# Digital Watermarking in Binary Document and Grayscale Images for Content Authentication

**Niladri Bihari Puhan**

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

**2007**

# Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Anthony T. S. Ho for his encouragement, guidance and support during my doctoral study. His valuable suggestions made it possible to complete the thesis within stipulated time. I am grateful to Prof. A. Makur for his useful comments and discussion during the development of algorithms. I would like to thank Prof. Pina Marziliano for her active help and guidance while designing of the perceptual model. I am grateful to Prof. Farook Sattar for his support and advice regarding the research procedure. Many thanks to Prof. Y. L. Guan for his help during subjective experiments and correction of the papers. I am thankful to Prof. I. J. Cox, University of London for his valuable suggestions regarding the exact authentication algorithm during his visit to the Center for Information Security, NTU, Singapore. I am thankful to Prof. Nasir Memon, Polytechnic University for sending his paper on binary image watermarking. A useful discussion regarding contour tracing procedure with Abeer George Ghuneim of McGill University through email helped me immensely during the implementation. The author would like to thank her a lot. I thank friends in my laboratory, Steve, Iris, Judy, Iqbal and Bappaditya for technical discussions. Without their timely cooperation, it would have been difficult for me to adjust in the new place. I should mention that without the support and care of my family members I could not have accomplished my education. Their sacrifice and nurturing always push me to strive for excellence. I dedicate my thesis to the supreme almighty who guides me with his blessings and gives me the hope to pursue for truth.

# Abstract

In recent times, information technology revolution has significantly transformed many parts of our lives. With increasing application of digital data in various forms such as image, video, music, documents and graphics, its security and authentic usage has posed important technological challenges for both industry and academia. This thesis discusses the issues regarding secure content authentication of binary document and grayscale images using digital watermarking techniques. By suitably hiding the watermark within a digital image, it is possible to verify whether it has been tampered by malicious attackers after the watermarking process. Using the hidden watermark, tamper localization and restoration of modified portions in the attacked image can be achieved. The difficulty in hiding the watermark of sufficient length within a binary document image arises due to its simple pixel statistics. A new perceptual model is designed towards the goal of selecting low-distortion pixels for imperceptible watermarking. Then an exact authentication method is proposed using the perceptual model such that any modification to the watermarked image could be detected with high accuracy. We have found that sufficient number of low-distortion pixels is not available in individual blocks of a binary document image for secure localization. A new localization method is proposed by constructing an erasable watermark and the method achieves high security after introducing erasable distortion in the watermarked image. A new restoration method is proposed for text document images using the error-control coding technique. Using this method, the original character sequence can be extracted after multiple attacks such as character deletion, insertion and substitution. The block-wise localization methods in binary document and grayscale image authentication suffer from a system level attack known as the

Holliman-Memon attack. To resist this attack, a new image index estimation algorithm is designed such that correct extraction of the unique image index is possible after fragile embedding. The advantage of this method is that a separate image index database is not necessary for user convenience. The authentication watermarking methods are found to be secure against various attacks and the security level is equivalent to the cryptographic authentication. Due to the use of fragile watermarks for content authentication, the proposed methods are particularly useful for binary document and grayscale images in electronic form.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

## Introduction

Due to the availability of systems for extensive use of digital data, significant interest in data hiding became perceptible in the last decade. It has become evident that intelligent hiding of a piece of data or information within another digital data could address many practical applications like covert communications, copyright protection and content authentication. In recent years, research and development of data hiding has made significant progress among the information technology community. Research in data hiding was initiated by its possible use for copyright protection of multimedia data. As the application domain of this evolving technology becomes broaden day by day, researchers use terms like data hiding or information hiding, steganography and digital watermarking to describe them. According to the definitions given in [1], data hiding is the general term encompassing a wide range of problems beyond that of embedding messages in content. The term hiding here can refer to either making information imperceptible or keeping the existence of the information secret. Steganography is a term derived from the Greek words *steganos*, which means "covered," and *graphia*, which means "writing". It is the art of concealed communication and the very existence of the message is secret. The hidden message is not related to the host signal which merely serves as a secret communication channel. Digital watermarking is defined as the practice of altering the data to embed a message about that data. Systems using data hiding can thus be divided into watermarking systems, in which the message is related to the original data, and non-watermarking systems, in which the message is unrelated to the original

data. The data could be an image, video, audio, vector graphics, documents, executable code or a combination of the above. In this thesis, we explore digital watermarking methods for binary document and grayscale images in particular. This introductory chapter describes properties of digital watermarking from a general perspective. Various methods developed for binary document and grayscale image watermarking are then discussed and the motivations addressed in this thesis are outlined.

## 1.1 A General Discussion On Digital Watermarking

One of the important reasons for significant interest shown towards digital watermarking is the risk of piracy in the digital domain. The availability of high-speed and inexpensive networks such as the Internet has brought convenient methods for the pirates to distribute copyright-protected digital data illegally. After digital recording and distribution over the Internet, the pirated copies do not have any quality degradation and hence find acceptability among users. Copyright protection has become a major concern among content producers and they are eagerly seeking technologies that can protect their rights against illegal copying. Traditionally, cryptography is widely used for secure distribution and protection of digital data. The data is encrypted before delivery and the encrypted data is transmitted to the legitimate users. Even if the pirates have access to the encrypted data, it is not useful without the decryption key. However, it is not possible to track the misuse by a legitimate user after decryption. The pirates could purchase a product which is in encrypted form and decrypt it using the appropriate key. Then illegal copies could be distributed without any difficulty and thus the content protection mechanism fails. Digital watermarking is seen as a promising alternative to the cryptographic

techniques for copyright protection, because the content producer can track piracy even after decryption of the product. The proprietary message is embedded within the data itself so that it is not destroyed after decryption and intentional or non-intentional processing. The message conveying the proprietary information is reliably extracted to track piracy or to establish the ownership issue in case of a conflict. Though copyright protection was the motivation behind the development of digital watermarking, it has been found that other applications like content authentication, broadcast monitoring, copy control, transaction tracking, proof of ownership and annotation can be addressed [2, 3, 4, 5].

Watermarking systems can be described in terms of a communication model consisting of three main elements: a transmitter, a communication channel and a receiver [6]. The embedding of the proprietary information within the original data plays the role of data transmission. Any intentional or non-intentional processing applied to the watermarked data represents the communication channel and the recovery of embedded information represents the receiver. The communication analogy of a watermarking system is illustrated in Figure 1.1. The data embedding module performs information coding and watermark embedding. As shown in Figure 1.1, the binary string $b$ is the input to the system and it is the watermark code. In many watermarking systems, $b$ is transformed into a watermark signal $w$ which is embedded within the original data $A$ imperceptibly and with variable robustness depending on the application it addresses. In some cases, the watermark code $b$ is directly embedded without any transformation. While embedding, proper perceptual modeling is performed to minimize any perceptual distortion in the watermarked data. After embedding, the watermarked data $A_w$ enters the channel where it undergoes

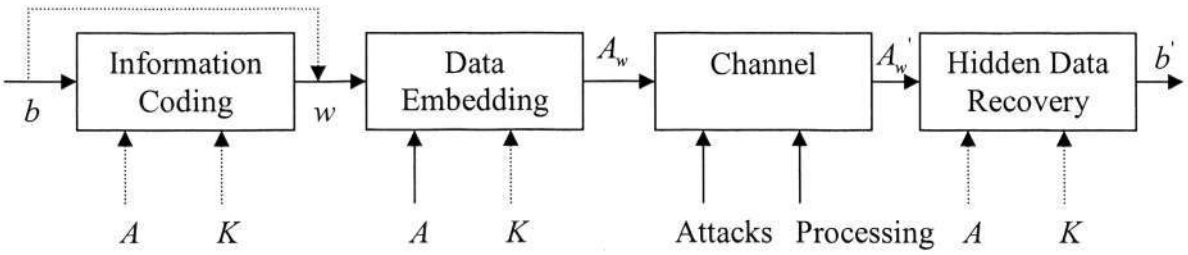



Figure 1.1: Communication model of a watermarking system.

various manipulations. Some manipulations may be performed explicitly by the hostile attacker aiming to destroy the watermark. There may be non-intentional processing such as compression, noise due to transmission or editing. In the recovery part of the system, the watermark code $b'$ is extracted from $A_w'$ by using the secret key $K$. In some cases, the recovery system gives a yes/no decision regarding the presence of the watermark. In watermarking terminology, three modules described above are known as the embedder, attacks and detector. A watermarking system is characterized by a number of important properties such as imperceptibility, capacity, robustness, blind and non-blind detection and security. These properties are briefly described below.

## (a) Imperceptibility

Imperceptibility of a watermarking system refers to the perceptual similarity between the original and watermarked data. Imperceptibility after watermark embedding is possible due to the imperfections of human senses. Therefore, still image and video watermarking methods rely on the characteristics of the human visual system, while audio watermarking exploit the properties of the human auditory system.

## (b) Capacity

The capacity of a particular watermarking technique is loosely identified with the number of information bits it is able to convey reliably, for a given class of attacks. Capacity is a fundamental property of any watermarking method which often determines whether a method is suitable for a given application or not. Different applications require different capacities and there exists a trade-off against imperceptibility and robustness, as shown in Figure 1.2. Higher capacity is obtained at the expense of either robustness or imperceptibility and a suitable trade-off is found depending on the application. For example, high capacity may not be necessary in copyright protection, but the embedded bits should be robust enough to survive various attacks. In secure authentication applications, higher capacity is necessary to embed sufficient information bits and robustness is not an issue. In fragile watermarking, the watermark should be destroyed after any kind of tampering to the content and so it is not robust to any class of attacks. In authentication scenario, the receiver has to make a binary decision (authentic vs. non-authentic). To arrive at this decision, it is necessary to embed a certain number of watermark bits, which is commonly termed as *capacity* in the watermarking literature. In the information theoretic perspective, there could be as little as one bit of information in authentication applications. So in this sense, the commonly used *capacity* definition in watermarking literature may not be a relevant notion.

## (c) Robustness

Watermark robustness is the ability to detect the embedded data after signal processing manipulations. Examples of various manipulations are lossy compression, geometric distortions (rotation, translation and scaling), printing and scanning,

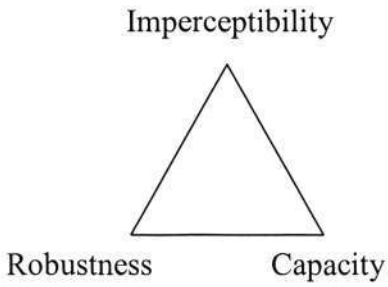



Figure 1.2: The watermarking trade-off.

filtering and noise addition. Based on robustness level of the watermark, there can be three types of watermarking methods such as robust watermarking, semi-fragile watermarking and fragile watermarking. In robust watermarking, the watermark should survive all signal processing manipulations that are likely to occur after the embedding process. Robust watermarking is useful in applications such as copyright protection. In semi-fragile watermarking, the watermark should be robust against a limited set of unintentional or system manipulations such as JPEG compression. A watermark is said to be fragile if it is destroyed as soon as any modification is applied to the watermarked data. The main application of fragile watermarking is in content authentication.

## (d) Blind and Non-blind Detection

A watermarking method is blind if it does not require the original data for watermark recovery. Conversely, a watermarking method is non-blind if the original data is needed to extract the watermark. Blind detection is useful in content authentication, copy control applications, while non-blind detection is used in transaction-tracking.

### (e) Security

The security of a watermark refers to its ability to resist hostile attacks. A hostile attack is any process specifically intended to defeat the watermark's purpose for a particular application. Unauthorized removal, unauthorized embedding and unauthorized detection of the watermark are considered as hostile attacks. The level of watermark security is decided on the particular type of application. According to the Kerckhoff's principle, the security of the watermarking algorithm should reside in the secrecy of the key, not in the obscurity of the algorithm [6]. Even if the attacker has complete knowledge of the algorithm, yet he/she should not be able to destroy or recover the watermark.

## 1.2 Overview of Image Watermarking Methods

### 1.2.1 Grayscale Image Watermarking

In this section, we shall describe various methods for watermarking of grayscale images whose pixels range from 0 to 255. Early work in this area was about modification of the least significant bits (LSB) in the image for watermark embedding [7]. The LSB modification method could achieve high capacity and imperceptibility due to the fact that LSB information is perceptually insignificant. However, the embedded watermark using the LSB modification method is not robust to any class of attacks. In [8], Bender *et al* introduced a spatial data hiding method known as "patchwork" algorithm. In this method, $n$ pairs of pixels were randomly selected and intensity of one pixel was enhanced while the other's intensity was reduced. The change in the brightness value of the pixel pair was used to embed one bit of watermark. However, the method was susceptible to noise and to any arbitrary scrambling of the LSB of an image. In [9], Pitas *et al* proposed another method which

is similar to patchwork. The watermark was embedded in the spatial domain by slightly modifying the intensity level of randomly selected pixels. Detection was performed by comparing the mean intensity value of the marked pixels against that of the unmarked pixels. The embedded watermark was found to be robust against JPEG compression and low-pass filtering.

Koch and Zhao proposed a method for embedding watermark in the discrete cosine transform (DCT) of the image [10]. The watermark was embedded in some regions of the image which were randomly selected and this method was useful for copyright labeling. Swanson *et al* proposed a robust watermarking method in the DCT domain [11, 12]. The original image was segmented into blocks and DCT of each block was computed. Perceptual mask was computed using a visual masking model which takes the frequency sensitivity and spatial masking property of the Human Visual System (HVS) into account. The perceptual mask was used to calculate the allowable alteration for the DCT coefficients. Block-wise embedding of the watermark was found to be robust against cropping and localized signal processing manipulations. The correlation detector was used for watermark detection and original image was necessary during detection.

In [13], Cox *et al* introduced the concept of embedding the watermark in the perceptually significant coefficients of an image. In this method, known as "spread spectrum watermarking" 2-D DCT of the entire image was computed and 1000 largest coefficients were selected for embedding. A Gaussian sequence of same length was multiplied with chosen scaling factor and then added to the selected DCT coefficients. Detection was performed by correlating the Gaussian sequence with the extracted

watermark sequence. The extracted watermark sequence was obtained after subtraction of the modified DCT coefficients from the corresponding DCT coefficients of the original image. This method was found to achieve significant robustness against various attacks as compared to its previous methods. The requirement of the original image for detection is the disadvantage of this method. In [14], Barni et al modified the original spread spectrum watermarking method so that the original image was not necessary for detection. After zigzag ordering of the DCT coefficients, a particular set of coefficients were chosen for embedding. This method was found to be robust against various attacks. In [15], block-wise DCT was used and the blocks with high activity were selected for embedding. The embedding was done in the mid-frequency DCT coefficients.

Podilchuck and Zeng suggested an image adaptive watermarking algorithm in [16]. The watermark was embedded in each block adaptively based on Watson's just noticeable difference (JND) model [17]. The embedding was performed in DCT and wavelet domains and the method was found to be robust against common image processing attacks and cropping. A linear correlation detector was used for watermark detection. For detection in wavelet domain, a normalized correlation coefficient was calculated separately for each sub-band. For each resolution level, the average value is calculated. Similarly the average correlation was calculated over a specified frequency orientation. The maximum correlation value over all possible levels as well as frequency orientation were chosen and compared to a threshold. A dynamic bit allocation based watermarking scheme was proposed in [18]. Mid-frequency coefficients in the DCT domain were selected for embedding and the choice of number and locations of the coefficients were flexible. The basic idea of the variable

bit allocation was that the blocks having more non-zero DCT components were embedded with more bits while others were used to embed with fewer bits. Preprocessing of the watermark like delta coding and block coding was performed to improve the detection performance. Fridrich proposed a hybrid watermarking scheme in [19] by embedding a low-frequency watermark in the low frequency DCT coefficients of an image. A spread-spectrum watermark was also embedded in the mid-frequency DCT coefficients. Both watermarks were embedded in a different portion of the frequency space to minimize interference. The resulting double watermarked image was robust against a wide range of attacks.

Kundur and Hatzinakos proposed a wavelet based image watermarking method in [20]. The wavelet decomposition of the watermark was added to the wavelet decomposition of the original image after scaling. The scaling factor was determined by using proper HVS modeling. The original image was necessary for detection and the method was found to be robust against JPEG compression, additive noise and linear filtering. In [21], Wang *et al* proposed a blind watermarking method in which significant wavelet coefficients were selected in significant wavelet subbands to embed the watermark. Experimental results demonstrated that the embedded watermark was robust against compression attacks. In [22], a multiresolution watermarking method was proposed using the discrete wavelet transform. The original image was decomposed in a hierarchical manner and the large coefficients at the high and middle frequency bands were embedded. This method was robust to common image distortions, such as the wavelet transform compression, rescaling and halftoning.

It is well known that phase spectrum preserves more information than magnitude [23, 24]. Since the phase is perceptually more significant than the magnitude, it seems logical to embed the watermark in the phase for achieving robustness. Runnaidh *et al* proposed a method in which the watermark was embedded in the phase of the original image [25]. It was proved that phase spectrum would be less affected after Additive White Gaussian Noise (AWGN) attack, if DFT coefficients having the largest magnitudes were modified. In [26], the authors introduced a rotation, scale and translation (RST) invariant watermarking method. Invariance to translation could be obtained by taking the Discrete Fourier Transform (DFT) of the original image and selecting the DFT magnitude. Then the DFT magnitude was mapped to log-polar coordinates. Translation invariance in the log polar domain was equivalent to scaling and rotational invariance in the spatial domain. So first taking the DFT of the log polar mapping and selecting the magnitude approximately could produce the RST invariant domain. The watermark was then embedded in the RST invariant domain and blind detection was possible using this method.

Based on the stochastic approach and texture masking property of the HVS, a robust algorithm against removal based attacks was proposed in [27]. The original image was modeled as either a non-stationary Gaussian or stationary generalized Gaussian process and the watermark was modeled as a stationary Gaussian process. For the non-stationary Gaussian model case, the original image was estimated using an adaptive Wiener filter or Lee filter [28]. For the second case, the closed form solution was known as soft-shrinkage [29]. It was found that estimation of the original image was difficult in high activity or texture region. This coincides with the fact that in texture areas noise visibility would be less than smooth areas of an image. Hence it

was proposed to embed the watermark in the texture region so that the watermark was not damaged in the estimation based attacks and imperceptibility could be maintained. Ramkumar and Akansu analyzed different watermarking methods in [30] with the aim to resolve the ownership issue. In order to defeat the purpose of the attacker, a new method was proposed which could effectively increase the computational complexity and defends against of an attack by a factor of over $10^{100}$. Different restrictions were also suggested in their method. These included the use of a fixed random sequence generator, a blind watermarking scheme and high detection statistics. In [31], Craver *et al* proposed an attack known as 'ambiguity attack' or 'Craver attack'. In this attack, the authors demonstrated the possibility of false ownership claim of a distributed image through reverse engineering of the watermarking process. The attacker could generate the fake watermark and fake original image by exploiting the invertibility property of a watermarking method. A watermarking method is said to be invertible if the inverse of the embedding process is computationally feasible. To counter this attack, the reference pattern should be dependent on the content of the original image by using one-way hash functions.

Lu *et al* proposed an image watermarking method based on vector quantization [32, 33]. Vector quantization (VQ) is a lossy data compression technique wherein the vectors are quantized rather than scalars [34]. The watermark was embedded into the codeword indices while ensuring that the distortion is less than a given threshold. The watermark was robust against the VQ compression with the same codebook. A watermarking method based on *Variable Dimension Vector Quantization* (VDVQ) [35] was suggested in [36]. While conventional VQ has fixed dimension code vectors, VDVQ has code vectors of varying dimension. Watermark bits were embedded in the

dimension information of the variable dimension reconstruction blocks of the original image. Both blind and non-blind variations of the method were possible. Since the embedded watermark could be removed after encoding and decoding, the method was less robust to attacks.

### 1.2.2 Binary Document Image Watermarking

Most previous image watermarking methods are for grayscale or color images in which the pixels take on a wide range of values. For those images, changing pixel values by a reasonable margin is imperceptible to the human eye. For images in which the pixels take on only a limited number of values, embedding the watermark without any visual distortion becomes more challenging. The methods developed for grayscale images cannot be directly applied to binary document images where the pixels can take on a value of either 0 (black) or 1 (white). A different class of watermarking methods has to be designed for binary document images due to different pixel statistics. The availability of suitable watermarking methods could address various applications using document images. Binary document images could potentially include digitized versions of text, circuit diagrams, signature, driver licenses, financial and legal documents, maps and drawings. Most document images are binary in nature and consist of a black foreground and a white background. For example, the foreground may contain characters of different fonts and sizes in text documents, lines and symbols in maps and drawings and text with signatures belonging to different persons in legal documents. In some binary documents, grayscale images are represented as halftone images. In such images, binary patterns are used to approximate gray level values of a grayscale image. The human visual system performs spatial integration of the fine binary patterns within local regions and

perceives them as different intensities. By using different image processing software, the distribution and editing of document images become easier. As such the ownership protection, authentication and annotation of binary document images have become important in recent years. In this section we review the watermarking methods in binary document images proposed in the literature.

Low *et al* [37, 38, 39, 40] introduced robust watermarking methods for formatted document images based on imperceptible line and word shifting. This method was applied to embed information in text images for bulk electronic publications. The detection process used a maximum-likelihood detector after the distortions and noise were corrected and removed. The line shifting method was found to have low capacity but the embedded data was robust to photocopying, scanning and printing process. The word shifting method could offer higher capacity than the line shifting method but the robustness was reduced to printing, photocopying and scanning. Brassil *et al* proposed a method in [41], where the height of the bounding box enclosing a group of words could be used for embedding. This method has a better data hiding capacity than line and word shifting methods. It was also robust to distortions caused by photocopying. For each mark, one or more adjacent words on an encodable text line were selected for displacement according to a selection criterion. The words immediately before and after the shifted words and a block of words on the reference line remain unchanged. The heights of characters in these unshifted blocks were used as "reference heights" for decoding. However this method was sensitive to document skewing. Chotikakamthorn [42] performed data embedding by adjusting widths of a few consecutive character spaces on the same text line. This method was proposed to overcome the shortcomings of word spacing method in application to many written

languages such as Chinese, Japanese and Thai, which do not have spaces with sufficiently large width as a word boundary. This method has embedding capacity comparable to that of the word shifting method and can also survive document duplication. In [43], the proposed algorithm slightly modified interword spaces so that different lines across a text act as sampling points of a sine wave. After the modification, the average spaces of various lines have the characteristics of a sine wave and the wave constituted a mark. This algorithm claimed to withstand different noise and distortion attacks.

Wu *et al* [44, 45] hid data in a binary image using a hierarchical model in which human perception was taken into consideration. Distortion that occurred due to flipping of a pixel was measured by considering the change in smoothness and connectivity of a 3×3 window centered at the pixel. In a block, the total number of black pixels is modified to be either odd or even for embedding the data bits. Shuffling [46] was used to equalize the uneven embedding capacity over the image. Koch and Zhao [47] proposed a data hiding algorithm in which a data bit '1' is embedded if the percentage of white pixels was greater than a given threshold, and a data bit '0' is embedded if the percentage of white pixels was less than another given threshold. A sequence of contiguous or distributed blocks was modified by flipping the pixels until such threshold was reached. This algorithm was not robust to attacks and the hiding capacity was low. Mei *et al* modified an eight-connected boundary of a connected component for data hiding [48]. A fixed set of pairs of five-pixel long boundary patterns have been identified for embedding data. One of the patterns in a pair required deletion of the center foreground pixel, whereas the other required the addition of a foreground pixel. A unique property of the method is that the two

patterns in each pair are dual of each other. This property allowed for blind detection of watermarking.

Amamo and Misaki proposed a feature calibration method in which text areas in an image were identified and the geometry of the bounding box of each text line was calculated in [49]. Each bounding box was divided into four partitions and grouped into two sets. The average width of the horizontal strokes of characters was computed as a feature. To calculate the feature, vertical black runs with lengths less than a threshold were selected and averaged. Two operations – "make fat" and "make thin" were defined by increasing and decreasing the lengths of the selected runs, respectively. To embed a '1' bit "make fat" operation was applied to partitions belonging to the first set and "make thin" operation was applied to partitions belonging to the second set. The opposite operations were used to embed '0' bit. During detection, the stroke width features were extracted from the partitions and added up for each set. If the sum was larger than a positive threshold, then the detection process would output '1'. If the difference was less than a negative threshold, the output would be '0'. An approach targeted on the copier system was presented in [50] for embedding data in text pages that might also contain color images or graphics. Small regions that consisted of text type pixels were first identified and the lightness of these regions was modulated to embed data. This change is imperceptible to human eyes yet detectable by scanners. To achieve robustness error correction coding was used and this method was found to be robust against printing and scanning.

Lu *et al* proposed an objective distortion measure for binary document images based on human perception [51, 52]. This method, known as the distance-reciprocal distortion measure (*DRDM*) measured the distortion due to data hiding in binary images. Distortion due to the flipping of a pixel is measured by taking two factors into consideration; the number of pixels (with same value) in a 5×5 window centered on the pixel to be flipped and each such pixel generated distortion equal to the inverse of its normalized distance to the center pixel. In [53], the watermark was embedded in the DC coefficients after the original image was transformed to the DCT domain. Watermark embedding in the frequency domain is not possible directly, because the embedded watermark is destroyed due to the post-embedding binarization process. To avoid this, the method employed binarization with a biased threshold and the watermark could survive some common image processing attacks. In [54], Lu *et al* proposed a method in which the watermark was embedded by enforcing the odd-even features of non-uniform blocks. During watermarking, the flippable pixels were selected based on the distance reciprocal distortion measure. To provide security in tamper-proofing and authentication, 2-D shifting was employed. Yang and Kot proposed a method in which the embedding was performed by using a smoothing technique [55]. Gold-like sequence was used as the watermark and the watermarked image was an enhancement to the original binary image in terms of smoothness. The noise suppression patterns were used in the embedding process which took smoothness in a 5×5 pixel neighborhood into consideration. This method had a moderate watermark capacity while preserving the image quality. In [56], a method for watermarking the text documents was proposed by using both inter-character and word spaces. In a text document, the inter-character spaces are smaller than the inter-word spaces. These two kinds of spaces were utilized to improve watermark capacity

and the word spaces could compensate for the insufficient inter spaces between characters. The proposed method claimed to achieve better watermark capacity as compared to conventional line shifting and word shifting methods.

Several watermarking methods have been suggested for halftone images that are used in the printed (analogue) media such as magazines, newspapers, printer outputs. These methods are particularly useful for halftone images, and are not suitable for other category of document images like text, drawings and cartoons which have sharply-contrasted boundaries. In [57], Baharav and Shaked proposed a method in which the watermark was embedded during the halftoning process. The basic idea was to use two dither matrices (instead of one) to encode the watermark. This method required an original grayscale image during the embedding process. Wang proposed modified ordered dithering and modified multi-scale error diffusion techniques for embedding the watermark in printed documents [58]. In the first method, one of the 16 neighboring pixels used in the dithering process was replaced in an ordered manner. The second method modified the binarization sequence of the error diffusion process based on the global and local properties of intensity in the original image. The results showed that the quality of the watermarked image was not affected after the embedding process. In [59], stochastic screen patterns were used where two screens formed two halftone images and the data was embedded through correlation between two screens. The embedded data could be recovered when two patterns were overlaid. Fu and Au proposed a method, named as DHST (data hiding by self toggling) for halftone image watermarking [60]. In this method, a pseudo-random number generator with a known seed was used to generate a set of pseudo-random locations. A watermark bit could be embedded at each location by making it black or white.

During detection, the same random number generator was used to read the pixel values at the embedded locations.

In [61], Fu and Au presented a new method for improving the visual quality of the watermarked image. The method is known as data hiding by pair toggling (DHPT). For DHPT, a pair of white and black pixels was chosen to change at the pseudo-random locations. This improved quality by preserving local average intensity and reducing the number of undesirable clusters of white or black pixels. In [62], a method called intensity selection (IS) was proposed to select the best location out of several candidate locations for watermark embedding. Improvement in visual quality could be obtained without sacrificing data hiding capacity using this approach. The method chose pixel locations that were either very bright or very dark. A data bit was embedded as the parity of the sum of the halftone pixels at $M$ pseudo-random locations and selected the best out of the $M$ possible locations.

Robustness to printing, scanning and photocopying is an important issue when the documents are distributed in analogue form. The line and word shifting approaches are found to be robust to printing, scanning, and photocopying operations. However, these methods have low capacity which does not make them suitable for authentication application. The methods using pixel flipping and feature modification approach are not robust to printing and scanning, but they offer higher capacity. These methods are useful in applications when documents are distributed in the electronic form and robustness to distortions is not necessary. An interesting discussion and review of various watermarking and data hiding methods for binary document images can be found in [63].

## 1.3 Objectives

Tampering in the digital domain using different readily available software tools has become increasingly easier and common nowadays. Perceptual similarity of the tampered data with the original makes it difficult for normal users to detect the forgery. The ease of forgery in the digital domain raises the question of authenticity and secure usage of the digital data as legal evidence in a court of law. To prove that the data in question is not forged, digital signature based techniques are being incorporated in cryptography. Digital signature is an encrypted summary of the data and a public key cryptography system [64] is used to create a digital signature. An asymmetric key encryption algorithm is used so that the private key required to encrypt the signature is different from the public key required to decrypt it. Unauthorized persons cannot create a new signature of the data and any modification of the data can be detected with high probability. These signatures are the metadata and they should remain along with the data for verification purpose. If the signatures are lost during usage, then the authentication task becomes impossible. For example if the signature of an image is stored in its header field, the signature is lost after converting the image to another format. Watermarking opens up the possibility to solve the problem of managing the metadata for authentication. Instead of storing the signature in the header file, embedding metadata as the watermark directly into the content brings important advantages into picture. We shall term the embedded watermark as the authentication signature. Any change in the representation of the data does not affect its authenticity and there is no need for separate space or specific format to accommodate the authentication signature. If the authentication signature is embedded block-wise in an image, then tamper localization is possible. By knowing the portions of data tampered, the attacker's motive behind such tampering can be

known. Either an exact or approximate version of the original data can be restored after modifications by making use of the extracted signature. If the authentication signature is suitably designed, then it can survive some innocuous manipulations like lossy compression. It is also possible to extract relevant information about the modifications that occurred after the watermarking process. Instead of just knowing whether the data has been tampered or not similar to the case of digital signature, the user has the choice to acquire more information by using the embedded watermarks.

In this thesis, we shall investigate content authentication application for binary document and grayscale images in electronic form using watermarking methods. Researchers have suggested various methods in this direction. In our investigation, we have found that most of the proposed authentication watermarking methods are suitable for grayscale images. While several important issues are still unsolved for content authentication in grayscale images, binary document image authentication has received limited attention amongst the watermarking community. Most of the existing binary image watermarking methods deals with the robust and annotation applications. However, the potential advantages of secure authentication watermarking for localizing changes, restoration of the original binary image and identifying the type of modifications have yet to be fully realized. In the following chapters, we shall propose new watermarking methods for secure authentication of binary document images. Our discussion in the following chapters does not include the case of halftone document images. In addition to binary document image authentication, we shall propose a new method to resist a system level attack, known as the Holliman-Memon attack [65]. The proposed method deals with the security of

block-wise independent watermarks against this attack and it is useful for both binary document and grayscale image authentication.

## 1.4 Thesis Organization

This thesis is organized as follows: In Chapter 1, a discussion of existing image watermarking methods and objectives of the thesis is presented. In Chapter 2, a new method for exact authentication of binary document images using perceptual watermarking is proposed. The proposed authentication watermarking methods in Chapter 3 use erasable watermarks for secure detection and localization of tampering. After tamper localization, it is possible to achieve approximate restoration in case of text document images. Authentication watermarking schemes for tamper localization are vulnerable to the Holliman-Memon attack. In Chapter 4, we propose a new method using unique image index to resist the Holliman-Memon attack. Chapter 5 concludes the thesis and some future works are discussed in this chapter.

# Chapter 2

# Exact Authentication in Binary Document Images Using Perceptual Modeling

## 2.1 Introduction

In the content authentication application, the most basic procedure is to determine whether the image under test has been modified or not after the watermarking process. If any of the pixels are modified, then the attacked image should be detected as inauthentic. This type of authentication procedure is known as *exact authentication* [1]. In exact authentication, fragile watermarks are suitably embedded such that any modification to the watermarked image easily destroys the watermark. As such, the detection algorithm will be able to detect every possible modification. Watermarking in the least-significant-bits of a grayscale image is an example of fragile watermarking. Most of the image processing operations alter the least significant bits; so the embedded watermark becomes undetectable after any such processing. The fragile watermark should be designed such that the attacker should not be able to forge a valid watermark within an attacked image. For this purpose, it is necessary for the watermark to be derived from the original image. Although fragile watermarks using predefined patterns (i.e. independent of the original image) can detect tampering, an attacker can easily forge such a fragile watermark. For example, in the case of an LSB watermark, forgery is a matter of copying the least significant bits from the authentic image to attacked image. For this reason, it is necessary for the watermark to be derived from the original image. A scheme with proper secrecy and

proper image feature selection can also be used for authentication application. This type of scheme is generally used in semi-fragile authentication scenario where it is desirable to resist common signal processing operations. In content authentication, the attacker tries to create the forged image out of the authentic image(s) to fulfill his/her interest. During forgery, the attacker has to ensure that the forged image contains a valid watermark and the aim is to keep the watermark intact while modifying the content. In contrast, the attacker seeks to destroy the watermark in robust watermarking while ensuring that the perceptual quality of the modified content remains acceptable. Fragile watermarking schemes in grayscale images have been proposed in [66, 67, 68].

In a typical cryptography-based authentication watermarking scheme, an authentication signature (either a message authentication code or digital signature) is computed from the whole image and embedded into the image itself. However, the very process of embedding a watermark alters the image, thus causing the subsequent authentication test to fail. To prevent this, it is necessary to partition the image into two parts, one of which is to be authenticated and the other part to be altered to accommodate a watermark. An example is to partition an image such that the least-significant-bit (LSB) plane holds the authentication signature computed from the remaining bits of the image. Cryptography- based authentication schemes have been proposed for grayscale images in [69, 70]. Our present work addresses the issue of exact authentication for binary document images in electronic form in conjunction with cryptography techniques. Incorporating cryptography makes it possible to design a secure authentication watermarking scheme. There are only a limited number of cryptography-based authentication watermarking methods available for binary

images. It is difficult to embed the signature in binary images as described above, because each pixel has only one bit. By modifying any pixel to embed a watermark would affect the signature of the image and the authentication test would fail. The challenging problem is how to divide the binary image into two parts such that the authentication signature can be embedded successfully.

Cryptography-based authentication watermarking schemes have been proposed for binary images in [71, 72]. Kim *et al* modified few bits in a binary image for embedding the authentication signature and the positions of those bits were known in both the embedding and the detection processes [71]. These pixels were cleared before computing the hash function. However this method of simple partitioning the binary image results in poor visual quality of the watermarked image. In [72], Kim *et al* shuffled the binary image and then partitioned the shuffled image into two equal parts. Authentication signature was computed from one part and then embedded into the other part using the block-wise data hiding technique developed in [44, 45]. A block in the second part of the image was embedded one bit of the authentication signature by modifying its total number of black pixels to be either odd or even. In this method, the first part is secure because the probability of undetected modification in this part is only $2^{-n}$ where $n$ is the length of the authentication signature. However the second part of the image which carries the signature is prone to a *'parity attack'*. The parity attack arises because the signature is embedded in the second part by considering the parity of the blocks, the number of black pixels. If two pixels that belong to the same block in the second part of the image change their values, the parity of this block may not change and so this modification will pass undetected. In the same paper, the proposed algorithm was modified to minimize the possibility of a

parity attack. Thus each block in the second part of the image would have different probabilities of suffering due to parity attack and without being detected. As such this method is not provably secure against any type of modification to the watermarked image. A new method has been proposed for text document images to tackle the issue of parity attack in [73, 74]. This method used block-wise hiding technique [44, 45] developed for watermark embedding. The signature was embedded only in the blocks that contain flippable pixels. The method was secure in using the non-interlaced blocks since the embeddability of the blocks was found to be invariant during the embedding process. However if interlaced blocks were used, a possibility of false tamper detection could arise because the embeddability of some blocks might change during embedding process. To increase the security in this case, the authors suggested applying shuffling to the original image or to the embeddable and unembeddable blocks.

In this chapter, we propose a new approach to embed the authentication signature in binary document images to detect any modifications to the watermarked image. Watermarking schemes for secure authentication application need high capacity. Hiding considerable amount of data in binary images is a difficult problem due to the simple pixel statistics of such images. In the embedding process, a perceptual model is necessary to minimize the visual distortion in the watermarked images. A new perceptual model is proposed towards selecting low-distortion pixels for watermarking and the subjective experimental results are presented to validate the perceptual model. The proposed perceptual model is used for designing a new authentication watermarking method to achieve security against every possible modification including the parity attack. In this new approach, necessary conditions

are suggested such that the original image can be partitioned into two parts. The pixel-wise embedding of the authentication signature will then remove any possibility of parity attack.

## 2.2 Proposed Perceptual Model

Traditional objective distortion measures that are widely used in image processing tasks include mean square error (*MSE*), signal-to-noise ratio (*SNR*), and peak signal-to-noise ratio (*PSNR*). For binary images these traditional distortion measures are not well correlated with human perception, because in this case all three measures only take the number of flipped pixels into account. The distortions in the binary images can be different even if the number of flipped pixels is the same. In this section, we propose a new perceptual model based on the curvature-weighted distance difference (*CWDD*) measure between two contour segments. This model is well correlated with the human perception in estimating distortion due to the flipping of a pixel. By using this model, it is possible to select suitable contour pixels for watermarking so that the watermarked image remains perceptually similar to the original image. We begin by recalling some definitions followed by model formulation and then subjective experiments to validate the proposed perceptual model.

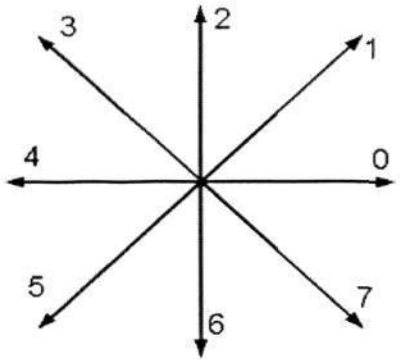
### 2.2.1 Definitions

**Contour segment:** The contour segment that passes through the flipping pixel is defined as an ordered set of contour pixels and it is represented by a set of 8-directional chain codes [75]. A contour segment with a set of $n$ pixels $\{p_i\}$, $i = 0,1,...,n-1$ can be represented by $(n-1)$ chain codes $\{c_i\}$, $i = 1,2,...,n-1$,

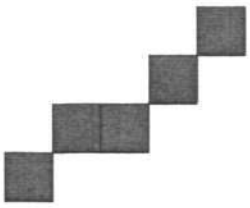
where chain code $c_i$ is the direction from pixel $p_{i-1}$ to $p_i$. An example of the 8-directional chain code representation of a contour segment is given in Figure 2.1.

**Euclidean distance:** Consider the two pixels $p_{i-1}$ and $p_i$ with coordinates $(x_1, y_1)$ and $(x_2, y_2)$, respectively, then the Euclidean distance between pixel $p_{i-1}$ and $p_i$ is defined by:

$$d_i = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad i = 1,2,...,n-1 \tag{2.1}$$




(a)



(b)

Figure 2.1: (a) 8-directional chain code representation, (b) $\{1,0,1,1\}$ and $\{5,5,4,5\}$ are the chain code representations for the contour segment, when starting from the bottom and top pixel respectively.

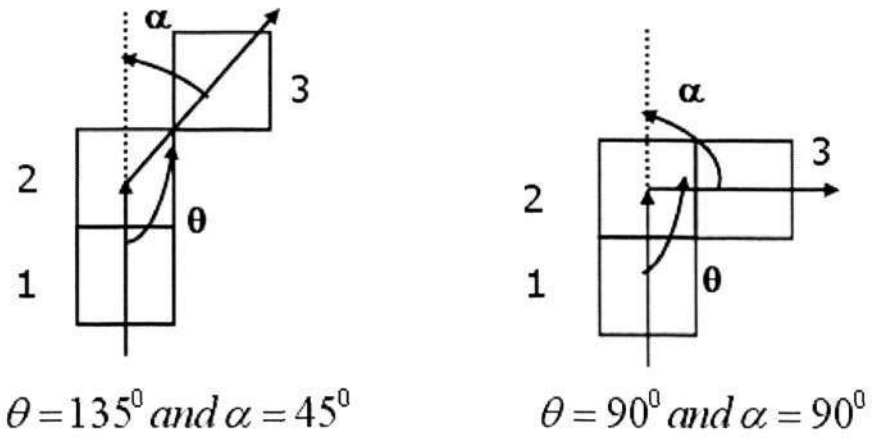
**Curvature:** Given an ordered set of pixels $P = \{p_i\}$, $i = 0,...,n-1$, the curvature at pixel $p_i$ is given by

$$\alpha_{p_i} = \begin{cases} (180^0 - \theta_i) & i = 1,2,...,n-2 \\ 0^0 & i = 0, n-1 \end{cases} \quad (2.2)$$

where $\theta_i \in [0,180^0]$ is the angle between the continuous segments $[p_{i-1}, p_i]$ and $[p_i, p_{i+1}]$. The curvature as defined above is symmetric in both clockwise and anticlockwise directions. The examples of curvature are shown in Figure 2.2.

### 2.2.2 Logic behind the Model

After flipping a contour pixel, the amount of visible distortion can be estimated by the change in the contour segment that passes through the pixel. If the length of the contour segment increases or decreases after flipping, the Euclidean distance along the contour segment would reflect this change. While traversing a contour segment from the start to end pixel, let us consider the following process steps. At each step there may or may not be a change in the direction with respect to the previous step. More changes in any direction would indicate that the contour segment is jagged. If there are fewer changes then it is considered to be a smoother contour segment. More changes in the directions at each step amounts to higher visibility of distortion. As such, the Euclidean distance at each step is weighted according to the curvature for indicating the level of visible distortion. Hence, in this model the distortion due to the flipping of a pixel is estimated by the *CWDD* measure between two contour segments.




$\theta = 135^0 \; and \; \alpha = 45^0$ $\theta = 90^0 \; and \; \alpha = 90^0$

Figure 2.2: The examples of curvature values '$\alpha$' at pixel-2.

### 2.2.3 Model Formulation

To calculate the distortion score for a contour pixel to be flipped, the 5-pixel length 'original contour segment' passing through this pixel is extracted by the contour tracing algorithm [76, 77]. Similarly after flipping the pixel, the 'watermarked contour segment' is also extracted. Since the contour segment is represented by chain code, this code can be efficiently applied to derive the curvature and distance values. The Euclidean distance $d_i$ between two pixels $p_{i-1}$ and $p_i$ can be computed from $c_i$ by using the following equation

$$d_i = \begin{cases} 1 & \text{mod}(c_i, 2) = 0 \\ \sqrt{2} & \text{mod}(c_i, 2) = 1 \end{cases} \qquad i = 1, 2, ..., n-1. \tag{2.3}$$

The curvature value $\alpha_{p_i}$ at pixel $p_i$ is 0 for $i = 0$ and $i = n-1$. For $i = 1, ..., n-2$, $\alpha_{p_i}$ as defined in Equation 2.2 may be computed from the chain codes $c_i$ and $c_{i+1}$ as:

$$\alpha_{p_i} = \begin{cases} \beta & \beta \le 180^0 \\ \left(360^0 - \beta\right) & otherwise \end{cases} \tag{2.4}$$

where

$$\beta = |c_{i+1} - c_i| \times 45^0. \tag{2.5}$$

After obtaining the curvature value at each pixel, a weight sequence $\{w_i\}$, $i = 0,1,...,n-1$ where

$$w_i = \begin{cases} 1 & \alpha_{p_i} = 0^0 \\ 1.5 & \alpha_{p_i} = 45^0 \\ 3 & \alpha_{p_i} = 90^0 \\ 4.5 & \alpha_{p_i} = 135^0 \\ 6 & \alpha_{p_i} = 180^0 \end{cases} \tag{2.6}$$

and $w_i$ is chosen to be monotonic to the curvature value $\alpha_{p_i}$. The curvature-weighted distance ($D_\alpha$) of a contour segment is then defined by:

$$D_\alpha = \sum_{i=1}^{n-1} w_{i-1} \square d_i \ . \tag{2.7}$$

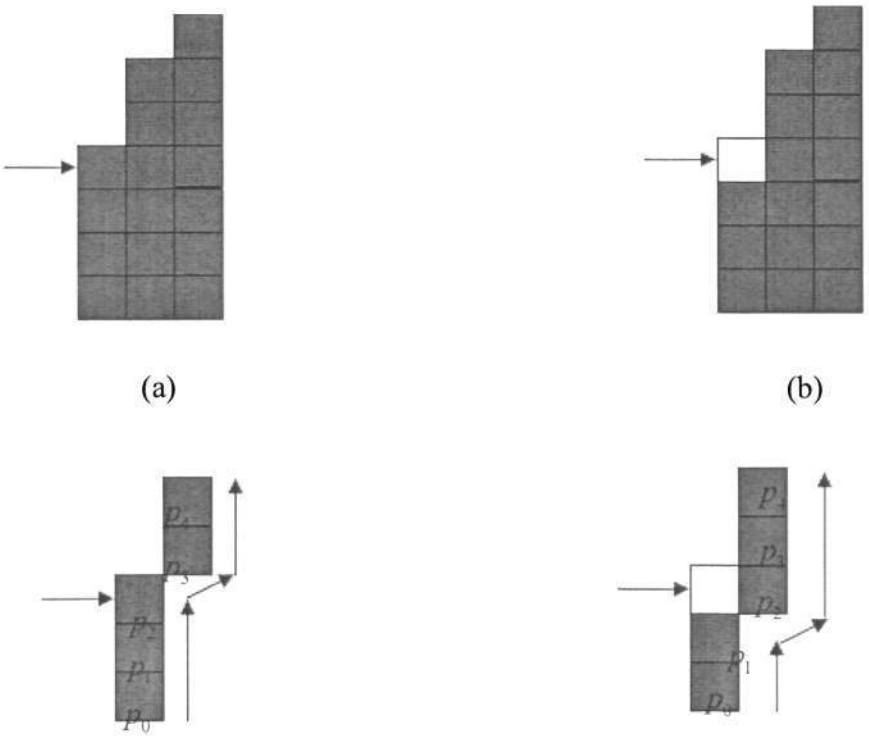
Let $D_\alpha^{original}$ and $D_\alpha^{watermarked}$ be the curvature-weighted distances of the original and watermarked contour segments, respectively. The *CWDD* measure for the flipped pixel is then given by:

$$CWDD = \left| D_\alpha^{original} - D_\alpha^{watermarked} \right| \ . \tag{2.8}$$

**Example:**

Using the proposed method, the *CWDD* measure computation is illustrated by an example as shown in Figure 2.3 (a)–(d). The arrow marked black pixel shown with its neighborhood in (a) is considered for the *CWDD* measure computation. In (b), the black pixel is flipped to white. The 'original contour segment' is shown with the

tracing sequence from the pixels in (c). Similarly the 'watermarked contour segment' is shown with the tracing sequence from the pixels in (d). The *CWDD* measure is then computed by first calculating the CWD of the original contour segment and then the CWD of the watermarked contour segment.



(a) (b)

(c) (d)

Figure 2.3: Example of *CWDD* computation: (a) The arrow marked black pixel shown with its neighborhood; (b) the black pixel is flipped to white; (c) tracing sequence in the 5-pixel length 'original contour segment'; (d) tracing sequence in the 'watermarked contour segment'.

**CWD of the original contour segment:**

The chain codes are $c_0 = 2,\ c_1 = 2,\ c_2 = 1,\ c_3 = 2$ and the distance values between the pixels are $d_1 = 1,\ d_2 = 1,\ d_3 = \sqrt{2},\ d_4 = 1$. The curvature values (in degrees) are $\alpha_{p_0} = 0,\ \alpha_{p_1} = 0,\ \alpha_{p_2} = 45,\ \alpha_{p_3} = 45,\ \alpha_{p_4} = 0$ and so the weight sequences are $w_0 = 1,\ w_1 = 1,\ w_2 = 1.5,\ w_3 = 1.5,\ w_4 = 1$. Therefore, the *CWD* for the original contour segment is

$$D_\alpha^{original} = w_0 d_1 + w_1 d_2 + w_2 d_3 + w_3 d_4$$

$$= 1\times 1 + 1\times 1 + 1.5\times\sqrt{2} + 1.5\times 1 = 5.62.$$

**CWD of the watermarked contour segment:**

The chain codes are $c_0 = 2,\ c_1 = 1,\ c_2 = 2,\ c_3 = 2$ and the distance values between the pixels are $d_1 = 1,\ d_2 = \sqrt{2},\ d_3 = 1,\ d_4 = 1$. The curvature values (in degrees) are $\alpha_{p_0} = 0,\ \alpha_{p_1} = 45,\ \alpha_{p_2} = 45,\ \alpha_{p_3} = 0,\ \alpha_{p_4} = 0$ and so the weight sequences are $w_0 = 1,\ w_1 = 1.5,\ w_2 = 1.5,\ w_3 = 1,\ w_4 = 1$. Therefore, the *CWD* for the watermarked contour segment is

$$D_\alpha^{watermarked} = w_0 d_1 + w_1 d_2 + w_2 d_3 + w_3 d_4$$

$$= 1\times 1 + 1.5\times\sqrt{2} + 1.5\times 1 + 1\times 1 = 5.62$$

So the *CWDD* measure of the pixel is

$$CWDD = \left| D_\alpha^{original} - D_\alpha^{watermarked} \right| = 0$$

The *CWDD* measure can not be computed in case of the isolated black contour pixel, because in this case contour tracing is not possible.

### 2.2.4 Subjective Experiments

The purpose of the subjective experiments is to validate the correlation between model ratings and human perception. Using the Adobe Photoshop software, four characters 'A', 'B', 'E', 'S' are converted to binary images of size 128×128 pixels shown in Figure 2.4. The choice of the characters is made on the basis of their different individual contour characteristics such as vertical, horizontal and curvature segments. In binary images, contours of different characters and symbols contain mainly these types of characteristics. To illustrate, the contour of the character 'A' has only horizontal and diagonal lines. Some part of the contour in the character 'B' is curved and the rest is horizontal and vertical. The contour of the character 'E' is totally smooth, i.e. there are only horizontal and vertical lines. The contour of the character 'S' is curved only. In case of binary images, different characters and symbols contain mainly these types of contour characteristics. The design of a number of independent test images is important for the purpose of subjective experiments. However, it is difficult to produce a test image for each value of *CWDD* measure because there may not be sufficient number of pixels of one particular value to flip in the original image and there will be a large number of such test images to handle. To overcome this difficulty, we use the technique called binning to produce the test images. We divide the *CWDD* range from [0, 8] into nine bins. With the exception of the first one, each bin has a length of one in terms of the *CWDD* measure. The first bin consists of the pixels with *CWDD* value equal to zero. For the second bin, the starting point of the bin is greater than zero and the end point is one. Then from the third bin to the last, there is an increment of one in the bin starting point. For each bin we choose to flip a maximum of ten pixels in the original image to produce one test image. To avoid the interference between flipping pixels, a minimum distance

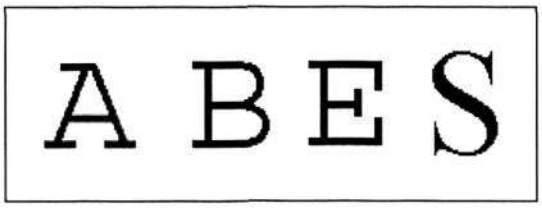



Figure 2.4: The set of binary images used in the subjective experiments.

between them is always maintained. Within a 7×7 pixel window, there cannot be more than one flipped pixel, even if the total number of available flipped pixels is less than 10. If the total number of flipped pixels in a test image is less than four, then the test image is not considered valid for subjective experiment. Because flipping an inadequate number of pixels may not be visible to the human eye for any value of *CWDD* measure. The distortion in the test image is computed as the mean of the *CWDD* values of flipped pixels and it is denoted by $CWDD_{mean}$. A total of 34 test images were produced and the subjective assessment was performed by 25 observers. Each observer was asked to rate the amount of impairment in the test image with reference to the original image, on a discrete 5-level scale ranging from 0 to 4 as shown in Table 2.1. For each test image, the mean opinion score (*MOS*) was computed by taking the mean of the 25 subjective scores.

### 2.2.5 Performance Attributes

To characterize the perceptual model in terms of its performance with respect to the subjective ratings, two performance attributes are used.

Table 2.1: Rating scale

| *Impairment* | *Score* |
|---|---|
| Imperceptible | 0 |
| Perceptible but not annoying | 1 |
| Slightly annoying | 2 |
| Annoying | 3 |
| Very annoying | 4 |

- Correlation coefficient ( $\rho$ ) for a set of $n$ data pairs $(x_i, y_i)$, $i = 0,...,n-1$ is a number between -1 and 1, which measures the degree to which two variables are linearly related and is defined as follows [78]:

$$\rho = \frac{n\sum_{i=0}^{n-1} x_i y_i - \sum_{i=0}^{n-1} x_i \sum_{i=0}^{n-1} y_i}{\sqrt{n\sum_{i=0}^{n-1} x_i^2 - \left(\sum_{i=0}^{n-1} x_i\right)^2}\sqrt{n\sum_{i=0}^{n-1} y_i^2 - \left(\sum_{i=0}^{n-1} y_i\right)^2}} \tag{2.9}$$

- Monotonicity measures the increase (decrease) in one variable is associated with increase (decrease) in other variable, independently of the magnitude of the increase (decrease). The degree of monotonicity can be defined by the Spearman rank-order correlation coefficient, which is defined as follows [78]:

$$r_s = 1 - \frac{6\sum_{i=0}^{n-1} d_i^2}{n(n^2-1)} \tag{2.10}$$

where

$$d_i = X_i - Y_i \tag{2.11}$$

and $X_i$ is the rank of $x_i$ and $Y_i$ is the rank of $y_i$ in the ordered data series.

### 2.2.6 Experimental Results

The results showed high correlation between the *CWDD* perceptual model and the subjective test data. The *MOS* was computed for each test image. The plots of subjective *MOS* versus $CWDD_{mean}$ are shown in Figure 2.5 (a)-(d) for the characters 'A', 'B', 'E' and 'S' respectively. In Figure 2.5 (e), the plots are obtained by combining the four data sets of all the characters and 26 points out of 34 remained within the 95% confidence interval. The performance attributes, Spearman rank-order correlation coefficients ($r_s$) and correlation coefficients ($\rho$) were computed between the subjective *MOS* and $CWDD_{mean}$ for all the test cases. The results summarized in Table 2.2 show that our perceptual model matches very well with the human perception in estimating distortion due to the flipping of a pixel. The reason for the poor performance of the model in case of the character 'E' may be due to the following reason. Since the contour of the character 'E' is smooth, hiding data is quite perceptible. The observer is able to see the difference even at low distortion because of the smooth contour. So even if the embedding distortion increases, it does not have much difference subjectively and the subjective ratings become random with respect to the model prediction. To illustrate the application of the proposed perceptual model for identifying the low-distortion pixels, we consider three binary images containing text (see Fig. 2.6a), drawing (see Fig. 2.7a) and signature (see Fig. 2.8a). After computing the *CWDD* measure, the pixels in the original images which have *CWDD* measure within the range from 0 to 1 are flipped. It was observed that a significant number of pixels could be modified in the selected images without noticeable visual

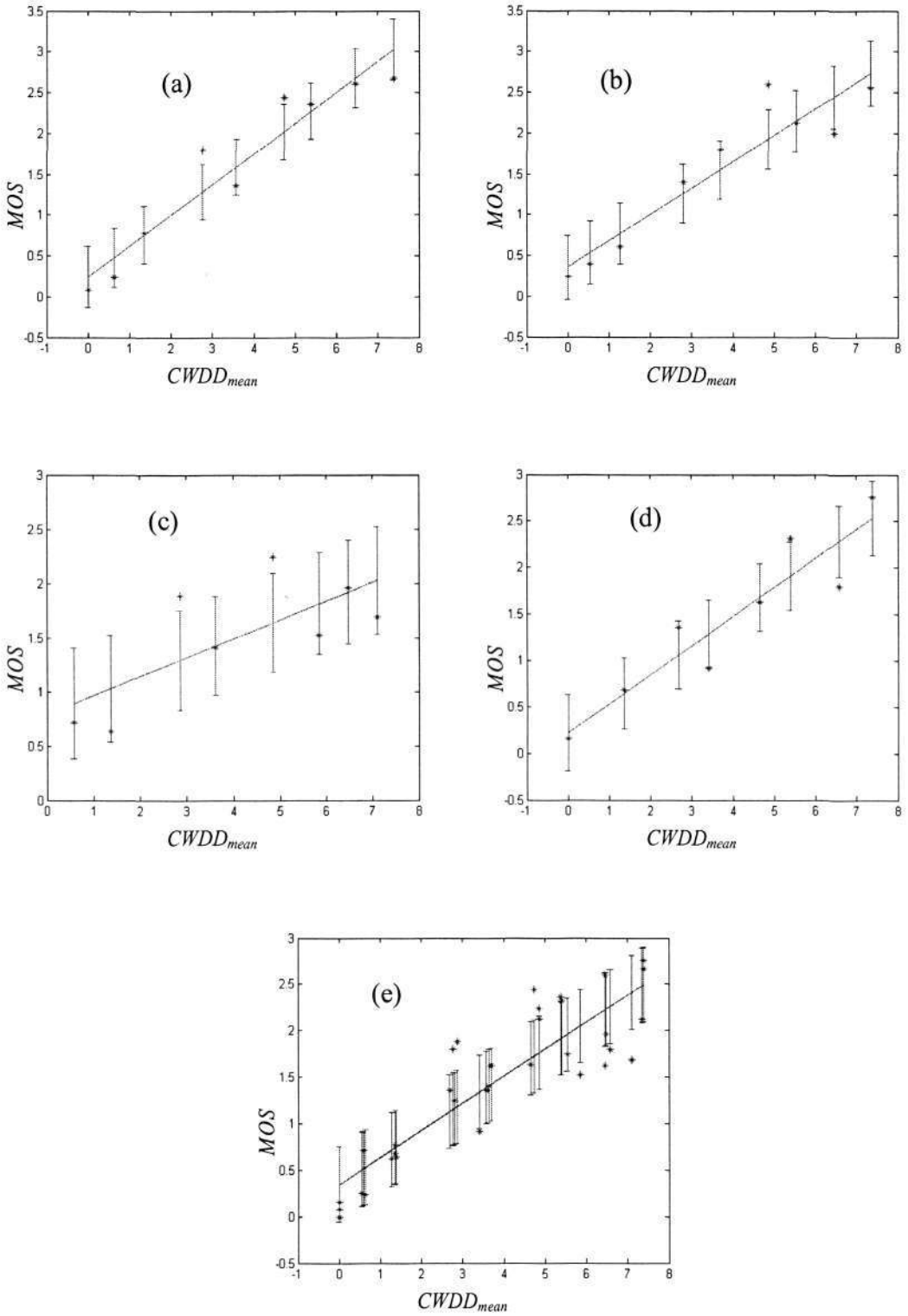



Figure 2.5: Subjective *MOS* vs. $CWDD_{mean}$ (a) character 'A', (b) character 'B', (c) character 'E', (d) character 'S', (e) combination of four data sets of all the characters. The error bars indicate the 95% confidence intervals of the subjective ratings.

Table 2.2: Performance attributes for all the test cases

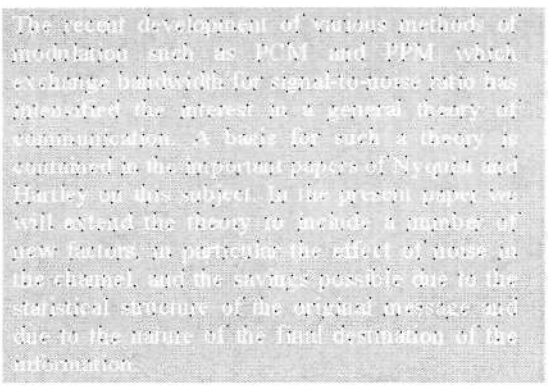
| *Character* | $r_s$ | $\rho$ |
|---|---|---|
| A | 0.97 | 0.96 |
| B | 0.88 | 0.94 |
| E | 0.60 | 0.72 |
| S | 0.95 | 0.93 |
| All | **0.85** | **0.88** |

The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist and Hartley on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

(a)

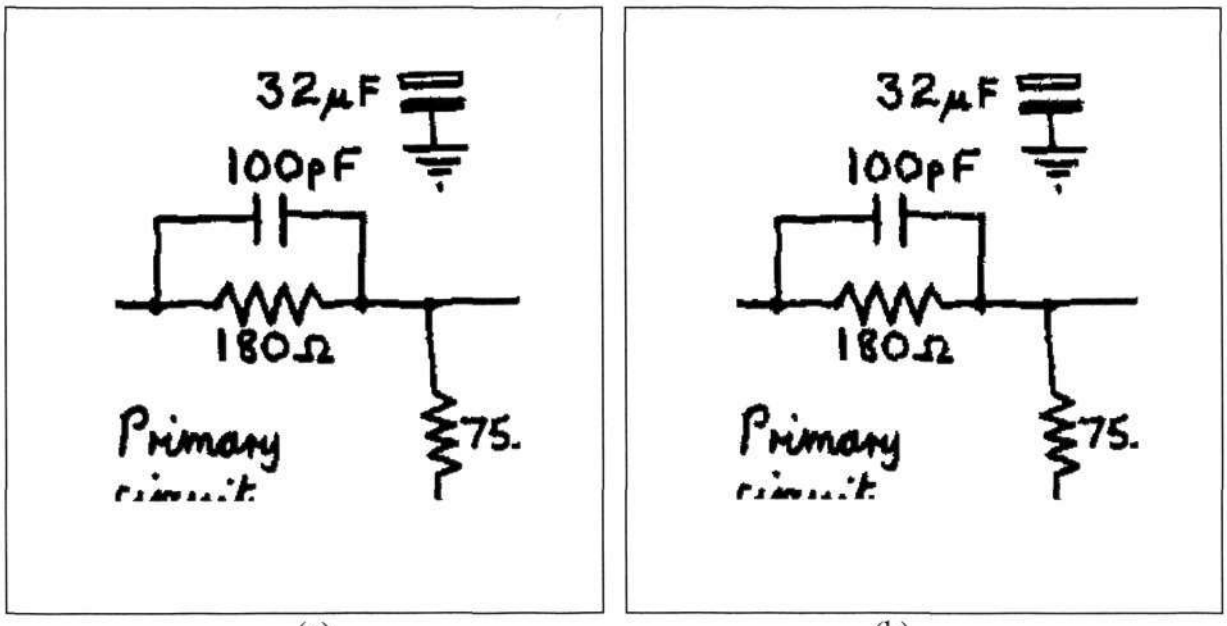
The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist and Hartley on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

(b)

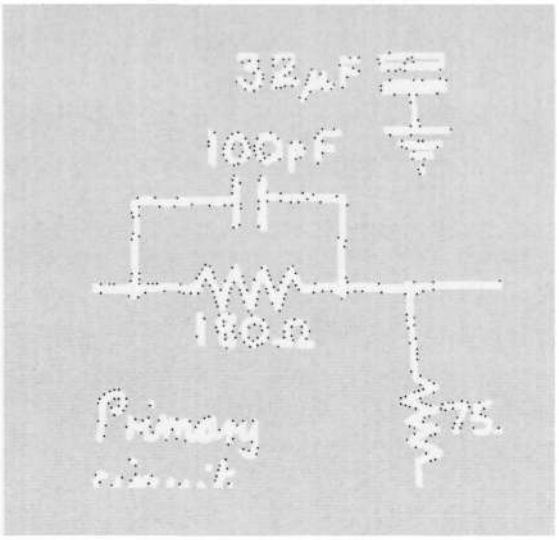


(c)

Figure 2.6: (a) The original text image of size 320×440 pixels; (b) image with 500 pixels flipped in the *CWDD* range from 0 to 1; (c) the difference image shows the flipped pixel positions.

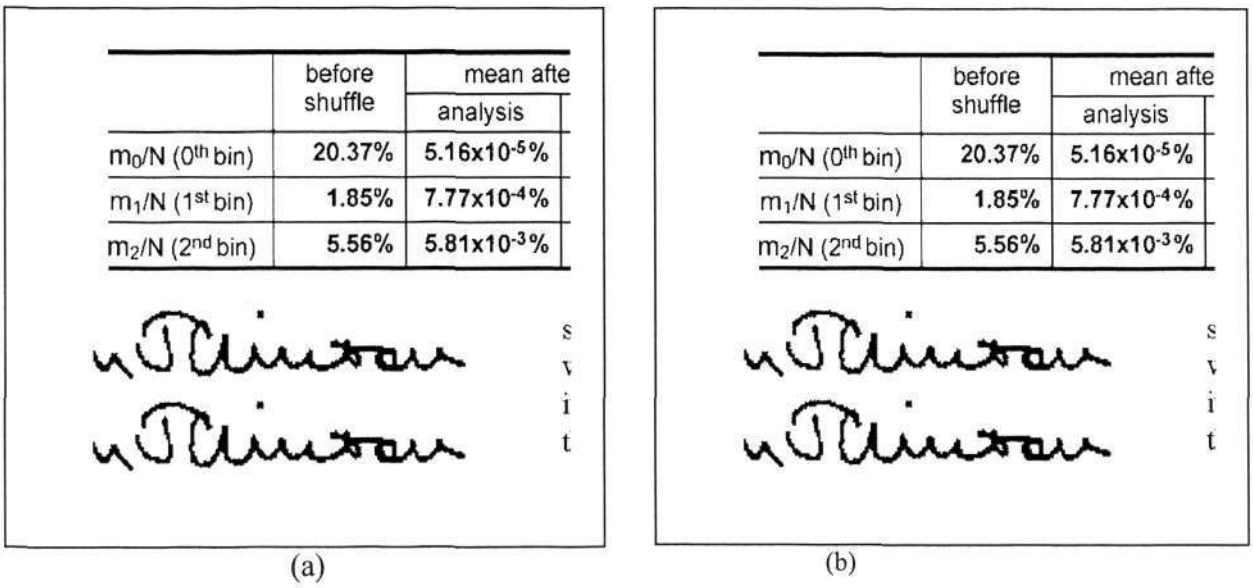



(a) (b)

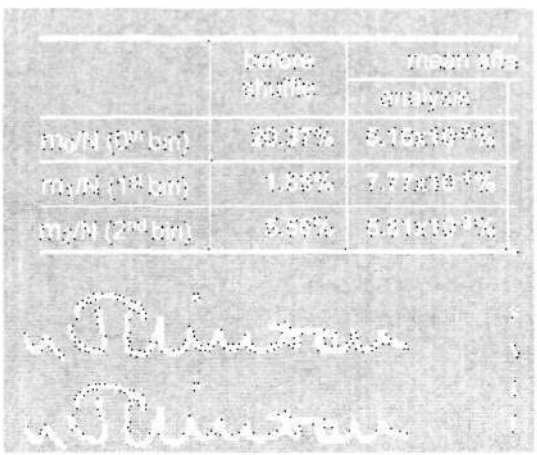


(c)

Figure 2.7: (a) The original drawing image of size 368×386 pixels; (b) image with 450 pixels flipped in the *CWDD* range from 0 to 1; (c) the difference image shows the flipped pixel positions.



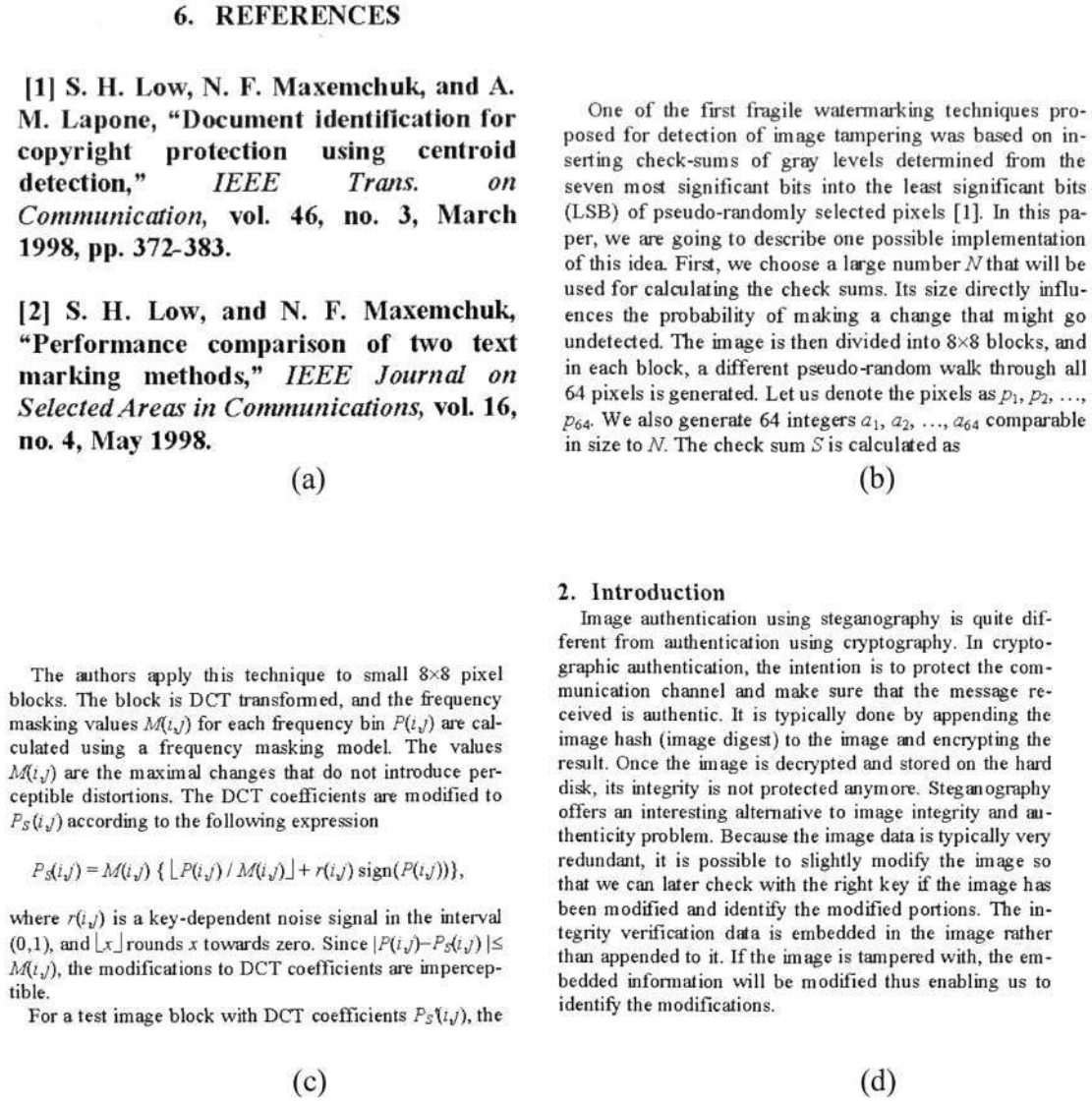

(a) (b)



(c)

Figure 2.8: (a) The original image of size 450×535 pixels containing text and signature; (b) the image with 600 pixels flipped in the *CWDD* range from 0 to 1; (c) the difference image shows the flipped pixel positions.

distortion. Figures 2.6-2.8 illustrate the original and modified images of different categories and the flipped pixel positions. The experimental results presented above demonstrate the satisfactory quality of the watermarked images generated by the proposed method. It is desirable to compare the quality of the watermarked images generated by the proposed perceptual method and the prior arts in [44, 45, 51, 52]. For this purpose, we conduct another subjective experiment using five original images. The first original image is shown in Figure 2.7(a) and other four original images are shown in Figure 2.9 (a)–(d). A certain number of low-distortion contour pixels are



6. REFERENCES

[1] S. H. Low, N. F. Maxemchuk, and A. M. Lapone, "Document identification for copyright protection using centroid detection," *IEEE Trans. on Communication*, vol. 46, no. 3, March 1998, pp. 372-383.

[2] S. H. Low, and N. F. Maxemchuk, "Performance comparison of two text marking methods," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, May 1998.

(a)

One of the first fragile watermarking techniques proposed for detection of image tampering was based on inserting check-sums of gray levels determined from the seven most significant bits into the least significant bits (LSB) of pseudo-randomly selected pixels [1]. In this paper, we are going to describe one possible implementation of this idea. First, we choose a large number $N$ that will be used for calculating the check sums. Its size directly influences the probability of making a change that might go undetected. The image is then divided into 8×8 blocks, and in each block, a different pseudo-random walk through all 64 pixels is generated. Let us denote the pixels as $p_1, p_2, \ldots, p_{64}$. We also generate 64 integers $a_1, a_2, \ldots, a_{64}$ comparable in size to $N$. The check sum $S$ is calculated as

(b)

The authors apply this technique to small 8×8 pixel blocks. The block is DCT transformed, and the frequency masking values $M(i,j)$ for each frequency bin $P(i,j)$ are calculated using a frequency masking model. The values $M(i,j)$ are the maximal changes that do not introduce perceptible distortions. The DCT coefficients are modified to $P_S(i,j)$ according to the following expression

$$P_S(i,j) = M(i,j)\{\lfloor P(i,j)/M(i,j)\rfloor + r(i,j)\,\mathrm{sign}(P(i,j))\},$$

where $r(i,j)$ is a key-dependent noise signal in the interval (0,1), and $\lfloor x \rfloor$ rounds $x$ towards zero. Since $|P(i,j)-P_S(i,j)| \le M(i,j)$, the modifications to DCT coefficients are imperceptible.

For a test image block with DCT coefficients $P_S(i,j)$, the

(c)

2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

(d)

Figure 2.9: Test images used in the subjective experiment.

flipped in each original image using the proposed *CWDD* measure, the *DRDM* model [51, 52] and the *Princeton* model [44, 45]. The image size and the number of flipped pixels for each original image are given in Table 2.3. A total of 15 test images were generated using the three methods and the subjective assessment was performed by 10 observers. Each observer was asked to rate the amount of impairment in the test image with reference to the original image, on a discrete 5-level scale ranging from 0 to 4 as shown in Table 2.1. For each test image, the mean opinion score (*MOS*) was computed by taking the mean of the 10 subjective scores. The mean opinion scores for the test images are given in Table 2.3. For each method, it is observed that the mean opinion scores lie in the range of [0, 1] in four out of five test cases. This shows the perceptual quality achieved by the three methods to be satisfactory. To compare between the proposed method and the Princeton model in terms of the perceptual quality, we observe that: (1) The Princeton model performs better than the proposed method in case of the original images 1, 2 and 5; and (2) both methods have similar performance for the original images 3 and 4. Between the proposed method and the *DRDM* model: (1) The proposed method performs better than the *DRDM* model for the original image 4; (2) the *DRDM* model performs better than the proposed method for the original image 1; and (3) both methods have similar performance for the original images 2, 3 and 5. Thus, it can be summarized that the Princeton model's performance is better than other two methods and the performance of the proposed method is comparable with the *DRDM* model.

Table 2.3: Mean opinion score for all the test cases

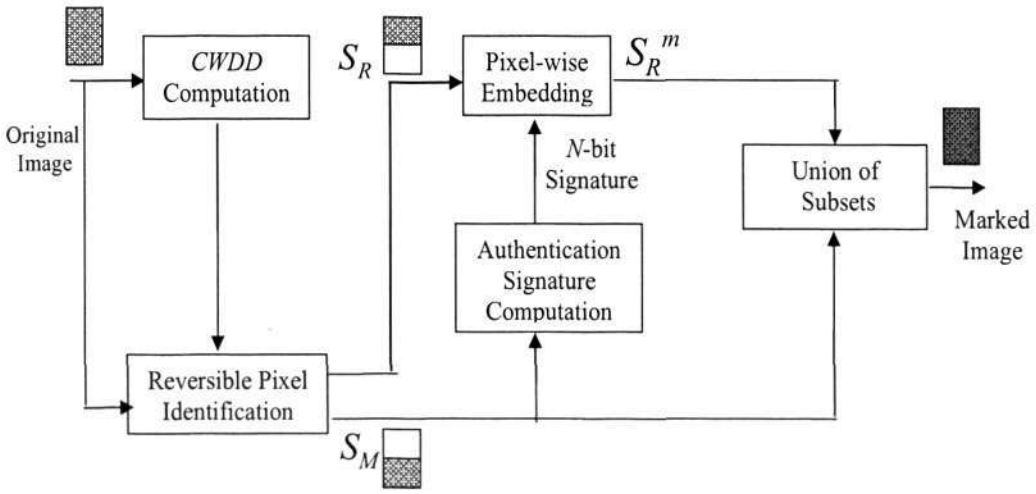
| Test image | Size | No. of pixels flipped | Mean opinion score | | |
|---|---|---|---|---|---|
| | | | *Princeton* | *DRDM* | *CWDD* |
| 1 | 368×386 | 350 | 0 | 0.1 | 0.6 |
| 2 | 430×495 | 500 | 0 | 0.7 | 0.6 |
| 3 | 337×519 | 750 | 0.7 | 1.0 | 0.8 |
| 4 | 360×508 | 850 | 1.1 | 1.6 | 1.0 |
| 5 | 463×535 | 950 | 0.5 | 1.0 | 1.2 |

## 2.3 Proposed Authentication Watermarking Algorithm

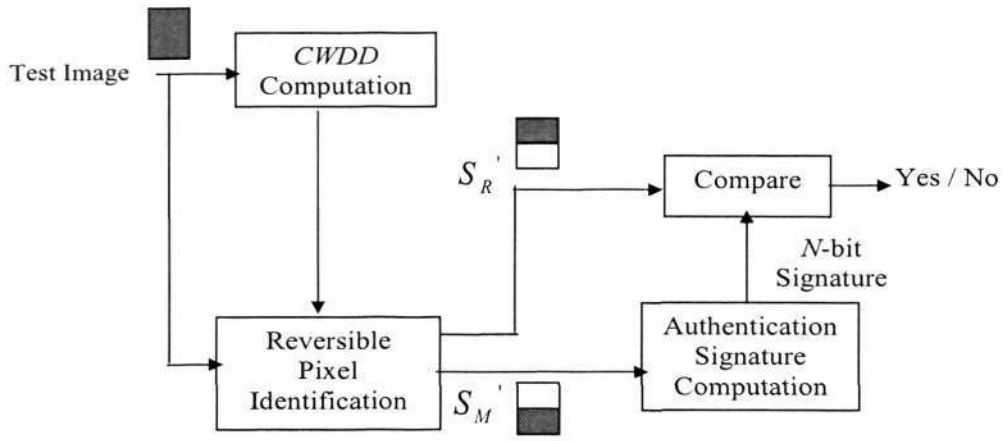
In this section, we propose a new algorithm for authentication watermarking in binary document images. According to Equation 2.8, the computation of *CWDD* measure takes the original and watermarked contour segments into account. The *CWDD* measure of a contour pixel remains the same before and after its flipping. This is because it is computed by considering the change occurred during the flipping process. This reversible property is particularly useful in identifying an ordered set of pixels in the original image. During watermarking each such pixel can carry one bit of the authentication signature computed from the remaining pixels in the image. For blind detection, these pixels should be detected in the same order before and after the watermarking process. However, direct application of the reversible property of the distortion measure is not sufficient for this purpose. In this new approach, necessary conditions for the correct detection of the ordered set of pixels are designed. We define each such pixel as the reversible pixel in an image. If a set of reversible pixels are found, the original image can be divided into two parts as necessary for correctly embedding the signature. The following steps explain the proposed authentication watermarking algorithm in binary images. The block diagrams of the proposed algorithm for embedding and detection are shown in Figure 2.10.

### 2.3.1 Conditions for Selecting the Reversible Pixels

To embed a $N$-bit signature within the original image, $N$ numbers of reversible pixels are searched in a sequential scanning order starting from left to right and then from




(a)




(b)

Figure 2.10: Block diagram of the proposed authentication watermarking algorithm (a) embedding process, (b) blind detection process.

top to bottom of the original image. Since the *CWDD* measure is defined for contour pixels in an image, only the contour pixels (both black and white) are examined for finding the reversible pixels.

**Definition:** A contour pixel in an image having the *CWDD* measure below a chosen threshold $T$ is defined as a suitable pixel. A pixel in an image is defined as a non-suitable pixel if it does not satisfy this criterion.

At the start of the sequential order search, each pixel in the original image is defined either as a suitable or non-suitable pixel. Among the suitable pixels, a sequential order search is performed until $N$ reversible pixels are found in the original image. The suitable pixels after flipping bring less visible distortion to the watermarked image and so only these pixels are examined for watermarking. In the following conditions, a suitable pixel is further categorized into either a reversible, pseudo-reversible or non-reversible pixel. A suitable pixel is defined as a reversible pixel, if it satisfies both Conditions $A$ and $B$. A suitable pixel is defined as a pseudo-reversible pixel, if it satisfies Condition $A$ but does not satisfy Condition $B$. If a suitable pixel does not satisfy Condition $A$, it is defined as a non-reversible pixel and it is not necessary to verify Condition $B$ for this pixel. The conditions are designed to ensure that after flipping the current suitable pixel, a reversible pixel should not be detected as a pixel which is not reversible and vice-versa at the blind detector.

Condition $A$. In an $M \times M$ pixel window centered on the current suitable pixel, there should not be any reversible or pseudo-reversible pixel already found in the original image.

Condition *B*. After flipping the current suitable pixel, in the 5×5 pixel neighborhood centered on it there should not be any suitable pixel which comes before in the scanning order and also satisfies the Condition *A*.

For the pixel coming first in the scanning order, the test condition is designed to be different. This is because, there is no other pixel coming before it in the scanning order. In this case, if the first pixel is a suitable pixel then it is declared as a reversible pixel. From second pixel onwards, the test is carried out using condition *A* and *B* as described above.

**Analysis:**

Condition *A* is necessary due to the following reasons. The flipping of the current suitable pixel may cause a change in the status of the already found reversible pixel. The pseudo-reversible pixel already found in its neighborhood may become a reversible pixel after the flipping process. Thus a change in the status of already found reversible and pseudo-reversible pixels may lead to wrong blind detection. The *CWDD* measure is computed using a 5-pixel long original contour segment and the original contour segment is centered on the current suitable pixel, i.e. 2 pixels are before and after it in a sequence. The flipping of the current suitable pixel could affect or change the *CWDD* measure of the pixels in a 5×5 pixel window centered on it. The *M* value should be chosen such that within the 5×5 pixel window centered on any pixel, there should not be more than one reversible pixel. This is because simultaneous flipping of multiple reversible pixels may convert a pixel (which is not reversible) into a reversible one. If the current suitable pixel is within the 5×5 pixel neighborhood of a pixel which in turn is in the 5×5 pixel neighborhood of a pseudo-

reversible pixel, its (current suitable pixel) flipping may cause a change in the status of the pseudo-reversible pixel. If $M$ is chosen to be equal or greater than 11, then the above possibilities of wrong detection are avoided.

Condition $B$ is necessary due to the following reason. If after flipping, any suitable pixel is generated among the neighbor pixels coming before in the scanning order and satisfy Condition $A$, it could become a (false) reversible pixel during detection. To verify this condition, the *CWDD* measure for the pixels in a 5×5 pixel neighborhood is computed after flipping the center pixel. However, any suitable pixel generated subsequently in the scanning order does not cause any error because of Condition $A$. The Condition $B$ is shown as an example in Figure 2.11.

### 2.3.2 Embedding

1. After $N$ reversible pixels are found, all pixels in the original image are divided into two disjoint subsets. The reversible pixels form the reversible subset $S_R$ and remaining pixels in the original image belong to the message subset $S_M$.
2. The authentication signature $A_S$ is computed from the pixels in $S_M$ using the key and embedded into the pixels of the reversible subset. The authentication signature to be used in this algorithm can be the hashed message authentication code (HMAC) using the secret key or the digital signature using the private / public key.
    (a) HMAC is found by computing the one way hash function of the data string that is a concatenation of the pixels belonging to the message subset $S_M$ and the secret key.

| | | | | |
|---|---|---|---|---|
| **3088** | **3089** | **3090** | **3091** | **3092** |
| **3528** | **3529** | **3530** | **3531** | **3532** |
| **3968** | **3969** | *3970* | 3971 | 3972 |
| 4408 | 4409 | 4410 | 4411 | 4412 |
| 4848 | 4849 | 4850 | 4851 | 4852 |

Figure 2.11: Condition $B$ is illustrated as an example for the current suitable pixel (the center pixel) at $10^{th}$ row and column in the original image of size 320×440 pixels. The scanning order of the center pixel and its 5×5 pixel neighborhood in the image are shown. After flipping, pixels in bold case are checked to detect whether any suitable pixel is generated which can satisfy the Condition $A$.

(b) For a digital signature, public key encryption and decryption technique is used. The digital signature is computed from the pixels in $S_M$ as follows. Let $H$ be a cryptographic hash function and we compute the hash

$$Q = H(S_M) \tag{2.12}$$

then $Q$ is encrypted with the encryption (private) key to generate the digital signature

$$A_S = E_{K_1}(Q) \tag{2.13}$$

where $E$ is the encryption function and $K_1$ is the private key.

3. Embedding is performed pixel-wise; so each reversible pixel in $S_R$ holds one bit of the authentication signature and the reversible pixel value is set equal to the signature bit it holds.
4. Set union operation of the embedded reversible subset $S_R^m$ and the message subset $S_M$ generates the watermarked image.

### 2.3.3 Detection

1. Similar to the embedding process, $N$ numbers of reversible pixels are searched in the test image at the blind detector in sequential scanning order and all pixels in this image are divided into two disjoint subsets. The reversible pixels form the reversible subset $S_R'$ and remaining pixels in the test image belong to the message subset $S_M'$.
2. The $N$-bit authentication signature is computed from the pixels in $S_M'$ using the key and compared with the extracted signature from $S_R'$.
   (a) If HMAC is used, then it is found by computing the one way hash function of the data string that is a concatenation of the pixels belonging to the message subset $S_M'$ and the secret key. If each bit of the computed HMAC matches with the corresponding reversible pixel value, then the image under question is authentic. Otherwise this image has been tampered after the watermarking process.
   (b) For the digital signature, the $N$-bit signature $A_S'$ is extracted from the pixels in reversible pixel subset $S_R'$. The signature is decrypted using the public key decryption algorithm $D$. The public key $K_2$ corresponding to the private key $K_1$ is used in the decryption process

$$P = D_{K_2}(A_S'). \tag{2.14}$$

The hash $Q_1$ is computed from $S_M'$ by the cryptographic hash function $H$ used in the embedding process

$$Q_1 = H(S_M'). \tag{2.15}$$

3. If $Q_1 = P$, then the image under question is authentic. Otherwise this image has been tampered after the watermarking process.

## 2.4 Results and Discussions

In this section, we present the simulation results by implementing the authentication watermarking algorithm proposed in the previous section. For demonstrating the effectiveness of the algorithm, HMAC and digital signature are computed as the authentication signature. In our method, security against any modification is obtained by using the cryptographic hash function. In the implementation, we have used MD5 [79] hash function and the DSA algorithm [80] for computing the authentication signature. The original image of size 320×440 pixels in Figure 2.6(a) is used to demonstrate the effectiveness of the algorithm against various modifications. The parameter $M$ is chosen to be 19 for keeping the reversible pixels separated by a distance and to satisfy condition $A$ for correct blind detection. The choice of higher $M$ value reduces visual interference among the reversible (flipping) pixels being separated by a larger distance. Thus the visual quality of the watermarked image is less affected. However, the number of available reversible pixels is reduced with the increase in $M$ value. The threshold parameter $T$ for choosing the suitable pixel by the $CWDD$ measure is 0.7. The choice of low value of parameter $T$ brings less visual distortion in the watermarked image. The user defined parameter $T$ can be suitably changed depending on the availability of reversible pixels in the original image. In our simulation, a maximum of 338 reversible pixels are found in the original image using the chosen parameters. If the value of $M$ is chosen to be 11 (the minimum value) instead of 19 with the same value of $T$, the maximum number of reversible pixels is found to be 623.

In the first case for embedding the 128-bit HMAC, a total of 128 reversible pixels are searched in the original image by following the sequential scanning order. Figure 2.12 shows the original and watermarked image after embedding the 128-bit HMAC. The watermarked image is visually similar to the original image. In order to prove correct blind detection, a total of 128 reversible pixels are searched in the watermarked image by following the sequential scanning order and using the same value of the parameters $M$ and $T$. In Figure 2.13, 128 reversible pixel positions in the original and watermarked image are shown to be identical. Since the position map of the reversible pixels in both the images is identical, correct blind detection is possible after the watermarking process. We perform multiple modifications such as deletion, insertion and substitution of characters in the watermarked image; (1) the only word *'information.'* in last line is deleted, (2) the word 'theory' is inserted into the last line, and (3) the word 'for' in line-5 is substituted by the word 'to'. The resulting attacked image is shown in Figure 2.14 (a). At the detector side the attacked image fails in the authentication test. In Figure 2.14(b), differences between the HMAC computed from the message subset and the reversible subset illustrate the failure of the attacked image to pass the authenticity test. In the second case, a 320-bit digital signature is generated using the digital signature algorithm (DSA). A total of 320 reversible pixels are searched in the original image by following the sequential scanning order. In Figure 2.15, the watermarked image is shown after embedding the digital signature. In Figure 2.16, 320 reversible pixel positions in the original and watermarked image are shown to be identical. Since the position map of the reversible pixels in both the images is identical, correct blind detection is possible. We perform multiple modifications in the watermarked image similar to the first case and the resulting attacked image fails in the authentication test. To test the effectiveness of proposed

method further, a total of 15 test images containing text, formulae, drawing and tables are generated. Table 2.4 shows the size of each test image and the number of reversible pixels found within each image. The value of $T$ and $M$ are chosen to be 0.7 and 11 respectively for finding the reversible pixels. From the table, it is evident that due to the availability of significant hiding space either a HMAC or a digital signature can be embedded for exact authentication purpose.

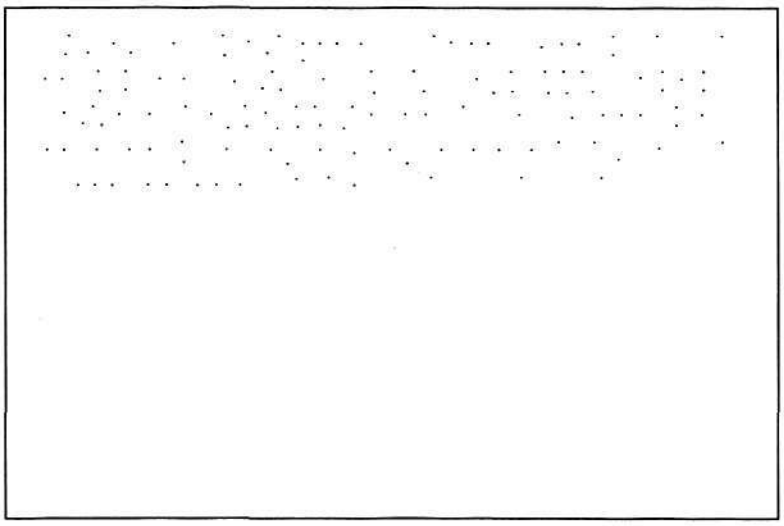
The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist and Hartley on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

(a)

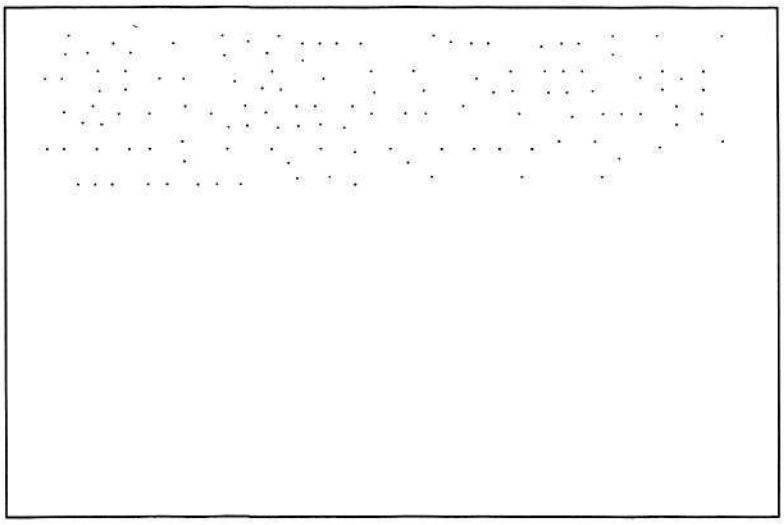
The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist and Hartley on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

(b)

Figure 2.12: (a) Original image of size 320×440 pixels; (b) watermarked image after embedding the 128-bit HMAC in the original image.



(a)

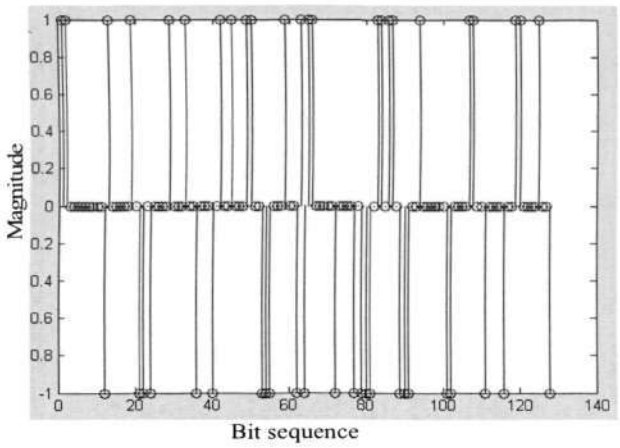


(b)

Figure 2.13: Position map of 128 reversible pixels, (a) original image (b) watermarked image.

The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis to such a theory is contained in the important papers of Nyquist and Hartley on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the theory

(a)




(b)

Figure 2.14: (a) Attacked image; (b) difference between the HMAC and the reversible subset of the attacked image.

The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist and Hartley on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

Figure 2.15: Watermarked image after embedding the 320-bit signature in the original image.

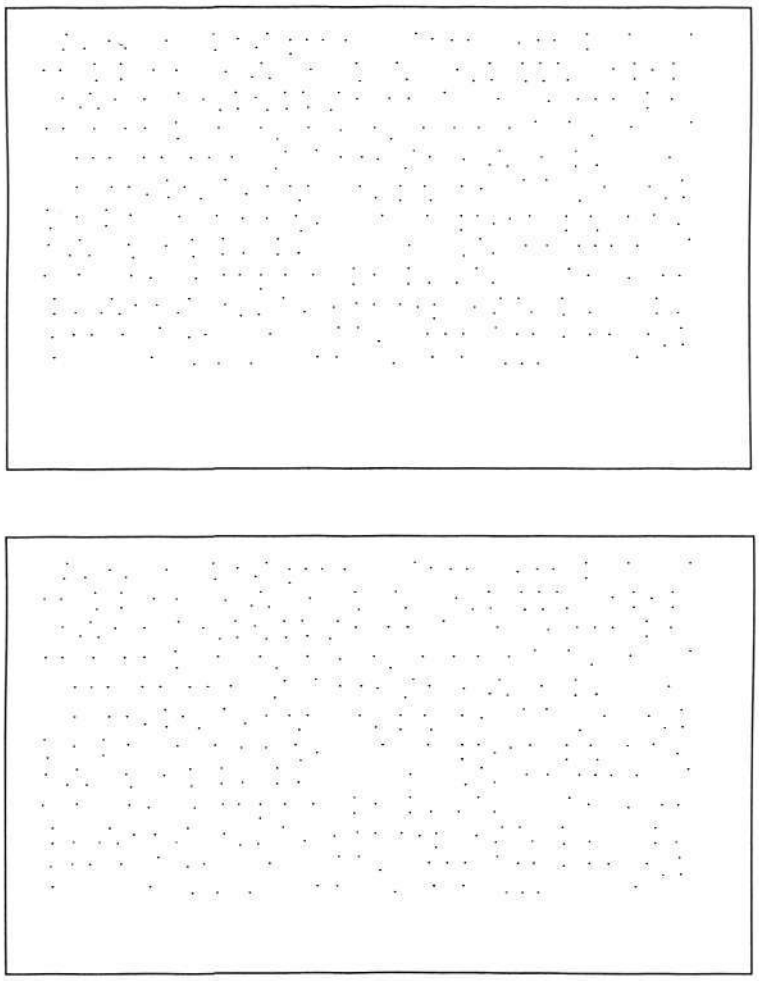


Figure 2.16: Position map of 320 reversible pixels, top: original image, bottom: watermarked image.

Table 2.4: Test image statistics

| Image number | Size | No. of reversible pixels |
|---|---|---|
| 1 | 463×535 | 1107 |
| 2 | 436×519 | 1020 |
| 3 | 442×508 | 978 |
| 4 | 461×508 | 776 |
| 5 | 513×542 | 662 |
| 6 | 495×547 | 611 |
| 7 | 425×533 | 869 |
| 8 | 548×796 | 1084 |
| 9 | 455×457 | 510 |
| 10 | 590×490 | 724 |
| 11 | 480×461 | 514 |
| 12 | 561×460 | 632 |
| 13 | 620×490 | 525 |
| 14 | 620×510 | 742 |
| 15 | 368×690 | 1099 |

The performance of the proposed authentication watermarking algorithm can be compared with previous methods with respect to their security level against content modifications. Kim *et al* proposed a method to detect any modification in binary images by block-wise embedding of the cryptographic signature [72]. As discussed earlier, this is vulnerable to parity attack but the visual quality is not degraded after watermarking. In [73, 74], the authentication method was secure against tampering when using the non-interlaced blocks, but contained a possibility of false tamper detection for interlaced blocks. In [71], the method was not vulnerable to parity attack due to the pixel-wise embedding of the cryptographic signature. However the visual quality of the watermarked image becomes degraded because relevant perceptual modeling was not performed. The security level of the new approach can be related to a secure cryptographic element such as a hash function. The probability of any undetected modification in the watermarked image is only $2^{-n}$ where $n$ is the length of the authentication signature being used. If the attacker wants to modify the message

subset such that the authentication signature remains the same, the chances of obtaining such a collision are removed by using a secure cryptographic key of length 128 bits or more. Furthermore, if the attacker alters any signature bearing reversible pixel, then the computed signature from the message subset will not match with the pixels in the reversible subset. It is not possible for an attacker to change the positions of the reversible pixels in an image without modifying the pixels belonging to the message subset. This is because a pixel in the reversible subset does not loose its status after flipping. So any such hostile attempt by the attacker can be successfully detected. Use of the *CWDD* measure ensures that the watermarked image remains visually similar to the original image. The possibility of parity attack is not present here because each bit of authentication signature is carried by a reversible pixel instead of a block. Thus, the new approach for suitable perceptual modeling and its application to pixel-wise embedding of the authentication signature achieves good visual quality and high security. To summarize, the ability of the proposed authentication algorithm for detecting any type of content modification in the watermarked image is equivalent to the security of cryptographic authentication without being susceptible to any kind of attacks.

## 2.5 Summary

In this chapter, we proposed a new approach for authentication watermarking in binary document images. The proposed algorithm can detect various modifications to the watermarked image with probability equivalent to that of the cryptographic authentication. For this purpose, an ordered set of low-distortion reversible pixels are selected for pixel-wise embedding of the authentication signature. The selection of the reversible pixels was performed by designing the necessary conditions and using the

reversible property of *CWDD* measure. The proposed algorithm did not suffer from parity attack like the block-wise hiding methods in binary images. The application of the proposed algorithm for binary images can be used in secure Fax transmission. After a transmission is performed, the sending Fax machine embeds the watermark using its own secret key. The receiver Fax machine can verify the received document whether it has not been modified after the transmission. Another application could be the legal usage of binary documents. If the legal documents are stored in a database, the user can verify their authenticity by using the appropriate secret or public key.

# Chapter 3

# Localization and Restoration in Binary Document Image Authentication Using Erasable Watermarks

## 3.1 Introduction

Many authentication watermarking methods can detect which portions of an image have been tampered after the embedding process. Using these methods it is possible to verify the regions of the image which are not tampered by the attacker. This capability of a watermarking method to localize tampering within a region is known as *localization*. If it is possible to localize tampering in an altered image, then the motive behind the tampering and possible attackers could be known. In the literature, there are two approaches to localization; sample-wise authentication and block-wise authentication. A fragile watermarking scheme for sample-wise authentication in grayscale and color images was proposed by Yeung and Mintzer in [66]. In this method, a binary function which could map each gray level of an image to either 0 or 1 was generated. The mapping function used to encode the binary logo was created by a pseudo-random number (PN) generator. To encode one bit in each pixel, the pixel value was changed to the closest match from the mapping table. The pixels were modified in a sequential manner and error diffusion technique was used to reduce visual artifacts. It was possible to determine which pixels have been modified during detection; thus the localization accuracy was at the level of a pixel. However, the scheme was vulnerable to the multiple stego-image attack if the same logo and key were used for multiple images. In [81], Fridrich *et al* proposed to replace the mapping

tables with an asymmetric encryption scheme to improve the security. This modification reduced the localization capability and the computational complexity was increased. It has been analyzed that the sample-wise authentication technique is not secure enough to detect every possible modification to the watermarked image with high probability [82, 83, 84].

One of the first block-wise localization methods was proposed by Wong in [70]. In this scheme, an image was divided into non-overlapping blocks and watermarking was performed for each block independently. The seven Most Significant Bits (MSBs) of all pixels in a block were hashed using a secure key-dependent hash function. The hash was then XORed with a chosen binary logo and inserted into the Least Significant Bits (LSB) of the same block. The watermark verification process started in the reverse order by calculating the key-dependent hash of the seven most significant bits in each block and XOR operation was performed with the LSB. Comparing the output with the used logo, the tampered blocks could be found. A public key version of this localization method was suggested in [85]. Though the block-wise authentication method could detect and localize tampering with high probability and accuracy, different attacks have been proposed in literature which exploits its block-wise independence. One weakness of this method is that it is possible to swap the blocks in an image without causing any detectable change. This problem could be avoided by including the block index while computing the signature [86]. However, there will still be a problem with swapping identically positioned blocks from a database of authentic images. This attack known as the Holliman-Memon attack has been described in [65]. More discussion about this attack and its countermeasures will be given in Chapter 4.

After localization, we are interested to know whether the modified portions of data in an attacked image could be restored. In literature, there are two types of restoration strategies: exact restoration and approximate restoration. In exact restoration, the original copy of the data can be restored without any bit error after modifications. In [87, 88], a Reed-Solomon Error-Correction code (ECC) was used to generate parity bytes for each row and column of an image. The generated parity bytes were embedded as a watermark in the two least significant bit planes of the image. If there were some changes in the watermarked image and within the error correction capability, the changes could be corrected by ECC decoding to restore the original image data. Even if the errors could not be corrected, it was possible to detect and localize them. In approximate restoration, the original copy of the data cannot be restored. However, an approximate but valuable version of the data is restored to provide information about the original data. In [89], a self-embedding technique was used for embedding a highly compressed version of the image into itself. In this method, the original image was compressed at JPEG 50% quality factor to generate the low-resolution image. For a particular block, the resulting binary sequence was inserted into the LSB plane of a block which is at a minimum distance away and in a randomly chosen direction. A minimum distance of 3/10 of the image size was used in this method. After tampering, it was possible to recover the low-resolution versions of the modified regions of the image. In [90], Kundur and Hatzinakos proposed a tell-tale watermarking method to estimate what kind of distortions had been applied to the image. This concept was extended for restoration purpose in [91]. This method assumed that the distortions to be modeled as linear blurring and then it could be inverted to restore the original image.

## 3.2 Localization in Binary Document Image Authentication

A method for localization for binary document images has been reported by Kim and Queiroz [72]. In this alteration locating method, the original image was subdivided into many sub-images and each sub-image was watermarked independently. A two-layer watermark was embedded imperceptibly using a block-wise data hiding technique to verify the integrity of watermarked image and localizing any modification in it. The disadvantage of the method was that the size of each sub-image was 128×128 pixels; so its localization accuracy was found to be low. The block-wise embedding technique used in this method also suffers from the parity attack. As discussed in the previous chapter, the parity attack arises because the signature is embedded by considering the parity of the blocks. If two pixels that belong to the same block change their values, the parity of this block may not change and so this modification will pass undetected. Recently, a new localization method has been proposed using a connectivity-preserving transition criterion [92]. The image is partitioned into multiple macro-blocks and an adaptive block identifier is embedded in selected macro-blocks for tamper localization. In this method, the localization accuracy is improved to 33×33 pixels block size. However, a possibility of false tamper detection and incorrect localization exists after various attacks. To overcome the shortcomings, we propose a new authentication watermarking method that is feasible and effective for localization in binary document images in electronic form. The proposed method achieves secure localization by using erasable watermarks in binary document image authentication which has not been reported so far in literature.

### 3.2.1 Reasons for Using Erasable Watermarks

The block-wise authentication technique proposed by Wong for grayscale images cannot be directly applied for binary images due to different pixel statistics. For embedding the authentication signature in each block imperceptibly, two basic requirements should be fulfilled. First, the number of low-distortion pixels in a block to embed the authentication signature should be high. Second, the watermark detection process should be blind. However, it is difficult to satisfy these requirements while watermarking each block in a binary image. Within a reasonable block-size there is insufficient number of low-distortion pixels available for embedding. The blind detection requirement of these pixels adds to the difficulty in achieving high watermark capacity in each block. Furthermore, an imperceptible watermark cannot be embedded in white regions of the image. The inability to watermark in the white regions makes the detector vulnerable to malicious tampering. For example, the attacker can tamper the targeted watermarked portions of an image to be white so that the detector would not be able to detect it. Due to these shortcomings, it is evident that unless the block size is large, imperceptible watermarking may not be suitable to embed the authentication signature.

We turn our attention to the possibility of embedding the signature in other such pixels that brings visual distortion in watermarked image. However, the resulting distortion due to the embedding process can be erased entirely at the blind detector. After erasing the embedded watermark, the original image can be restored at the blind detector. This particular concept is known as erasable, invertible or reversible watermarking in the literature and the watermark thus embedded is termed as an erasable watermark. In some applications such as military, legal and medical imaging,

any distortion introduced by embedding an authentication signature might be unacceptable. A suitable way to deal with such scenarios is to restore the original image from the watermarked image after verification. This has led to an interest in using erasable watermarks for authentication so that the watermarks can be removed or erased to obtain the exact copy of the original image. The algorithms for designing erasable watermarking systems have been suggested in [93, 94, 95, 96]. The algorithm proposed by Fridrich *et al* [93] for exact authentication of natural images is of particular interest to this paper. The proposed algorithm can be summarized as follows: let $A$ represent the information that is altered in the cover work when we embed a message of $N$ bits. Fridrich *et al* have shown that the erasability is possible provided $A$ is compressible. If $A$ can be losslessly compressed to $M$ bits, $N$-$M$ additional bits can be erasably embedded in the cover work. In the implementation of this algorithm for natural images, it is observed that the neighboring pixels are highly correlated. This leads to correlations between neighboring bits within a bit plane. Thus some bit planes in the whole image can be sufficiently compressed to implement an erasable watermark.

For binary document images, each pixel is represented by one bit and it can be considered that there is only one bit plane in the image. If all pixels in the bit plane are losslessly compressed to construct the erasable watermark, then the compressed block does not have perceptual correlation with the original and hence the user is not able to know what relevant information is present in the watermarked image. Therefore, creating an erasable watermark by directly compressing the bit plane is not relevant in document images. If a set of suitable pixels within a block with high

correlation can be found, they can be losslessly compressed and an erasable watermark can be constructed for binary document images.

### 3.2.2 Constructing an Erasable Watermark

To construct an erasable watermark, we find a set of pixels in each block of the binary image such that; (1) there exists a high correlation among the pixels, (2) the same set of pixels can be found at blind detector and (3) the relevant information is preserved after the embedding process so that the user can determine whether the particular watermarked image is useful for his/her purpose. We analyze pixels in the binary document image based on their 8-neighborhood. As shown in Figure 3.1, there can be six categories of pixel neighborhood in such images. The center pixel can be either black or white with its eight neighbor pixels possibly all black, or all white, or mixture of black and white. Pixels whose neighborhoods have both white and black pixels are contour pixels like in Figure 3.1 (c) and (f) and they convey important visual information in the document image. The center pixel in Figure 3.1 (b) is called a *foreground* pixel and the center white pixel in (e) can represent a hole, so these pixels convey some information. The black pixel whose all neighbor pixels are white is termed as an *isolated* pixel like in Figure 3.1 (a). These pixels are perceived as noise in a binary image. As shown in Figure 3.1 (d), a white pixel whose all neighbor pixels are white is termed as a *background* pixel. Among these pixel categories, we choose the isolated and background pixels for embedding an erasable watermark due to following reasons: Isolated and background pixels do not convey important information within document images; If these pixels are altered, a background noise will be formed in the image which is similar to the salt-and-pepper noise found in

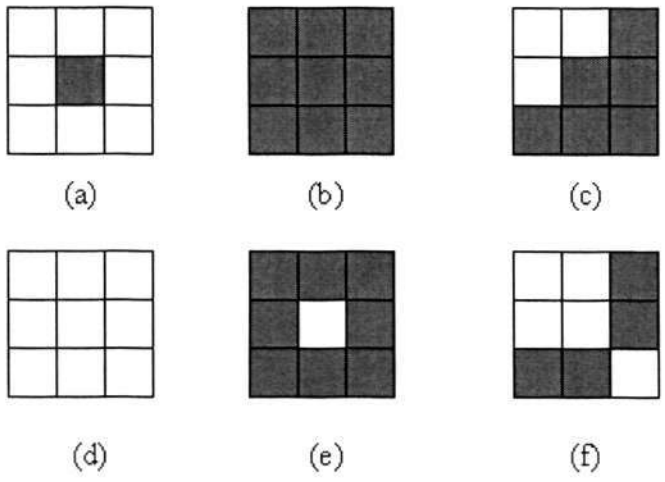



Figure 3.1: Different categories of pixels in a binary document image based on their 8-neighborhood.

natural images. In document images, we obtain information by recognizing various patterns such as characters, symbols, lines and curves. It is known that human vision has remarkable ability to recognize such patterns even in the presence of noise. So after embedding an erasable watermark in these pixels, the user can still obtain relevant information about the document. The background pixels occur in long sequences and isolated pixels occur in between them with less probability. Such a set of pixels can be significantly compressed using the run-length coding scheme [97]. Flipping of a background pixel creates an isolated pixel and vice-versa; so blind detection of the embedded pixels is possible.

To construct the erasable watermark in each block, an ordered set of *insignificant* pixels are searched in a sequential scanning order starting from left to right and from top to bottom. Pixels which are in the border with other blocks are not included in this search to maintain block independence. A pixel within a block is defined as an insignificant pixel, if it satisfies the following conditions 1, 2 and 3. A pixel is defined

as a pseudo-insignificant pixel, if it satisfies conditions 1 and 2 but does not satisfy condition 3. The conditions are designed to ensure that after flipping the current pixel, an insignificant pixel should not be detected as a pixel which is not insignificant and vice-versa during blind detection. A pixel in a block is defined to be an insignificant pixel if,

Condition 1. The pixel is either a background pixel or an isolated pixel.

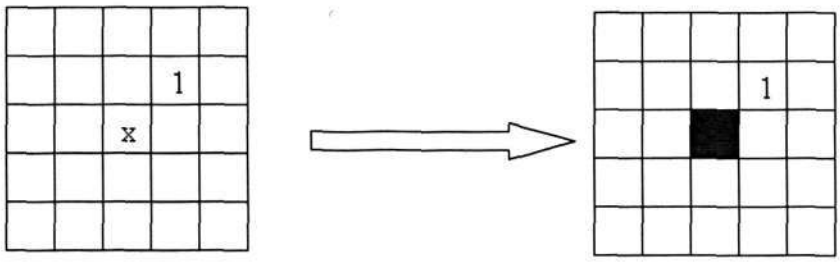
Condition 2. In an $M \times M$ pixel window, there should not be any insignificant or pseudo-insignificant pixel already found in the block.

Condition 3. After flipping the current pixel, there should not be any pixel in its 8-pixel neighborhood which comes before in the scanning order and satisfies the above two conditions.

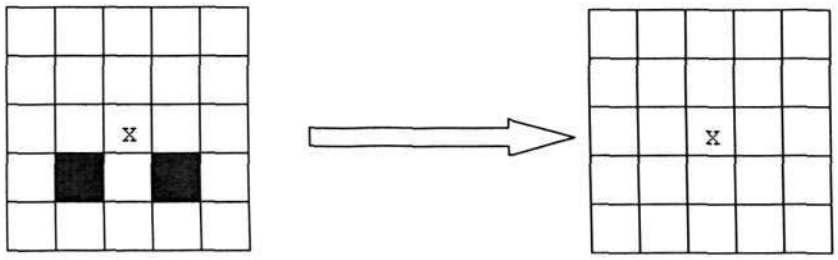
**Analysis:**

The reason for selecting the background and isolated pixels for constructing the erasable watermark has already been explained in this section. Condition 2 is necessary due to the following reasons: flipping of the current pixel may cause a change in the status of the already found insignificant pixel; the pseudo-insignificant pixel already found in its neighborhood may become an insignificant pixel after the flipping process. Thus a change in the status of already found insignificant and pseudo-insignificant pixels can lead to wrong blind detection; the flipping of the current pixel could change the status of its 8-neighborhood insignificant pixel. This possibility is shown in Figure 3.2 (a) as an example. Flipping the center pixel (marked by x) to black will affect the neighboring pixel (marked by 1). This pixel no longer satisfies the condition 1 after flipping the center pixel. If it has been decided as an

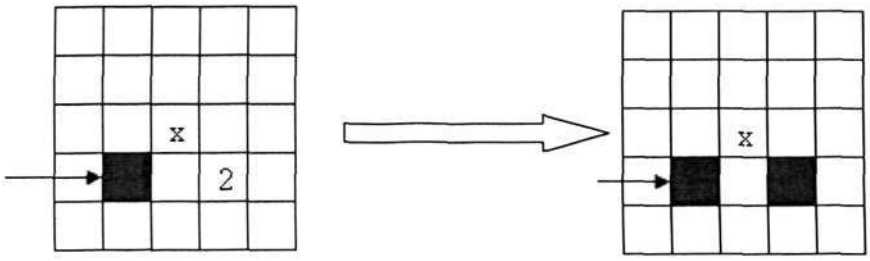
insignificant pixel for embedding, during blind detection its status will be changed. In a 3×3 pixel window centered on a pixel, there should not be more than one insignificant pixel. This is because flipping of multiple insignificant pixels may convert a pixel (which is not insignificant) into an insignificant one. This is shown as an example in Figure 3.2 (b). If both black insignificant pixels are flipped, then there is a possibility that the center pixel (marked x) will be detected as an (false) insignificant pixel. In Figure 3.2 (c), an example is shown where flipping of a white pixel could convert a pseudo-insignificant pixel into an insignificant pixel. When $M$ is equal to 3, the black pixel (arrow marked) satisfies conditions 1 and 2. However, it does not satisfy condition 3, because the pixel in its 8-neighborhood (marked with x) satisfies conditions 1 and 2 after its flipping. This black pixel is therefore detected as a pseudo-insignificant pixel and it is not considered for embedding. The white pixel (marked with 2) is detected as an insignificant pixel and it may be flipped to black for embedding. The black pixel, (arrow marked) which is actually a pseudo-insignificant pixel, would then be detected as an (false) insignificant pixel during blind detection. If $M$ is chosen to be equal or greater than 5, the above possibilities of wrong detection would be avoided. Condition 3 is necessary due to the following reason: If after flipping, any pixel is generated among the 8-neighborhood pixels coming before in the scanning order and satisfy conditions 1 and 2, it could become a (false) insignificant pixel during detection; however, any pixel generated subsequently in the scanning order does not cause any error because of condition 2.




(a)




(b)




(c)

Figure 3.2: Examples of pixel patterns to illustrate the wrong blind detection of insignificant pixels. (Left) patterns before flipping process, (right) – patterns after flipping process.

### 3.2.3 Erasable Watermark Embedding for Localization

We shall outline the proposed localization method in following steps. The block diagram of the proposed method for embedding is shown in Figure 3.3 (a).

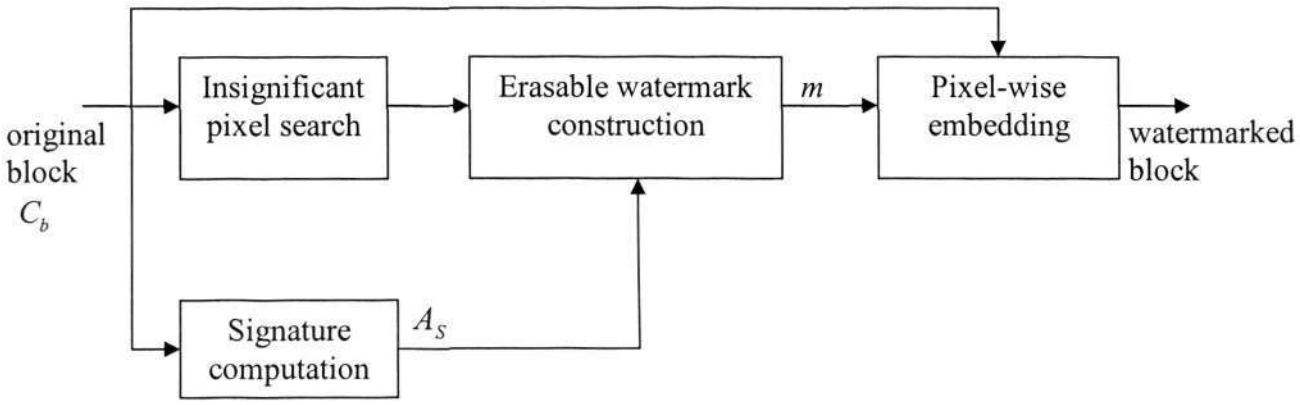



Figure 3.3 (a): Block diagram of the proposed localization method for embedding process.

1. The whole image is divided into non-overlapping blocks of $Y \times Z$ pixels. Watermarking is performed for each block independently and in a sequential order starting from left to right and from top to bottom of the image.
2. In each block, an ordered set of insignificant pixels are searched in a sequential scanning order as described by conditions 1, 2 and 3 in Section 3.2.2. The insignificant pixel set is losslessly compressed using the run-length coding scheme. Let the compressed data be denoted as $C_D$.
3. Authentication signature $A_S$ is computed from the block according to the following equation

$$A_S = H(C_b, K, I_b, I_K) \tag{3.1}$$

where, $H$, $C_b$, $K$, $I_b$ and $I_K$ denote hash function, current block in the original image, secret key, block index and image index, respectively.

The block index is used in the computation of signature to resist block-swapping by a hostile attacker and the image index is necessary to resist the Holliman-Memon attack [86].

4. The compressed data and authentication signature are concatenated to create the message '*m*', which is embedded in the insignificant pixel set producing the watermarked block. The embedding is performed pixels-wise; so an insignificant pixel holds one bit of *m* and its pixel value is set equal to the signature bit it holds. Likewise all blocks in the image are watermarked.

### 3.2.4 Erasable Watermark Detection for Localization

The block diagram of the proposed method for detection is shown in Figure 3.3 (b).

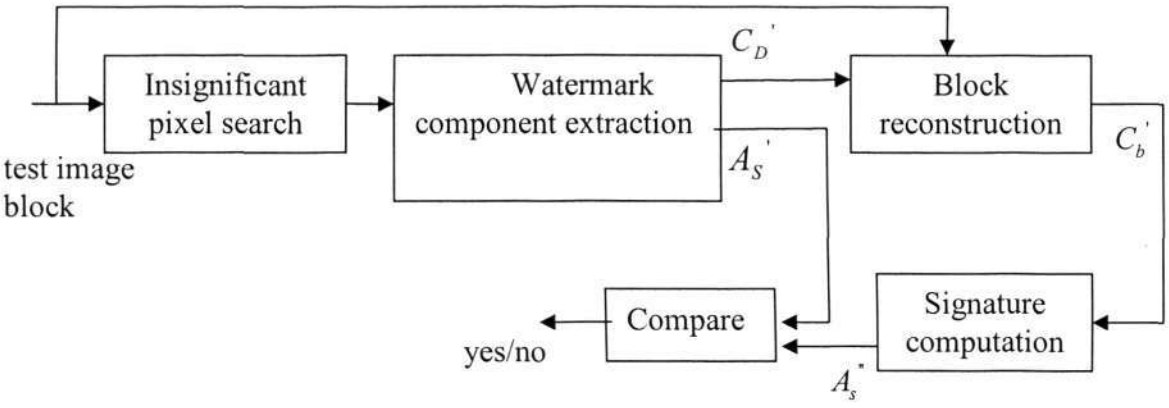



Figure 3.3 (b): Block diagram of the proposed localization method for blind detection process.

1. To verify each block in the test image, the message $m'$ is extracted by finding the insignificant pixel set like in the embedding process. Its component pieces, the compressed version of insignificant pixel set $C_D'$ and the authentication signature $A_S'$ are extracted. The compressed version of the

insignificant pixel set together with the current block is used to reconstruct the block $C_b^{'}$.

2. The authentication signature of the reconstructed block is computed as follows and compared with the extracted signature.

$$A_S^{*} = H(C_b^{'}, K, I_b, I_K) \tag{3.2}$$

where, $H$, $C_b^{'}$, $K$, $I_b$ and $I_K$ denote hash function, reconstructed block, secret key, block index and image index, respectively.

3. If the signatures $A_S^{'}$ and $A_S^{*}$ match, then the reconstructed block is authentic. Verification of each block is performed independently to localize any tampering in the watermarked image.

### 3.2.5 Results and Discussions

In this section, we present simulation results by constructing the erasable watermark for the proposed block-wise localization method. The authentication signature to be used in this algorithm is the Hashed Message Authentication Code (HMAC). The HMAC is found by computing the one way hash function of the data string that is a concatenation of the pixel set and secret key. In our method, high security against content modification is obtained by using the cryptographic hash function. We have implemented the message-digest (MD5) [79] hash function to compute the HMAC. The output 128-bit HMAC is used as the authentication signature and the message '$m$' is constructed for each block as described in the proposed method. The value of $M$ is chosen to be 5 for achieving high watermark capacity and correct watermark detection. The first ten bits of $m$ represent the size of the compressed data. While compressing the insignificant pixel set by run-length coding, a 10-bit representation is

used for the number of white pixels and a 1-bit representation for the number of black pixels. This is because the possibility of occurrence of isolated pixels is less as compared to the background pixels. The first ten bits giving the size information and the run-length encoded data represent the compressed data in $m$. We have chosen a block size of 40×40 pixels in our simulation and the block-size can be suitably modified if the length of the authentication signature is changed. The original and watermarked images are shown in Figure 3.4-3.6 after the pixel-wise embedding of $m$ in each block. The original images are padded with white pixels if necessary such that the image dimensions become multiples of the chosen block size. It is observed that even though some background noise is present in the watermarked images, the content in the documents can readily be read and understood by the user. In the watermarked images, it is observed that the background noise appears to be more random and different well-structured patterns can be recognized due to the inherent ability of human vision. For secure embedding, each block in the original image should have high watermark capacity. The capacity of a block is the number of bits that can be embedded within it. To analyze the performance of the proposed method in different images, we define the term *redundancy* ($R$) in Equation 3.3 as the number of bits available in a block to accommodate the signature,

$$R = \text{Size of the insignificant pixel set} - \text{Compressed data size.} \qquad (3.3)$$

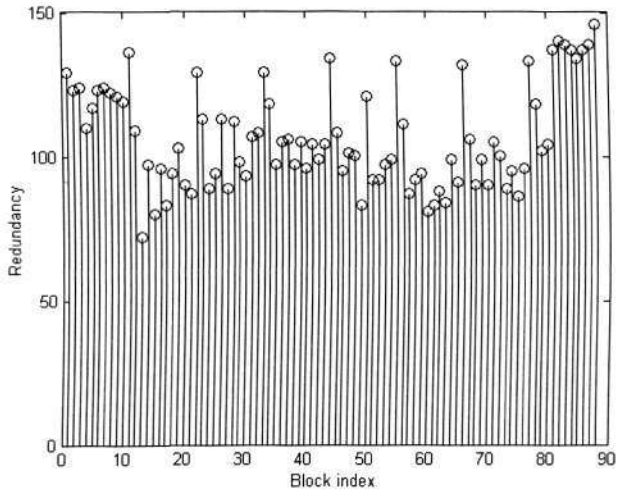
A similar definition of redundancy has been used for natural images in [93]. In Figure 3.4-3.6 (c), it is shown that most blocks in the images have high redundancy for embedding $m$. If $R$ in a block is less than 128 bits, e.g. 90 bits, then the first 90 bits of HMAC will be used to construct $m$ and authenticate the current block. Similarly at detector, the comparison between computed and extracted signature is performed only for the first 90 bits. With no tampering, all blocks in the watermarked images are

The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist and Hartley on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

(a)

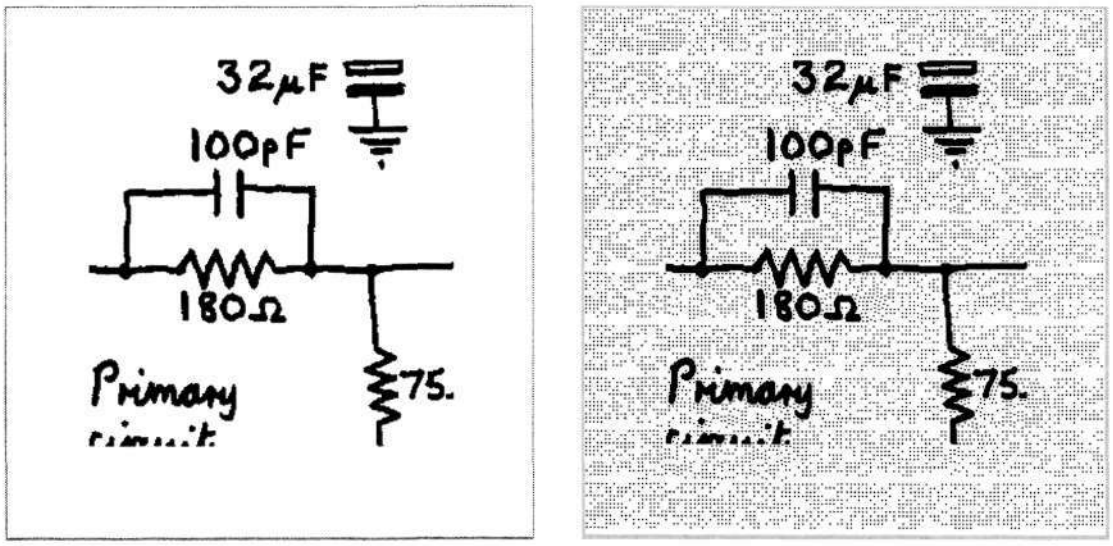
The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist and Hartley on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

(b)

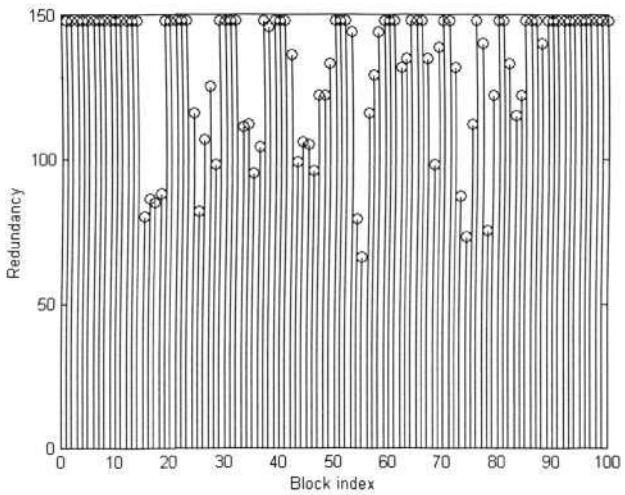



(c)

Figure 3.4: (a) Original text document image of size 320×440 pixels; (b) watermarked image after embedding the erasable watermark in each block; (c) redundancy ($R$) for each block of the original image.



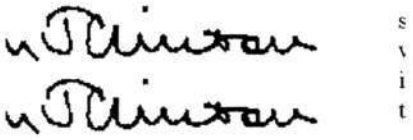

(a) (b)




(c)

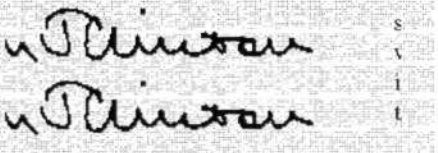
Figure 3.5: (a) Original drawing image of size 400×400 pixels; (b) watermarked image after embedding the erasable watermark in each block; (c) Redundancy ($R$) for each block of the original image.

| | before shuffle | mean afte analysis |
|---|---|---|
| $m_0/N$ ($0^{th}$ bin) | 20.37% | $5.16\times10^{-5}$% |
| $m_1/N$ ($1^{st}$ bin) | 1.85% | $7.77\times10^{-4}$% |
| $m_2/N$ ($2^{nd}$ bin) | 5.56% | $5.81\times10^{-3}$% |

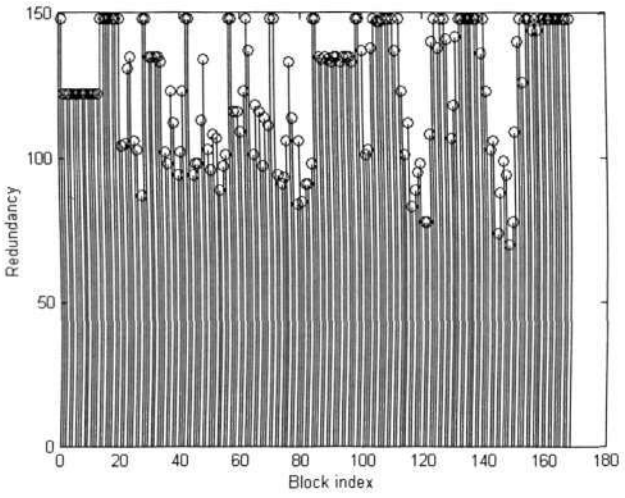


(a)


| | before shuffle | mean afte analysis |
|---|---|---|
| $m_0/N$ ($0^{th}$ bin) | 20.37% | $5.16\times10^{-5}$% |
| $m_1/N$ ($1^{st}$ bin) | 1.85% | $7.77\times10^{-4}$% |
| $m_2/N$ ($2^{nd}$ bin) | 5.56% | $5.81\times10^{-3}$% |

(b)




(c)

Figure 3.6: (a) Original image of size 480×560 pixels containing text and signature; (b) watermarked image after embedding the erasable watermark in each block; (c) Redundancy ($R$) for each block of the original image.

verified. After verification, the exact copy of original image can be restored at the blind detector. The watermark erasing process is shown in Figure 3.7 for a text document image. To illustrate the localization capability of the proposed method, we perform the following modifications in the watermarked image of Figure 3.5 (b). The characters 'Primary' at the left and bottom portion of the image are removed and the

attacked image is shown in Figure 3.8. The detection is performed on the attacked image and the detector correctly localizes the tampered blocks.

Although the localization accuracy and cryptographic security offered by the block-wise localization method is high, it is however vulnerable to a counterfeiting attack known as Holliman-Memon attack [65]. To counteract this attack, Wong and Memon suggested including a unique image index while computing the signature [86]. The use of a unique image index in Equations 3.1 and 3.2 for computing the signature removes the possibility of this attack entirely. Such an approach is feasible for some practical applications such as selling photographs in Internet; however it may not be always possible because managing such indices brings extra overhead to the user. We propose a new method in Chapter 4 to counteract this attack such that it is not necessary to manage the image index database.

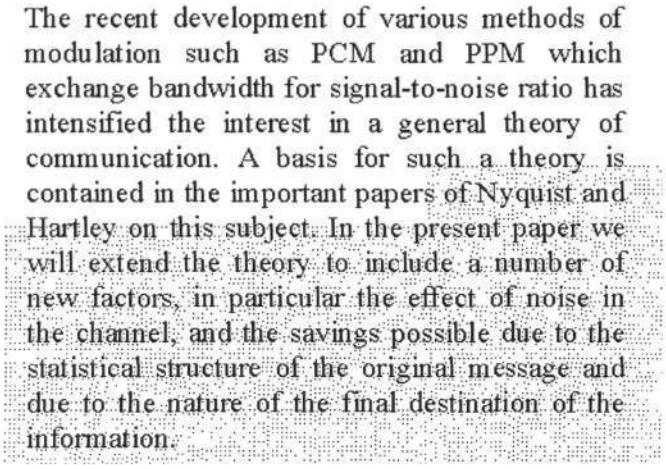
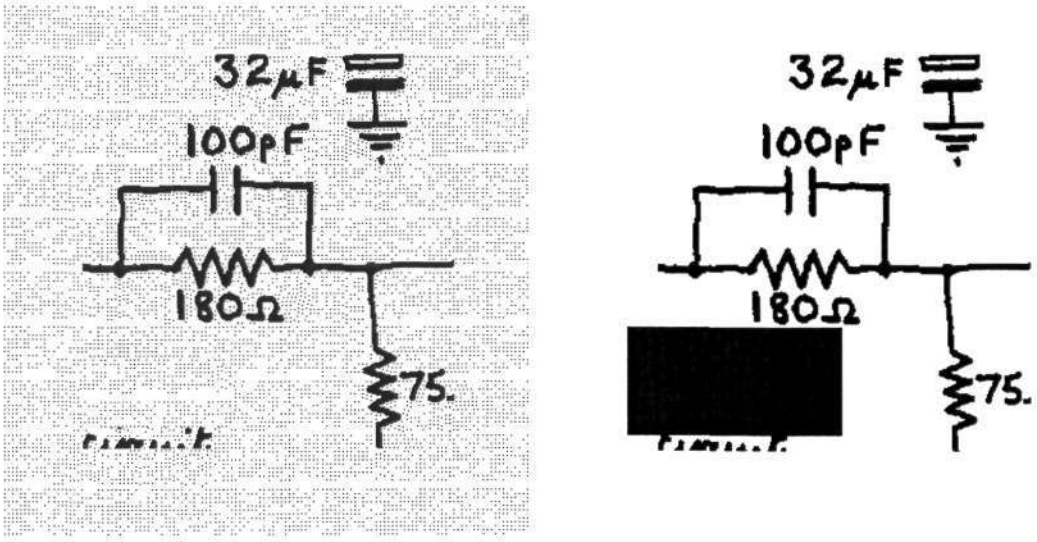



Figure 3.7: The watermark erasing process is shown in which 40 blocks out of 88 blocks have been restored after verification.




(a) (b)

Figure 3.8: (a) Attacked image; (b) image showing the authentic reconstructed blocks and a total of 8 tampered blocks are shown in dark region.

To test the effectiveness of our proposed method further, a total of 15 test images containing text, formulae, drawing and tables are generated. In Table 3.1-3.3, $R_{mean}$ for each test image is given for the block sizes of 36×36, 40×40 and 48×48 pixels where $R_{mean}$ is the mean of redundancy of all the blocks in a test image. The test image is padded with white pixels if necessary, such that the image dimension becomes multiples of the chosen block size. By taking the mean of the mean redundancy of all 15 test images, an approximate account of available watermark capacity in binary document images is obtained. The mean of the mean redundancy is found to be 90.22, 113.27 and 139.06 bits for 36×36, 40×40 and 48×48 pixels block size respectively. Thus it is shown that a hash output of 96 bits or 128 bits can be suitably embedded in blocks of various binary document images in the proposed method. The performance of the proposed algorithm can be compared with the previous methods. The localization accuracy in the previous method [72] was

approximately at the block size of 128×128 pixels. In the proposed method it has been significantly improved to approximately the block-size of 40×40 pixels. The block-wise embedding in the previous method suffers from parity attacks as discussed in Section 3.2. The possibility of parity attack is not present in the proposed method because each message bit is embedded in an insignificant pixel instead of a block. Due to the pixel-wise embedding of the cryptographic signature, there is no possibility of false tamper detection in the proposed method.

Table 3.1: Redundancy in test images with 36×36 pixels block size.

| Image number | Number of blocks | $R_{mean}$ (bits) |
|---|---|---|
| 1 | 195 | 86.11 |
| 2 | 195 | 87.16 |
| 3 | 195 | 88.34 |
| 4 | 195 | 92.25 |
| 5 | 240 | 101.57 |
| 6 | 224 | 100.60 |
| 7 | 180 | 88.21 |
| 8 | 368 | 97.82 |
| 9 | 169 | 85.54 |
| 10 | 238 | 86.45 |
| 11 | 182 | 86.02 |
| 12 | 208 | 84.16 |
| 13 | 238 | 92.06 |
| 14 | 238 | 92.54 |
| 15 | 220 | 84.51 |

Table 3.2: Redundancy in test images with 40×40 pixels block size.

| Image number | Number of blocks | $R_{mean}$ (bits) |
|---|---|---|
| 1 | 168 | 111.95 |
| 2 | 143 | 105.51 |
| 3 | 156 | 110.82 |
| 4 | 156 | 115.25 |
| 5 | 182 | 123.19 |
| 6 | 182 | 124.78 |
| 7 | 154 | 114.31 |
| 8 | 280 | 119.90 |
| 9 | 144 | 110.02 |
| 10 | 180 | 105.78 |
| 11 | 144 | 107.56 |
| 12 | 180 | 109.43 |
| 13 | 192 | 115.88 |
| 14 | 208 | 118.16 |
| 15 | 180 | 106.52 |

Table 3.3: Redundancy in test images

with 44×44 pixels block size.

| Image number | Number of blocks | $R_{mean}$ (bits) |
|---|---|---|
| 1 | 143 | 136.48 |
| 2 | 120 | 130.25 |
| 3 | 132 | 136.75 |
| 4 | 132 | 141.90 |
| 5 | 156 | 151.51 |
| 6 | 156 | 152.14 |
| 7 | 130 | 142.32 |
| 8 | 247 | 148.81 |
| 9 | 121 | 136.19 |
| 10 | 154 | 132.43 |
| 11 | 121 | 133.26 |
| 12 | 143 | 132.24 |
| 13 | 154 | 140.29 |
| 14 | 168 | 142.76 |
| 15 | 144 | 128.63 |

## 3.3 Restoration in Text Document Image Authentication

After localization of tampered regions, the next relevant question arises about the possibility of restoring the modified portions in a binary document image. In this section, a new method to address the issue of restoration is proposed. Previously described restoration methods cannot be applied to binary images due to limited data hiding capacity. The proposed method is particularly effective for text document images in electronic form. In text document images, several characters from a finite set convey the necessary information. The user could extract information regarding the document from the shape of the characters and their particular appearance such as size and font is less important. Each character of the English alphanumeric set can be represented by a 7-bit ASCII (American Standard Code for Information Interchange) code. The previous work for restoration for text document images has been reported by Makur [98]. In this method, self-embedding was used for restoration of the original

character sequence. The ASCII code of a character was used as its watermark and embedded imperceptibly in another character of the document. For watermark embedding, a particular character was selected through a random permutation or cyclic shift function. During watermark verification, each character was compared with its corresponding watermark to localize tampering in the document image and to restore the original character sequence. This method is effective for restoration against the alterations like character substitution, deletion and insertion. However, there exists a possibility of false tamper detection and restoration failure after only a few (even 1) individual alterations. For multiple alterations such as combined deletion and insertion of characters in the document, restoration is not possible due to a loss of synchronization. To overcome these shortcomings, we propose a new method for restoration of the original character sequence using erasable watermarks in conjunction with error correction coding and cryptography techniques. The proposed method belongs to the category of approximate restoration. In the proposed method, embedding an erasable watermark in each block of an image will introduce visible noise in watermarked images that can be made available for different users. In the embedding process, the relevant information contained in the text document is preserved so that the user can read or understand the documents. The user can localize any tampering after watermarking with high probability and accuracy. After localization, the original character sequence can be restored by using error correction coding technique. The watermark can then be erased from the authenticated images to retrieve the distortion-free original images for further analysis and application.

### 3.3.1 Finding Insignificant Pixels After Preprocessing

For restoration applications, a substantial amount of watermark capacity is required to embed necessary information bits. In the proposed restoration method, the embedded watermark is not designed to be robust against any class of attacks. The availability of high capacity through constructing an erasable watermark is illustrated in the results of the localization method. In our investigation, we found that it is necessary to further increase the available watermark capacity for achieving effective restoration capability. The procedure to finding insignificant pixels in a block is described in Section 3.2.2. For correct and blind detection, it was shown that the minimum value of $M$ should be chosen as 5. To increase the redundancy in a block, the value of $M$ should be decreased such that more number of insignificant pixels could be found. At the same time to ensure correct watermark detection, certain modifications are necessary in the watermarking process. In a text document image, the number of isolated pixels is significantly less than the number of background pixels. If the value of $M$ is chosen to be 3, then correct detection is not possible due to the following reasons: As shown in Figure 3.2 (b), there is a possibility of false insignificant pixel generation after flipping two black insignificant pixels; In Figure 3.2 (c), flipping of a white pixel (marked by 2) could convert a pseudo-insignificant pixel (marked by an arrow) into an insignificant pixel. To avoid such possibilities of wrong detection, we modify the procedure for finding the insignificant pixels in a block after performing a preprocessing step. In each block of the original image, if two isolated pixels are found within $3 \times 3$ pixel window centered on any pixel, one of them will be flipped to become a white pixel. Isolated pixels in a text document image do not carry significant information and the probability is low in finding two isolated pixels in a $3 \times 3$ pixel window. Flipping few isolated pixels does not have much impact on the

text document. After this preprocessing step, an ordered set of *insignificant* pixels are searched in a sequential scanning order starting from left to right and top to bottom. Pixels which are near the border with other blocks are not included in this search to maintain block independence. A pixel within a block of the preprocessed image is defined to be an insignificant pixel if the following three conditions are satisfied. A pixel is defined as a pseudo-insignificant pixel, if it satisfies Conditions 1 and 2 but does not satisfy Condition 3.

Condition 1. The pixel is either a background pixel or an isolated pixel.

Condition 2. In a 3×3 pixel window centered on the current pixel, there should not be any insignificant pixel and in a 5×5 pixel window there should not be any pseudo-insignificant pixel, already found in the block.

Condition 3. After flipping the current pixel, there should not be any pixel in its 8-pixel neighborhood which comes before in the scanning order and satisfies the above two conditions.

### 3.3.2 Erasable Watermark Embedding for Restoration

Before embedding, preprocessing is performed on the text document image as described in Section 3.3.1. The block diagrams of the proposed restoration method are shown in Figure 3.9. We outline the proposed restoration method in the following steps.

1. The original image is divided into non-overlapping blocks of 32×40 pixels. The size of the original image is chosen to be the multiple of individual block

size. Watermarking is performed for each block independently and in a sequential order starting from left to right and top to bottom of the image.

2. All characters in the original image are extracted in the sequential order and a binary sequence $A_c$ is obtained after each character is converted into the corresponding ASCII code. The character extraction process can be implemented by using optical character recognition (OCR) techniques [99].
3. Another binary sequence $E_s$ is obtained by encoding $A_c$ with an [*n, k*] Bose, Ray-Chaudhuri, Hocquenghem (BCH) encoder [100]. Before ECC encoding, zero-padding can be performed such that the number of bits in the sequence $A_c$ becomes an integral multiple of $k$. The parameters $n$ and $k$ are chosen such that the total number of bits in $E_s$ is less than or equal to $108 \times B$, where $B$ is the total number of blocks in the image. The number 108 is chosen here because a maximum of 108 bits in $E_s$ will be embedded in a single block, as described in the following steps.
4. After ECC encoding, zero-padding can be performed in the sequence $E_s$ to make the number of bits exactly equal to $108 \times B$. $E_s$ is then randomly permuted using the secret key $K$. Let the permuted sequence be denoted as $P_s$. The errors due to tampering in a localized region of the image often come as bursts. The random permutation distributes the burst type of error over the whole image; thus increasing the correction capability of ECC coding.
5. The sequence $P_s$ is divided into $B$ non-overlapping segments of size 108 bits each. Let each such segment be denoted by $S_c$. A maximum of 108 bits of a particular segment $S_c$ is used in the embedding process of the corresponding block in the sequential order.

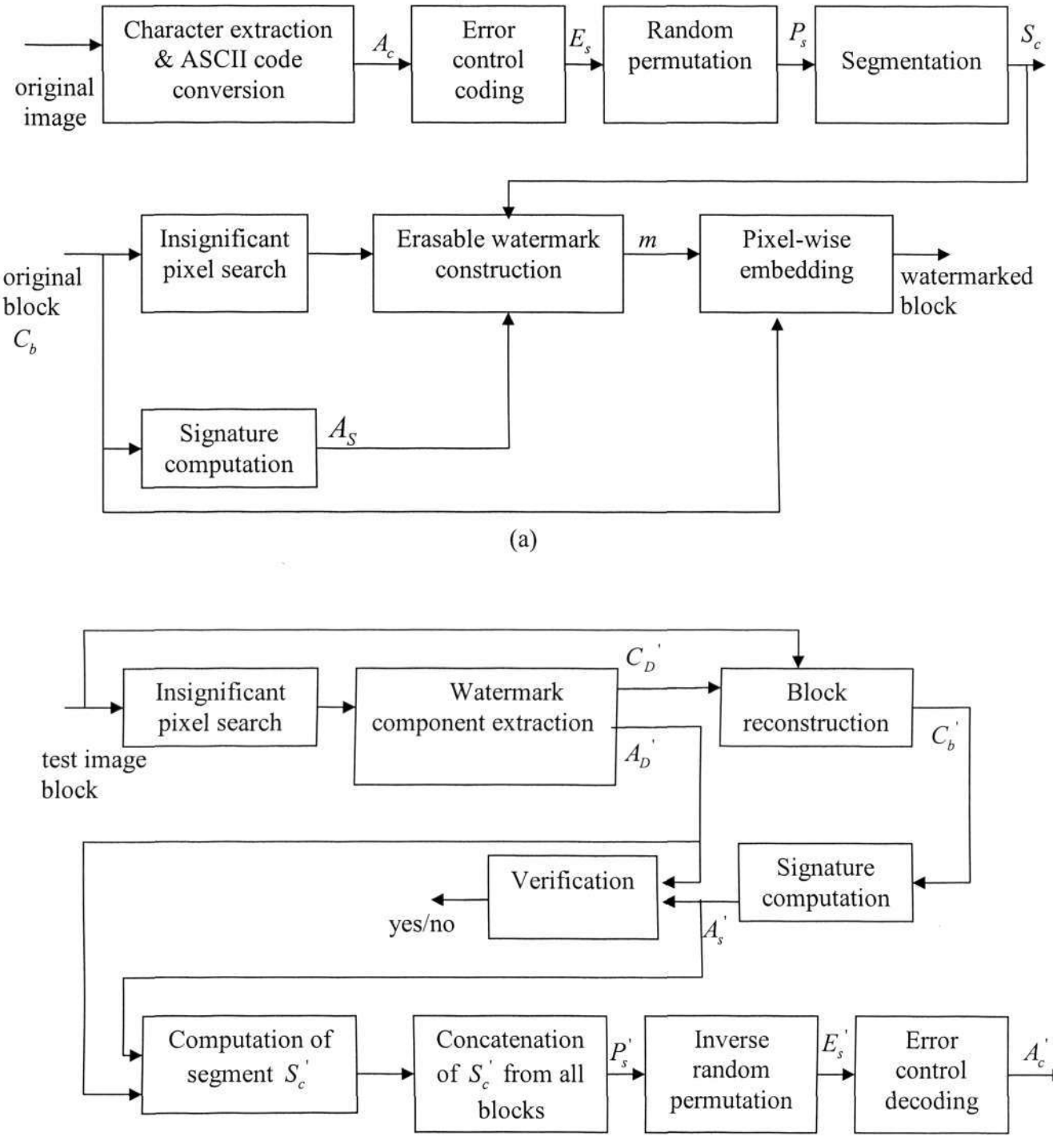



Figure 3.9: Block diagram of the proposed restoration method: (a) Embedding process; (b) blind detection process.

6. In each block, an ordered set of insignificant pixels are searched in the sequential scanning order by using Conditions 1, 2 and 3 described in Section 3.3.1. The insignificant pixel set is losslessly compressed using the run-length coding scheme. While compressing the insignificant pixel set by run-length coding, 10-bit representation is used for the number of white pixels and 1-bit for the number of black pixels. A total of ten bits are used to represent the size of the run-length encoded data. These ten bits along with the run-length encoded data represent the compressed data $C_D$. The term *redundancy* ($R$) is computed according to Equation 3.3 as number of bits available in a block to embed necessary information.

7. Let the sets $S_1$ and $S_2$ be defined such that $S_1$ contains the first 32 bits and $S_2$ contains the next $n_s$ bits of the segment $S_c$ where

$$n_s = \min(R - 64, 76). \tag{3.4}$$

In Equation 3.4, $n_s$ is defined to adjust the size of embedded data if the current block has insufficient redundancy.

8. The 64-bit authentication signature $A_s$ defined by

$$A_s = H(C_b, K, I_b, I_K, S_2) \tag{3.5}$$

is computed from the current block, where $H, C_b$, $I_b$ *and* $I_K$ denote hash function, current block in the original image, block index and image index, respectively.

The block index is used in the computation of signature to resist block-swapping by the attacker and the image index is necessary to resist the Holliman-Memon attack. The reason for using $S_2$ in signature computation is to secure the embedded information against hostile attacks. If these bits used

for restoration purpose are altered by the attacker in a block, then the block becomes inauthentic.

9. The authentication data $A_D$ of size $(n_s + 64)$ bits consists of three sets. Let the sets be denoted as $A_D{}^1, A_D{}^2$ and $A_D{}^3$ where $A_D{}^1$ contains the first 32 bits, $A_D{}^2$ contains the next 32 bits and $A_D{}^3$ contains the remaining $n_s$ bits of $A_D$. The three sets are computed according to Equation 3.6

$$\begin{aligned} A_D{}^1 &= A_s{}^1 \oplus S_1 \\ A_D{}^2 &= A_s{}^2 \oplus S_1{}^c \\ A_D{}^3 &= S_2 \end{aligned} \tag{3.6}$$

where $\oplus$ is the exclusive OR operation, $A_s{}^1$ contains the first 32 bits and $A_s{}^2$ contains the remaining 32 bits of $A_s$.

10. The compressed data $C_D$ and authentication data $A_D$ are concatenated to create the message *m,* which is embedded in the insignificant pixel set. The embedding is performed pixel-wise; so an insignificant pixel holds one bit of *m* and its pixel value is set equal to the message bit it holds. According to Equation 3.4, the maximum size of $S_2$ is 76 bits. Thus the maximum size of $A_D$ in a block is equal to 140 bits. In an ideal case, each block of the original image should have $R \geq 140$. Likewise all blocks in the image are watermarked.

### 3.3.3 Erasable Watermark Detection for Restoration

1. The test image is divided into non-overlapping blocks of $32 \times 40$ pixels. Verification is performed for each block independently and in a sequential order starting from left to right and top to bottom of the image.

2. To verify each block, the message $m$ is extracted by finding the insignificant pixel set using Conditions 1, 2, and 3 as described in Section 3.3.1. During detection, it was found that preprocessing was not necessary. The parameters $R$ and $n_s$ are computed using Equation 3.3 and 3.4, respectively. The compressed data $C_D'$ and the authentication data $A_D'$ are extracted from the message $m$. The compressed data $C_D'$ together with the current watermarked block is used to reconstruct the block $C_b'$.

3. The authentication data $A_D'$ of size ($n_s$ + 64) bits is separated into three sets. Let the sets be denoted as $A_D{}^a, A_D{}^b$ and $A_D{}^c$ where $A_D{}^a$ contains the first 32 bits, $A_D{}^b$ contains the next 32 bits and $A_D{}^c$ contains the remaining $n_s$ bits of $A_D'$.

4. The 64-bit authentication signature $A_s'$ is computed from the reconstructed block $C_b'$ according to

$$A_s' = H(C_b', K, I_b, I_K, A_D{}^c) \tag{3.7}$$

where the two sets $A_s{}^a$ and $A_s{}^b$ are defined such that $A_s{}^a$ contains the first 32 bits and $A_s{}^b$ contains the remaining 32 bits of $A_s'$.

5. The segment containing information bits for restoration purpose; $S_c'$ of size ($n_s$ +32) bits is computed. It consists of two sets: $S_1'$ contains the first 32 bits, which is computed

$$S_1' = A_D{}^a \oplus A_s{}^a \tag{3.8}$$

The other set $S_2'$ contains the remaining $n_s$ bits and is equal to $A_D{}^c$

6. Another set $S_1''$ is computed according to

$$S_1'' = A_D'^{b} \oplus A_s^{b}. \tag{3.9}$$

The reconstructed block $C_b'$ is authentic, if the following condition

$$S_1' = \left(S_1''\right)^c \tag{3.10}$$

is satisfied. Otherwise the current block in the test image has been tampered after watermarking.

7. If necessary, zero-padding can be performed such that the size of $S_c'$ becomes equal to 108 bits. During verification of all blocks in the sequential order, the segments $S_c'$ are concatenated to obtain the sequence $P_s'$.

8. The sequence $E_s'$ is obtained after inverse permutation of $P_s'$ by using the secret key $K$. The $[n, k]$ BCH decoder is applied in each $n$-bit segment of $E_s'$ to obtain $A_c'$. If the number of bits in the last segment of $E_s'$ is less than $n$, then it is not considered for ECC decoding. If the number of error bits in $E_s'$ is within the correction capability of ECC decoder, the original character sequence can be correctly extracted from the ASCII code sequence $A_c'$.

### 3.3.4 Results and Discussions

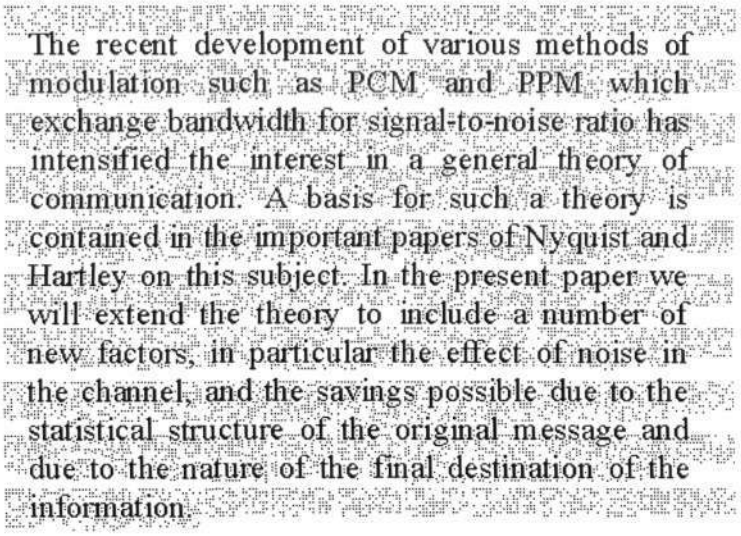
In this section, we present simulation results by constructing the erasable watermark in the proposed restoration method. The authentication signature to be used in this method is the Hashed Message Authentication Code (HMAC). The 64-bit HMAC is used as the authentication signature and the message $m$ is constructed for all blocks as described in the proposed method. The length of HMAC is reduced to 64 bit from 128 bits used in case of localization to accommodate extra information bits for restoration purpose. Using the modified procedure, there exists a number of insignificant pixels

available within a block size of 32×40 pixels. The choice of higher block size can produce higher capacity, but after modifications the number of error bits in a block is also increased. Moreover, restoration would be affected. The values of $n$ and $k$ are chosen to be 63 and 18, respectively for BCH error correction coding. Using this scheme, a total of 10 error bits can be corrected in a 63-bit ECC encoded segment. The original and the watermarked image after pixel-wise embedding of $m$ in each block are shown in Figure 3.10. It is observed that although background noise is present in the watermarked image, the text can still be read and understood by the user. Without any tampering, all blocks in the watermarked images are verified. After verification, the original image can be restored at the blind detector. In Figure 3.11, it is shown that most blocks in the image have high redundancy for embedding $m$. Due to insufficient redundancy in some blocks, a total of 31 error bits are corrected during ECC decoding and the original character sequence is extracted. Figure 3.12 illustrates the number of error bits in each 63-bit segment of the extracted sequence $E_s^{'}$ without any attacks.

We perform multiple modifications such as character deletion, block swapping, character insertion and character substitution in the watermarked image: (1) the only word *'information.'* in last line is deleted; (2) first three blocks are swapped with the last three blocks; (3) the word 'theory' is inserted in the last line; and (4) the word 'for' in line-5 is substituted by the word 'to'. The resulting attacked image is shown in Figure 3.13. Blind detection is performed on the attacked image to restore the original character sequence after tamper localization. In Figure 3.14, the authentic reconstructed blocks are shown after the watermark erasing process. The detector

The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist and Hartley on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

(a)




(b)

Figure 3.10: (a) Original image of 320×440 pixels, (b) the watermarked image after embedding the message *m* in 110 blocks.

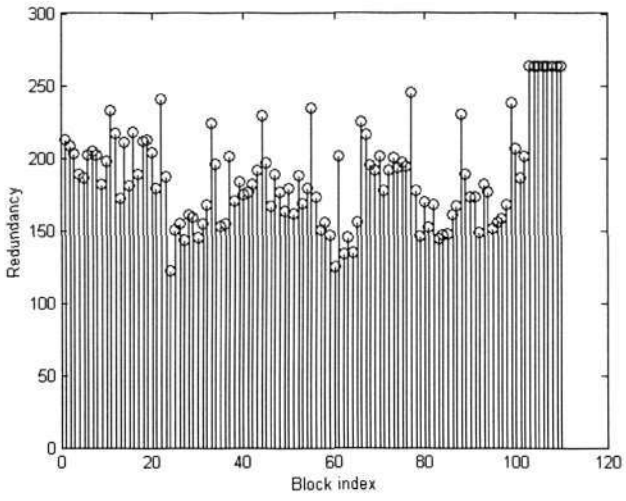



Figure 3.11: Redundancy ($R$) for each block of the original image.

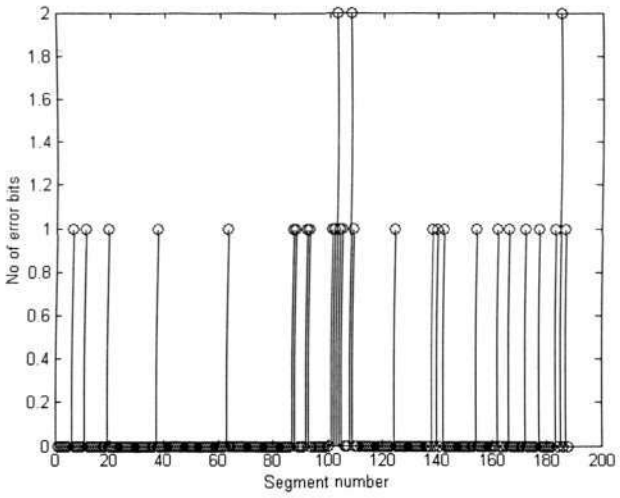



Figure 3.12: The number of error bits in each 63-bit segment of the bit sequence $E_s'$ extracted from the watermarked image.

correctly localizes 13 tampered blocks shown as dark regions. A total of 753 error bits are corrected during ECC decoding and the original character sequence is extracted. The number of error bits in each 63-bit segment of the extracted sequence $E_s'$ is shown in Figure 3.15. Since the number of the error bits in each segment does not

exceed 10, correct extraction of original character sequence is possible after ECC decoding.

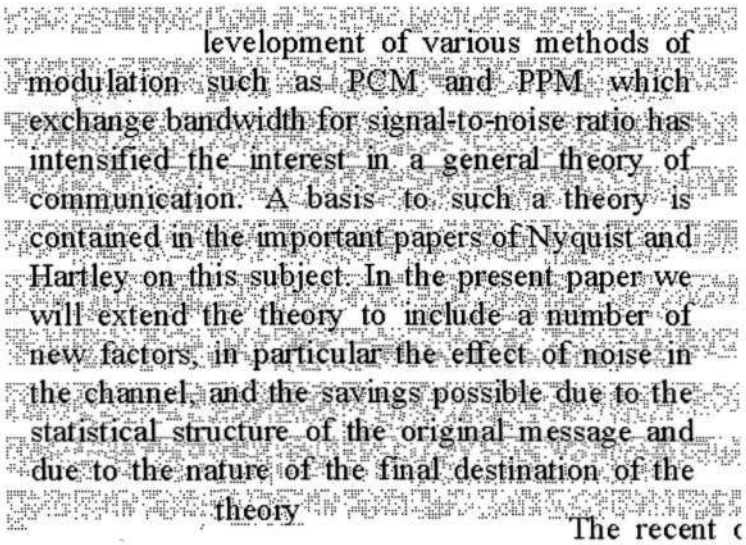



Figure 3.13: Attacked image after multiple alterations like character deletion, block swapping, character insertion and character substitution.

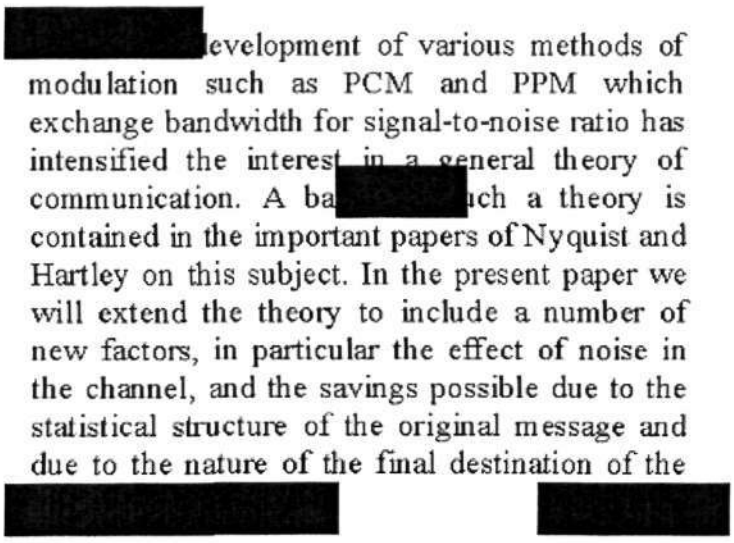



Figure 3.14: Image showing the authentic reconstructed blocks and a total of 13 tampered blocks are shown in dark region.

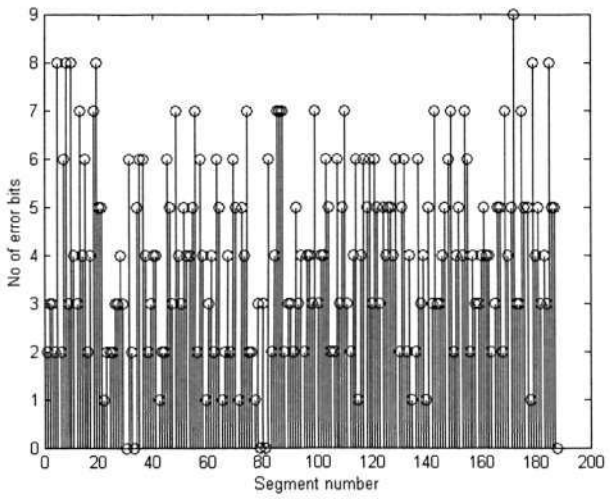
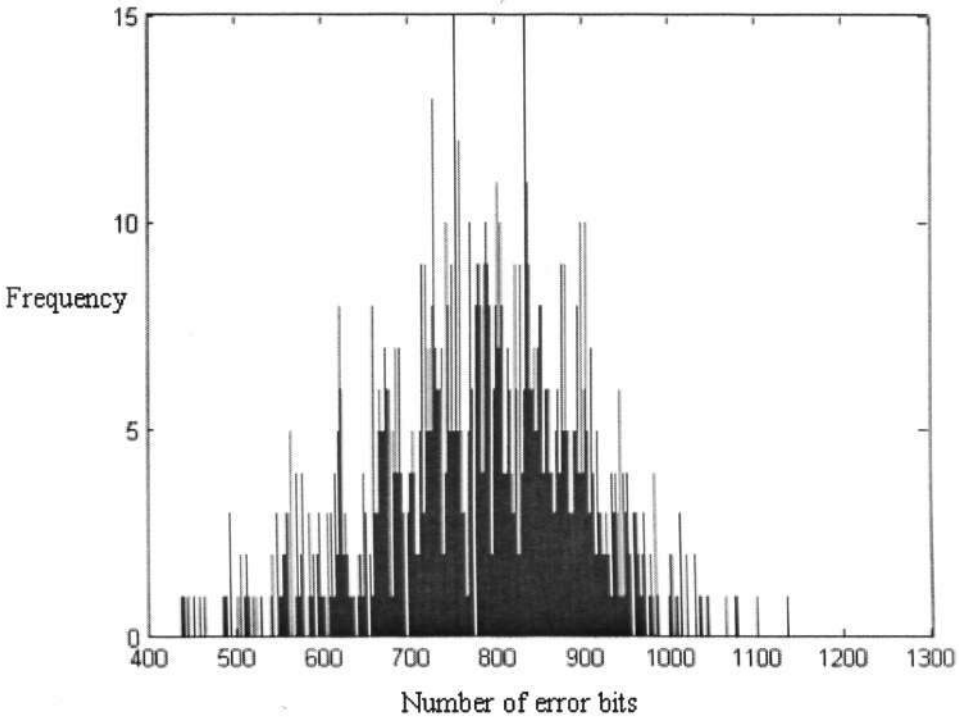



Figure 3.15: The number of error bits in each 63-bit segment of the bit sequence $E_s^{'}$ extracted from the attacked image.

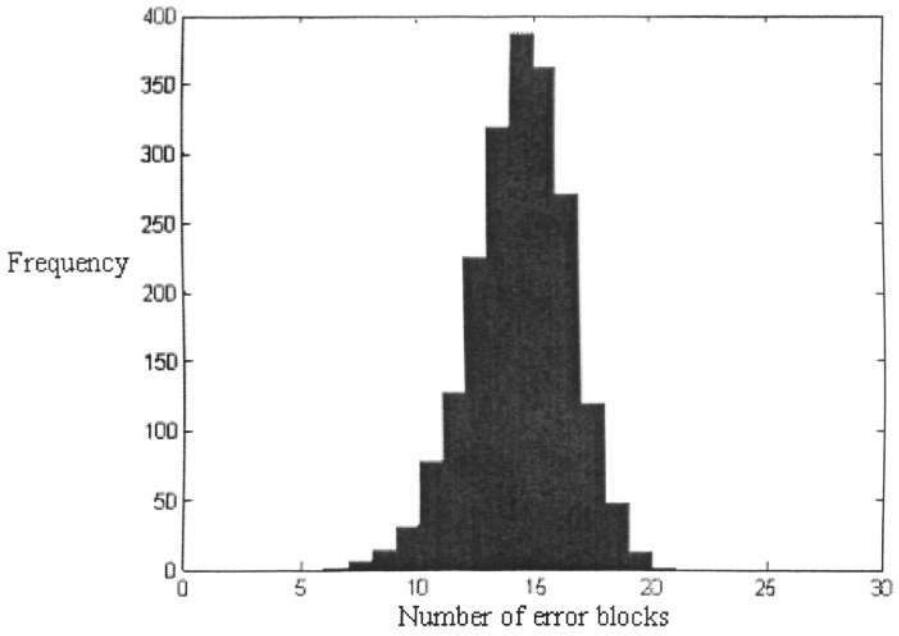
Restoration of the original character sequence is not possible during detection if the number of error bits exceeds 10 in any 63-bit segment of the extracted sequence $E_s^{'}$. The error bits are induced in $P_s^{'}$ after tampering and the error bits are randomly distributed in $E_s^{'}$ due to the random permutation. To study restoration failure, we performed various attacks using the original image shown in Fig. 3.10 (a). During each attack, a unique key is used to watermark the original image using the embedding method. Then we select different combination of blocks in the watermarked image by using the key and the selected blocks are tampered. Then the sequence $P_s^{'}$ is extracted and an inverse permutation is performed on $P_s^{'}$ using the key to obtain $E_s^{'}$. The number of error bits in each 63-bit segment of $E_s^{'}$ is then computed. The number of blocks selected for tampering is increased sequentially till the restoration failure occurs, i.e. the total number of error bits exceeds by 10 in any 63-bit segment of $E_s^{'}$. The threshold value (i.e. the number of error bits and error

blocks that can be corrected) is found just before reaching the restoration failure in each attack. By varying the key, a total of 2000 attacks are performed. The mean value of the number of error bits and error blocks is found to be 778.08 and 13.88 respectively. In Figure 3.16, the histograms of the number of error bits and error blocks are shown.

The performance of the proposed algorithm can be compared with the previous restoration method [98]. As discussed in Section 3.3, in this method there exists a possibility of false tamper detection and restoration failure. The restoration of a character at any location depends on the watermark which is embedded in another character. The relevant information for restoration is concentrated at certain specific locations rather than being distributed in the whole image. The restoration capability is thus reduced for certain alterations. After multiple alterations such as combined deletion and insertion of characters, restoration would not be possible using this method due to a loss of synchronization. In the proposed method, relevant information is embedded in the whole image. As such, there is no restoration failure even after multiple alterations as shown in Figure 3.15. The restoration capability of the proposed method is limited by the number of error bits in the extracted sequence. If the document image is most edited, then ECC coding could not correct the errors beyond a limit. However, the tampering in the document image can still be localized with an accuracy of $32 \times 40$ pixel block size. For the proposed restoration method, we discuss about two cases of false tamper detection. First, a block is declared to be inauthentic when there is no tampering in the block at all. The insignificant pixel positions are selected in such a manner that correct watermark detection is possible at the receiver without using the original image. So when there is no tampering, each block in the test image is verified to be authentic. Second, a block is declared to be




(a)




(b)

Figure 3.16: Histogram: (a) The error bits; (b) the error blocks.

inauthentic when another block in the test image is tampered. In the proposed method, watermarking is performed for each block independently; hence authenticity status of each block remains unaffected by the attacker's activity within other blocks. Thus there is no possibility of false tamper detection due to the proposed method.

## 3.4 Summary

In this Chapter, we proposed new methods for localization and restoration in binary document image authentication using erasable watermarks. The proposed localization method can localize different modifications in the image with an accuracy of $40 \times 40$ pixel block size. To construct the erasable watermark in each block of the original image, an ordered set of insignificant pixels was selected and then compressed using the run-length coding scheme. After embedding process, the user could interpret the document easily in the presence of noise incorporating the inherent ability of human vision. After verifying each block, the user could restore the original image for further analysis. The localization accuracy of the proposed method is significantly improved and does not suffer from any parity attack. The new restoration method is particularly useful for restoration of the original character sequence after localization in text document images. An erasable watermark was constructed by combining the compressed data, HMAC of the block and ECC encoded ASCII code sequence. In the proposed method, it is possible to restore the original character sequence after multiple alterations.

# Chapter 4

# Secure Authentication Watermarking for Localization Against the Holliman-Memon Attack

## 4.1 Introduction

Authentication watermarking schemes using block-wise independent watermarks for localization are vulnerable to the Holliman-Memon attack [65]. Though the localization accuracy and cryptographic security offered by a block-wise localization method is high, its block-wise independence was used by Holliman and Memon to design a counterfeiting attack in [65]. If a set of images are watermarked with the same key, it is possible to modify an arbitrary image to be authentic using this attack. The attacker divides the image into non-overlapping blocks and for each block performs a search in the set of authentic blocks. The original block is replaced with the most similar block to maintain perceptual quality of the forged image. Thus the attacker creates the forged image by a collage of authentic blocks and the forged image is authenticated using block-wise independent watermarks. This particular counterfeiting attack is known as the Holliman-Memon attack or collage attack or vector quantization attack. To illustrate the Holliman-Memon attack, we present an example given in [101]. A database of 19 fingerprint images has been used in this attack. The images of size $640\times640$ are watermarked with $8\times8$ block size using Wong's localization scheme [70]. While the original unwatermarked image is shown in Figure 4.1(a), the counterfeit image for Wong's scheme is shown in Figure 4.1(b). As a result of the Holliman-Memon attack, the image shown in Figure 4.1(b) is

(a)

(b)

Figure 4.1: Example of the Holliman-Memon attack: (a) Original unwatermarked fingerprint image; (b) counterfeit image using the Holliman-Memon attack from 19 watermarked images.

verified to be authentic by Wong's scheme. A number of countermeasures have been proposed in the literature to resist this attack [81, 84, 86, 101]. However, most of these methods can resist this attack at the cost of localization accuracy. In this chapter, we suggest a new method so that authenticity of individual blocks can be verified without sacrificing localization accuracy.

## 4.2 Countermeasures Against the Holliman-Memon Attack

In this section, we describe various countermeasure methods proposed in the literature to resist the Holliman-Memon attack.

### 4.2.1 Neighborhood Dependent Blocks

In [65], a practical approach is suggested to remove the block-wise independence of the watermark. The signature embedded in each block is calculated using some of the surrounded data from neighboring blocks, as well as the data within the block itself. Using this method, a collage of watermarked blocks cannot be authenticated by the detector, because the neighborhood relationship between the blocks is not preserved in the fake image. However, this introduces some ambiguity to the localization, because a change in one block can change the signature that should be embedded in its neighbors.

### 4.2.2 Image Index and Block Index in Signature Computation

In [86], Wong and Memon suggested including a block index and a unique image index while computing the signature for each block. The use of block index solves the problem of block swapping in a watermarked image. The attacker needed to search for the most similar blocks only at identical block positions of all database images. Although this complicates the attacker's task, it is still possible to launch an attack if the number of database images is high. The use of a unique image index completely eliminates the possibility of this attack. However, the image index is also necessary for verification at the detector.

In some applications such as e-commerce it may be possible to make such indices publicly available. However, in many cases, managing such indices would create additional overheads and may not be operational feasible. To solve this problem, the authors suggested extracting the image index from the image itself by computing the hash of its most significant bits (MSBs). However if any of the MSBs are altered due to tampering in the image, it will lead to a complete loss of localization. Another approach to solve the problem of image index management has been proposed by Fridrich *et al.* in [81]. In this method, an image index is embedded in the image at multiple locations in a robust manner. In case of a tampering, the multiple copies increase the chance of extracting the correct image index at the detector. However, this does not guarantee the correct index extraction in all cases of tampering and the embedding process increases visual distortion in the watermarked image.

### 4.2.3 Separation of Content Origin and Authentication

In [84], the information about the content origin was embedded in each block of the image. This method was based on Wong's scheme and a symmetric logo was embedded instead of a fixed logo. The symmetric logo contained information about image dimensions, camera ID, block index, author ID, image index and other ancillary data. The logo structure was used to verify the block integrity and the logo content provided the information about the block origin. However, due to the use of a symmetric logo the probability of false authentication increased from $2^{-N}$ to $2^{-N/2}$, where $N$ is the size of the logo. Though each block in the fake image was authenticated by this method, the origin information in the extracted logos could detect that the image has been constructed using the Holliman-Memon attack.

### 4.2.4 Hierarchical Watermarking

In [101], Celik *et al* proposed a hierarchical watermarking method based on Wong's scheme. Using this method, the image was divided into blocks in a multilevel hierarchy and block signatures were calculated in each hierarchy. While signatures of small blocks on the lowest level of the hierarchy ensure superior tamper localization accuracy, higher level block signatures would resist this attack through a trade-off between security and the localization accuracy.

The objective of a localization method is to verify whether each block in an image is authentic or not. Wong's scheme could achieve this objective against any kind of tampering with the accuracy of a chosen block size. However its drawback lies in verifying each block in the fake image to be authentic against the Holliman-Memon attack. The above discussion shows that the proposed countermeasures could resist this attack with the performance reduction as compared to Wong's scheme. The countermeasure using the unique image index in [86] is of particular interest to this Chapter. If the correct image index can be extracted for verification after the fragile embedding process, then the Holliman-Memon attack could be resolved without the loss of localization accuracy. In the next section, we address this motivation by proposing a new localization method using a unique image index. In the proposed method, the localization accuracy remains at the level of chosen block size and it is possible to determine the authenticity of each block in the fake image resulting from the Holliman-Memon attack.

## 4.3 Proposed Method

In this section, we propose a new method for enhancing the security of Wong's scheme to resist the Holliman-Memon attack. The idea behind the method is based on the use of a unique image index in the computation of the authentication signature for every block. The image index along with the signature is embedded in a fragile manner so that the visual distortion is minimized. By designing an informed detector, the correct image index is estimated from the fake image. The proposed embedding and detection methods are described as follows:

### Embedding:

1. The original image $X$ is partitioned into non-overlapping blocks of $12 \times 12$ pixels. Let $\{X_1, X_2, ..., X_B\}$ denote the individual blocks in a sequential order, starting from left to right and top to bottom of the image. Watermarking is performed for each block independently and in a sequential order.
2. For each block $X_r$, a corresponding block $X_r^e$ is formed by setting the least significant bit (LSB) of each pixel to zero. The authentication signature to be embedded in this method is the 128-bit Hashed Message Authentication Code (HMAC). The HMAC ($H_r$) is computed according to the following equation:

$$H_r = H(X_r^e, K, r, I_X) \qquad (4.1)$$

where $H, K, r, I_X$ denote hash function, secret key, block index and image index, respectively.

3. Out of the 144 least significant bits, $H_r$ is inserted in 128 positions and the rest LSBs hold the 16-bit image index. Using the 16-bit image index, it is possible to securely watermark $2^{16}$ or 65536 images with one secret key.

## Detection:

After embedding the authentication signature, the following side information about the watermarked image is available to the detector:

- In a block, the authentication signature computed using the embedded image index should match with the extracted signature.
- All blocks contain the same image index.
- All blocks are connected with each other, i.e. it is possible to move from one pixel to any other pixel in the image using an 8-connected path [75].

It is possible to estimate the correct image index at the detector by using the above side information.

1. The test image $X'$ is partitioned into non-overlapping blocks of $12 \times 12$ pixels and detection is performed for each block in a sequential order.
2. The 128-bit authentication signature $H_r^{\,d}$ and the 16-bit image index $I_X^{\,d}$ are extracted from the least significant bits of each block $X_r'$ and $X_r^{\,d}$ is formed by setting the LSBs of $X_r'$ to zero. The HMAC ($H_r^{\,c}$) is computed according to the following equation

$$H_r^{\,c} = H(X_r^{\,d}, K, r, I_X^{\,d}) \qquad (4.2)$$

where $H, K, r, I_X^{\,d}$ denote hash function, secret key, block index and extracted image index, respectively.

3. All the bits in a block can be divided into three groups: (a) MSBs (b) LSBs containing the authentication signature and (c) LSBs containing the image

index. If any of these bits are changed after an attack, either the extracted signature or the computed signature will be altered. A matrix ($R$) of *(M/12)* rows and *(N/12)* columns is constructed while computing $H_r^c$ for each block. Each entry of $R$ represents a particular block in image $X'$ and this relationship is shown in Figure 4.2. The magnitude of an entry in $R$ is '1' if $H_r^d$ and $H_r^c$ matches in the corresponding block; otherwise it is equal to '0'.

4. Different image indices are extracted from all the blocks in $X'$; e.g. 1, 14, 156, 1089 etc. Let each such index be termed as the candidate image index ($I_X^T$). The image index estimation algorithm is used to determine the authenticity score for all the candidate image indices. So the algorithm is run for each candidate image index once and the candidate image index is matched with $I_X^d$ in each block. For every $I_X^T$, the authenticity score ($A_S$) is computed according to

$$A_S = \frac{1}{B}\sum_{u=1}^{M/12}\sum_{v=1}^{N/12} S(u,v) \tag{4.3}$$

where $B$ is the total number of blocks, $M$ and $N$ are size of the test image $X'$ and $S$ is the matrix computed using the image index estimation algorithm.

5. The candidate image index with the highest authenticity score is chosen to be the estimated image index. Let the estimated image index be denoted as $I_E$. A block in the test image will be authenticated if the following conditions are satisfied:

| | | | |
|---|---|---|---|
| **1** | **2** | **3** | **4** |
| **5** | **6** | **7** | **8** |
| **9** | **10** | **11** | **12** |
| **13** | **14** | **15** | **16** |

| | | | |
|---|---|---|---|
| 1 (**1**) | 0 (**2**) | 1 (**3**) | 1 (**4**) |
| 0 (**5**) | 1 (**6**) | 1 (**7**) | 0 (**8**) |
| 1 (**9**) | 1 (**10**) | 1 (**11**) | 1 (**12**) |
| 1 (**13**) | 0 (**14**) | 1 (**15**) | 0 (**16**) |

Figure 4.2: (Left) an image of 48×48 pixels is partitioned into blocks of 12×12 pixels and the block numbers are shown in a sequential order. (Right) The matrix '*R*' having the entries either 0 or 1 and the corresponding block number are written within the bracket.

(a) The image index extracted from the block ($I_X{}^d$) is equal to the estimated image index ($I_E$).

(b) The corresponding entry in matrix $R$ for the block is 1.

6. The authenticity measure ($A_M$) of the test image is defined to quantify the attack severity and its value is equal to the authenticity score of $I_E$. The maximum value of $A_M$ is 1 when all blocks in the test image are authenticated and 0 when all blocks in the test image are inauthentic.

## Image Index Estimation Algorithm:

For every $I_X{}^T$, a matrix '*T*' of *(M/12)* rows and *(N/12)* columns is constructed and each entry of *T* corresponds to an individual block in a sequential order as found in $R$.

*for* $u = 1, 2, ..., M/12$ and $v = 1, 2, ..., N/12$

*if* ($R(u,v) = 1$ and $I_X^{\,d} = I_X^{\,T}$), then
$T(u,v) = 1$
*else*
$T(u,v) = 0$
*end*

*end*

A score matrix '*S*' of *(M/12)* rows and *(N/12)* columns is then computed from *T*.

*for* $u = 1, 2, ..., M/12$ and $v = 1, 2, ..., N/12$

*if* $T(u,v) = 1$, then
$S(u,v) = (G+1)/B$ (4.4)
*else*
$S(u,v) = 0$
*end*

*end*

where $G$ is the total number of 1's in $T$ that is connected to the present entry at *(u, v)* through the 8-connected path and $B$ is the total number of blocks. The 8-connected path consists of 1's in $T$ and it does not take 0's into account. The matrix $G$ is computed by treating $T$ as the binary image and then applying the connected component labeling procedure [75]. The authenticity score ($A_S$) for the candidate image index $I_X^{\,T}$ is then computed according to Equation 4.3.

**Example: Authenticity Score Computation**

Using the proposed method described above, the authenticity score computation for a candidate image index is illustrated by an example. Consider the matrix $T$ constructed for a candidate image index:

$$T=\begin{bmatrix}1 & 1 & 0 & 1\\ 0 & 0 & 0 & 1\\ 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 1\end{bmatrix}.$$

For this 4×4 matrix, the score matrix $S$ is computed as follows: For the first entry at $T$ (1, 1), the total number of 1's connected to it is 1; so $G$ is equal to 1. Hence $S$ (1, 1) = 2/16. For the entry at $T$ (3, 1), the total number of 1's connected to it is 4; so $G$ is equal to 4. Hence $S$ (3, 1) = 5/16. For the entry at $T$ (4, 4), the total number of 1's connected to it is 4; so $G$ is equal to 4. Hence $S$ (4, 4) = 5/16.

After computation at each entry of $T$, the score matrix $S$ is found as follows:

$$S=\begin{bmatrix}2/16 & 2/16 & 0 & 2/16\\ 0 & 0 & 0 & 2/16\\ 5/16 & 5/16 & 0 & 0\\ 5/16 & 0 & 5/16 & 5/16\end{bmatrix}.$$

The authenticity score for the candidate image index is,

$$A_S=\frac{1}{16}\sum_{u=1}^{4}\sum_{v=1}^{4}S(u,v)=0.1289.$$

## 4.4 Results and Discussions

In this section, we present simulation results to demonstrate the effectiveness of the proposed method. In our implementation, we have used the hashed message authentication code as the authentication signature. The message-digest hash function (MD5) [79] is used to compute the 128-bit HMAC. It is evident that any cryptographic hash function can be used to achieve similar results for the proposed watermarking method. For the first test case, we demonstrate the localization capability of this method against tampering in the watermarked image. The 'Barbara' image of size 300×300 pixels is used as the original image. The decimal equivalent of

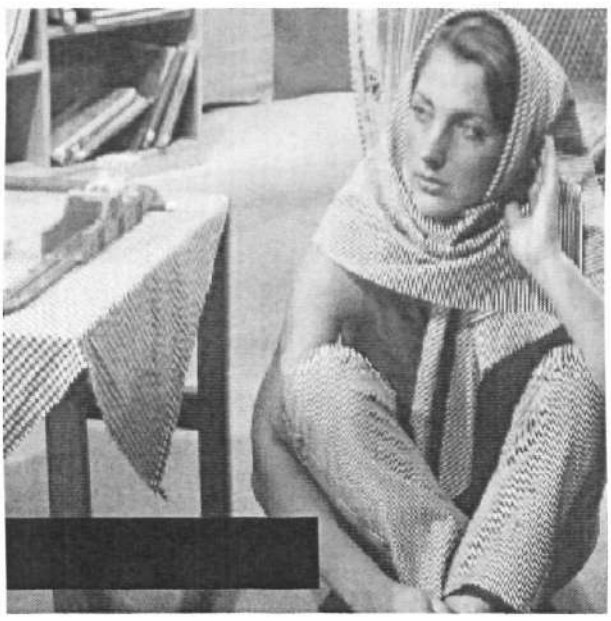
the 16-bit image index used in watermarking is 23159. Using the proposed method, the 128-bit authentication signature and 16-bit image index are embedded in 625 blocks of the original image. The original image and watermarked image are shown in Figure 4.3. Without any tampering, all blocks in the watermarked image are authenticated by the proposed detection method. The estimated image index is 23159 and authenticity measure of the watermarked image is 1. The watermarked image is then tampered with the words 'copyright image' inserted in it. The resulting attacked image and authenticated image are shown in Figure 4.4. The dark region in the authenticated image indicates the tampered blocks. The estimated image index is 23159 and the authenticity measure of the attacked image is approximately 0.88. The effectiveness of the proposed method against the Holliman-Memon attack is demonstrated in our second test case. As in [65, 101], a database of fingerprint images is used for this attack. A total of 64 fingerprint images of size 300×300 pixels are watermarked using the proposed embedding method. The images are watermarked using the 16-bit image indices whose decimal equivalent ranges from 1 to 64. The unwatermarked fingerprint image and the fake fingerprint image constructed using the Holliman-Memon attack is shown in Figure 4.5. For this attack, the most similar block is searched at the identical block position of 64 database images using the mean square error (MSE) criterion. For the fake image, the PSNR is approximately 22.58 dB. The visual quality of the fake image is degraded as compared to the method in [65] and blocking artifacts are visible. Since the block index is used in the signature computation step, the codebook construction domain is restricted in the proposed method. The visual quality of the fake image is therefore reduced. As the size of the database increases, it is possible to improve the visual quality of the fake image and the number of blocking artifacts may diminish. The proposed detection method is

(a)

(b)

Figure 4.3: (a) Original 'Barbara' image of size 300×300 pixels, (b) watermarked image after embedding the authentication signature and the image index in each block.


(a)



(b)

Figure 4.4: (a) Attacked image in which the words 'Copyright Image' are inserted, (b) image showing tamper localization in dark regions.

(a)

(b)

Figure 4.5: (a) The unwatermarked fingerprint image of size 300×300 pixel, (b) fake image constructed using Holliman-Memon attack.

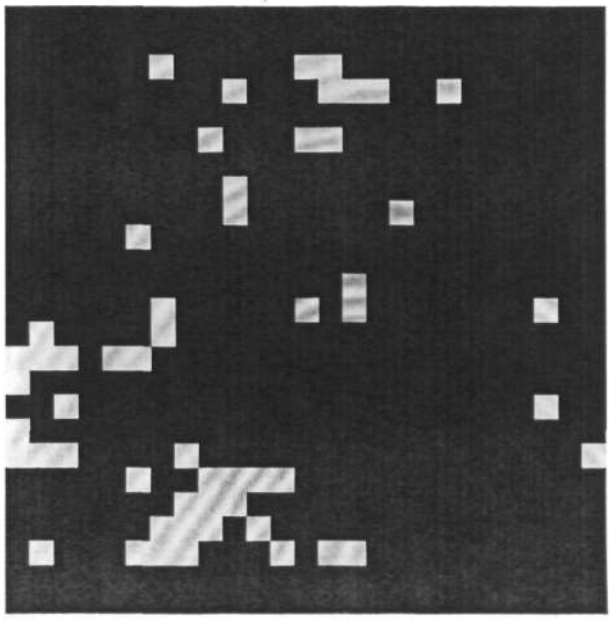


Figure 4.6: Detection output after verifying the fake image using the proposed method. Dark region in the image shows the inauthentic blocks.

used to verify each block in the fake image and the result is shown in Figure 4.6. A total of 55 blocks out of 625 blocks in the fake image are authenticated. The estimated image index is 8 and the authenticity measure of the fake image is approximately $0.94\times10^{-3}$ which indicates the attack severity.

To demonstrate the correlation between the proposed authenticity measure and attack severity, we perform the following experiment using 64 watermarked fingerprint images. Various fake images are generated for the fingerprint test image in Fig. 4.5 using the Holliman-Memon attack. For generating a fake image, a set of fingerprint images are randomly chosen from 64 images by using a key. A total of 381 fake images are generated by varying the key and the number of watermarked images used for performing the Holliman-Memon attack. The proposed method is used to verify the fake images and the relationship between the percentage of number of inauthentic

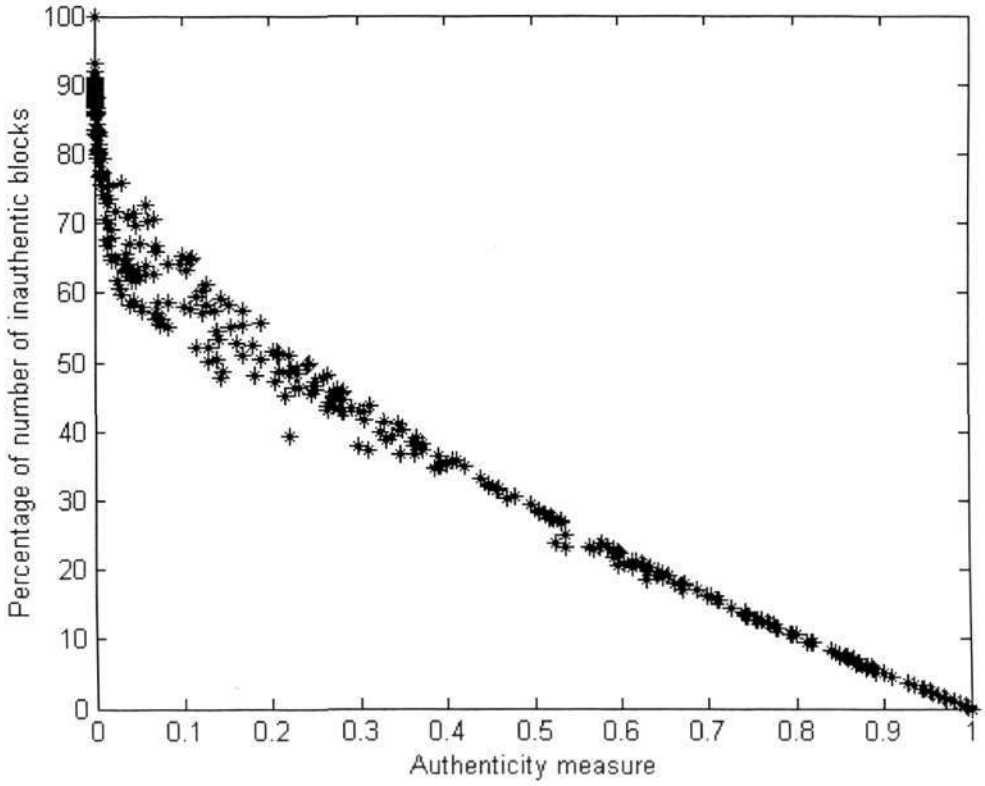



Figure 4.7: Percentage of number of inauthentic blocks ($P_I$) vs. the authenticity measure ($A_M$) for the fingerprint test image.

blocks ($P_I$) and the authenticity measure ($A_M$) is shown in Fig. 4.7. As $P_I$ increases (attack severity increases), the authenticity measure has a decreasing trend and vice versa. The correlation coefficient [78] between $P_I$ and $A_M$ is found out to be -0.96 approximately.

To test the effectiveness of the proposed method further, a total of 164 images are chosen from the databases [102, 103] and the images are resized to generate the original images of size $300 \times 300$ pixels. The proposed embedding method is used to generate 164 watermarked images using the 16-bit image indices whose decimal equivalent ranges from 1 to 164. Six original images are chosen for demonstrating the

effectiveness of the proposed method and the images are shown in Fig. 4.8 (a)-(f). For each original image, the corresponding fake image is constructed using the Holliman-Memon attack. While constructing a fake image, the watermarked image corresponding to the original image is not considered during the attack and thus each fake image is generated from 163 watermarked images. The fake images are shown in Fig. 4.9 (a)-(f). The proposed detection method is used to verify the fake images and the detection output is shown in Fig. 4.10 (a)-(f). The authenticity measure, PSNR and number of inauthentic blocks for the fake images are shown in Table 4.1. The detection of large portions of inauthentic regions and low authenticity measure for the fake images shows that the Holliman-Memon attack is practically infeasible against the proposed method.

To show the correlation between the authenticity measure and attack severity, various fake images are generated for each original image through the procedure used in case of the fingerprint test image. Then the proposed detection method is used to verify the fake images. The relationship between percentage of number of inauthentic blocks and the authenticity measure is shown in Fig. 4.11 (a)-(f) for six test cases. The high magnitude of the correlation coefficients between $P_I$ and $A_M$ summarized in Table 4.2 shows that the proposed authenticity measure quantifies the attack severity.

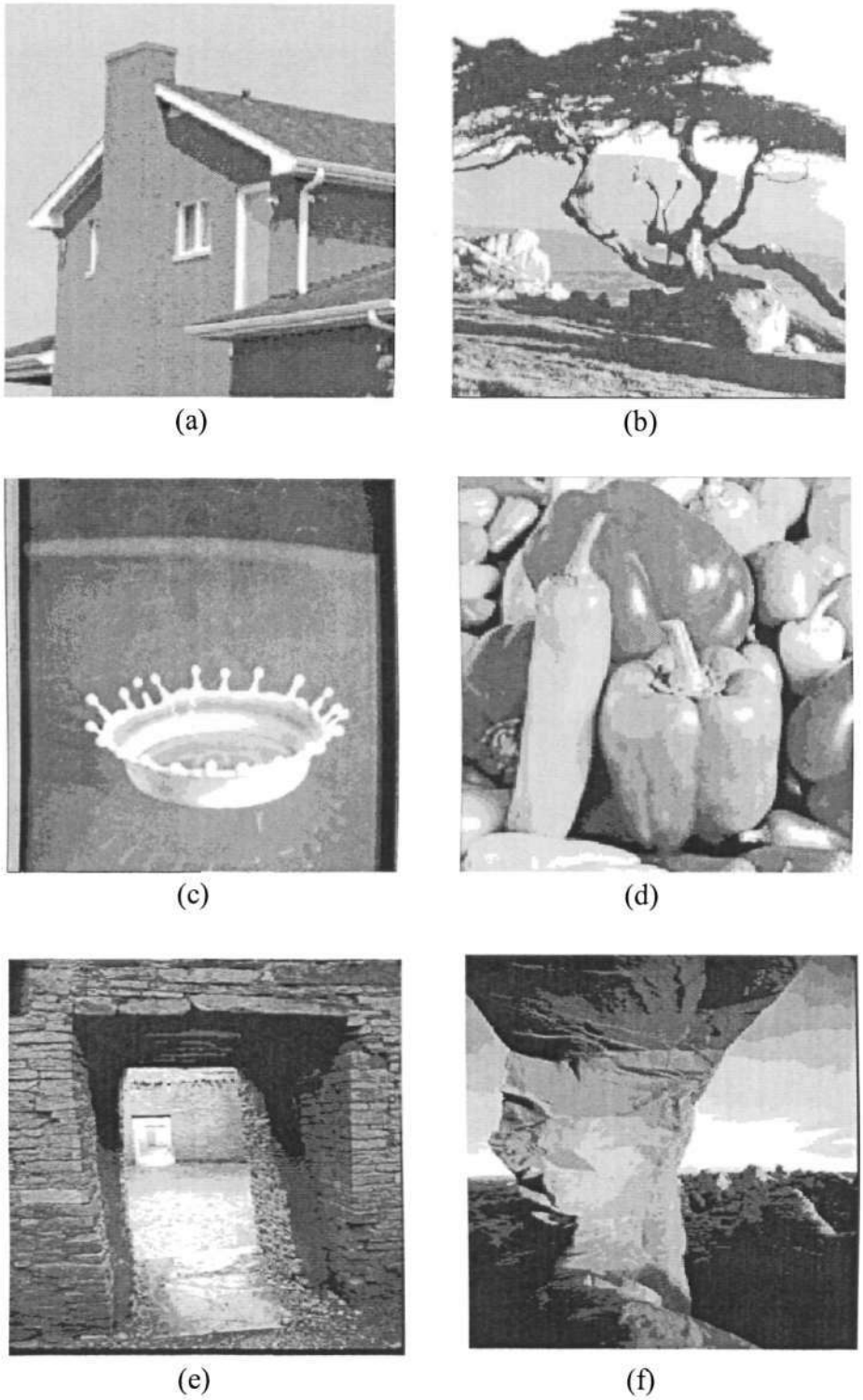



Figure 4.8: Original test images.

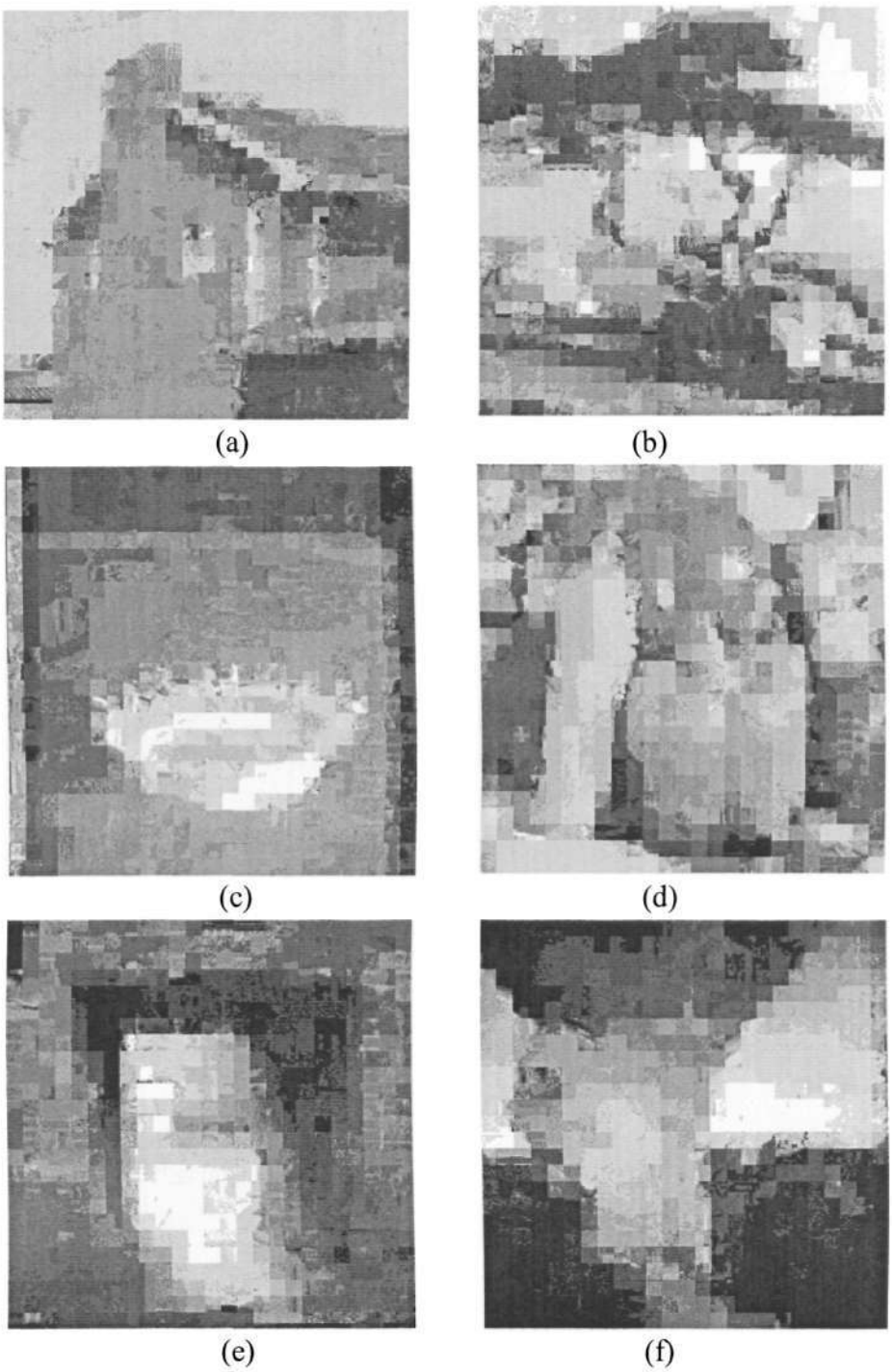



Figure 4.9: The fake images constructed by performing the Holliman-Memon attack using 163 watermarked images.

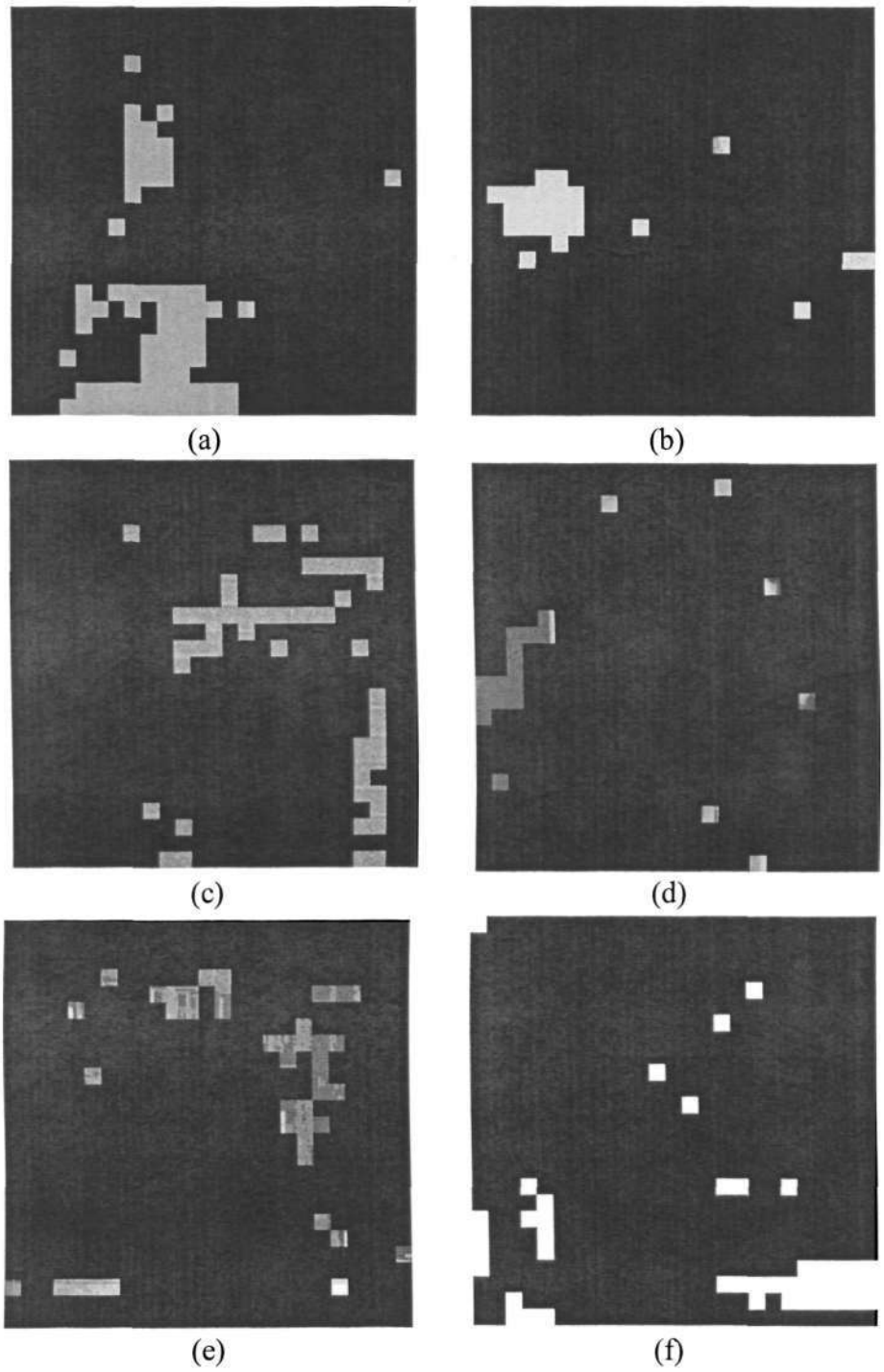



Figure 4.10: Detection output after verifying authenticity of the fake images. Dark region in the image shows the inauthentic blocks.

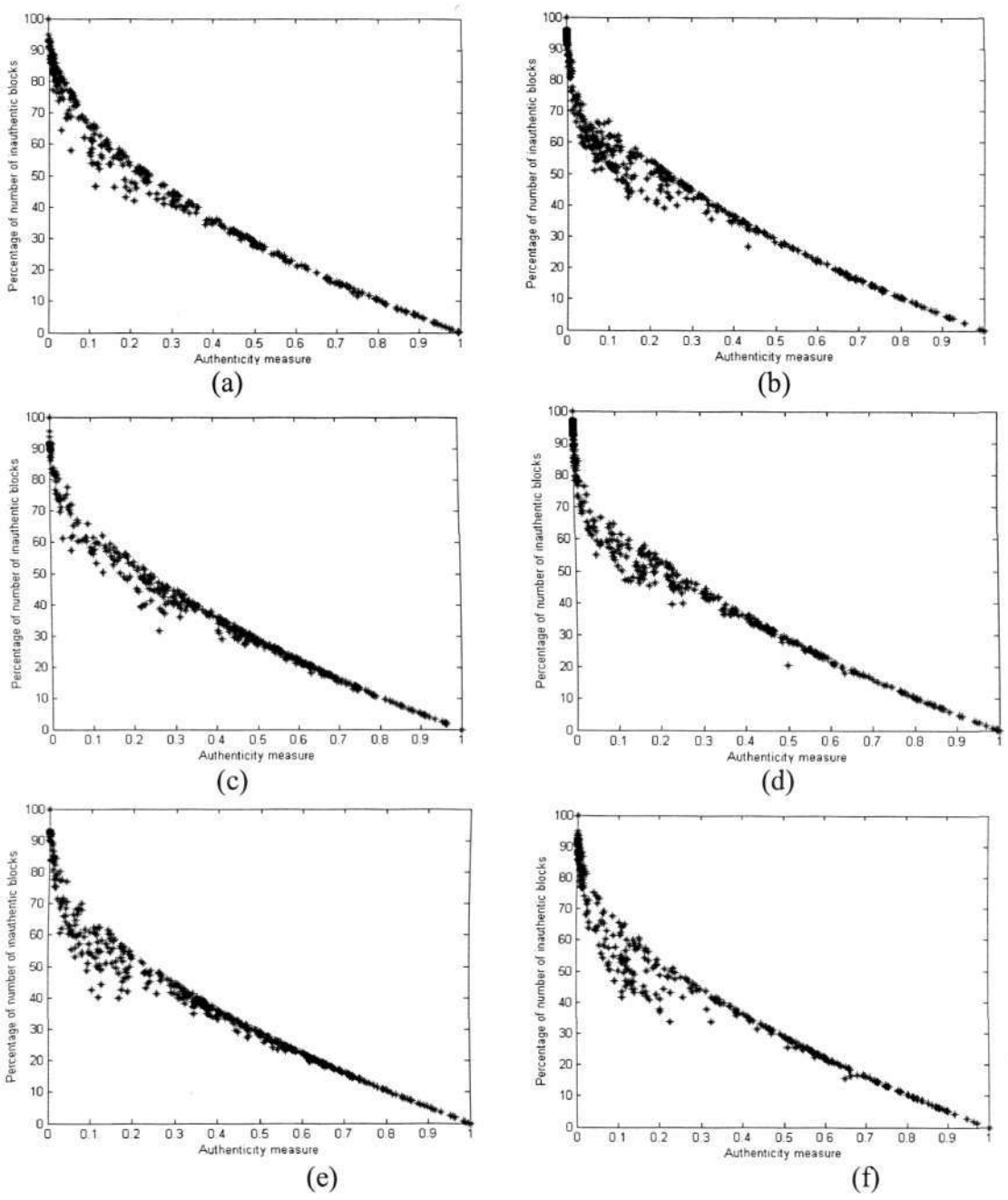



Figure 4.11: Percentage of number of inauthentic blocks ( $P_I$ ) vs. the authenticity measure ( $A_M$ ) for six test cases.

Table 4.1: Performance attributes for all test cases

| Test cases | PSNR (dB) | Number of inauthentic blocks | Authenticity measure |
|---|---|---|---|
| a | 22.28 | 556 | 0.0069 |
| b | 18.21 | 600 | $9.44\times10^{-4}$ |
| c | 21.55 | 575 | 0.0015 |
| d | 20.59 | 605 | $4.5\times10^{-4}$ |
| e | 20.15 | 583 | 0.0011 |
| f | 21.15 | 581 | 0.0014 |

Table 4.2: Correlation coefficients between $P_I$ and $A_M$

| Test cases | Number of fake images | Correlation coefficient |
|---|---|---|
| a | 445 | -0.96 |
| b | 449 | -0.91 |
| c | 436 | -0.95 |
| d | 417 | -0.93 |
| e | 478 | -0.94 |
| f | 414 | -0.93 |

The performance of the proposed method to resist the Holliman-Memon attack can be compared with previous localization methods. As discussed in Section 4.2, the possibility of this attack is entirely eliminated using a unique image index in [86]. However, for verification it is necessary to manage the database of such indices which may not be possible in many practical applications. In [101], the hierarchical watermarking method could detect this attack at a larger region instead of the chosen block size. In the proposed method, after this attack each block in the fake image can be verified, without any loss or ambiguity in localization. After both Holliman-Memon attack and tampering, localization accuracy of the proposed method remains at the level of chosen block size similar to Wong's scheme. Since the correct image index can be estimated at the detector, it is not necessary to manage the database of

image indices. The authenticity measure quantifies the attack severity in an image by taking connectivity among the authentic blocks into account. As the attacker tries to approximate an unwatermarked image by finding similar blocks in large number of database images, the authenticity measure for the fake image decreases significantly as shown in Table 4.1. More blocks in the fake image are verified to be authentic when this measure is of high value. The blocks for the fake image are to be chosen within a less number of database images and the blocks from any such image should be connected with each other to maximize this measure. This would make the attacker's task more difficult to generate the fake image of reasonable perceptual quality. All blocks can be authenticated only if the authenticity measure of the fake image is 1. In that case, the fake image should be exactly equal to one of the database images.

Localization accuracy of the proposed method is bounded by the chosen block size. As the authentication signature is embedded into the least significant bit-plane of the block, minimum block size is determined by the length of the authentication signature. High security against content modification is obtained by using the cryptographic hash function in the signature computation. The only method to break the hash function is a brute-force attack which, according to the birthday paradox requires roughly $2^{n/2}$ attempts to be successful, where $n$ is the number of bits in the hash output. In Wong's scheme, a block size of $8 \times 8$ pixels is chosen to embed the 64-bit signature. In this case approximately $2^{32}$ attempts are needed to find a block whose hash output is same as the original block. This scale of computation is potentially feasible using present day's technology. For this reason, the HMAC of length 128 bits is used in the proposed method to attain cryptographic security. The probability of

false authentication in the proposed method is approximately equal to $2^{-128}$, which is negligibly small. The smallest block size for embedding the 128-bit HMAC and 16-bit image index would be about 12×12 pixels; thus the localization accuracy is equal to this block size. In [86], the chosen block size is also 12×12 pixels for the 128-bit signature and the block size in [101] is 10×10 pixels for the 64-bit signature. For this method, to accommodate the 128-bit signature and the payload of higher hierarchies, the block size would be greater than 12×12 pixels. Cropping is one of the image manipulations in which the watermark detection method fails to authenticate regions due to the loss of synchronization of block boundaries. To detect cropping, a sliding-window search is utilized to regain synchronization with the block boundaries in [101]. This search method can also be used in the proposed method to synchronize block boundaries in the cropped image. In [101], it has been analyzed that the computational complexity of a localization method depends on the number of signature operations which is equal to total number of blocks in the image. In the proposed method, additional processing is necessary to compute the authenticity measure of the fake image. In our implementation using the MATLAB software, both the embedding and the detection processes are performed approximately within 43 seconds for the image size of 300×300 pixels. The authenticity measure computation is performed within 2 seconds approximately.

## 4.5 Application in Binary Document Image Authentication

Since a localization method in binary document image authentication uses block-wise independent watermarks, the Holliman-Memon attack is also applicable in this case. In the previous chapter, a new method for localization in binary document images is proposed based on the erasable watermarking approach. A unique image index was

used for computing the signature in each block to resist this attack. To solve the issue of image index estimation for the binary image case, the proposed localization method is modified as follows:

## Embedding:

1. The original image $X$ is divided into non-overlapping blocks of $Y \times Z$ pixels. Watermarking is performed for each block independently and in a sequential order starting from left to right and top to bottom of the image.
2. In each block, an ordered set of insignificant pixels are searched in a sequential scanning order as described by Conditions 1, 2 and 3 in Section 3.2.2. The insignificant pixel set is losslessly compressed using the run-length coding scheme. Let the compressed data be denoted as $C_D$.
3. Authentication signature $A_S$ is computed from the block according to the following equation:

$$A_S = H(C_b, K, I_b, I_K) \tag{4.5}$$

where, $H$, $C_b$, $K$, $I_b$ and $I_K$ denote hash function, current block in the original image, secret key, block index and image index, respectively.

4. The message $m$ which is embedded in the insignificant pixel set producing the watermarked block is constructed. There are three parts in $m$: (1) the compressed data $C_D$; (2) 16-bit image index $I_K$; and (3) the authentication signature $A_S$. The structure of the message $m$ is shown in Figure 4.12.
5. The embedding is performed pixel-wise; so an insignificant pixel holds one bit of $m$ and its pixel value is set equal to the signature bit it holds. Likewise all blocks in the image are watermarked.

Detection:

1. To verify each block in the test image $X'$, the message $m'$ is extracted by finding the insignificant pixel set similar to the embedding process. Its component pieces, the compressed version of the insignificant pixel set $C_D'$, 16-bit image index $I_K'$ and the authentication signature $A_S'$ are extracted. The compressed version of the insignificant pixel set together with the current block is used to reconstruct the block $C_b'$.

2. The authentication signature $A_S''$ of the reconstructed block is computed as follows and compared with the extracted signature as follows:

$$A_S'' = H(C_b', K, I_b, I_K'). \qquad (4.6)$$

3. A matrix $(R)$ of *(M/Y)* rows and *(N/Z)* columns is constructed while computing $A_S''$ for each block. Each entry of $R$ represents a particular block in image $X'$. The magnitude of an entry in $R$ is 1 if $A_S'$ and $A_S''$ match in the corresponding block; otherwise it is equal to 0.

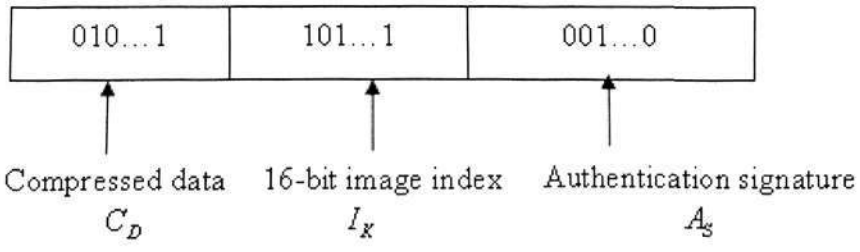



Figure 4.12: The structure of message *m* consisting of three parts.

4. Different image indices can be extracted from all the blocks in $X'$. Let each such index be termed as the candidate image index ($I_X{}^T$). For every $I_X{}^T$, the authenticity score ($A_S$) is computed by applying the image index estimation algorithm described in Section 4.3.
5. The candidate image index with the highest authenticity score is chosen to be the estimated image index. Let the estimated image index be denoted as $I_E$. A block in the test image will be declared as authentic if the following conditions are satisfied:
    - The image index extracted from the block ($I_K{}'$) is equal to the estimated image index ($I_E$).
    - The corresponding entry in $R$ for the block is 1.
6. The authenticity measure ($A_M$) of the test image is defined to quantify the attack severity and its value is equal to the authenticity score of $I_E$. The maximum value of $A_M$ is 1 when all blocks in the test image are authenticated and 0 when all blocks in the test image are not authenticated.

### 4.5.1 Results and Discussions

From an attacker's perspective, constructing a fake image by using the Holliman-Memon attack in binary document images is easier than the case of grayscale images. If suitable blocks containing required symbols and characters are found in a database of binary images, the visual quality of the fake image would be excellent. In this section, we presented simulation results by implementing the modified localization method as described above. We consider two binary document images for demonstrating the effectiveness of the proposed method against the Holliman-Memon

attack. In our implementation, we have used MD5 hash function to compute the HMAC. The output 128-bit HMAC is used as the authentication signature and the message $m$ is constructed for each block as described in the proposed method. The value of $M$ is chosen to be 5 for achieving high watermark capacity and correct watermark detection. For both images, the message $m$ is embedded within each block of size 40×40 pixels. The first original image of size 480×520 pixels and its corresponding watermarked image are shown in Figure 4.13. Similarly, the second original image 480×520 pixels and its corresponding watermarked image are shown in Figure 4.14. While computing the authentication signature, the 16-bit image indices with the decimal equivalent of 48056 and 56273 were used for the first and second images respectively. The fake image was constructed from two watermarked images as follows. The equation in the last rows of the second watermarked image was added into the first watermarked image. For this purpose, 8 blocks with the block indices of 146 to 153 in the first watermarked image were replaced with the corresponding blocks in the second watermarked image. To illustrate, the block having the index of 146 in the first watermarked image was replaced with the block of index 146 from the second watermarked image. Similarly, the block replacing operation was performed for all 8 blocks and the fake image is shown in Figure 4.15. According to the principle of Holliman-Memon attack, the blocks between the two watermarked images are swapped at identical positions; thus each block of the fake image should be authenticated. The proposed detection method was used to verify each block in the fake image and the result is shown in Figure 4.16. A total of 8 blocks out of 156 blocks in the fake image were verified to be inauthentic. The estimated image index is 48056 and the authenticity measure of the fake image is approximately 0.9.

## 4.6 Summary

In this chapter, we proposed a new method in authentication watermarking to resist the Holliman-Memon attack. In this method, a unique image index was used while computing the signature and embedded in the least significant bit-plane to minimize visual distortion. By using side information about the watermarked image, the informed detector estimated the correct image index from the fake image. Authenticity of each block in the fake image was verified without any loss or ambiguity in localization. The localization accuracy of the proposed method remained at the chosen block size as compared to Wong's scheme. The authenticity measure was defined to quantify the attack severity and the number of authentic blocks in the fake image depended on its magnitude. The attacker's task to generate the fake image became more difficult with the inclusion of the authenticity measure. All blocks in the fake image would be verified only if it was exactly the same as one of the watermarked images in the database. The proposed countermeasure is also shown to be effective against the Holliman-Memon attack in binary document image authentication.

The authors apply this technique to small 8×8 pixel blocks. The block is DCT transformed, and the frequency masking values $M(i,j)$ for each frequency bin $P(i,j)$ are calculated using a frequency masking model. The values $M(i,j)$ are the maximal changes that do not introduce perceptible distortions. The DCT coefficients are modified to $P_S(i,j)$ according to the following expression

$$P_S(i,j) = M(i,j)\{\lfloor P(i,j)/M(i,j)\rfloor + r(i,j)\,\mathrm{sign}(P(i,j))\},$$

where $r(i,j)$ is a key-dependent noise signal in the interval (0,1), and $\lfloor x \rfloor$ rounds $x$ towards zero. Since $|P(i,j)-P_S(i,j)| \le M(i,j)$, the modifications to DCT coefficients are imperceptible.

For a test image block with DCT coefficients $P_S(i,j)$, the masking values $M'(i,j)$ are calculated. The error at $(i,j)$ is estimated by the following equation

(a)

The authors apply this technique to small 8×8 pixel blocks. The block is DCT transformed, and the frequency masking values $M(i,j)$ for each frequency bin $P(i,j)$ are calculated using a frequency masking model. The values $M(i,j)$ are the maximal changes that do not introduce perceptible distortions. The DCT coefficients are modified to $P_S(i,j)$ according to the following expression

$$P_S(i,j) = M(i,j)\{\lfloor P(i,j)/M(i,j)\rfloor + r(i,j)\,\mathrm{sign}(P(i,j))\},$$

where $r(i,j)$ is a key-dependent noise signal in the interval (0,1), and $\lfloor x \rfloor$ rounds $x$ towards zero. Since $|P(i,j)-P_S(i,j)| \le M(i,j)$, the modifications to DCT coefficients are imperceptible.

For a test image block with DCT coefficients $P_S(i,j)$, the masking values $M'(i,j)$ are calculated. The error at $(i,j)$ is estimated by the following equation

(b)

Figure 4.13: (a) Original binary document image of size 480×520 pixels, (b) watermarked image after embedding the authentication signature in each block.

$Q_{\Delta,1}(f) = 0$ if $\lfloor f/(\Delta 2^l) \rfloor$ is even,
$Q_{\Delta,1}(f) = 1$ if $\lfloor f/(\Delta 2^l) \rfloor$ is odd

at the quantization level $l$. If a wavelet coefficient $f_{k,l}(m,n)$ is chosen for watermark embedding, it is modified so that

$$Q_{\Delta,s}(f_{k,l}(m,n)) = w(\iota) \text{ XOR } qkey(m,n),$$

where $w(\iota)$ is the $\iota$-th watermark bit and $qkey$ is a bit generated from the image and a secret key. The construction of the quantization function $Q$ guarantees that one will never have to modify the coefficient at the level $l$ by more than $\pm\Delta 2^l$. The watermark is extracted by evaluating the expression

$$w(\iota) = Q_{\Delta,s}(f'_{k,l}(m,n)) \text{ XOR } qkey(m,n),$$

(a)

$Q_{\Delta,1}(f) = 0$ if $\lfloor f/(\Delta 2^l) \rfloor$ is even,
$Q_{\Delta,1}(f) = 1$ if $\lfloor f/(\Delta 2^l) \rfloor$ is odd

at the quantization level $l$. If a wavelet coefficient $f_{k,l}(m,n)$ is chosen for watermark embedding, it is modified so that

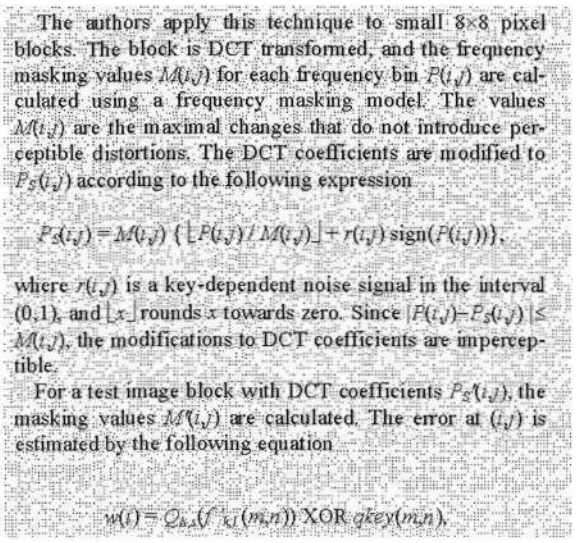
$$Q_{\Delta,s}(f_{k,l}(m,n)) = w(\iota) \text{ XOR } qkey(m,n),$$

where $w(\iota)$ is the $\iota$-th watermark bit and $qkey$ is a bit generated from the image and a secret key. The construction of the quantization function $Q$ guarantees that one will never have to modify the coefficient at the level $l$ by more than $\pm\Delta 2^l$. The watermark is extracted by evaluating the expression

$$w(\iota) = Q_{\Delta,s}(f'_{k,l}(m,n)) \text{ XOR } qkey(m,n),$$

(b)

Figure 4.14: (a) Original binary document image of size 480×520 pixels, (b) watermarked image after embedding the authentication signature in each block.



The authors apply this technique to small 8×8 pixel blocks. The block is DCT transformed, and the frequency masking values $M(i,j)$ for each frequency bin $P(i,j)$ are calculated using a frequency masking model. The values $M(i,j)$ are the maximal changes that do not introduce perceptible distortions. The DCT coefficients are modified to $P_S(i,j)$ according to the following expression

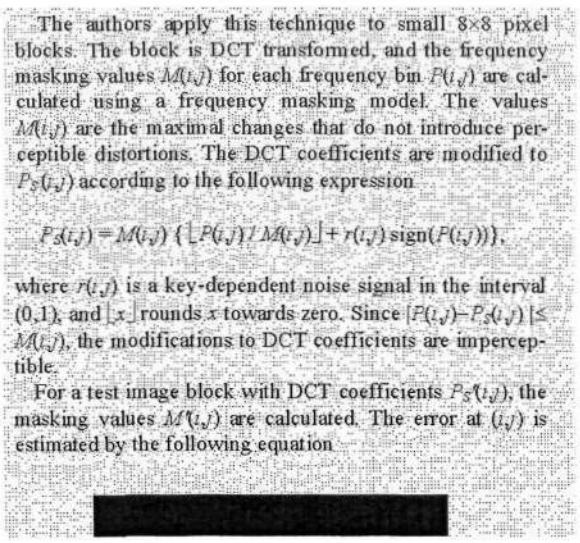
$$P_S(i,j) = M(i,j)\,(\lfloor P(i,j)/M(i,j) \rfloor + r(i,j)\,\mathrm{sign}(P(i,j)))$$

where $r(i,j)$ is a key-dependent noise signal in the interval (0,1), and $\lfloor x \rfloor$ rounds $x$ towards zero. Since $|P(i,j)-P_S(i,j)| \leq M(i,j)$, the modifications to DCT coefficients are imperceptible.

For a test image block with DCT coefficients $P_S'(i,j)$, the masking values $M'(i,j)$ are calculated. The error at $(i,j)$ is estimated by the following equation

$$w(i) = Q_{\Delta,x}(f'_{x1}(m,n)) \text{ XOR } qkey(m,n),$$

Figure 4.15: The fake image constructed using the Holliman-Memon attack.



The authors apply this technique to small 8×8 pixel blocks. The block is DCT transformed, and the frequency masking values $M(i,j)$ for each frequency bin $P(i,j)$ are calculated using a frequency masking model. The values $M(i,j)$ are the maximal changes that do not introduce perceptible distortions. The DCT coefficients are modified to $P_S(i,j)$ according to the following expression

$$P_S(i,j) = M(i,j)\,(\lfloor P(i,j)/M(i,j) \rfloor + r(i,j)\,\mathrm{sign}(P(i,j)))$$

where $r(i,j)$ is a key-dependent noise signal in the interval (0,1), and $\lfloor x \rfloor$ rounds $x$ towards zero. Since $|P(i,j)-P_S(i,j)| \leq M(i,j)$, the modifications to DCT coefficients are imperceptible.

For a test image block with DCT coefficients $P_S'(i,j)$, the masking values $M'(i,j)$ are calculated. The error at $(i,j)$ is estimated by the following equation

[redacted]

Figure 4.16: Detection output after verifying the fake image using the proposed method. Dark region in the image shows the inauthentic blocks.

# Chapter 5

# Conclusions and Future Work

This thesis presents various aspects of digital watermarking in content authentication application. We have proposed new methods that can be useful for exact authentication, localization and restoration purpose in binary document and grayscale images. Towards the goal of identifying low-distortion pixels in binary document images, a new *CWDD* perceptual model is designed. To detect any modifications to the watermarked image including the parity attack, an exact authentication method is proposed by applying the reversible property of the *CWDD* model. Due to simple pixel statistics of binary document images, it is difficult to find a large number of low-distortion pixels within a block of reasonable size. To meet the challenge of embedding a cryptographic signature of sufficient length within each block of the image, a new localization method is designed by constructing the erasable watermark. The method achieves high security after introducing the erasable distortion in the watermarked image. Then, a new restoration method is designed such that it is possible to extract the original character sequence after tamper localization in text document images. Against multiple attacks such as the block swapping, character insertion, deletion and substitution, it is possible to achieve effective restoration capability using the error-control coding technique. The proposed methods achieve security against various modifications that is equivalent to the cryptographic authentication. To withstand the Holliman-Memon attack in binary document and grayscale image authentication, a new image index estimation algorithm is proposed. The correct extraction of the image index by taking the authenticity and connectivity

among the blocks during blind detection foils any attempt by the attacker to authenticate the fake image.

In the thesis, the low-distortion contour pixels and the isolated noise pixels are used for different applications. In Chapter 2, the low-distortion contour pixels are used for exact authentication while in Chapter 3, the isolated noise pixels are used for localization and restoration. As discussed, the proposed exact authentication algorithm falls under the category of non-erasable watermarking while the localization and restoration algorithms are in the erasable watermarking category. We discuss in following why two types of pixels are used for different problems addressed in the thesis. Similar to the low-distortion contour pixels, the isolated noise pixels can also be used for the non-erasable exact authentication since they both allow blind detection. However, the watermarked image in this case will suffer from visible distortion. For this reason, the isolated noise pixels are not considered for the non-erasable exact authentication in Chapter 2. Instead, the low-distortion contour pixels are used for this purpose. The isolated noise pixels can be used for the erasable exact authentication application. In this case, the proposed localization algorithm in Chapter 3 can be used to design and embed the watermark in the whole image instead of each block. However, there will be visual distortion in the watermarked image and it is necessary to erase the watermark at the detector additionally.

For localization and restoration application, the low-distortion contour pixels are not considered due to the following reason: Within a reasonable block-size, there is an insufficient number of low-distortion pixels available for embedding. The blind detection requirement of these pixels adds to the difficulty in achieving sufficient

watermark capacity in each block. Further, the low-distortion contour pixels are not available for watermark embedding in white regions of the image. The inability to watermark in the white regions makes the detector vulnerable to malicious tampering. Thus, it is evident that unless the block size is large, the low-distortion contour pixels may not be suitable to embed the authentication signature for localization and restoration. The low-distortion contour pixels are not suitable for erasable watermarking. Erasable watermarking method achieves high capacity in natural images due to the fact that its neighboring pixels are highly correlated. This leads to correlations between neighboring bits within a bit plane. Thus some bit planes in the whole image can be sufficiently compressed to implement an erasable watermark. For binary document images, it has been found that the correlation among the low-distortion contour pixels is low. So it is difficult to compress them losslessly and obtain sufficient information space to embed a watermark.

From the above discussion it can be summarized that it is possible to use the isolated noise pixels for authentication problems in Chapter 2 and 3. On the other hand, the low-distortion contour pixels are not suitable for localization and restoration application. The criterion for point selection is low impact on visual perception and blind detection, independent of the purposes of the watermark. A question arises if it is possible to design generic point selection schemes that can be applied to different contexts. In other words, is it plausible to simultaneously use both types of pixels for embedding the watermark? In non-erasable watermarking case, there exists a possibility to use both types of pixels for embedding a watermark. This is because both types of pixels allow blind detection. However, the visual quality of the watermarked image will be degraded as compared to the case of using only low-

distortion contour pixels. In erasable watermarking case, blind detection is possible if both types of pixels are used for embedding; however in this case watermark capacity will be reduced. As discussed in Chapter 3, high watermark capacity is achieved in using isolated noise pixels because the white pixels occur in long sequences and black pixels occur in between them with less probability. Among such a set of pixels, there exists a high correlation which is exploited during lossless compression. If only low-distortion contour pixels are used along with only isolated noise pixels, the increasing occurrence of black contour pixels will reduce the correlation in the pixel set. So the information space available for embedding due to compression will be reduced. To design a new erasable watermarking method, which can achieve high watermark capacity even when the correlation in the pixel set is low, constitutes an interesting line for future research.

Information theoretic approaches focus mainly on the theoretical analysis of watermarking systems [104]. They deal with abstract mathematical models for watermark encoding, attacks and decoding. These models enable studying watermarks at a high level without resorting to any specific application such as image authentication. Therefore, the results obtained by using these techniques are potentially useful in a wide variety of applications by suitably mapping the application to an information theoretic model. Information theoretic methods have been successfully applied to information storage and transmission [105]. Here, messages and channels are modeled probabilistically, and their properties are studied analytically to estimate the capacity of various channels, i.e. the maximum amount of information that can be transmitted through a channel so that decoding with a small probability of error is possible. Using the analogy between communication and

watermarking channels, it is possible to compute fundamental information-carrying capacity limits of watermarking channels using information theoretic analysis.

In [106], a relevant and interesting discussion has been given on the theoretical aspect of the authentication problem. In general data hiding problems, the decoder knows that one of $|M|$ possible messages is embedded in the data, and attempts to reliably decode the message. However, the problem is quite different during signature verification when the receiver must perform the simpler binary decision: Is the received signal marked using a given signature $m \in M$ or not? To detect any tampering of the data, a fragile watermarking technique is often used. An example of fragile watermarking would be a LSB method in which the LSB plane is a signature known to the detector, and the detector declares an error even if one bit in the LSB plane has been modified. To analyze more general signature verification problems involving admissible attacks (e.g., transmission noise and/or desynchronization operations), an appropriate class of channels $p_{Y|X}$ is defined. The receiver has access not just to the degraded data $y$ and the key $k$, but also to the signature $m \in M$. The decoding function is a binary decision rule $g(y,k,m)$ taking values in $\{0,1\}$ and indicating the absence or presence of the tested signature, respectively. The challenge in a signature verification problem is to design a good embedding code. During detection test, the output of the decoder is compared with a predetermined threshold. By varying the threshold, the receiver operating characteristics (ROC) giving the probability of true positives versus the probability of false positives is obtained.

The fundamental limits of signature verification schemes with distortion constraints have been studied by Steinberg and Merhav [107]. They proved that the detection

problem is easier than the full decoding problem due to the small size of the decision space. In the analysis, they assumed a class of distortion-constrained memoryless channels for modeling the attacks. In [108], the application of a recently developed quantization based watermarking scheme to image authentication has been investigated. To prevent image manipulations and fraudulent use of modified images, the embedding of semi-fragile digital watermarks has been proposed. The watermark should survive modifications introduced by random noise or compression, but should not be detectable from non-authentic regions of the image. The proposed method allows reliable blind watermark detection from a small number of pixels, and thus enables the detection of local modifications to the image content. Using the proposed method, it was demonstrated that authentication of the compressed data has low error probabilities. In [109], a formulation for the general problem of authentication with a semantic model has been proposed. The model is particularly useful for the cases in which the content may have to undergo a variety of legitimate editing or distortions prior to authentication. Under the proposed model, the general authentication problem is different from other authentication methods such as self-embedding and fragile watermarking. In the new method, the editor is constrained according to a reference channel model that can be freely chosen independently of any semantic model. The proposed method has been improved as compared to the robust and fragile watermarking techniques in terms of security.

The methods we have described above deal with general authentication problem from an information theoretic perspective. The main aim of such analysis is to find the capacity of a technique, i.e. the number of information bits that can be the conveyed reliably for a given class of attacks. The above authentication methods find the

capacity under the condition that the watermarking technique is insensitive to a set of legitimate attacks or distortions while detecting all other cases. So, these methods are particularly applicable to semi-fragile authentication watermarking techniques. In the thesis, we are designing the authentication watermarking techniques using cryptographic signatures as the fragile watermark. The embedded watermark is designed such that it can detect every possible alteration to the content. So the above information-theoretic analysis methods is not applicable to the proposed techniques. As discussed previously in Chapter 2 and 3, the proposed fragile watermarking methods can detect any kind of alterations with low false positive probability. To achieve maximum fragility, the comparison between embedded and extracted watermark sequences is done on a bit-by-bit basis. So, the detection threshold for the proposed methods is equal to the size of the embedded authentication signature. The capacity of the proposed authentication watermarking methods is limited by the amount of perceptual distortion during the embedding process, since the watermark is not expected to tolerate any class of attacks. It is an important future research work to design information theoretic analysis methods for semi-fragile authentication of binary document images. In this direction, Voloshynovskiy *et al* reports some first theoretical results in semi-fragile authentication watermarking of electronic and printed document images using robust hashing [110].

In [111], a theoretical analysis for reversible data hiding method has been presented. Reversible data hiding is achieved by lossless compression of a correlated pixel set, then concatenating the compressed bit stream with auxiliary data and replacing the original set. Maximum information space will be obtained when the pixel set is compressed at its entropy rate. Thus, the capacity is dependent on the type of

compression scheme being used in the watermarking method. In our proposed method, we have used the run-length coding compression scheme to achieve low complexity though the run-length coding scheme does not achieve optimum compression. A better capacity can be achieved using a sophisticated compression scheme; however this will increase the implementation time. In the paper, it has been shown that the above described approach of compression and bit replacement is not optimal for Bernoulli binary sources. A new reversible method has been proposed for Bernoulli binary sources which outperforms the existing method in [93] in terms of capacity. However, the proposed method is designed for simple binary sources and is not applicable for complex signals such as image, video and text. The reversible schemes considered in previous papers have a fragile nature: in those schemes, changing a single bit in the watermarked data would prohibit recovery of both the original host signal as well as the embedded auxiliary data. In [112], theoretical capacity limit of a robust reversible data-hiding method have been analyzed. A practical code construction algorithm is suggested in the paper that outperforms the classical time-sharing solution for a simple Bernouilli binary source.

In terms of future work, we think that understanding of human visual system characteristics to perceive distortion in binary images is a significant research area. The methods including ours, discussed in the thesis for visual distortion modeling for binary images are based on a heuristic approach. It will be interesting to pursue this line of research such that a model can be designed similar to Watson's just noticeable difference (JND) model [17]. For this purpose, experimental study of human vision's response to binary image distortion is necessary. Comparative study of the proposed *CWDD* model with other perceptual models for watermarking and other applications

like compression will be pursued. The proposed localization method introduces erasable visual distortion in the watermarked image. Enhancement study of the proposed method will be performed to reduce the distortion level. The proposed restoration method will be extended such that it can be applied to binary document images in general. The methods designed in the thesis use fragile watermarks; so the watermarks cannot survive innocuous manipulations such as print and scan, noise addition and geometrical alterations. A less number of semi-fragile watermarking techniques are proposed in case of binary images as compared to grayscale images. Recently, various semi-fragile techniques in binary images are suggested in literature [113, 114, 115]. However, the potential to apply digital watermarking for content authentication application in print or analogue media is yet to be fully realized. The perspective of identifying and estimating the distortions after watermarking process (known as the telltale watermarking) is important in binary document and grayscale image authentication. In another research direction, security of the embedded watermark in digital data is an important issue and it is being actively pursued by researchers. The design of novel methods to achieve watermark security can make digital watermarking useful for various practical applications.

# Bibliography

[1] I. J. Cox, Matthew L. Miller, and Jeffrey A. Bloom, "Digital Watermarking," Morgan Kaufmann Publishers Inc., San Francisco, 2001.

[2] M. D. Swanson, M. Kobayashi and A. H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies, " *Proc. of the IEEE*, vol. 86, no. 6, pp. 1064-1087, June 1998.

[3] F. Hartung and M. Kutter, "Multimedia Watermarking Techniques," *Proc. of the IEEE*, vol. 87, no. 7, pp. 1079 -1107, July 1999.

[4] I. J. Cox and M. L. Miller, "A Review of Watermarking and the Importance of Perceptual Modeling," *Proc. SPIE*, vol. 3016, pp. 92-99, Feb. 1997.

[5] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual Watermarks for Digital Images and Video," *Proceedings of the IEEE,* vol. 87, no. 7, pp. 1108-1126, July 1999.

[6] M. Barni and F. Bartolini, "Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications," Marcel Dekker Inc., 2004.

[7] R. van Schyndel, A. Z. Tirkel and C. F. Osborne, "A Digital Watermark," *IEEE International Conference on Image Processing*, vol. II, pp. 86-90, Austin, USA, 1994.

[8] W. Bender, D. Gruhl and N. Morimoto, "Techniques for Data Hiding," *IBM Systems Journal*, vol. 35, 1996.

[9] I. Pitas and N. Nikolaidis, "Copyright Protection of Images Using Robust Digital Signatures," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 2168-2171, May 1996.

[10] E. Koch and J. Zhao, "Towards Robust and Hidden Image Copyright Labeling," *Proc. IEEE Workshop on Nonlinear Signal and Image Processing*, pp. 452-455, Greece, 1995.

[11] M. D. Swanson, B. Zhu and A. H. Tewfik, "Transparent Robust Image Watermarking," *IEEE International Conference on Image Processing*, vol. 3, pp. 211-214, 1996.

[12] M. D. Swanson, B. Zhu and A. H. Tewfik, "Robust Data Hiding for Images," *IEEE Digital Signal Processing Workshop*, pp. 37-40, Norway, 1996.

[13] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, pp. 1673-1687, 1997.

[14] Mauro Barni, Franco Bartolini, Vito Cappellini and Alessandro Piva, "A DCT-Domain System for Robust Image Watermarking," *Signal Processing*, vol. 66, pp. 357-372, 1998.

[15] Adrian G. Bors and Ioannis Pitas, "Image Watermarking Using DCT Domain Constraints," *IEEE International Conference on Image Processing*, vol. 3, pp. 231-234, Switzerland, 1996.

[16] C. I. Podilchuk and W. Zeng, "Image-Adaptive Watermarking Using Visual Models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 525-539, 1998.

[17] A. B. Watson, "DCT Quantization Matrices Visually Optimized For Individual Images," *Proc. SPIE Conf. Human Vision, Visual Processing and Digital Display IV*, vol. 1913, pp. 202–216, 1993.

[18] A. Sinha, "Digital Watermarking: A Dynamic Bit Allocation Based Approach," *SPCOM 2001: 6th Biennial Conference Proceedings*, pp. 49-56, India, 2001.

[19] J. Fridrich, "Combining Low-frequency and Spread Spectrum Watermarking," *SPIE International Symposium on Optical Science, Engineering, and Instrumentation*, pp. 203-212, San Diego, 1998.

[20] Deepa Kundur and Dimitrios Hatzinakos, "A Robust Digital Image Watermarking Method using Wavelet-Based Fusion," *IEEE International Conference on Image Processing*, vol. 1, pp. 544-547, Santa Barbara, 1997.

[21] H. M. Wang, P. C. Su and C. C. Jay Kuo, "Wavelet Based Digital Image Watermarking," *Optics Express*, vol. 3, no. 12, pp. 491-496, 1998.

[22] X. G. Xia, C. G. Boncelet, and G. R. Arce, "Wavelet Transform Based Watermark for Digital Images," *Optics Express*, vol. 3, no. 12, pp. 497-511, 1998.

[23] A. V. Oppenheim and J. S. Lim, "The Importance of Phase in Signals," *Proc. of IEEE*, vol. 69, no. 5, pp. 529-541, 1981.

[24] A. V. Oppenheim, J. S. Lim, G. Kopec and S. C. Pohlig, "Phase in Speech and Pictures," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 27, pp. 632-637, 1979.

[25] J. J. K. O. Runnaidh, W. J. Powling and F. M. Boland, "Phase Watermarking Of Digital Images," *IEEE International Conference on Image Processing*, vol. 3, pp. 239-242, Switzerland, 1996.

[26] J. J. K. O. Runnaidh and T. Pun, "Rotation, Scale and Translation Invariant Spread Spectrum Digital Image Watermarking," *Signal Processing*, vol. 66, no. 3, pp. 303-317, 1998.

[27] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner and T. Pun, "A Stochastic Approach to Content Adaptive Digital Image Watermarking," *Proc. of the Third International Workshop on Information Hiding, Lecture Notes in Computer Science,* vol. 1768, pp. 211-236, 1999.

[28] J. S. Lim, "Two Dimensional Signal and Image Processing," *Englewood Cliffs*, NJ, Prentice-Hall, 1990.

[29] P. Moulin and J. Liu, "Analysis of Multiresolution Image Denoising Schemes Using Generalized- Gaussian Priors," *Proc. IEEE Signal Processing Symposium on Time-Freq. And Time-Scale Analysis*, pp. 909-919, 1998.

[30] M. Ramkumar and A. N. Akansu, "Image Watermarks, and Counterfeit Attacks: Some problems and Solutions," *Proc. of Content Security and Data Hiding in Digital Media*, USA, pp. 102-112, 1999.

[31] S. Craver, N. Memon, B. L. Yeo and M. M. Yeung, "Resolving Rightful Ownership with Invisible Watermarking Techniques: Limitations, Attacks and Implications," *IEEE Journal of Selected Areas in Communications*, vol. 16, no. 4, pp.573-586, 1998.

[32] Z. M. Lu and S. H. Sun, "Digital Image Watermarking Technique Based on Vector Quantization," *Electronics Letters*, vol. 36, no. 4, pp. 303–305, 2000.

[33] Z. M. Lu, J. S. Pan and S. H. Sun, "VQ-Based Digital Image Watermarking Method," *Electronics Letters*, vol. 36, no. 14, pp. 1201–1202, 2000.

[34] Y. Linde, A. Buzo and R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[35] S. S. Selvi and A. Makur, "A Review of Variable Dimension Vector Quantizers and their Applications to Speech and Image Coding," *Electro Technology, Journal of Society of Electronic Engineers*, vol. 41, no. 3 & 4, pp. 38-70, 1997.

[36] A. Makur and S. S. Selvi, "Variable Dimension Vector Quantization Based Image Watermarking," *Signal Processing*, vol. 81, no. 4, pp. 889-893, 2001.

[37] S. H. Low, N. F. Maxemchuk and A. M. Lapone, "Document Identification for Copyright Protection Using Centroid Detection," *IEEE Trans. on Communication,* vol. 46, no. 3, pp. 372-383, 1998.

[38] N. F. Maxemchuk and S. H. Low, "Marking Text Documents," *Proc. IEEE International Conference on Image Processing,* Santa Barbara, 1997.

[39] S. H. Low and N. F. Maxemchuk, "Performance Comparison of Two Text Marking Methods," *IEEE Journal on Selected Areas in Communications,* vol. 16, no. 4, pp. 561-572, 1998.

[40] J. T. Brassil, S. Low and N. F. Maxemchuk, "Copyright Protection for the Electronic Distribution of Text Documents," *Proc. of the IEEE (Invited Paper)*, vol. 87, no. 7, pp. 1181-1196, 1999.

[41] J. Brassil and L. O'Gorman, "Watermarking Document Images with Bounding Box Expansion," *Proc. 1st International Workshop on Information Hiding,* pp. 227-235, Cambridge, UK, 1996.

[42] N. Chotikakamthorn, "Document Image Data Hiding Techniques Using Character Spacing Width Sequence Coding," *Proc. IEEE Intl. Conference on Image Processing,* pp. 250-254, Japan, 1999.

[43] D. Huang and H. Yan, "Interword Distance Changes Represented by Sine Waves for Watermarking Text Images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1237-1245, 2001.

[44] M. Wu, E. Tang, and B. Liu, "Data Hiding in Digital Binary Images," *Proc. IEEE International Conference on Multimedia and Expo,* New York, 2000.

[45] Min Wu, Bede Liu, "Data hiding in binary image for authentication and annotation," *IEEE Transactions on Multimedia*, vol. 6, no. 4, pp. 528-538, 2004.

[46] M. Wu and B. Liu, "Digital Watermarking Using Shuffling," *IEEE International Conference on Image Processing,* Kobe, Japan, 1999.

[47] E. Koch and J. Zhao, "Embedding Robust Labels into Images for Copyright Protection," *Proc. International Congress on Intellectual Property Rights for Specialized Information, Knowledge & New Technologies,* Vienna, 1995.

[48] Q. Mei, E. K. Wong and N. Memon, "Data Hiding in Binary Text Documents," *SPIE Proc Security and Watermarking of Multimedia Contents III*, San Jose, 2001.

[49] T. Amamo and D. Misaki, "Feature Calibration Method for Watermarking of Document Images," *Proc. 5th Int'l Conf on Document Analysis and Recognition*, pp. 91-94, Bangalore, India, 1999.

[50] A. K. Bhattacharjya and H. Ancin, "Data Embedding in Text for a Copier System," *Proc. IEEE International Conference on Image Processing,* vol. 2, pp. 245-249, 1999.

[51] H. Lu, J. Wang, Alex C. Kot, and Yun Q. Shi, "An Objective Distortion Measure for Binary Document Images Based on Human Visual Perception," *Proc. Int. Conference on Pattern Recognition*, vol. 4, pp. 239-242, Canada, 2002.

[52] H Lu, A C Kot and Y Shi, "Distance-Reciprocal Distortion Measure for Binary Document Images", *IEEE Signal Processing Letter*, Vol. 11, No. 2, pp. 228-231, February 2004.

[53] H. Lu, X. Shi, Y.Q. Shi, A.C. Kot and L. Chen, "Watermark Embedding in DC Components of DCT for Binary Images," *IEEE Workshop on Multimedia Signal Processing*, pp. 300-303, 2002.

[54] H. Lu, A. C. Kot and J. Cheng, "Secure Data Hiding in Binary Document Images for Authentication," *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 3, pp. 806-809, 2003.

[55] H. Yang and A. C. Kot, "Data Hiding for Bi-level Documents Using Smoothing Techniques," *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 5, pp. 692-695, 2004.

[56] H. Yang and A. C. Kot, "Text Document Authentication by Integrating Inter Character and Word Spaces Watermarking," *IEEE International Conference on Multimedia and Expo. (ICME),* 2004.

[57] Z. Baharav and D. Shaked, "Watermarking of Dither Half-toned Images," *Proc. of SPIE Security and Watermarking of Multimedia Contents,* vol. 1, pp. 307-313, 1999.

[58] H-C A. Wang, "Data Hiding Techniques for Printed Binary Images," *International Conference on Information Technology: Coding and Computing,* pp. 55-59, 2001.

[59] M. S. Fu and O. C. Au, "Data Hiding in Halftone Images by Stochastic Error Diffusion," *IEEE Int. Conference* Acoustics, *Speech and Signal Processing*, 2001.

[60] M. S. Fu and O. C. Au, "Data Hiding by Smart Pair Toggling for Halftone Images," *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, pp. 2318-2321, 2000.

[61] M. S. Fu and O. C. Au, "Data Hiding Watermarking for Halftone Images," *IEEE Transactions On Image Processing*, vol. 11, no. 4, pp. 477- 484, 2002.

[62] M. S. Fu and O. C. Au, "Improved Halftone Image Data Hiding with Intensity Selection," *IEEE International Symposium on Circuits and Systems (ISCAS),* vol. 5, pp. 243-246, 2001.

[63] M. Chen, E. Wong, N. Memon and S. Adams, "Recent Developments in Document Image Watermarking and Data Hiding," *Proc. SPIE Multimedia Systems and Applications IV*, vol. 4518, pp.166-176, 2001.

[64] R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Commun. ACM.* vol. 21, pp.120-126, 1978.

[65] M. Holliman and N. Memon, "Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes," *IEEE Trans. Image Processing*, vol. 9, no. 3, pp. 432-441, March 2000.

[66] Yeung M. M. and Mintzer F., "An Invisible Watermarking Technique for Image Verification," *Proc. ICIP*, Santa Barbara, California, 1997.

[67] Coppersmith, D., Mintzer, F., Tresser, C., Wu, C. W. and Yeung, M. M., "Fragile Imperceptible Digital Watermark with Privacy Control," *Proc. SPIE, Security and Watermarking of Multimedia Contents*, pp. 79-84, San Jose, California, 1999.

[68] Walton, S, "Information Authentication for a Slippery New Age," *Dr. Dobbs Journal,* vol. 20, no. 4, pp. 18–26, 1995.

[69] Friedman, G. L., "The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image," *IEEE Transactions on Consumer Electronics,* vol.39, no.4, pp. 404-408, 1993.

[70] P. Wong, "A Watermark for Image Integrity and Ownership Verification," *Proc. IS&T PIC*, Portland, Oregon, 1998.

[71] H. Y. Kim and A. Afif, "Secure Authentication Watermarking for Binary Images," *Proc. Sibgraphi – Brazilian Symposium on Computer Graphics and Image Processing*, pp 199-206, 2003.

[72] H. Y. Kim and R. L. de Queiroz, "Alteration-Locating Authentication Watermarking for Binary Images," *Proc. Int. Workshop on Digital Watermarking 2004,* (Seoul), LNCS-2939, 2004.

[73] H Yang and A C Kot, "Data Hiding for Text Document Image Authentication by Connectivity Preserving," *IEEE ICASSP*, pp. 505-508, Philadelphia, March 2005.

[74] Huijuan Yang and Alex, C. Kot, "Pattern-Based Data Hiding for Binary Image Authentication by Connectivity-Preserving", *IEEE Transactions On Multimedia*, Vol. 9, No. 3, pp. 475-486, April 2007.

[75] R. C. Gonzalez, R. E. Woods, "Digital Image Processing," 2nd Edition, Prentice Hall.

[76] T. Pavlidis, "Algorithms for Graphics and Image Processing," Computer Science Press, Rockville, Maryland, 1982.

[77] Abeer George Ghuneim, "Tutorial on Contour Tracing," available at *http://www.cs.mcgill.ca/~aghnei/index.html*.

[78] D. Anderson, D. J. Sweeney and T. A. Williams, "Introduction to Statistics: An Applications Approach," West Publishing Company, 1981.

[79] R. L. Rivest, "RFC 1321: The MD5 Message-Digest Algorithm," *Internet Activities Board*, 1992.

[80] B. Schneier, "Applied Cryptography", John Wiley & Sons, 1996.

[81] Fridrich, J., Goljan, M. and Baldoza, A. C., "New Fragile Authentication Watermark for Images," Proc. *ICIP*, Vancouver, Canada, 2000.

[82] Memon, N., Shende, S. and Wong, P., "On the Security of the Yeung-Mintzer Authentication Watermark," *Proc. of the IS & T PICS Symposium*, Georgia, 1999.

[83] Fridrich, J., Goljan, M. and Memon, N., "Further Attacks on Yeung-Mintzer Fragile Watermarking Scheme," *Proc. SPIE, Security and Watermarking of Multimedia Contents*, pp. 428-437, San Jose, California, 2000.

[84] J. Fridrich, "Security of Fragile Authentication Watermarks with Localization," *Proc. SPIE*, vol. 4675, no. 75, 2002.

[85] P. W. Wong, "A Public Key Watermark for Image Verification and Authentication," *Proc. IEEE Int. Conference on Image Processing*, pp. 425-429, Chicago, Oct. 4-7, 1998.

[86] P.W. Wong and N. Memon, "Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification," *IEEE Trans. Image Processing*, vol. 10, no. 10, October 2001.

[87] J. Lee and C. S. Won, "Authentication and Correction of Digital Watermarking Images," *Electronics Letters*, pp. 886-887, vol. 35, no. 11, 1999.

[88] J. Lee and C. S. Won, "Image Integrity and Correction Using Parities of Error Control Coding," *IEEE International Conference on Multimedia and Expo,* pp. 1297-1300, vol. 3, 2000.

[89] J. Fridrich and M. Goljan, "Images with Self-correcting Capabilities," *IEEE International Conference on Image Processing*, pp. 792-796, vol. 3, 1999.

[90] D. Kundur and D. Hatzinakos, "Digital Watermarking for Telltale Tamper-Proofing and Authentication," *Proc. of the IEEE Special Issue on Identification and Protection of Multimedia Information*, vol. 87, no. 7, pp. 1167-1180, July 1999.

[91] D. Kundur and D. Hatzinakos, "Semi-Blind Image Restoration Based on Telltale Watermarking," *Proc. 32nd Asilomar Conference on Signals, Systems and, Computers*, Pacific Grove, California, vol. 2, pp. 933-937, November 1998.

[92] Huijuan Yang and Alex, C. Kot, "Binary Image Authentication With Tampering Localization By Embedding Cryptographic Signature and Block Identifier," *IEEE Signal Processing Letters,* vol. 13, no. 12, pp. 741-744, Dec. 2006.

[93] J. Fridrich, M. Goljan and M. Du, "Invertible Authentication," *Proc. of SPIE, Security and Watermarking of Multimedia Contents*, 2001.

[94] M. Goljan, J. Fridrich and M. Du, "Distortion-free Data Embedding for Images," *4th International Information Hiding Workshop*, 2001.

[95] J. Tian, "Wavelet-based Reversible Watermarking for Authentication," *Proc. of SPIE Security and Watermarking of Multimedia Content IV*, vol. 4675, no. 74, Jan 2002.

[96] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni and W. Su, "Distortionless Data Hiding Based on Integer Wavelet Transform," *IEE Electronics Letters*, pp. 1646-1648, vol. 38, no. 25, 2002.

[97] K. Sayood, "Introduction to Data Compression," Morgan Kauffmann Publishers, 2000.

[98] A. Makur, "Self-embedding and Restoration Algorithms for Document Watermark," *Proc. IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 2, pp. 1133-1136, 2005.

[99] Stephen V. Rice, George Nagy and Thomas A. Nartker, "Optical Character Recognition: An Illustrated Guide to the Frontier," Kluwer Academic Publishers, Norwell, Massachusetts, 1999.

[100] R. E. Blahut, "Theory and Practice of Error Control Codes," Addison-Wesley Publishing Company, 1983.

[101] M. U. Celik, G. Sharma, E. Saber and A. M. Tekalp, "Hierarchical Watermarking for Secure Image Authentication with Localization," *IEEE Trans. Image Processing*, vol. 11, no. 6, pp. 585-595, June 2002.

[102] Image database:
www.petitcolas.net/fabien/watermarking/image_database/index.html.

[103] The USC-SIPI image database: http://sipi.usc.edu/database/index.html.

[104] R Chandramouli, N Memon and M Rabbani, "Digital Watermarking", *Encyclopedia of Imaging Science and Technology*, pp. 158-172, 2002.

[105] C. E. Shannon, "A mathematical theory of communication," *Bell System Tech. Journal* 27, pp. 379–423, 1948.

[106] Pierre Moulin and Ralf Koetter, "Data Hiding Codes", vol. 93, issue 12, pp. 2083- 2126, *Proc. of the IEEE*, 2005.

[107] Y. Steinberg and N. Merhav, "Identification in the presence of side information with application to watermarking," *IEEE Trans. Information Theory,* vol. 47, no. 4, pp. 1410–1422, 2001.

[108] J. J. Eggers and B. Girod, "Blind watermarking applied to image authentication," pp. 1977–1980, vol. III, *Proc. ICASSP*, 2001.

[109] Emin Martinian, Gregory W. Wornell and Brian Chen, "Authentication with Distortion Criteria", *IEEE Transactions on Information Theory,* vol. 51, issue 7, pp. 2523- 2542, 2005.

[110] S. Voloshynovskiy *et al*, "Information-theoretic analysis of electronic and printed document authentication", *Proceedings of the SPIE*, pp. 516-535, vol. 6072, 2006.

[111] Ton Kalker and Frans Willems, "Capacity Bounds and Code Constructions for Reversible Data-Hiding", *Proc. Int. Conf. on DSP*, Santorini, Greece, July 2002.

[112] Ton Kalker and Frans M.J. Willems, "Capacity bounds and code constructions for reversible data-hiding", *Proceedings SPIE Conference*, Santa Clara, California, Jan. 2003.

[113] H.Yang and A. C. Kot, "Semi-fragile Watermarking for Text Document Images Authentication," *IEEE International Symposium on Circuits and Systems (ISCAS)*, Kobe, Japan, May 2005.

[114] M. A. Masry, "A Watermarking Algorithm for Map and Chart Images," *SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents VII*, January 2005.

[115] Shiyan Hu, "Document Image Watermarking Algorithm Based on Neighborhood Pixel Ratio," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, USA, 2005.

# Publications


1. Niladri B. Puhan, Anthony T. S. Ho, "Secure authentication watermarking for localization against the Holliman-Memon attack," *ACM Multimedia Systems*, vol. 12, no. 6, pp. 521-532, May 2007.
2. Niladri B. Puhan, A.T.S. Ho, F. Sattar, "Erasable Authentication Watermarking in Binary Document Images," *Accepted in IEEE International Conference on Innovative Computing, Information and Control* (ICICIC-07), Kumamoto, Japan, September 2007.
3. Niladri B. Puhan, Anthony T. S. Ho, "Secure Exact Authentication in Binary Document Images," *Proc. IET Intl. Conference on Visual Information Engineering* (VIE), pp. 29 - 34, Bangalore, India, September 2006.
4. Niladri B. Puhan, Anthony T. S. Ho, "Restoration in Secure Text Document Image Authentication Using Erasable Watermarks," *Lecture Notes in Artificial Intelligence,* pp. 661-668, vol. 3802, 2005.
5. Niladri B. Puhan, Anthony T. S. Ho, "Secure Tamper Localization in Binary Document Image Authentication," *Lecture Notes in Artificial Intelligence*, pp. 263-271, vol. 3684, 2005.
6. Niladri B. Puhan, Anthony T. S. Ho, "Binary Document Image Watermarking for Secure Authentication Using Perceptual Modeling," *Proc. Fifth IEEE International Symposium on Signal Processing and Information Technology* (ISSPIT), pp. 393-398, Athens, Greece , July 2005.
7. Anthony T. S. Ho, Niladri B. Puhan, P. Marziliano, A. Makur, Y. L. Guan, "Perception Based Binary Image Watermarking," *Proc. IEEE International Symposium on Circuits and Systems* (ISCAS), pp. 37-40, vol. 2, Vancouver, Canada, May 2004.
8. Anthony T. S. Ho, Niladri B. Puhan, P. Marziliano, A. Makur, Y. L. Guan, "Imperceptible Data Embedding in Sharply-Contrasted Binary Images," *Proc. IEEE International Conference on Control, Automation, Robotics and Vision* (ICARCV), pp. 958-963, vol. 2, Kunming, China, December 2004.
9. Anthony T. S. Ho, Niladri B. Puhan, P. Marziliano, Y. L. Guan, "A Novel Curvature-Weighted Distance Difference (*CWDD*) Perceptual Model for Binary Image Data Hiding," *Proc. Fifth International Conference on Advances in Pattern Recognition* (ICAPR), Calcutta, India, December 2003.
10. Niladri B. Puhan, Anthony T. S. Ho, "Public Key Authentication of Binary Document Images Using Perceptual Watermarking," *Accepted in the Workshop Proceedings of International Conference on Computational Intelligence and Security* (CIS), China, 2005.
11. Niladri B. Puhan, Anthony T. S. Ho, F. Sattar "Secure Exact Authentication in Binary Document Image Watermarking," *To be submitted in EURASIP Journal on Applied Signal Processing*.
12. Niladri B. Puhan, Anthony T. S. Ho, F. Sattar, "Localization in Binary Document Image Authentication Watermarking after Tampering and Holliman-Memon Attack," *To be submitted in LNCS Transactions on Data Hiding and Multimedia Security.*