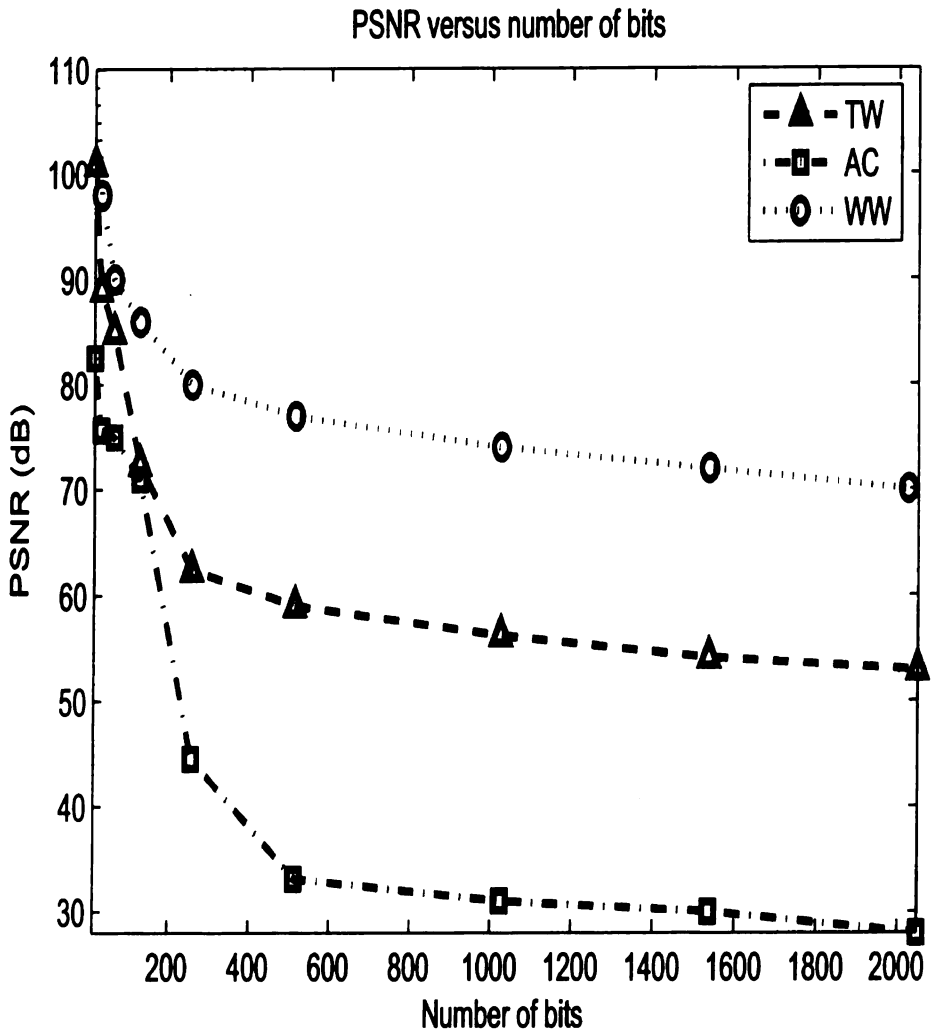

Figure 6.2. Comparison between Time-Wigner (TW), Wigner-Wigner (WW), and autocorrelation (AC) methods in terms of PSNR versus number of watermark bits

Wigner method. A binary watermark sequence of length 256 is embedded inside the Lena image. The performance measure used is the correlation coefficient between the extracted and the original watermark. Figure 6.3 shows a sample result under AWGN attack. The figure shows that the proposed Time-Wigner method outperforms the other two methods. This is expected, since the watermark in the autocorrelation method is embedded in a recursive way. This way of embedding will cause any error in detecting the watermark bit to affect the detection of the neighboring bits.

Moreover, the embedded watermark in the Wigner-Wigner method is $\{0, 1\}$, while it is $\{-1, 1\}$ in the Time-Wigner method. Thus, distortions on the watermarked image in the Wigner-Wigner case will affect the extraction of the watermark more than in the Time-Wigner case.

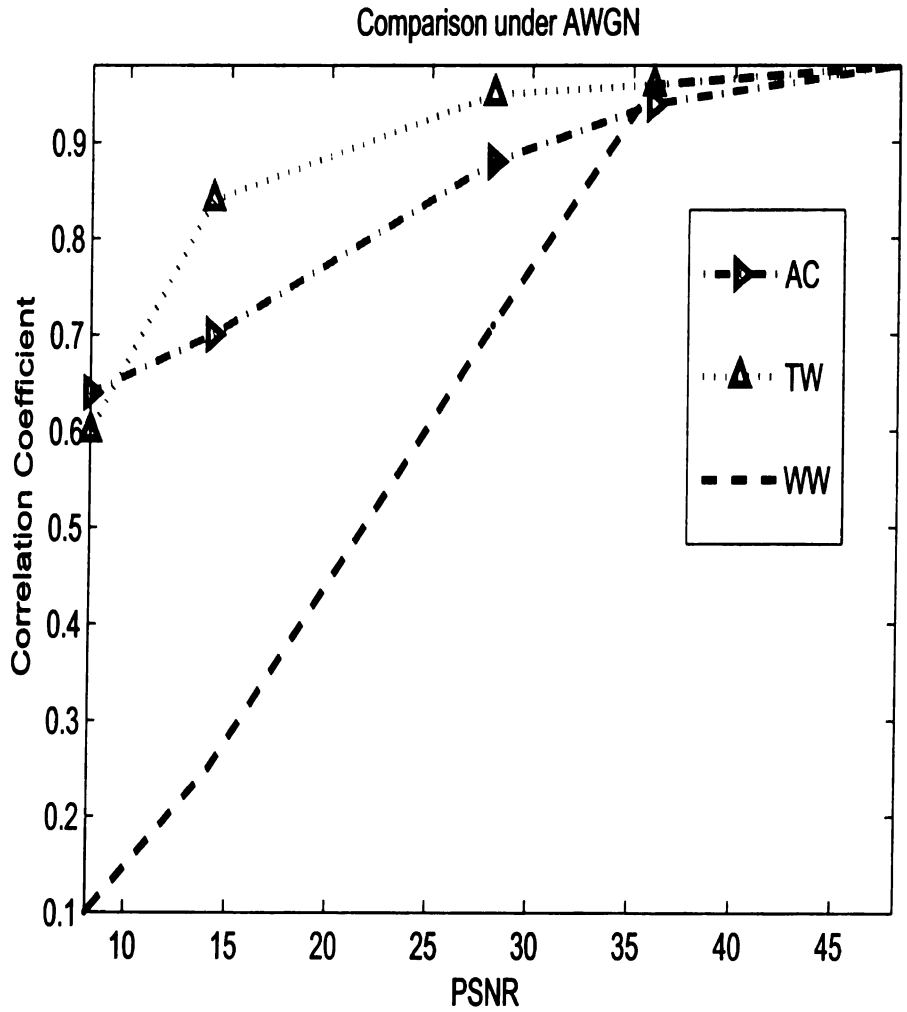

Figure 6.3. Comparison between Time-Wigner (TW), Wigner-Wigner (WW), and autocorrelation (AC) methods under AWGN attack

As a summary for this section, Table 6.1 summarizes the main comparisons, between the Wigner-based and the autocorrelation methods. It is important to mention

that due the high computational complexity of the proposed algorithms, their usage may be limited to certain applications like copyright protection and data hiding.

Table 6.1. A comparison between the Wigner-based methods and the autocorrelation method.

	Wigner-Based	Autocorrelation
Capacity	M^2	$\frac{M(M-1)^2}{4}$
Multi-bit	Yes	Yes
Blind	Non-blind	Semi-blind
Computationally complex	$O(M^2 log(M))$	$O(M^2)$

6.2 Techniques for Performance Improvement

In this section, we discuss some techniques that can be used to improve the performance of the proposed methods. This includes the use of the the pseudo-random watermark generator and the reference watermark discussed in Chapter 4.

6.2.1 Pseudo-random Watermark Generator

In order to provide more secure and robust watermarks, the watermark can be generated using a pseudo-random generator. This idea has been used in [97] and has been applied to other multi-bit watermarking algorithms. The original watermark sequence w of length N is spread out to generate another sequence $\bar{W}$ of length M according to,

$$\bar{W} = \text{sgn}\left(\alpha \sum_{j=1}^{N} w_j \cdot \bar{P}_j\right). \tag{6.3}$$

where $\bar{P}_j$ is a pseudo-random sequence of length M and α is a gain factor that determines the watermark magnitude.

The set P of N reference marks is constructed such that the pseudo-random

sequences are orthogonal to each other.

$$P = \{\bar{P}_1, \bar{P}_2, ..., \bar{P}_N\}, \tag{6.4}$$

$$\bar{P}_j = \left[P_{j1}, P_{j2}, ..., P_{jM}\right], \tag{6.5}$$

where $P_{ji} \in \{-1, 1\}$.

The initial state of the random sequence generator should be known by the embedder and the detector in order to produce the same P. After $\bar{W}$ is created, it is embedded instead of w and at the receiver we recover $\hat{\bar{W}}$ of length M. In order to reconstruct the original watermark sequence, a binary decision is made on the decision variable D_k

$$D_k = \frac{1}{M} \sum_{i=1}^{M} \hat{\bar{W}}_i \cdot P_{ki}, \tag{6.6}$$

$$\hat{w}_k = \text{sgn}(D_k). \tag{6.7}$$

Generating the watermark this way will increase the security of the embedding algorithm, since the created watermark, $\bar{W}$, rather than the original watermark is embedded in the host data. Moreover, it will increase the robustness of the watermarking algorithm, since the initial state of the random sequence generator is known at the receiver which carries some side information about the original watermark.

In order to illustrate the effect of using the pseudo-random watermark generator, we apply this generator to the autocorrelation method and compare the output with the output obtained by embedding the watermark directly. The watermark is a binary sequence of length 256 and $c = 0.2$. Table 6.2 shows that using the pseudo-random watermark generator reduces the bit error rate to zero. This improved efficiency, however, comes at the expense of using the extra seed key, which is transmitted to the receiver. The results in Table 6.2 agree with the results for the method proposed

in [97]. In [97], the authors embed a multi-bit watermark inside an image using a deterministic embedding scheme that ensures total embedding efficiency, i.e. zero bit error rate.

Table 6.2. Bit error rate in detecting the watermark with and with out using pseudo-random watermark generator

Generator	No	Yes
AWGN (PSNR=48.1db)	0.01	0.00
AWGN (PSNR=28.1db)	0.05	0.00
JPEG (CR=1.7)	0.02	0.00
JPEG(CR=7.7)	0.08	0.00
MF (3×3)	0.10	0.00
MF (5×5)	0.11	0.00

6.2.2 Reference Watermark

In order to increase the robustness of the proposed watermarking algorithms, we may use a reference watermark. The reference watermark, which is assumed to be known at the receiver, and the desired watermark are embedded in an orthogonal way, which means that they are not embedded in the same pixels. The bit error rate for the reference watermark is expected to be equal to the bit error rate for the desired watermark. The following equation is used to determine if an error occurred in the i^{th} bit of the reference watermark,

$$\mathrm{BE}(i) = w_r(i) \oplus \hat{w}_r(i), \tag{6.8}$$

where w_r and $\hat{w}_r$ are the original and extracted reference watermarks respectively and $\oplus$ is the exclusive X-OR operator. If $BE(i)$ equals 1 then an error occurred at the i^{th} otherwise the extracted bit is correct.

If more than one reference watermark is used, a majority rule can be used to determine if an error has occurred at the i^{th} bit. For the case of R reference watermarks, this can be written as,

$$\text{Majority}(i) = \left(\sum_{k}^{R} w_{rk}(i) \oplus \hat{w}_{rk}(i) \right), \tag{6.9}$$

where if Majority$(i) = 1$, the i^{th} bit of the extracted desired watermark is shifted.

Deciding the use of reference watermark is dependent on the targeted PNSR value and the number of watermark bits. For example, if the number of watermark bits is large and the targeted PSNR is high, using reference watermark is not recommended. On the other hand, if the goal is to provide more accurate detection/extraction, the use of reference watermark is desirable. Therefore, the use of the reference watermark in the Wigner-Wigner method is justified, because the square operation in the simplified function,

$$\hat{P}(y) = \sqrt{P^2(y) + \left(\sum_{\omega y} A_P(y, \omega_y) \right) * w^2(y)}, \tag{6.10}$$

makes the watermark less robust and harder to extract. Therefore, we can use the reference watermark to add more robustness to the Wigner-Wigner method. To compare the results between using and not using the reference watermark, we test the Wigner-Wigner method under the two cases for different attacks. Figure 6.4 shows the simulation results for the AWGN attack. The results, as expected, show the superior performance of using the reference watermark.

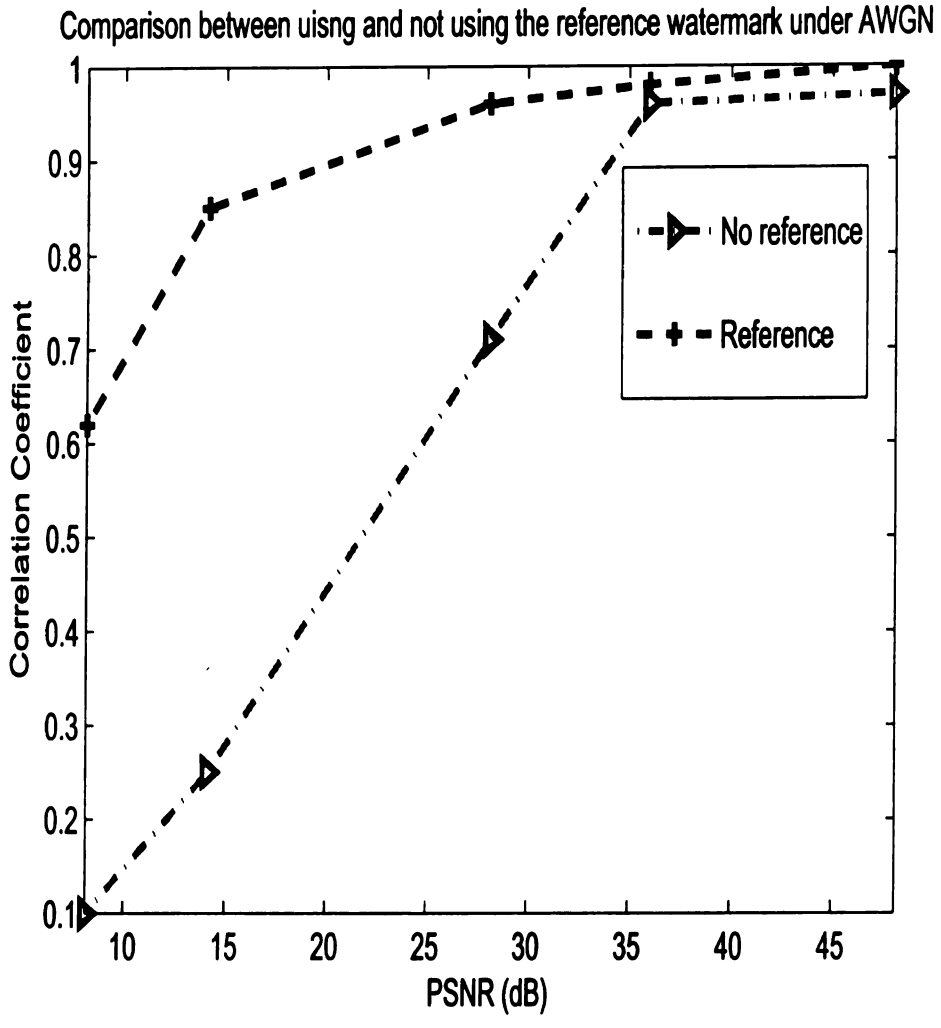

Figure 6.4. Comparison between using and not using the reference watermark for the Wigner-Wigner method under AWGN.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Summary of the Dissertation

In this dissertation, three new watermarking algorithms based on the time-frequency representation of the image has been introduced. It has been shown that for positive and real signals, the signal can be retrieved from its Wigner distribution without any error. This realization inspired the implementation of two time-frequency watermark embedding methods; one uses the time-frequency distribution of the image, the Time-Wigner method, and the other uses the time-frequency information for both the image and the watermark, the Wigner-Wigner method. For both methods, under special conditions as described in this dissertation, the embedding algorithm in the joint domain can be simplified to a non-linear embedding function in the time domain as long as the modified distribution is still a valid Wigner distribution. This result reduces the computational complexity of embedding and detecting the watermark. A non-blind correlation based detector is derived using the non-linear embedding function and the probability of error is found in the case of Gaussian and binary watermark sequences. The proposed algorithms are shown to be transparent and robust under attacks through experiments.

The third method introduces a robust watermarking algorithm based on the local autocorrelation function. The autocorrelation function is modified such that the watermarked function is still a valid autocorrelation function. A semi-blind detection algorithm is derived and its performance is quantified by deriving the probability of error. The proposed algorithm is shown to be transparent and robust under attacks. The proposed algorithm performs better than conventional transform domain algorithms as illustrated through our comparison with a DWT based method and a

spread spectrum method.

All of the proposed methods have the ability to embed multi-bit watermarks. Each of the proposed methods has a different advantage. The Time-Wigner method is the best in terms of the perceptibility and PSNR values. The Wigner-Wigner method should be used when the robustness is the main concern as it has superior performance with the use of the reference watermark. The autocorrelation method has the highest capacity and is semi-blind. Therefore, it should be used when the data to be hidden is large. These variations make the proposed methods applicable to a wide range of watermarking applications.

7.2 Future Work

The proposed Wigner-based methods assume the watermarked distribution to be a valid Wigner distribution. However, this is not always true and an error is introduced in the inversion process. Therefore, we used the error between the Wigner of the watermarked pixels and the watermarked distribution as a key and send it as a side information to the receiver. This error key added robustness to the proposed algorithms. Watermarking the Wigner distribution in a way that keeps it as a valid Wigner distribution, would be a possible extension of this work and will save us from sending the extra error key. Moreover, in the Wigner-based methods, due to the computational complexity and the difficulty of implementing the Wigner distribution for the image, which is a two dimensional signal, we randomly choose a subset of pixels from the image and do the watermarking on the Wigner distribution of this subset, which is a one dimensional vector. However, finding the Wigner distribution of the whole image or some blocks from the image, may reveal new characteristics and information about the image which can improve the watermarking in this domain.

The proposed watermarking methods have been applied to gray-scale images. Future research may modify the methods to make them suitable for other types of

media, i.e. video and audio. For example, the fact that a video is a sequence of images, suggests that the algorithms should be applicable for this media type. However, in audio signals, where there are some negative and positive values, the implementation is not that trivial. The proposed algorithms use the fact that positive real-valued signals can be synthesized from their Wigner distributions without any error. This fact does not apply in the audio case and other methods for finding the inverse of the Wigner distribution for signals which are not necessarily positive or real-valued, should be used. In [84], the authors established an algorithm for synthesizing the signal from its Wigner distribution by finding the discrete-time signal whose Wigner distribution best matches a specified time-frequency distribution in the sense of the least mean squared error. One may apply the proposed Wigner-based methods on the audio signals and use the algorithm developed in [84] to find the inverse of the watermarked distribution.

APPENDICES

In this appendix, we give the derivation for the detection statistics for the two algorithms, i.e. z_1 and z_2 in (3.14) and (4.11) for the Time-Wigner method, and in (3.17) and (4.14) for the Wigner-Wigner method. We assume that w and $\hat{w}$ are independent Gaussian random variables with zero means and variances of σ_1^2 and σ_2^2, respectively.

A.2 Detector derivation for the Time-Wigner method

Let,

$$z_1 = \sum_y A_P(y) w^2(y). \tag{A.1}$$

The mean of z_1 is,

$$\mu_{z_1} = \sigma_1^2 \sum_y A_P(y). \tag{A.2}$$

The variance of z_1 is,

$$\sigma_{z_1}^2 = E\left[z_1^2\right] - \mu_{z_1}^2. \tag{A.3}$$

where,

$$\begin{aligned} E\left[z_1^2\right] &= E\left[\sum_y \sum_{\hat{y}} A_P(y) A_P(\hat{y}) w^2(y) w^2(\hat{y})\right] \\ &= E\left[\sum_y A_P^2(y) w^4(y)\right] + E\left[\sum_y \sum_{\hat{y} \neq y} A_P(y) A_P(\hat{y}) w^2(y) w^2(\hat{y})\right] \\ &= 3\sigma_1^4 \sum_y A_P^2(y) + \sigma_1^4 \sum_y \sum_{\hat{y} \neq y} A_P(y) A_P(\hat{y}), \end{aligned} \tag{A.4}$$

where we have used the fact that $E\left[w^4(y)\right] = 3\sigma_1^4$ for a normal random variable with

zero mean [98]. Noting that,

$$\mu_{z_1}^2 = \sigma_1^4 \left(\sum_y A_P(y) \right)^2 ,$$

$$= \sigma_1^4 \sum_y A_P^2(y) + \sigma_1^4 \sum_y \sum_{\hat{y} \neq y} A_P(y) A_P(\hat{y}), \tag{A.5}$$

the variance of z_1 is given by,

$$\sigma_{z_1}^2 = 2\sigma_1^4 \sum_y A_P^2(y). \tag{A.6}$$

Similarly for,

$$z_2 = \sum_y A_P(y) w(y) \hat{w}(y). \tag{A.7}$$

It is apparent that z_2 has zero mean,

$$\mu_{z_2} = 0, \tag{A.8}$$

and the variance of z_2 is,

$$\sigma_{z_2}^2 = E\left[z_2^2 \right],$$

$$= E\left[\sum_y \sum_{\hat{y}} A_P(y) A_P(\hat{y}) w(y) \hat{w}(y) w(\hat{y}) \hat{w}(\hat{y}) \right],$$

$$= E\left[\sum_y A_P^2(y) w^2(y) \hat{w}^2(y) \right],$$

$$= \sigma_1^2 \sigma_2^2 \sum_y A_P^2(y). \tag{A.9}$$

where the independence of w and $\hat{w}$ is used in simplifying the first equality to the second one.

A.3 Detector derivation for the the Wigner-Wigner method

In this method $Y_1(k)$, $Y_2(k)$ and $C(k)$ are the Fourier transforms of $w^2(m)$, $\hat{w}^2(m)$ and $A_P(y)$ respectively. Therefore,

$$Y_1(k) = \sum_{m=0}^{N-1} w^2(m) e^{-j\frac{2\pi mk}{N}}, \tag{A.10}$$

and

$$Y_2(k) = \sum_{m=0}^{N-1} \hat{w}^2(m) e^{-j\frac{2\pi mk}{N}}. \tag{A.11}$$

Since $w(m)$ and $\hat{w}(m)$ are independent random variables, $w^2(m)$ and $\hat{w}^2(m)$ are independent too. Therefore, $Y_1(k)$ and $Y_2(k)$ are independent. For large N, $Y_1(k)$ and $Y_2(k)$ can be assumed to be Gaussians using the central limit theorem. The mean of $Y_1(k)$ is,

$$\begin{aligned} \mu_{Y_1(k)} &= E\left[\sum_{m=0}^{N-1} w^2(m) e^{-j\frac{2\pi mk}{N}}\right], \\ &= E\left[w^2(m)\right] \sum_{m=0}^{N-1} e^{-j\frac{2\pi mk}{N}}, \\ &= N\sigma_1^2 \delta(k), \end{aligned} \tag{A.12}$$

where $\sum_{m=0}^{N-1} e^{-j\frac{2\pi mk}{N}} = N\delta(k)$.

In order to find the variance of $Y_1(k)$, we need to find $E\left[Y_1^2(k)\right]$,

$$E\left[Y_1^2(k)\right] = E\left[\sum_m \sum_{\hat{m}} w^2(m)\hat{w}^2(\hat{m})e^{-j\frac{2\pi k(m-\hat{m})}{N}}\right],$$

$$= E\left[\sum_k w^4(m) + \sum_m \sum_{\hat{m}\neq m} w^2(m)\hat{w}^2(\hat{m})e^{-j\frac{2\pi k(m-\hat{m})}{N}}\right],$$

$$= 3\sigma_1^4 N + \sigma_1^4\left[\sum_m \sum_{\hat{m}\neq m} e^{-j\frac{2\pi n(m-\hat{m})}{N}}\right], \qquad \text{(A.13)}$$

where

$$\sum_m \sum_{\hat{m}\neq m} e^{-j\frac{2\pi k(m-\hat{m})}{N}} = \sum_m \sum_{\hat{m}} e^{-j\frac{2\pi k(m-\hat{m})}{N}} - N,$$

$$= \sum_{-(N-1)}^{(N-1)} (N-|l|)e^{-j\frac{2\pi kl}{N}} - N,$$

$$= -2N + 2N^2\delta(k) - \sum_{-(N-1)}^{(N-1)} |l|e^{-j\frac{2\pi kl}{N}},$$

$$= -2N + 2N^2\delta(k) - 2\sum_{0}^{(N-1)} l\cos(\frac{2\pi kl}{N})$$

$$= -2N + 2N^2\delta(k) - 2\left[\frac{-N}{2} + \frac{N^2}{2}\delta(k)\right],$$

$$= -N + N^2\delta(k). \qquad \text{(A.14)}$$

where $l = m - \hat{m}$

Therefore,

$$E\left[Y_1^2(k)\right] = \left[2N + N^2\delta(k)\right]\sigma_1^4, \qquad \text{(A.15)}$$

and the variance of $Y_1(k)$ is given by,

$$\sigma^2_{Y_1(k)} = 2N\sigma_1^4. \tag{A.16}$$

Following the same procedure the mean and the variance for $Y_2(k)$ are given by,

$$\mu_{Y2(k)} = N\sigma_2^2\delta(k), \tag{A.17}$$

$$\sigma^2_{Y2(k)} = 2N\sigma_2^4. \tag{A.18}$$

For,

$$z_1 = \sum_k C(k)Y_1^2(k) \tag{A.19}$$

The mean is,

$$\begin{aligned} \mu_{z_1} &= E\left[\sum_k C(k)Y_1^2(k)\right], \\ &= \sum_k C(k)E\left[Y_1^2(k)\right], \\ = \sum_k C(k)\left[2N + N^2\delta(k)\right]\sigma_1^4 &= 2N\sigma_1^4\sum_k C(k) + N^2\sigma_1^4 C(0), \end{aligned} \tag{A.20}$$

and the variance can be obtained by computing $E\left[z_1^2\right]$ as follows,

$$E\left[z_1^2\right] = E\left[\sum_k \sum_{\hat{k}} C(k)C(\hat{k})Y_1^2(k)Y_1^2(\hat{k})\right],$$

$$= E\left[\sum_k C^2(k)Y_1^4(k) + \sum_k \sum_{\hat{k}\neq k} C(k)C(\hat{k})Y_1^2(k)Y_1^2(\hat{k})\right],$$

$$= \sum_k C^2(k)\left[12N^2 + (12N^3 + N^4)\delta(k)\right]\sigma_1^8 + E\left[\sum_k \sum_{\hat{k}\neq k} C(k)C(\hat{k})Y_1^2(k)Y_1^2(\hat{k})\right],$$

$$= \sum_k C^2(k)\left[12N^2 + (12N^3 + N^4)\delta(k)\right]\sigma_1^8 +$$

$$\sigma_1^8 \sum_k \sum_{\hat{k}\neq k} C(k)C(\hat{k})\left(2N + N^2\delta(k)\right)\left(2N + N^2\delta(\hat{k})\right) \quad \text{(A.21)}$$

where we have used the fact that $E\left[Y_1^4(k)\right] = \left[12N^2 + (12N^3 + N^4)\delta(k)\right]\sigma_1^8$ for a gaussian random variable with non-zero mean [98].

By noting that,

$$\mu_{z_1}^2 = \sum_k C^2(k)\left[4N^2 + (4N^3 + N^4)\delta(k)\right]\sigma_1^8 +$$

$$\sigma_1^8 \sum_k \sum_{\hat{k}\neq k} C(k)C(\hat{k})\left(2N + N^2\delta(k)\right)\left(2N + N^2\delta(\hat{k})\right), \quad \text{(A.22)}$$

the variance of z_1 is given by,

$$\sigma_{z_1}^2 = E\left[z_1^2\right] - \mu_{z_1}^2 = 8N^2\sigma_1^8\left[\sum_k C^2(n) + NC^2(0)\right]. \quad \text{(A.23)}$$

Similarly for,

$$z_2 = \sum_k C(k)Y_1(k)Y_2(k). \quad \text{(A.24)}$$

The mean is,

$$\mu_{z_2} = E\left[\sum_k C(k)Y_1(k)Y_2(k)\right],$$
$$= N^2\sigma_1^2\sigma_2^2 E\left[\sum_k C(k)\delta(k)\right],$$
$$= N^2\sigma_1^2\sigma_2^2 C(0). \tag{A.25}$$

and,

$$E\left[z_2^2\right] = E\left[\sum_k\sum_{\hat{k}} C(k)C(\hat{k})Y_1(k)Y_1(\hat{k})Y_2(k)Y_2(\hat{k})\right],$$
$$= E\left[\sum_k C^2(k)Y_1^2(k)Y_2^2(k) + \sum_k\sum_{\hat{k}\neq k} C(k)C(\hat{k})Y_1(k)Y_1(\hat{k})Y_2(k)Y_2(\hat{k})\right],$$
$$= E\left[\sum_k C^2(k)Y_1^2(k)Y_2^2(k)\right],$$
$$= \sum_k C^2(k)\left(2N + N^2\delta(k)\right)^2\sigma_1^2\sigma_2^2. \tag{A.26}$$

where the independency of $Y_1(k)$ and $Y_2(k)$ is used to simplify the second equality. The variance of z_2 is given by,

$$\sigma_{z_2}^2 = 4N^2\sigma_1^2\sigma_2^2\left[\sum_k C^2(n) + NC^2(0)\right]. \tag{A.27}$$

BIBLIOGRAPHY

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Academic Press, 2002.

[2] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding: Steganography and Watermarking- Attacks and Countermeasures*, Kluwer Academic Publishers, 2000.

[3] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking Digital Image and Video Data: A State of-the-art Overview," *IEEE Signal Processing Magazine*, vol. 36, no. 9, pp. 20-46, Sep. 2000.

[4] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, "Information Hiding-A Survey," *Proceedings of IEEE special issue on Protection of Multimedia Content*, vol. 87, pp. 1062-1078, July. 1999.

[5] F. Hartung, M. Kutter, "Multimedia Watermarking Techniques," *Proceedings of IEEE special issue on Protection of Multimedia Content*, vol. 87, pp. 1079-1107, July. 1999.

[6] A. Gurijala, and J. R. Deller, "Detector Design for Parametric Speech Watermarking," *IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 251-255, July 2005.

[7] L. Wen-Nung, and C. Li-Chun, "Robust and High-quality Time-domain Audio Watermarking Based on Low-frequency Amplitude Modification," *IEEE Transactions on Multimedia*, vol. 8, pp. 46- 59, Feb. 2006.

[8] L. Wei, X. Xiangyang, and L. Peizhong, "Localized Audio Watermarking Technique Robust Against Time-scale Modification," *IEEE Transactions on Multimedia*, vol. 8, pp. 60- 69, Feb. 2006.

[9] S. Jiande, and L. Ju, "A Temporal Desynchronization Resilient Video Watermarking Scheme Based on Independent Component Analysis," *IEEE International Conference on Image Processing*, vol. 1, pp. 265-268, Sep. 2005.

[10] K. Su, D. Kundur, and D. Hatzinakos, "Statistical Invisibility for Collusion-resistant Digital Video Watermarking," *IEEE Transactions on Multimedia*, vol. 1, pp. 43-51, Feb. 2005.

[11] G. Doerr, and J. Dugelay, "Security Pitfalls of Frame-by-frame Approaches to Video Watermarking," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2955-2964, Oct. 2004.

[12] E. Praun, H. Hoppe, A. Finkelstein, "Robust Mesh Watermarking," *Proceedings of Computer Graphic*, vol. 1, pp. 49-56, Aug. 1999.

[13] J. Cox, M. L. Miller, and J. A. Bloom, "Watermarking Applications and their Properties," *Proceedings of International Conference on Information Technology: Coding and Computing 2000*, vol. 2729, pp. 6-10, March. 2000.

[14] S. Craver, M. Wu, and B. Liu, "What can we Reasonably Expect from Watermark?," *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, vol. 1, pp. 223–226, Oct. 2001.

[15] F. Huang, Habib M. Hosseini, H. C. Chua, and Y.L. Guan, "Watermarking of Streaming Video for Finger-Printing Applications," *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 452-455, May 2002.

[16] L. D. Strycker, P. Termont, J. Vandewege, J. Haitsma, A. A. C. M. Kalker, M. Maes, and G. Depovere, "Implementation of a Real-Time Digital Watermarking Process for Broadcast Monitoring on a Trimedia Processor," *IEEE Proceedings of Vision, Image and Signal Processing*, vol. 147, pp. 371-376, 2000.

[17] G.W. Braudaway, K.A. Magerlein, F. Mintzer: "Protecting Publicly-Available Images With A Visible Image Watermark," *SPIE Conference on Optical Security and Counterfeit Deterrence Techniques*, vol. 2659, pp. 126-133, Feb. 1996.

[18] J. Meng, S-F. Chang, "Embedding Visible Video Watermarks in the Compressed Domain," *IEEE International Conference on Image Processing*, vol. 1, pp. 474 - 477, Oct. 1998

[19] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[20] G. W. Braudaway, K. A. Magerlein, and F. Mintzer, "Protecting Publically Available Images with a Visible Image Watermark," *Proceedings of SPIE, Optical Security and Counterfeit Deterrence Techniques*, vol. 2659, pp. 126-133, July 1996.

[21] W. W. Ping, and N. Memon, "Secret and Public Key Image Watermarking Schemes for Iimage Authentication and Ownership Verification," *IEEE Transactions on Image Processing*, vol. 10, pp. 1593-1601, Oct. 2001.

[22] K. Gopalakrishnan, N. Memon, and P. L. Vora, "Protocols for Watermark Verification," *IEEE Transactions on Multimedia*, vol. 8, pp. 66-70, Oct. 2001.

[23] C. Chin-Chen, and C. Chi-Yien, "An Enhanced Buyer Seller Watermarking Protocol," *International Conference on Communication Technology Proceedings*, vol. 2, pp. 1779- 1783, April 2003.

[24] Y. Lim, C. Xu, and D. D. Feng, "Web Based Image Authentication using Invisible Fragile Watermark," *Proceedings of the Pan-Sydney Area Workshop on Visual information Processing*, vol. 11, pp. 31-34, 2001.

[25] S. Katzenbeisser, and F. A. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, Inc, 2000.

[26] J. M. Acken, "How Watermarking adds Value to Digital Contents," *Communications of the ACM*, vol. 41, pp. 75-77, July 1998.

[27] J. Zhao, E. Koch, and C. Luo, "In Bussniess Today and Tomorrow," *Communications of the ACM*, vol. 41, pp. 67-72, July 1998.

[28] S. Wang, and Y. Lin, "Wavelet Tree Quantization for Copyright Protection Watermarking," *IEEE Transactions on Image Processing*, vol. 13, pp. 154 - 165, Feb 2004.

[29] A. Jain, and L. Hong, and R. Bolle, "On-Line Fingerprint Verification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 302-314, April 1997.

[30] C. Fei, D. Kundur, and R. Kwong, "Analysis and Design of Secure Watermark-based Authentication Systems," *IEEE Transactions on Information Forensics and Security*, vol. 1, pp. 43-55, March 2006.

[31] S. Ravi, A. Agarwal, and S. Ganesan, "Frequency Domain Real Time Digital Image Watermarking," *IEEE International Conference on Electro Information Technology*, vol. 1, pp. 1-6, May 2005.

[32] I.J. Cox, M.L. Miller, J.M.G. Linnartz, T. Kalker, "A Review of Watermarking Principles and Practices," *IEEE Digital Signal Processing for Multimedia Systems*, vol. 1, pp. 461-482, 1999.

[33] M. Kutter, F. Hartung, "Introduction to Watermarking Techniques," *Information Techniques for Steganography and Digital Watermarking*, vol. 1, pp. 97-119, Dec. 1999.

[34] P. Meerwald, A. Uhl, "Watermark Security via Wavelet Filter Parameterization," *International Conference on Image Processing*, vol. 3, pp. 1027-1030, 2001.

[35] S. Voloshynovskiy, O. Koval, M. Kivanc Mihcak and T. Pun, "The Edge Process Model and Its Application to Information Hiding Capacity Analysis", *IEEE Transactions on Signal Processing*, vol. 54, pp. 1813-1825, May 2006.

[36] H. S. Bassali, J. Chhugani, S. Agarwal, A. Aggarwal, and P. Dubey, "Compression Tolerant Watermarking for Image Verification," *IEEE International Conference on Image Processing*, vol. 1, pp. 430-433, Sep. 2000.

[37] P. Meerwald, A. Uhl, "A Survey of Wavelet-Domain Watermarking Algorithms," *Proceedings of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents*, vol. 4314, pp. 505-516, Jan. 2001

[38] A.H. Tewfik, "Digital Watermarking," *IEEE Signal Processing Magazine*, vol. 17, pp 17-88, Sep. 2000.

[39] J. Dugelay, S. Roche, "A Survey of Current Watermarking Techniques," *Information Techniques for Steganography and Digital Watermarking*, vol. 1, pp. 121-145, Dec. 1999.

[40] O. Bruyndonckx, J. J. Quisquater, and B. Macq, "Spatial Method for Copyright Labeling of Digital Images," *Proceedings of IEEE Workshop on Nonlinear Signal and Image Processing*, vol. 1, pp. 456–459, June 1995.

[41] J. R. Hernández, M. Amado, and F. P rez Gonzalez, "DCT-Domain Watermarking Techniques For Still Images: Detector Performance Analysis and a New Structure," *IEEE Transactions on Image Processing*, vol. 9, pp. 55-68, Jan. 2000.

[42] T. Holotyak, J. Fridrich and S. Voloshynovskiy, "Blind Statistical Steganalysis of Additive Steganography Using Wavelet Higher Order Statistics," *Conference on Communications and Multimedia Security*, vol. 2578, pp. 273-274, Sep. 2005.

[43] A. Nikolaidis, and I. Pitas, "Asymptotically Optimal Detection for Additive Watermarking in the DCT and DWT Domains," *IEEE Transactions on Image Processing*, vol. 12, pp. 563-571, May 2003.

[44] F. Y. Shih, and S. Y. T. Wu, " Combinational Image Watermarking in the Spatial and Frequency Domains," *Pattern Recognition*, vol. 36, pp.969-975, April 2003.

[45] P. Yang Zhao Campisi, and D. Kundur, "Dual Domain Watermarking for Authentication and Compression of Cultural Heritage Images," *IEEE Transactions on Image Processing*, vol. 13, pp. 430-448, March 2004.

[46] W. Dietl, P. Meerwald, and A. Uhl, "Protection of Wavelet-based Watermarking Systems using Filter Parametrization," *ACM Signal Processing*, vol. 83, pp. 2095-2116, Oct. 2003.

[47] R. Bangaleea and H. C. S Rughooputh, "Performance Improvement of Spread Spectrum Spatial-Domain Watermarking Scheme through Diversity and Attack Characterisation," *IEEE AFRICON, Africon Conference in Africa*, vol. 1, pp. 293–298, Oct. 2002.

[48] D. P. Mukherjee, S. Maitra, and S. T. Acton, "Spatial Domain Digital Watermarking of Multimedia Objects for Buyer Authentication," *IEEE Transactions on Multimedia*, vol. 6, pp. 1-15, Feb. 2004.

[49] N.F. Johnson, S.C. Katezenbeisser, "A Survey of Steganographic Techniques," *Information Techniques for Steganography and Digital Watermarking*, pp. 43-75, Dec. 1999.

[50] M. Barni, F. Bartolini, A. De Rosa, and A. Piva, "Capacity of the Watermark Channel: How Many Bits can be Hidden within a Digital Image?," *Proceedings of SPIE*, vol. 3657, pp. 437-448, Jan. 1999.

[51] M. Barni, F. Bartolini, A. De Rosa, and A. Piva, "A DWT-based Technique for Spatio-frequency Masking of Digital Signatures," *Proceedings of International Conference on Security and Watermarking of Multimedia Contents*, vol. 3657, pp. 31–39, Jan. 1999.

[52] M. Barni, F. Bartolini, and A. Piva, " Improved Wavelet-based Watermarking through Pixel-wise Masking," *IEEE Transactions on Image Processing*, vol. 10, pp. 783–791, May. 2001.

[53] J. J. K. O'Ruanaidh and T. Pun, "Rotation, Scale and Translation Invariant Digital Image Watermarking," *IEEE Signal Processing: Special Issue on Copyright Protection and Control*, vol. 66, pp. 303–317, May 1998.

[54] S. Pereira, S. Voloshynoskiy, and T. Pun, "Optimal Transform Domain Watermark Embedding via Linear Programming," *IEEE Transactions on Signal Processing*, vol. 81, no. 6, pp. 1251-1260, June 2001.

[55] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Images, Audio and Video," *Proceedings of International Conference on Image Processing*, vol. 3, pp. 243–246, Sep. 1996.

[56] Ching-Yung Lin, Min Wu, Jeffrey A. Bloom, Ingemar J. Cox, Matt L. Miller, and Yui Man Lui, "Rotation, Scale, and Translation Resilient Watermarking for Images," *IEEE Transactions on Image Processing*, vol. 10, pp. 767–782, May 2001.

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

[58] H. S. Malvar, and D. A. F. Florencio, "Improved Spread Spectrum: A New Modulation technique for Robust Watermarking," *IEEE Transactions Signal Processing*, vol. 51, pp. 898-905, April 2003.

[59] A. Briassouli, and P. Moulin, "Detection-theoretic Analysis of Warping Attacks in Spread-spectrum Watermarking," *Proceedings of Acoustics, Speech, and Signal Processing*, vol. 3, pp. 53-6, April 2003.

[60] L. Hua, and J. E. Fowler, "A Performance Analysis of Spread-spectrum Watermarking Based on Redundant Transforms," *IEEE International Conference on Multimedia and Expo*, vol. 2, pp. 553-556, 2002.

[61] W. C. Chu, "DCT-based Image Watermarking using Subsampling," *IEEE Transactions on Multimedia*, vol. 5, pp. 34- 38, March 2003.

[62] M. A. Suhail, and M. S. Obaidat, "Digital Watermarking-based DCT and JPEG Model," *IEEE Transactions on Instrumentation and Measurement*, vol. 52, pp. 1640-1647, Oct. 2003.

[63] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Transactions on Image Processing*, vol. 6, pp. 1673-1687, 1997.

[64] J. Cox, M. L. Miller, and A. L. Mckellips, "Informed Embedding: Exploiting Image and Detector Information During Watermark Insertion," *IEEE International Conference on Image Processing*, vol. 3, pp. 1-4, 2000.

[65] J. Mayer and R. A. Silva, "Efficient Informed Embedding of Multi-bit Watermark," *International Conference on Acoustic, Speech and Signal Processing*, vol. 3, pp. 389-392, 2004.

[66] J. Mayer, A. V. Silverio, and J. C. M. Bermudez, "On the Design of Pattern Sequences for Spread Spectrum Image Watermarking," *Telecommunication Symposium*, 2002.

[67] D. Kundur, *Multiresolution Digital Watermarking: Algorithms and Implications for Multimedia Signals*, PhD dissertation, 1999.

[68] D. Kundur and D. Hatzinakos, "Digital Watermarking using Multiresolution Wavelet Decomposition," in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, vol. 5, pp. 2969–2972, 1998.

[69] P. G. Flikkema, "Spread-Spectrum Techniques for Wireless Communication," *IEEE Signal Processing Magazine*, vol. 14, pp. 26-36, May 1997.

[70] S. Stankovic, I. Djurovic, and I. Pitas, "Watermarking in the Space/Spatial-frequency Domain using Two-dimensional Radon-Wigner Distribution," *IEEE Transactions on Image Processing*, vol. 10, pp. 650-658, Apr. 2001.

[71] B. G. Mobasseri, "Digital Watermarking in the Joint Time-frequency Domain," in *IEEE International Conference on Image Processing*, vol. 3, pp. 481–484, 2002.

[72] S. Erkucuk, S. Krishnan, and M. Zeytinoglu, "Robust Audio Watermarking using a Chirp Based Technique," in *IEEE International Conference on Multimedia and Expo*, vol. 2, pp. 513–516, July 2003.

[73] B. Barkat and F. Sattar, "A New Time-frequency Based Private Fragile Watermarking Scheme for Image Authentication," in *IEEE International Symposium on Signal Processing and Applications*, vol. 2, pp. 363–366, 2003.

[74] H. zer, B. Sankur, and N. Memon, "An SVD-based Audio Watermarking Technique", *International Multimedia Conference, Proceedings of the 7th workshop on Multimedia and security*, vol. 1, pp. 51–56, 2005.

[75] Y. Steinberg and N. Merhav, "Identification in the Presence of Side Information with Application to Watermarking," *IEEE Transactions on Information Theory*, vol. 47, pp. 1410–1422, May 2001.

[76] L. Cohen, *Time-Frequency Analysis*, Prentice Hall, New Jersey, 1995.

[77] http://www.wavelet.org/tutorial/tf.htm.

[78] http://www.cbi.dongnocchi.it/glossary/TimeFrequency.html.

[79] G. F. Boudreaux-Bartels and T. W. Parks, "Signal Estimation using Modified Wigner Distribuiton," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, vol. 22, pp. 3.1-3.4, March 1984.

[80] G. F. Boudreaux-Bartels and T. W. Parks, "Time-varying Filtering and Signal Estimation using Wigner Distribution Synthesis Techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing* , vol. 34, pp. 442-451, June 1986.

[81] K. B. Yu and S. Cheng, "Signal Synthesis from Wigner Distribution," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 1037-1040, March 1985.

[82] B. V. K. Kumar, C. P. Neuman, and K. J. Devos, "Discrete Wigner Synthesis," *IEEE Transactions on Signal Processing*, vol. 11, pp. 277-304, 1986.

[83] T. J. McHale and G. F. Boudreaux-Bartels, "An Algorithm for Synthesizing Signals from Partial Time-frequency Models using the Cross Wigner Distribution," *IEEE Transactions on Signal Processing*, vol. 41, pp. 1986-1990, May 1993.

[84] S. Rao Nelatury, B. G. Mobasseri, "Synthesis of Discrete-Time Discrete-Frequency Wigner Distribution", *IEEE Signal Processing Letters*, vol. 10, pp. 221-224, 2003.

[85] G. Cristobal, C. Gonzalo, and J. Bescos, "Image Filtering and Analysis Through the Wigner Distribution function," in *Advances in Electronics and Electron physics*, W. Hawkes, Ed., pp. 309–397. Academic Press, 1991.

[86] M. Al-khassaweneh and S. Aviyente, "Robust Watermarking on the Joint Spatial-spectral Domain," *Proceedings of IEEE Digital Signal Processing Workshop*, vol. 1, pp. 297–301, 2004.

[87] M. Al-khassaweneh and S. Aviyente, "A time-frequency Inspired Robust Image Watermarking," *IEEE Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 392–396, 2004.

[88] M. Al-khassaweneh and S. Aviyente, "Image Watermarking in the Autocorrelation Domain," *Proceedings of the 4th ACM international workshop on Contents protection and security*, vol. 1, pp. 53–58, Oct. 2006.

[89] M. Zeng and B. Liu, "A Statistical Watermark Detection Technique without using Original Images for Resolving Rightful Ownerships of Digital Images," *IEEE Transcations on Image Processing*, vol. 6, pp. 1534–1548, Nov. 1999.

[90] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 1991.

[91] "http://hlab.phys.rug.nl/imlib/index.html," .

[92] C. H. Lee and H. K. Lee, "Improved Autocorrelation Function Based Watermarking with Side Information," *Journal of Electronic Imaging*, vol. 14, pp. 0130121–01301213, Mar. 2005.

[93] P. Dong and N.P. Galatsanos, "Geometric Robust Watermarking through Watermark Pattern Shaping," in *Proceedings of International Conference on Image processing*, vol. 1, pp. 493–496, Sep. 2003.

[94] C. K. Chui, *Wavelets: A Tutorial in Theory and Applications*, Academic Press, California, 1992.

[95] D. Kundur and D. Hatzinakos, "Toward Robust Logo Watermarking using Multiresolution Image Fusion Principles," *IEEE Transactions on Multimedia*, vol. 6, pp. 185–198, 2004.

[96] G. Chang, B. Yu, and M. Vetterli, "Adaptive Wavelet Thresholding for Image Denoising and Compression," *IEEE Transactions on Image Processing*, vol. 9, pp. 1532–1546, Sept. 2000.

[97] J. Mayer and R. A. Silva, "Efficient Informed Embedding of Multi-bit Watermark," in *Proceedings of IEEE International Conference on Acoustics,Speech and Signal Processing*, vol. 3, pp. 389–392, 2004.

[98] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*, Wiley, New York, 2000.