ABSTRACT

A STUDY AND SOME EXPERIMENTAL WORK OF DIGITAL IMAGE AND VIDEO WATERMARKING

by
Tsui-Feng Liu

The rapid growth of digitized media and the emergence of digital networks have created a pressing need for copyright protection and anonymous communications schemes. Digital watermarking (or data hiding in a more general term) is a kind of steganography technique by adding information into a digital data stream. Several most important watermarking schemes applied to multilevel and binary still images and digital videos were studied. They include schemes based on DCT (Discrete Cosine Transform), DWT (Discrete Wavelet Transform), and fractal transforms. The question whether these invisible watermarking techniques can resolve the issue of rightful ownership of intellectual properties was discussed. The watermarking schemes were further studied from malicious attack point of view, which is considered an effective way to advance the watermarking techniques. In particular, the StirMark robustness tests based on geometrical distortion were carried out.

A binary watermarking scheme applied in the DCT domain is presented in this research project. The effect of the binarization procedure necessarily encountered in dealing with binary document images is found so strong that most of conventional embedding schemes fail in dealing with watermarking of binary document images. Some particular measures have to be taken. The initial simulation results indicate that the proposed technique is promising though further efforts need to be made.

A STUDY AND SOME EXPERIMENTAL WORK OF DIGITAL IMAGE AND VIDEO WATERMARKING

by
Tsui-Feng Liu

A Project
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the
Degree of Engineer

Department of Electrical and Computer Engineering

January 2001

APPROVAL PAGE

A STUDY AND SOME EXPERIMENTAL WORK OF DIGITAL IMAGE AND VIDEO WATERMARKING

Tsui-Feng Liu

Dr. Yun-Qing Shi, Project Advisor — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Edwin Hou, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Mengchu Zhou, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Tsui-Feng Liu

Degree: Degree of Engineer

Undergraduate and Graduate Education:

- Degree of Engineer in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2001
- Master of Science in Electrical Engineering,
  Polytechnic University, Brooklyn, NY, 1996
- Bachelor of Science in Electrical Engineering,
  Cheng Kung University, Tainan, Taiwan.

Major: Electrical Engineering

Presentations and Publications:

A. Tan and Tsui-Feng Liu,

" Performance of the Adaptive LMS Orthogonal Filters," Proc. of the 35th Midwest Symposium on Circuits and Systems, 1994, Volume 2, pp. 1469-1472.

Tsui-Feng Liu and A. Tan,

Adaptive Wave and Orthogonal Digital Filters Based on Least-Mean-Square (LMS) and Normalized LMS Algorithms (M. S. Thesis).

To my beloved family

ACKNOWLEDGEMENT

I would like to thank Dr. Shi, my advisor, who guided me in doing the research and preparing my project. I also would like to thank Dr. Hou and Dr. Zhou for their evaluation of my project. Above all, I am very much obliged to my parents and friends for their concern and assistance.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 The Historical View of Steganography

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means "covered writing". It includes a lot of secret communications methods that hide the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications [1]. For example, the childhood practice of writing message in 'invisible ink' would qualify as Steganography since the writing is hidden in the sense that it is not obvious that it is there unless you know to look for it. That means that Steganography is the way that can hide any kinds of writing anywhere that people is difficult to find.

Some Steganographic methods are introduced here. The Internet provides an increasingly broad band of communication as a means to distribute information to people. Such information includes text, images, and audio to convey ideas for mass communication. Hiding information in text by using invisible inks is early popular approach. Adding spaces and "invisible" characters to text provides a method to pass hidden information. An interesting way to see this is to add spaces and extra line breaks in an HTML file. Web browsers ignore these "extra" spaces and lines, but revealing the source of the web page displays the extra characters. Many different methods of hiding information in images exist. These methods range from Least Significant Bit (LSB) or noise insertions, manipulation of image and compression algorithms, and modification of image properties such as luminance. Other more robust methods of hiding information in

images include application of the transform domain that uses its components to hide information. These methods hide messages in significant areas of the cover image which make them more robust to attacks. The LSBs and transforms can also applied to hide information in audio system. In audio, small echoes can be added or subtle signals can be masked by sounds of higher amplitude [2]. Unused space in file headers of image and audio can be used to hold "extra" information. Taking advantage of unused or reserved space to hold covert information provides a means of hiding information without perceptually degrading the carrier. Selecting the proper combination of Steganography tools and carriers is the key to successful information hiding.

1.2 Project Content

In the chapter 2, we propose a binary watermarking encoding scheme based on DCT (Discrete Cosine Transform) domain and do some survey of the other binary and multilevel watermarking schemes of the still image and video. For the binary image watermarking encoding scheme, we introduce a binary watermarking encoding scheme and describe the paper from AT&T Bell Laboratories [3]. For multilevel image watermarking encoding scheme, we introduce and discuss the watermarking scheme applied on DCT (Discrete Cosine Transform) domain, DWT (Discrete Wavelet Transform) and Fractal coding. For the watermarking scheme applied on DCT domain, we summarize the Cox's paper [4] and introduce Huang's image watermarking scheme [5]. For video, we describe the paper " Copyright Protection Video Delivery Networks by Watermarking of Pre-Compressed Video".

In chapter 3, we introduce and discuss the paper – IBM Research Report " Can Invisible Watermarks Resolve Rightful Ownerships? ".

In chapter 4, we introduce the StirMark tool developed for simple robustness testing of image watermarking algorithms and do the experiment by using StirMark testing tool. Finally the conclusions and discussions are included in chapter 5.

CHAPTER 2

DIGITAL IMAGE AND VIDEO WATERMARKING

Digital watermarking is a means of protecting multimedia data from intellectual piracy. It is achieved by modifying the original data to insert a "signature". Several watermarking schemes have been proposed in recent years. We do some survey of digital watermarking schemes for still image and video. For the binary watermark encoding scheme of the still image we introduce a binary watermarking encoding scheme and discuss the other binary watermarking encoding scheme from the AT&T Bell Laboratories. For multilevel image watermarking scheme, we discuss the watermarking schemes applied on DCT (Discrete Cosine Transform) domain, DWT (Discrete Wavelet Transform) domain, Fractal transform domain, and video.

2.1 Binary Watermarking Scheme

Many invisible watermarking schemes have been reported in recent years. These techniques can be classified in two categories: spatial-domain and transform-domain based. The earlier watermarking techniques reported were based on spatial domain. The transform-domain-based techniques can be employed with common image transforms like discrete cosine transform (DCT), Wavelets, Fourier transforms.

Brassil et al [3] in AT&T Bell Laboratories proposed three methods appropriate for document images. We describe in next section. Based on Cox. et al.'s paper [4], they placed a length n watermark into N x N DCT of the image and placed the watermark into the n highest magnitude coefficients of the DCT matrix, excluding the DC component.

They applied their encoding scheme on grayscale images. We introduce a watermarking scheme applied to DCT domain of binary document images.

2.1.1 Electronic Marking and Identification Techniques

Copyright protection is becoming more necessary when the global Internet are increasingly used to deliver electronic documents. While the photocopy has hurt the publishers, the need for document security is much greater for electronic document distribution. When a document is marked in an invisible way by a codeword identifying the registered owner to whom the document is sent, and if a document copy is found, that copy can decoded and the registered owner identified. We introduce the techniques by a modified and summarized version of the AT&T paper "Electronic Marking and Identification Techniques to Discourage Document Copying" proposed by J. Brassil, S. Low, N. Maxemchuk, L. O'Gorman. [3].

Their proposed encoding techniques can also make paper copies of documents traceable. In particular, the codeword embedded in each document survives plain paper copying. Document marking can be achieved by altering the text formatting, or by altering certain characteristics of textual elements (e. g., characters). The goal in the design of coding methods is to develop alterations that are reliably decodable (even in the presence of noise) yet largely invisible to the reader. The marking techniques they describe can be applied to either an image representation of the document or to a document format file. The document format file is a computer file describing the document content and page layout (or formatting), using standard format description languages such as PostScript, Tex, troff, etc. The image representation describes each

page of a document as an array of pixels. The image may be bitmap (also called binary or black-and-white), gray-scale, or color. For this work, they describe both document format file and image coding techniques. They restrict the image coding techniques to bitmaps encoded within the binary-valued text regions. Common to each technique is that a codeword is embedded in the document by altering particular textual features.

2.1.1.1 Line-Shift Coding This is a method of altering a document by vertically shifting the locations of text lines to encode the document uniquely. This encoding may be applied either to the format file or bitmap of a page image. The embedded codeword may be extracted from the format file or bitmap. In certain cases this decoding can be accomplished without need of the original image, since the original is known to have uniform line spacing between adjacent lines within a paragraph.

2.1.1.2 Word-Shift Coding This is a method of altering a document by horizontally shifting the locations of words within text lines to encode the document uniquely. This encoding can be applied to either the format file or to the bitmap of a page image. Decoding may be performed from the format file or bitmap. The method is applicable only to documents with variable spacing between adjacent words. Variable spacing in text documents is commonly used to distribute white space when justifying text. Because of this variable spacing, decoding requires the original image – or more specifically, the spacing between words in the unencoded document. See Figure 2.1 for an example of word-shift coding.

Consider the following example of how a document might be encoded with word-shifting. For each text line, the largest and smallest spacing between words are found. To code a line, the largest spacing is decreased by some amount and the smallest is augmented by the same amount. This maintains the text line length, and produces little qualitative change to the text image.

2.1.1.3 Feature Coding This is a coding method that is applied either to a format file or to a bitmap image of a document. The image is examined for chosen text features, and those features are altered, or not altered, depending on the codeword. Decoding requires the original image, or more specifically, a specification of the change in pixels at a feature. There are many possible choices of text features; here, we choose to alter upward, vertical endline – that is the tops of letters, b,d,h, etc. These endlines are altered by extending or shortening their lengths by one (or more) pixels, but otherwise not changing the endlines feature. See Figure 2.2 for an example of feature coding.

Among their proposed encoding techniques, line-shifting is likely to be the most easily discernible by readers. However they also expect line-shifting to be the most robust type of encoding in the presence of noise. This is because the long lengths of text lines provide a relatively easily detectable feature. For this reason, line shifting is particularly well suited to marking documents to be distributed in paper form, where noise can be introduced in printing and photocopying. As they show in the next section, their experiments indicate that they can easily encode documents with line shifts that are sufficiently small that are not noticed by the casual reader, while still retaining the ability to decode reliably.

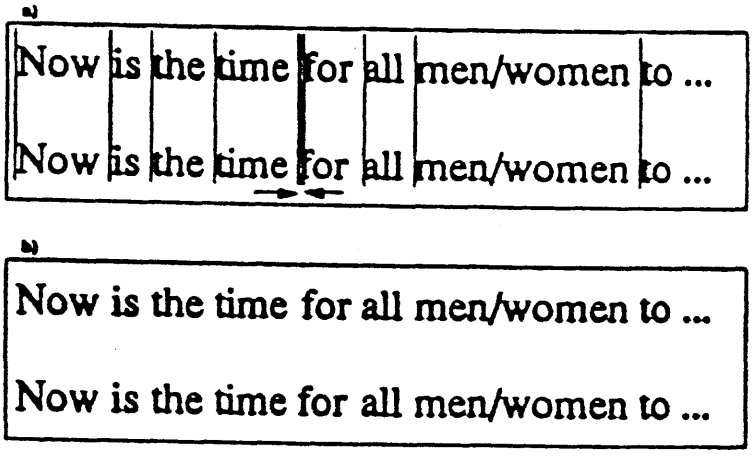


Fig. 2.1 Example of word-shift coding. In a). the top text line has added spacing before the "for" the bottom text line has the same spacing after the "for" In b), these same text lines are shown again without the vertical lines to demonstrate that either spacing appears natural.

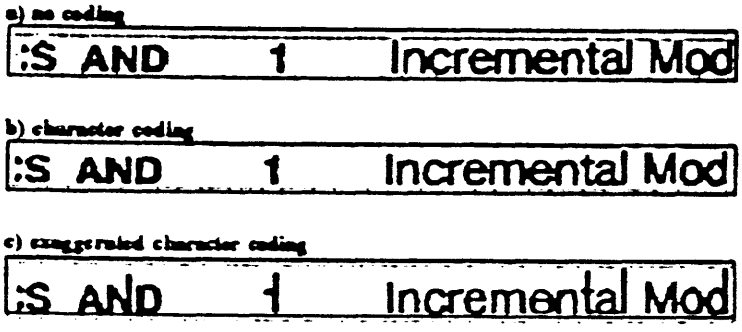


Fig. 2.2 Example shows feature coding performed on a portion of text from a journal table of contents. In a), no coding has been applied. In b), feature coding has been applied to select characters. In c), the feature coding has been exaggerated to show feature alterations.

They expect that word-shifting will be less discernible to the reader than line-shifting, since the spacing between adjacent words on a line is often varied to support text justification. Feature encoding can accommodate a particularly large number of sanctioned document recipients, since there are frequently two or more features available for encoding in each word. Feature alterations are also largely indiscernible to reader. Feature encoding also has the additional advantage that it can be applied directly to image files, which allows encoding to be introduced in the absence of a format file.

2.1.1.4 Implementation and Experimental Results In this section we describe the method for coding and decoding they used for testing the line-shift coding method. Each intended document recipient was given a unique codeword. Each codeword specifically a set of text lines to be moved in the document specifically for that recipient. The length of each codeword equaled the maximum number of lines that were displaced in the area to be encoded. In their line-shift encoder, each codeword element belonged to the alphabet {-1,+1,0}, corresponding to a line to be shifted up, down or remain unmoved.

Though their encoder was capable of shifting an arbitrary text line either up or down, they found that the decoding performance was greatly improved by constraining the set of lines moved. In the results presented in this section, they used a differential encoding technique. With this coding they kept every other line of text in each paragraph unmoved, starting with the first line of each paragraph. Each line between two unmoved lines was always moved either up or down. That is, for each paragraph, the 1^{st}, 3^{rd}, 5^{th}, etc. lines were unmoved, while the 2^{nd}, 4^{th}, etc. lines were moved. Note that the consequence of using differential encoding is that the length of each codeword is cut

approximately in half. In each of their experiments they displaced at least 19 lines, which corresponds to a potential of at least $2^{19} = 524,288$ distinct codewords / page. More than a single page per document can be coded for a large number of codeword possibilities.

The line-shift decoder measured the distance between each pair of adjacent text line profiles (within the page profile). This was done by one of two approaches—either they measured the distance between the baselines of adjacent line profiles, or they measured the difference between centroids of adjacent line profiles. A baseline is the logical horizontal line on which characters sit; a centroid is the center of mass of a text line profile. To define the centroid of a text line precisely, suppose the text line profile runs from scan line y, y+1,...,to y+w, and the respective number of ON bits/scan line are h(y), h(y+1),...,h(y+w). Then the text line centroid is given by

$$(Y\, h(y) + \ldots + (y+w)\, h(y+w)) \;/\; (h(y) + \ldots + h(y+w)).$$

The measured interline spacings (i.e., between adjacent centroids or baselines) were used to determine if white space has been added or subtracted because of a text line shift as seen in Fig.2.3. This process, repeated for every line, determined the codeword of the document — this uniquely determined the original recipient.

They describe their decision rules for detection of line shifting in a page with differential encoding. Suppose text lines i-1 and i +1 are not shifted and text line i is either shifted up or down. In the unaltered document, the distance between adjacent baselines, or baseline spacings, are the same. Let s_{i-1} and s_i be the distances between

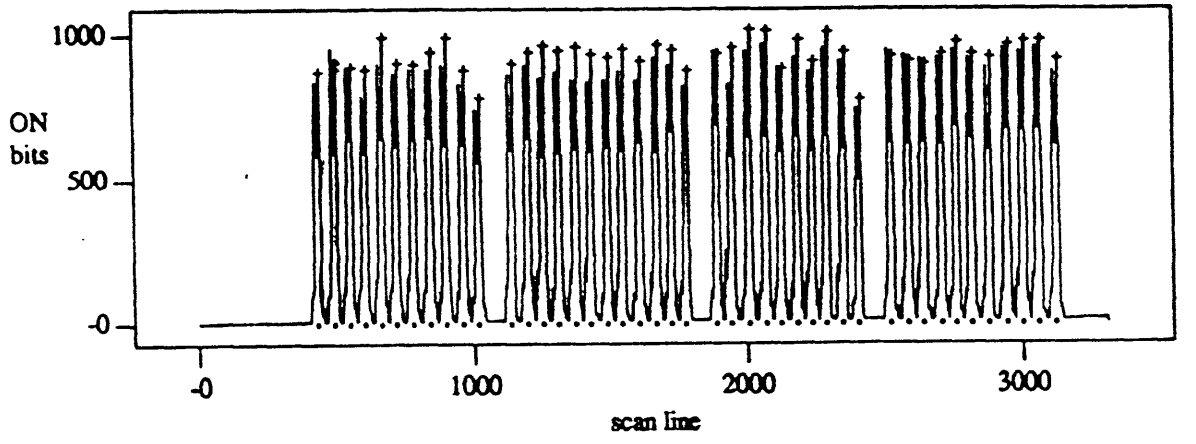


Fig. 2.3 Profile of a recovered document page. Decoding a page with line shifting requires the distances between adjacent text line centroids (marked with •) or base lines (marked with +) and deciding whether white space has been added or subtracted.

baselines i-1 and i, and between baselines i and i+1, respectively, in the altered document. Then the baseline detection decision rule is:

If $s_{i-1} > s_i$: decide line i shifted down

If $s_{i-1} < s_i$: decide line i shifted up

Otherwise : uncertain

The experimental result tested how well a fixed line spacing shift could be detected as document degradation become increasingly severe. The equipment they used in the experiments was as follows: a Ricoh FSIS 400 dpi Flat Electronic Scanner, Apple LaserWriter IIntx 300 dpi laser printer, and a Xerox 5052 plain paper. The printer and copier were selected in part paper copier. They wrote the software routine to implement a rudimentary line-shift encoder for a PostScript input file. They encoded a single-spaced page of text using different encoding. They used a 10- point Times-Roman font, and a pixel line shift. Twenty-one lines were shifted on the page. They then made repeated copies (the 1^{st}, ... , 10^{th} copy) of the page, and used each copy in a separate experiment. Hence, each successive experiment used a slightly more degraded version of the same text page.

Detection results are as follows. The centroid detection method successfully detected all 21 line shifts for each generation of photocopy (through the 10^{th} copy). The baseline detection method successful detected all lines on every copy generation, with the following exceptions: 1 error was made on the 4^{th}, 5^{th}, 6^{th}, 7^{th} and 10^{th} copy, 2 errors on the 9^{th} copy; 1 uncertain on the 3^{rd}, 4^{th}, and 10^{th} copy, 2 uncertains on the 7^{th} copy and 4

uncertains on the 8th copy. In summary, the baseline detection method successfully detected at last 16 lines shifts on each copy generation.

Their results indicate that, for baseline decoding, detection errors and uncertainties do not increase monotonically with the number of copies. Further, the line spacings that could not be detected correctly varied somewhat from copy to copy.

2.1.2 Binary Image Watermarking in DCT Domain

In DCT domain, there are two requirements for selecting suitable coefficients to embed watermarks: (1) The coefficients should have a large perceptual capacity that allows strong watermarks to be embedded in without causing any perceptual distortion. (2) The coefficients should be hardly changed by common image processing. The proposed watermarking encoding scheme embeds the watermark in the DC term of DCT. The experimental results demonstrate the proposed watermarking algorithm can work well on formatted text images and other formats that appear in the binary document.

2.1.2.1 Overview of Proposed Watermarking Algorithm In the proposed binary watermarking encoding scheme, the original binary image is processed first; then the preprocessed image is split into non-overlapped blocks of 8×8 and each block is DCT transformed. The watermark, which is a random number sequence and obeys Gaussian distribution, is inserted into the DC coefficients of each block. After the modification of the DC component, an inverse DCT transform is performed for each block. Finally the image is performed to binary document image.

In watermark detection, the corrupted watermark is extracted by comparing the original binary image and corrupted watermarked binary image. The correlation of the extracted watermarked and the original watermark sequence is computed to decide if a watermark exists. A block diagram of the proposed algorithm in DCT domain is illustrated in Figure 2.4.

2.1.2.2 Watermark Embedding The watermark embedding of the proposed watermarking technique in DCT domain consists of the following steps:

1. The original binary image f (x, y) is preprocessed .
2. The preprocessed image g (x, y) is split into non-overlapped blocks of 8×8, denoted as B_k, k = 0, 1, …, n-1, which can be described as:

   $$g(x, y) = \cup_k B_k = \cup_k g_k (x', y') \quad 1 \le x' \le 8, 1 \le y' \le 8$$

3. Each B_k is DCT transformed:

   $$G_k (u, v) = DCT \{ g_k (x', y') \}$$

4. The watermark $W = \{x_i, 0 \le i < n \}$ is composed of a random number sequence of length n which obeys the Gaussian distribution N(0,1).
5. The watermark is inserted by modifying the DC components as follows.

   $$G'_k (u, v) = G_k (u, v) \cdot (1 + \alpha \cdot x) \quad \text{if } u = v = 0$$

   $$G'_k (u, v) = G_k (u, v) \quad \text{otherwise}$$

   Where α is a scaling factor and x_k is the kth element of the watermark W.

   We choose $\alpha = 0.5$ as the scaling factor in all experiments.
6. The inverse DCT is performed as follows.

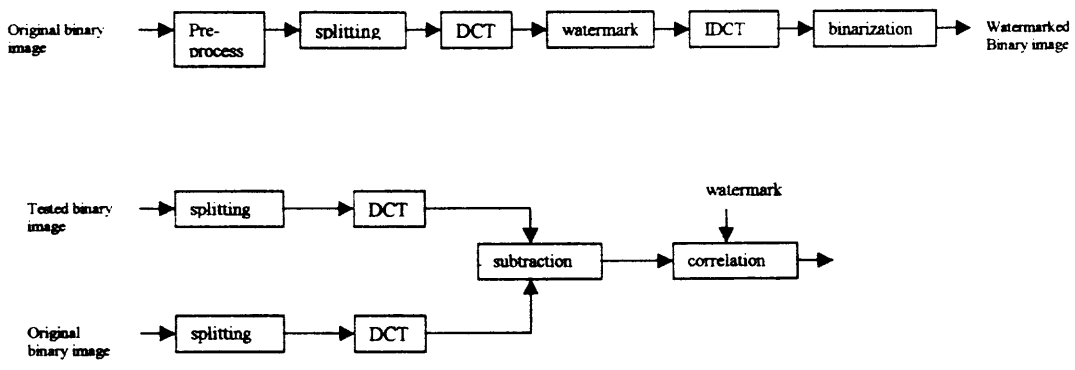


Fig. 2.4 Proposed watermarking algorithm

$$g(x, y) = \cup_k \text{ IDCT } \{ G'_k(u, v), \quad 1 \le u \le 8, 1 \le v \le 8 \}$$

7. Binarization will be made to the inverse DCT transformed image to obtain the watermarked binary image.

2.1.2.3. Watermark Detection The detection of the watermark is done by using correlation technique.

1. The original binary image f (x , y) and the test binary image f'(x', y') are split into non-overlapped blocks of 8 × 8 pixels.
2. Each block in these two images is DCT transformed. Let F_k (u, v) and F^*_k (u, v) denote the DCT coefficients of f (x , y) and f' (x' , y') respectively.
3. The corrupted watermark W^* is extracted by comparing the corresponding DC component in each block.

   $x^*_i = (F^*_k(0, 0) - F_k(0, 0)) \; / \; \alpha \cdot F_k(0, 0) \qquad k = 0, 1, \ldots, n-1$

   Then $W^* = \{ x^*_i , 0 \le i < n \}$, where is the corrupted version of x_i.
4. To determine whether a watermark exists, we compute the similarity by computing the correlation:

   $Corr(W^*, W) = \Sigma_i \; (x^*_i \cdot x_i) \; / \; \text{sqrt}(\Sigma \; x^*_i \cdot x^*_i)$

If $Corr(W^*, W) >$ T, it indicates that a watermark exists in the tested image. Otherwise watermark is not detected. Where T is a predefined threshold.

2.1.2.4 Experimental Results In addition to text, binary documents also contain tables, graphics, images etc. In this project, we do simulations with 4 different kinds of binary image. In the proposed watermarking algorithm, we generate 52 random number

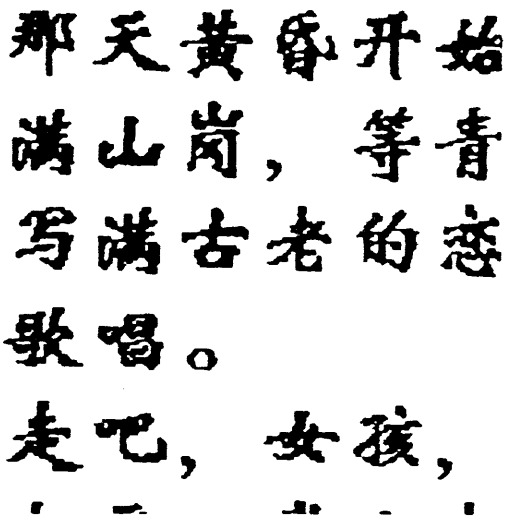
sequences from the normal distribution N (0,1). The third sequence is used as the true watermark. The experimental results of four different binary images are shown in Fig. 2.5 - Fig. 2.9.

Fig. 2.5(a), Fig. 2.6(a), Fig. 2.7(a), and Fig. 2.8(a) are four original binary images. Fig. 2.5(b), Fig. 2.6(b), Fig. 2.7(b), and Fig. 2.8(b) are four watermarked gray-scaled images embedded the watermark on dc component of DCT domain. The invisibility of our watermarked images is shown on Fig. 2.5(c), Fig. 2.6(c), Fig. 2.7(c), and Fig.2.8(c). The response of detection to the watermarked image is shown on Fig. 2.5(d), Fig. 2.6(d), Fig. 2.7(d), and Fig. 2.8(d). Finally Fig. 2.9(a) and Fig. 2.9(b) are shown to demonstrate the robustness of the watermarked image against digital signal processing.

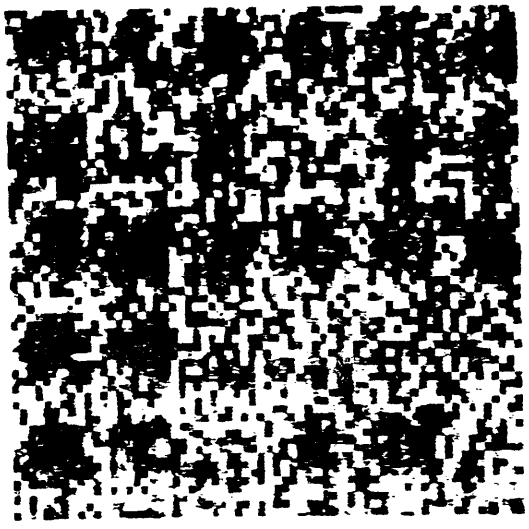
2.2 Secure Spread Spectrum Watermarking for Multimedia

A watermark may contain additional information, which can be identified the purchaser of the particular copy of the material. An effective watermark should be perceptually invisible and robust (impossible to remove). The digital watermark schemes applied on the document images have introduced in last section. We will introduce a digital watermark applied to DCT (Discrete Cosine Transform) domain for multilevel images.

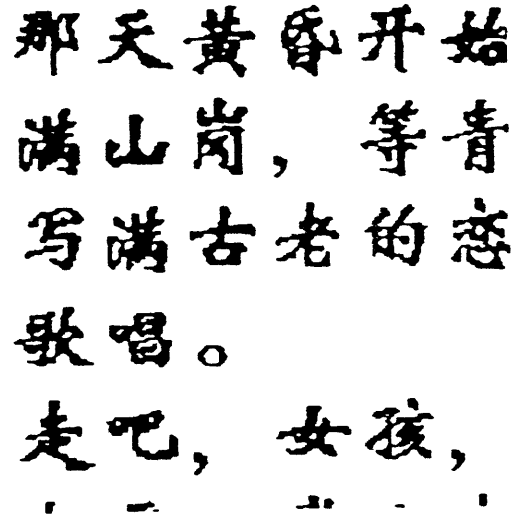
A modified and summarized version of the IEEE Trans. On Image Processing, proposed by Ingemar J. Cox, Joe Killant, Tom Leighton and Talal Shamoon [4] is described as follows.



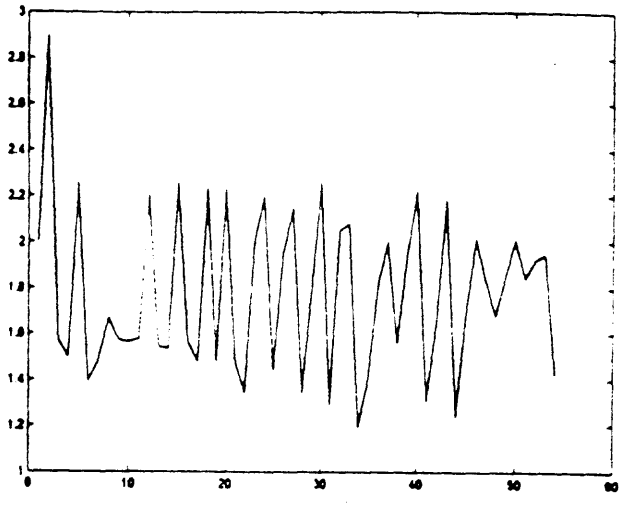
(a)



(b)

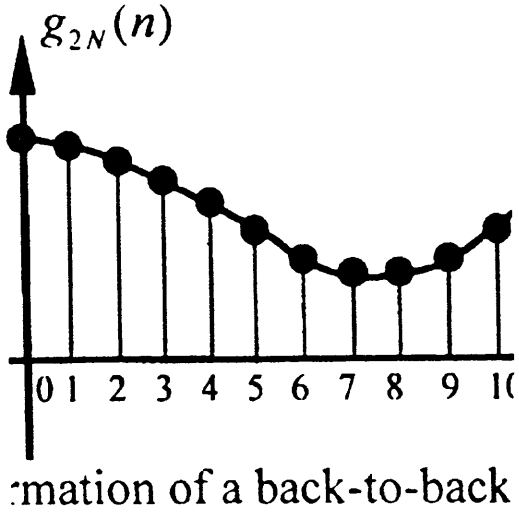


(c)



(d)

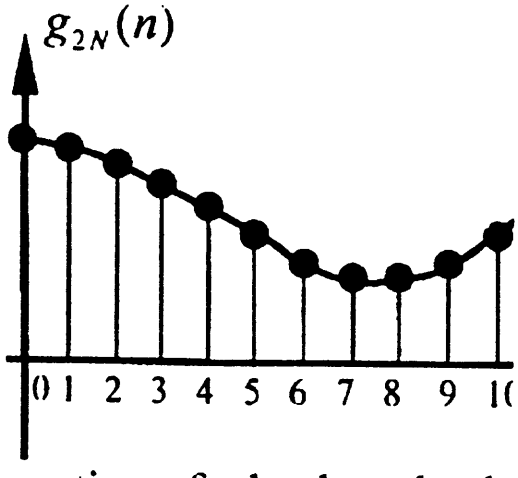
Fig. 2.5 Experimental results for "Chinese.tif". (a) original image
(b) watermarked gray-scale image (c) watermarked binary image
(d) response of detection for watermarked binary "chinese.tif" image



:mation of a back-to-back

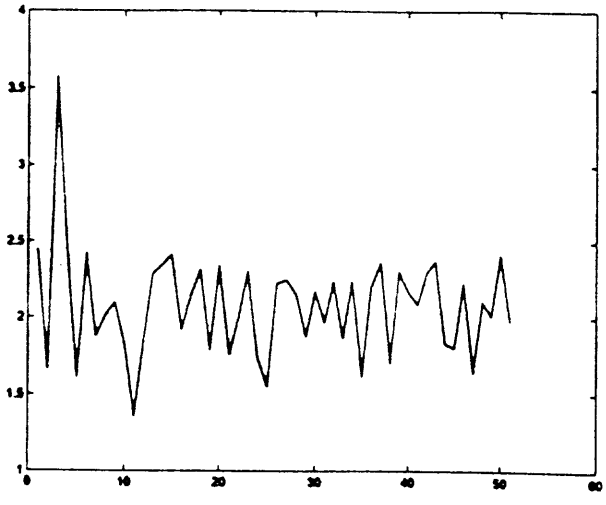
(a)

(b)



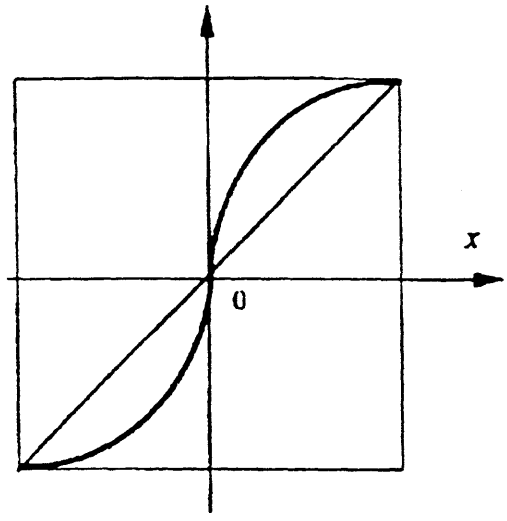
:mation of a back-to-back

(c)

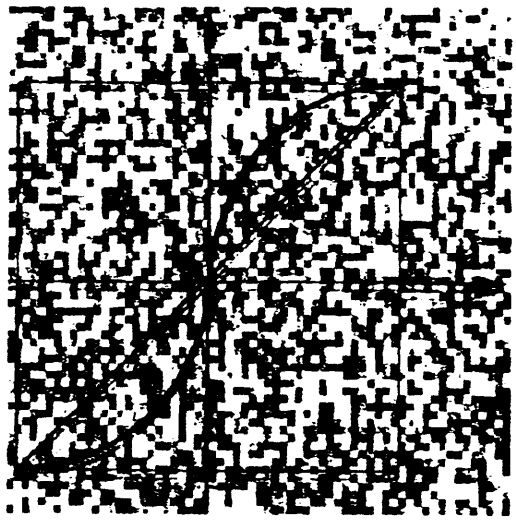


(d)

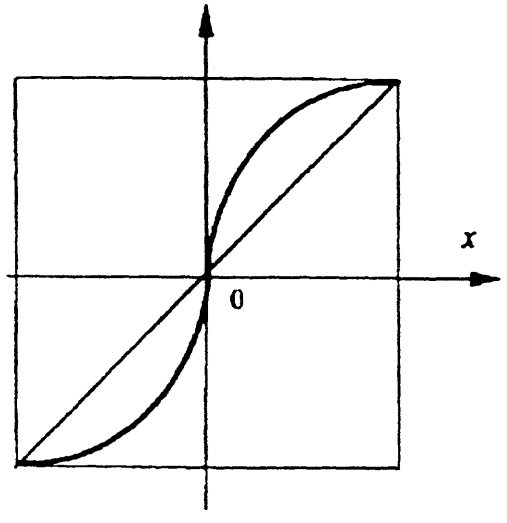
Fig. 2.6 Experimental results for "graph2.tif". (a) original image
(b) watermarked gray-scale image (c) watermarked binary image
(d) response of detection for watermarked binary "graph2.tif" image



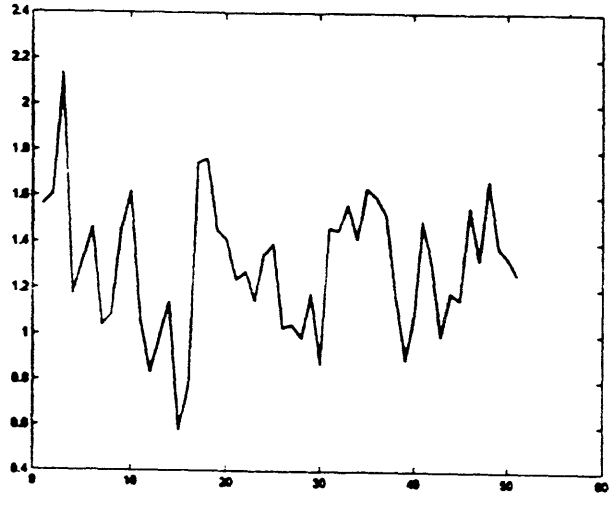
(a)



(b)



(c)

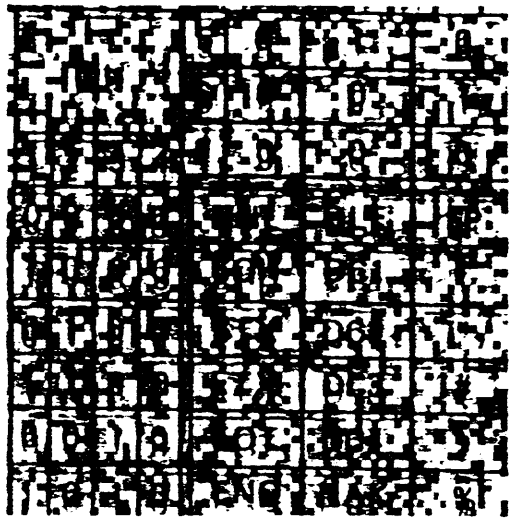


(d)

Fig. 2.7 Experimental results for "drawing.tif". (a) original image
(b) watermarked gray-scale image (c) watermarked binary image
(d) response of detection for watermarked binary "drawing.tif" image

Bits				5	0	1	0
				6	0	0	1
1	2	3	4	7	0	0	0
0	0	0	0		NUL	DLE	SP
1	0	0	0		SOH	DC1	!
0	1	0	0		STX	DC2	"
1	1	0	0		ETX	DC3	#
0	0	1	0		EOT	DC4	$
1	0	1	0		ENQ	NAK	%

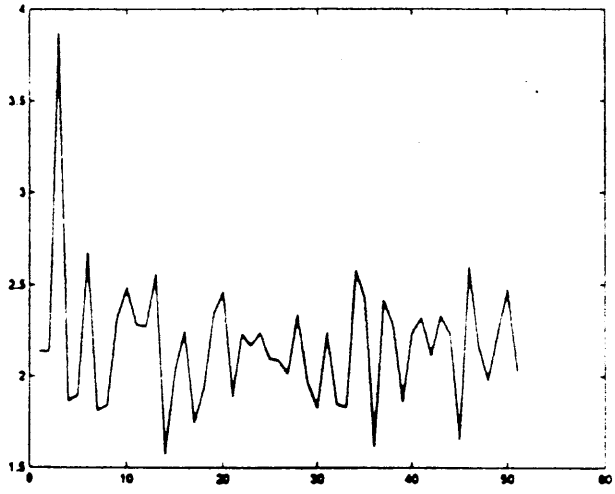
(a)



(b)

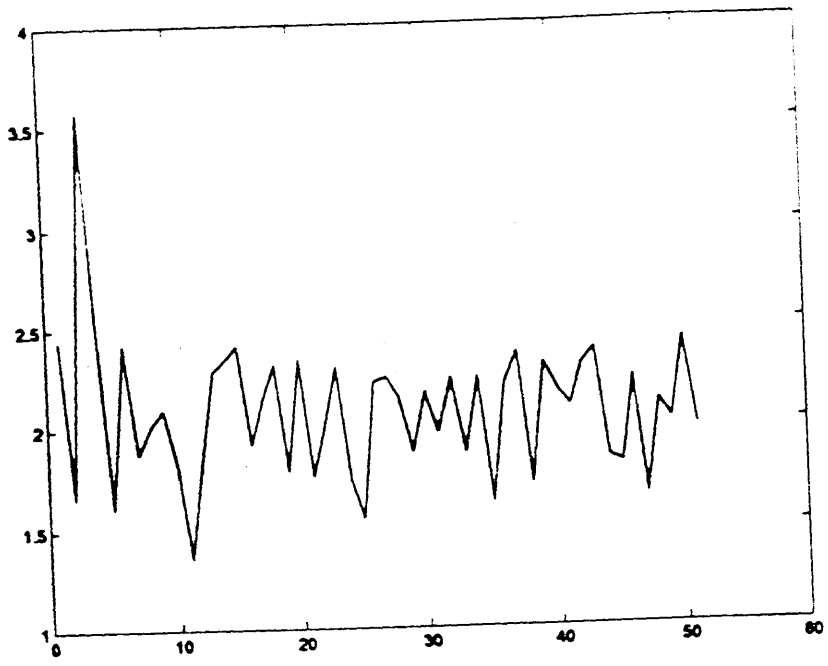
Bits				5	0	1	0
				6	0	0	1
1	2	3	4	7	0	0	0
0	0	0	0		NUL	DLE	SP
1	0	0	0		SOH	DC1	!
0	1	0	0		STX	DC2	"
1	1	0	0		ETX	DC3	#
0	0	1	0		EOT	DC4	$
1	0	1	0		ENQ	NAK	%

(c)

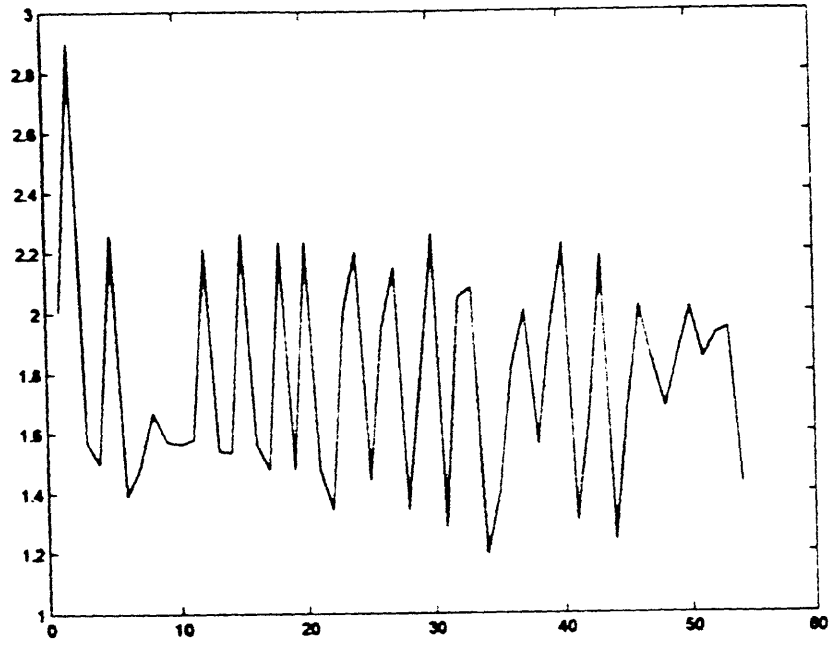


(d)

Fig. 2.8 Experimental results for "table.tif". (a) original image
(b) watermarked gray-scale image (c) watermarked binary image
(d) response of detection for watermarked binary "table.tif" image



(a)



(b)

Fig. 2.9 Demonstration of robustness (a) Response of detection to unsharp contrast enhancement filtered watermarked binary image "graph2.tif" (b) Response of detection to unsharp filtered watermarked binary image "chinese.tif"

2.2.1 Spread Spectrum Coding of a Watermark

Spreading the watermark throughout the spectrum of an image ensures a large measure of security against unintentional or intentional attacks. A watermark placed in the high frequency spectrum of an image can be easily eliminated with little degradation to the image by any process that directly or indirectly performs low pass filtering. The problem then becomes how to insert a watermark into the most significant regions of the spectrum in a preserving fashion.

The general procedure for frequency domain watermarking is illustrates in Figure 2.10. They use a frequency domain method based on the discrete cosine transform (DCT), then choose some perceptually significant regions in the spectrum. In practice, in order to place a length n watermark into an N x N image, they computed N x N DCT of the image and placed the watermark into the n highest magnitude coefficients of the transform matrix, excluding the DC component. More generally, n randomly chosen coefficients could be chosen from the M, M > n most perceptually significant coefficients of the transform.

They extract from each document D a sequence of values $V = v_1,\ldots,v_n$, into which they insert a watermark $X = x_1,\ldots,x_n$ to obtain an adjusted sequence of values $V' = v_1',\ldots,v_n'$ is then inserted back into the document in place of V to obtain a watermarked document D′. One or more attackers may then alter D′, producing a new document D^*. Given D and D^* a possibly corrupted watermark X^* is extracted and is compared to X for statistical significance. They extract X^* by first extracting a set of values $V^* = v_1^*,\ldots,v_n^*$ from D^* (using information about D) and then generating X^* from V^* and V.

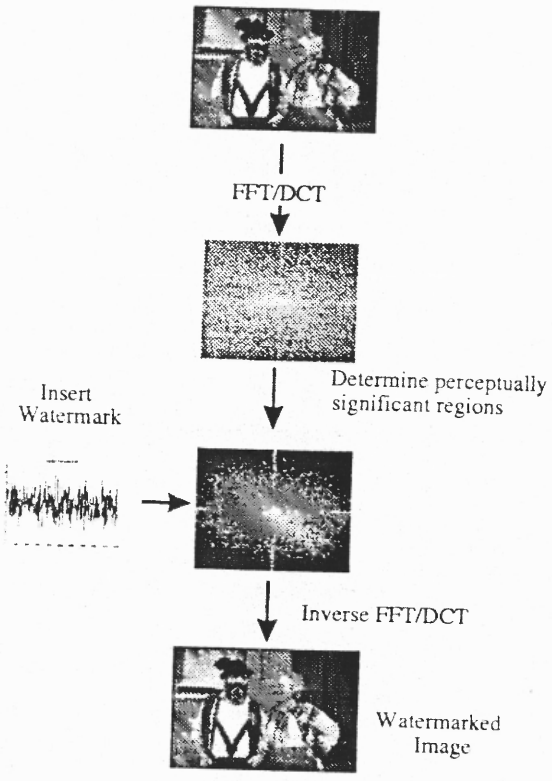


Fig. 2.10 Stages of watermark insertion process

2.2.2 Inserting and Extracting the Watermark

When they insert X into V to obtain V′ they specify a scaling parameter α which determines the extent to which X alters V. Three natural formula for computing V′ are:

$$v'_i = v'_i + \alpha x_i \quad (1)$$

$$v'_i = v_i (1 + \alpha x_i) \quad (2)$$

$$v'_i = v_i (e^{\alpha x_i}) \quad (3)$$

Equation 1 is always invertible, and Equation 2 and 3 are invertible if $v_i \neq 0$, which holds in all of their experiments. Given V^* they can therefore compute the inverse function to derive X^* from V^* and V. Equation 1 may not be appropriate when the v_i values vary widely. If $v_i = 10^6$ then adding 100 may be insufficient for establishing a mark, but if v_i = 10 adding 100 will distort this value unacceptably. Insertion based on Equations 2 or 3 are more robust against such differences in scale. They note that Equations 2 and 3 give similar results when αx_i is small.

2.2.3 Determining Multiple Scaling Parameters

A single scaling parameter α may not be applicable for perturbing all of the values v_i, since different spectral components may exhibit more or less tolerance to modification. More generally one can have multiple scaling parameter$\alpha_1, \ldots, \alpha_n$ and use update rules such as $v'_i = v_i (1 + \alpha x_i)$. They viewed α_i as a relative measure of how much one must alter v_i to alter the perceptual quality of the document. A large α_i means that one can

perceptually "get away" with altering v_i by a large factor without degrading the document.

In general, one may have little idea of how sensitive the image is to various values. One way of empirically estimating these sensitivities is to determine the distortion caused by a number of attacks on the original image. For example, one might compute a degraded image D* from D, extract the corresponding values $v^*_{ii,...,} v^*_n$ and choose α_i to be proportional to the deviation $| v^*_i - v_i |$. For greater robustness, one should try many forms of distortion and make α_i proportional to the average value of $| v^*_i - v_i |$.

One may combine this empirical approach with general global assumptions about the sensitivity of the values. For example, one might require that $\alpha_i \geq \alpha_j$, whenever $v_i \geq v_j$. One way to combine this constraint with the empirical approach would be to set α_i according to

$$\alpha_i \sim \max | v^*_j - v_j |.$$

In all experiments they simply use Equation 2 with a single parameter $\alpha = 0.1$. When they computed JPEG-based distortions of the original image they observed that the higher energy frequency components were not altered proportional to their magnitude (the implicit assumption of Equation 2).

2.2.4 Choosing the Length, n, of the Watermark

The choice of n depends on what the watermark is spread out among the relevant components of the images. For a more quantitative assessment of this tradeoff, they

consider watermarks of the form $v'_i = v'_i + \alpha x_i$ and model a white noise attack by $v^*_i = v'_i + r_i$ where r_i are chosen according to independent normal distributions with standard deviation σ. For the watermarking procedure they described below one can recover the watermark when σ is proportional to $\sigma / (n)^{2/1}$. Note that the number of bits of information associated with the watermark can be arbitrary.

2.2.5 Evaluating the Similarity of Watermarks

It is highly that the extracted mark X* will be identical to the original watermark X. They measure the similarity of X and X* by

$$Sim(X,X^*) = (X^* \cdot X) \ / \ sqrt(X^* \cdot X^*) \qquad (4)$$

Many other measures are possible, including the standard correlation coefficient. To decide whether $sim(X,X^*) > T$, where T is some threshold.

2.2.6 Experimental Results

In order to evaluate their proposed watermarking scheme, they took the "Bavarian Couple" image of Figure 2.11 and produced the watermarked version of Figure 2.12. They then subjected the watermarked image to a series of image processing and collusion style attacks. Their experiments are preliminary, but show resilience to certain types of common processing. Of note is their method's resistance to compression such as JPEG, and data conversion (printing and scanning).

Fig. 2.11 "Bavarian Couple" courtesy of Corel Stock Photo Library.

Fig. 2.12 Watermarked version of "Bavarian Couple".

In all experiments, a watermark length of 1000 was used. They added the watermark to the image by modifying 1000 of the more perceptually significant components of the image spectrum using Equation (2). More specifically, the 1000 largest coefficients of DCT (excluding the DC term) were used. A fixed scale factor of 0.1 was used throughout. Figure 2.13 shows the response of the watermark detector to 1000 randomly generated watermarks of which only one matches the watermark present in Figure 2.12. The positive response due to the correct watermark is very much stronger that the response to incorrect watermarks.

An important open problem is the construction of a method that identifies perceptually significant components from an analysis of the image and the human perceptual system. Such a method may include additional considerations regarding the relative predictability of a frequency based on its neighbors. The latter property is important in combating attacks that that may use statistical analyses of frequency spectra to replace components with their maximum likelihood estimate.

2.3 Adaptive Image Watermarking Scheme Based on Visual Masking

An efficient watermark should be invisible and robust (impossible to remove). We have introduced I. J. Cox's image watermarking scheme in DCT domain. In this section, we will introduce another invisible and robust image watermarking scheme in DCT domain based on a summarized version of the paper proposed by J. Huang and Y. Q. Shi. [5].

They propose an adaptive watermarking algorithm in discrete cosine transform (DCT) domain based on visual masking. They embed watermarks in some significant DCT coefficients so that watermarks cannot be damaged easily. In order to embed

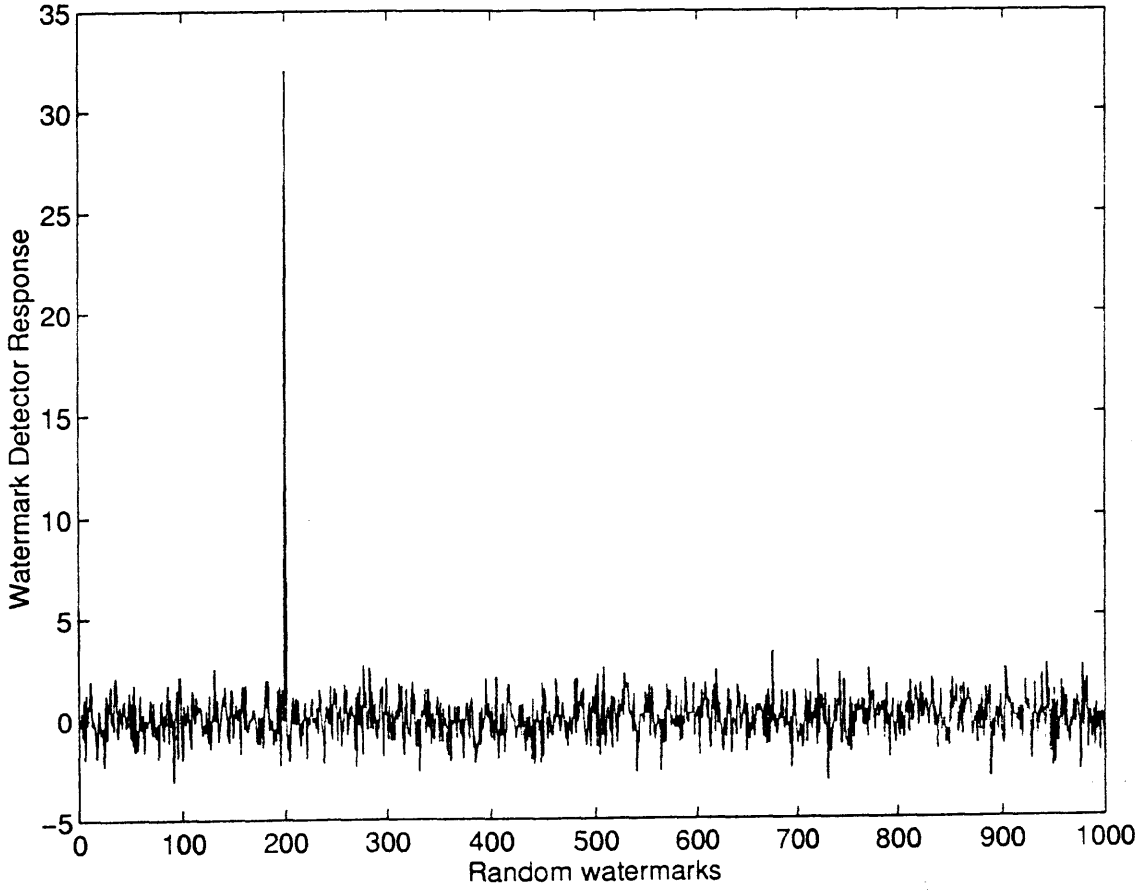


Fig. 2.13 Watermark detector response to 1000 randomly generated watermarks. Only one watermark (the one to which the detector was set to respond) matches that present in Figure 2.12.

watermarks in images under the constraint of being invisible, classification based on visual masking is applied to image blocks prior to watermark embedding. Watermarks are embedded adaptively according to the classification. Simulation results demonstrate that the generated watermarks are very robust against some typical image processing techniques such as compression, low-pass filtering and subsampling.

2.3.1 Overview of the Adaptive Image Watermarking Scheme

The proposed watermarking algorithm is shown in Fig. 2.14. The original image is first split into non-overlapped 8 x 8 blocks and then do DCT for each block. In DCT domain, they estimate the average brightness and texture complexity of each block. Based on the estimation, all blocks are classified into three categories: dark and weak texture (Class 1), bright and strong texture (Class 3), and remains (Class 2). The watermark, a random number sequence, is divided into groups. Each group, having three numbers, is inserted into three significant DCT coefficients of one of the blocks in the image, with different strength: the strongest for Class 3 and the weakest for Class 1. After the modification of DCT coefficients, inverse the DCT transform for each block to obtain the watermarked image.

The detection of watermark is quite straightly forward. That is, with the original image, the possible watermark is extracted from the tested image. The correlation of the extracted watermark and the original watermark is computed and used to decide whether a watermark exists. Watermark embedding and detection are discussed in more detail in the next two sections, respectively.

2.3.2 Watermark Embedding

Watermark embedding is implemented via the following four steps: (i) image splitting and taking DCT; (ii) block classification in DCT domain; (iii) watermark generation and casting; (iv) inverse DCT.

They first split the original image, f(x, y), into non-overlapped 8 x 8 blocks, denoted as B_k, k=0, 1,.... After splitting the image, taking the Discrete Cosine transform of each block (B_k), and then $F_k(u, v)$ is computed.

Embedding a watermark can be viewed as superimposing a weak signal onto a strong background. According to the visual features of human vision system (HVS), there are two observations: (i) the brighter the background, the higher the embedded signal could be (luminance masking) (ii) the stronger the texture in the background, the lower the visibility of the embedded signal would be (texture masking [6]). Based on the visual masking, they classify each B_k into one of three categories; Class1 (dark and weak texture), Class2 (the remains) and Class3 (bright and strong texture).

The average brightness information of the background is represented by DC component in DCT domain. To estimate the texture complexity, they quantize the DCT coefficients using the same method as used in JPEG. Then the number of non-zero coefficients is computed. The larger the number, the stronger the texture in the blocks is.

If $F_k(0,0) < T_1$ and number $\{ \text{int} (F_k(u, v) / Q(u, v)) \neq 0\} < T_2$, then B_k is classified into Class 1. If $F_k(0,0) > T_1$ and number $\{ \text{int} (F_k(u, v) / Q(u, v)) \neq 0\} > T_2$, then B_k is classified into class 3. Otherwise, classified into Class 2. Here, int (·) means taking integer and Q (u, v) denotes quantization step at (u,v). T_1 and T_2 are two predefined thresholds.

The watermark W is chosen to be a random number sequence of length n, i.e., $W = \{x_i, 0 \leq i < n\}$, that draw from the normal distribution N(0,1). To embed the watermark, the DCT coefficients are modified follows:

$$F'_k(u, v) = F_k(u, v) + \alpha_k \cdot x_i \;,\; 3k \leq i < 3(k+1);\; (u, v) \in \{(0,1),(1,0),(1,1)\}$$

$$F'_k(u, v) = F_k(u, v) \qquad ;\ \text{otherwise}$$

Where α_k is a scaling factor, that varies according to the classification mentioned above. Based on many experiments, α_k is set to be 2, 6 and 9 for Class 1, Class 2 and Class 3, respectively.

The watermarked image is obtained by

$$f'(x, y) = \cup_k f'_k(x, y) = \cup_k \textbf{IDCT}\ \{ F'_k(u, v), 0 \leq u, v < 8\}$$

They embed a part of the watermark in first three low-frequency AC components in each block in order that the watermark can be robust. Experiments indicate that if more DCT coefficients are modified for watermark embedding, it may have negative effect on either robustness or invisibility.

2.3.3 Watermark Detection

The watermark can be detected by using correlation technique. Let $F^*_k(u,v)$ denote the DCT coefficients of the corrupted watermarked image in block B_k. The corrupted watermark W* is extracted by

$$W^* = \cup_k W^*_k, \quad W^*_k = \{ x^*_i, 3k \le i < 3(k+1) \}$$

$$W^*_k = F^*_k(u, v) - F_k(u, v), \quad (u, v) \in \{ (0,1), (1,0), (1,1) \}$$

Where x_i^* is the corrupted version of x_i, $0 \le i < n$.

To determine if a watermark exists, we compute the correlation between W* and W:

$$Corr(W^*, W) = \Sigma_i \, x_i^* \cdot x_i \; / \; \Sigma_i \, (x_i^*)^2$$

If $corr(W^*,W) > T_3$, it indicates that there is a watermark existing in tested image. Experiments indicate that response of the detector to false marks does not exceed 4. So T_3 is selected to be 6 in our work.

2.3.4 Simulation Results

To test the proposed algorithm, they generate 2000 random number sequences that draw from the normal distribution N (0,1). The 1010^{th} sequence is utilized as the true watermark. The experimental results on "Lena" image of 256 x 256 are shown in Fig. 2.15 – Fig. 2.18.

Fig. 2.15 demonstrates that the watermark generated with the proposed algorithm is invisible, where Fig. 2.15 (a) and Fig. 2.15 (b) are the original image and watermarked image, respectively. The response of the detector to the watermarked image is shown in Fig. 2.16 (a), where the response to the true watermark is much stronger than to the false watermarks. Fig. 2.16 (b) is the response of the detector to subsampled (2:1 in both

horizontal and vertical directions) watermarked image. Fig. 2.17 shows the reconstructed watermarked image after JPEG compression (0.24 bpp. 26.3 db in PSNR) and the response of the detector to it. Even though the watermarked image is distorted seriously, the watermark can still be detected robustly. Fig. 2.16 – Fig. 2.18 demonstrate the robustness of the watermark against subsampling, compression, and low-pass filtering, respectively.

2.4 Wavelet-Based Watermarking Schemes

A digital watermark should be invisible. It is hidden in an image as a traditional watermark, i. e. to claim the copyright or authorship of an image. Even though the digital watermark is invisible, it is still important to cast the watermark in a way such that it is difficult to remove. A digital watermark system consists of three components: watermark generation, casting and retrieval. In this section, a scheme to search perceptually significant coefficients is developed to hide the watermark. They focus on watermark casting in the frequency domain and to be more specific in the wavelet domain.

A modified and summarized version of the paper "Watermark Design for Embedded Wavelet Image Codec" proposed by P.-C. Su, H.-J. M. Wang and J. Kuo.[7] is introduced as follows.

2.4.1 Significant Coefficient Search

To prevent the watermark from attack, they cast the watermark in significant coefficients after the transform. In general, these coefficients will not change a lot after signal processing or compression attacks. The way to search significant wavelet coefficients is motivated by their previous work on the multi-threshold wavelet codec (MTWC) [8]. The

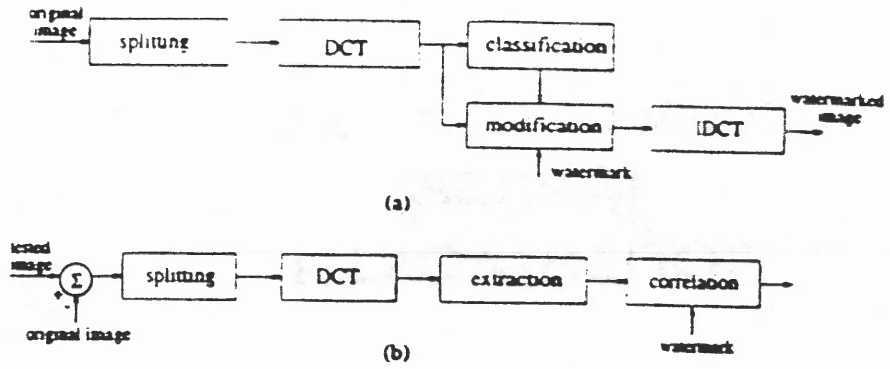


Fig. 2.14 Proposed watermarking algorithm

Fig. 2.15 Demonstration of invisibility (a) Original image (b) Watermarked image. Both are of 256 x 256

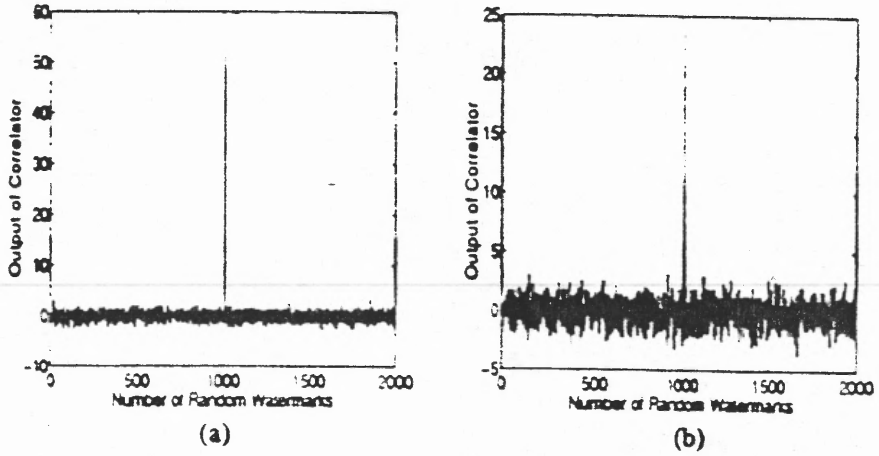


Fig. 2.16 Demonstration of robustness (a) Response of detector to Fig. 2.15 (b) (b) Response of detector to subsampled watermarked image

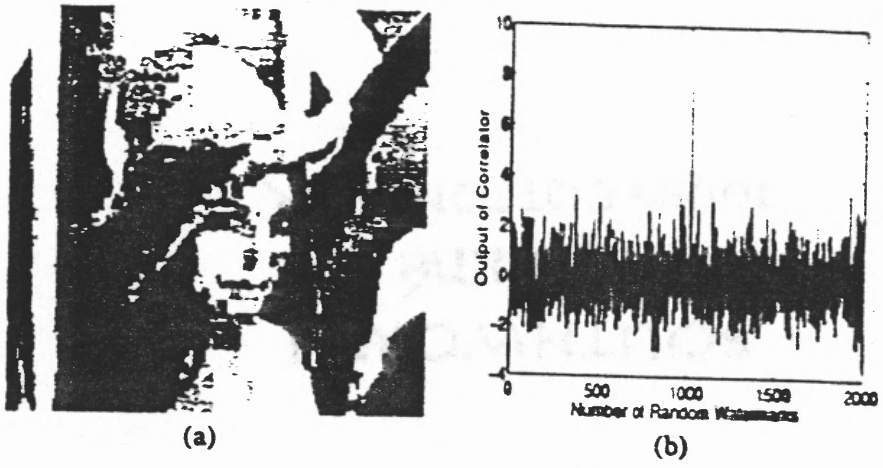


Fig 2.17 Demonstration of robustness (a) Reconstructed watermarked image after JPEG compression at 0.24 bpp, PSNR= 26.3 dB (b) Response of detector

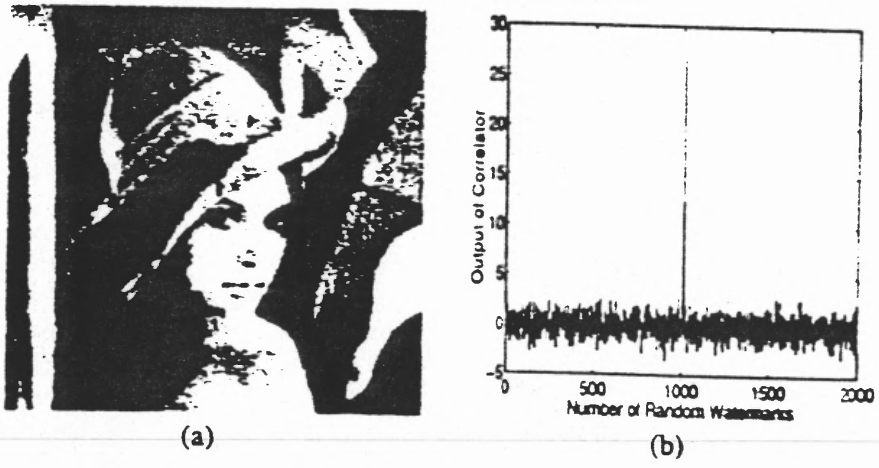


Fig 2.18 Demonstration of robustness (a) Mean-filter watermarked image (b) Response of detector

successive subband quantization (SSQ) scheme was adopted in MTWC to choose perceptually significant coefficients for watermark casting. They can find out which subband contains the most significant coefficients by successive subband quantization (SSQ) from MTWC [9] [10] and search significant coefficients in the selected subband only. Each coefficient in subband s is presented by C_s (x, y) and T is the initial threshold of subband s and calculated by

$$T = C_{max,s} / 2,$$

Where $C_{max,s}$ is the maximum absolute coefficient value in subband s.

The significant coefficient searching scheme is designed as follows.

1. Initialization: Set the initial threshold Ts of each subband to one half of its maximum absolute value of coefficients inside the same subband. Set all coefficients un-selected.
2. Select the subband (except the DC term) with the maximum value of $\beta_s \times T_s$, where β_s is the weighting factor of subband s. For the selected subband, they examine all un-selected coefficients $C_s(x, y)$ within the current threshold T_s and choose those coefficients which are greater than T_s to be significant coefficients.
3. Update the new threshold in subband via $T_s^{new} = T_s /2$.
4. The watermark is cast in the selected significant coefficients obtained in Step 2.
5. Repeat Step 2 to Step 4 until all watermark symbols are cast.

Note that they do not cast the watermark in the DC term since it may lead a serious fidelity loss in the protected image. The value of β_s is used to control the tendency of protected frequency region. In their experiments, they set all of β_s to 1.0 expect for the weighting factor β_0 for the DC term, which is set to 0.

2.4.2 Adaptive Watermark Casting and Retrieval

Figure 2.19 shows the block diagram of invisible watermark embedding. They perform the watermark casting by using the spread spectrum technique. The formula of watermark casting is

$$C_s'(x, y) = C_s(x, y) + \alpha_s \beta_s T_s W_k, \quad (1)$$

Where C is the selected original coefficient. C′ is the watermarked coefficient, and T_s is the current threshold of subband s. W_k is the *k*th watermark element in a watermark sequence of length N_w. W_k takes values between 1 and – 1. The value of α_s and β_s are scaling factors. The value of α_s is adjustable by the user to increase (or decrease) the watermarked image fidelity and decrease (or increase) the security of watermark protection. It is chosen that $\alpha_s \in (0.0, 1.0]$. The parameter β_s is used to control the subband selection order. For example, a larger value of β_s in higher frequency subbands can give significant coefficients in the higher frequency components a higher priority.

For the watermark detection part, the extracted error can be written as

$$E^*_{s,k}(x, y) = C^*_{s,k}(x, y) - C_s(x, y),$$

Where $C^*_{s,k}$ is the coefficient coming from the possible attacked image I* and C_s is the coefficient of the original image I. The detection is performed by computing the similarity between C* and C as

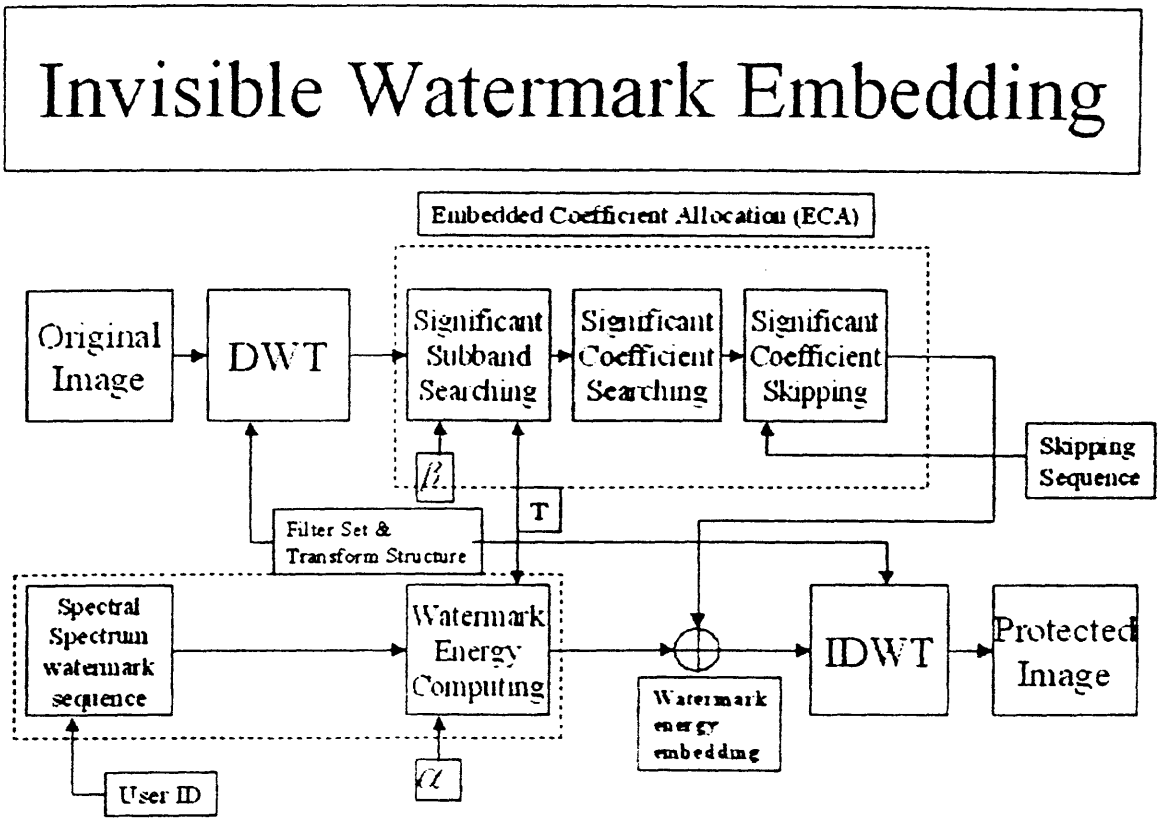


Fig. 2.19 The block diagram of invisible watermark embedding

$$SIM(I^*, I) = N_w (\Sigma E^*_{s,k}(x, y) \cdot E_{s,k}(x, y)) / (\| E^*_{s,k}(x, y) \| \| E_{s,k}(x, y) \|), \quad (2)$$

Where N_w is the amount of watermark symbols. In the experiment, they choose the length of the watermark sequence to N_w = Height × Width / 32. $E_{s,k}(x, y)$ is the original watermark and $E^*_{s,k}$ is the attacked watermark with respect to wavelet coefficient $C_s(x, y)$. Note that they use the inner product of $E^*_{s,k}$ and $E_{s,k}$ normalized by their norms as the similarity measure, which measures the cosine function of the angle of two vectors. Any similarity measure less than zero is treated as zero, where these two vector $E^*_{s,k}$ and $E_{s,k}$ are actually in the reverse direction. The maximum value of the similarity is 1 if the watermark is perfectly extracted.

2.4.3 Watermark Protection

Four watermark protection types are described as follows.

An attacker who knows this algorithm very well but without the watermark information (the selected wavelet transform structure and the transform level L) could damage the watermark with little probability by destroying selected significant coefficients directly.

Different wavelet filters and different numbers of transform level will produce a different energy compaction result. In other words, it will lead to different values of T_s and different locations of significant coefficients. As a result, the exact locations of significant coefficients are difficult to estimate by the attacker.

To increase watermark's robustness, a significant coefficient skipping scheme is used to achieve the non-invertible function. A random integer sequence with a value from

0 to M is generated by another seed to indicate the skipping number. If the skipping value is Q, we skip Q selected significant coefficients before watermark insertion.

The initial threshold of each subband plays an important role in watermark casting. It is important to prevent the initial threshold value of each subband from being taken by the attacker. The proposed algorithm do not embed any watermark in the maximum coefficient of each subband $C_{max,s}$ that is used to calculated the initial threshold.

2.4.4 Experimental Results

They performed the watermark protection on the Lena image of size 512 x 512 with N = 8,192 and α = 1.0. In Figure 2.20(a), they show the watermark retrieval result without any attack. They compare the similarity performance with other 1,000 different watermark sequences. They show the watermark retrieval results after the soften, sharpen and median filtering attacks in Figure 2.20 (b), (c) and (d). Watermark retrieval results after 6 x 6 block mosaic low pass filter attack and 50% uniform random noise addition attack were shown in Figure 2.21 (a) and (b) respectively. It is clear that the embedded watermark with ID number 450 is retrieved successfully under these attacks. The proposed algorithm can survive well under a DCT-based JPEG compression attack. The experimental result after the JPEG compression with the quality factor set to 5% is shown in Figure 2.21 (c). This is equal to around 46:1 compression ratio. Moreover, Figure 2.21 (d) shows that their algorithm can survive even under a 512:1 compression attack by wavelet-based compression codec.

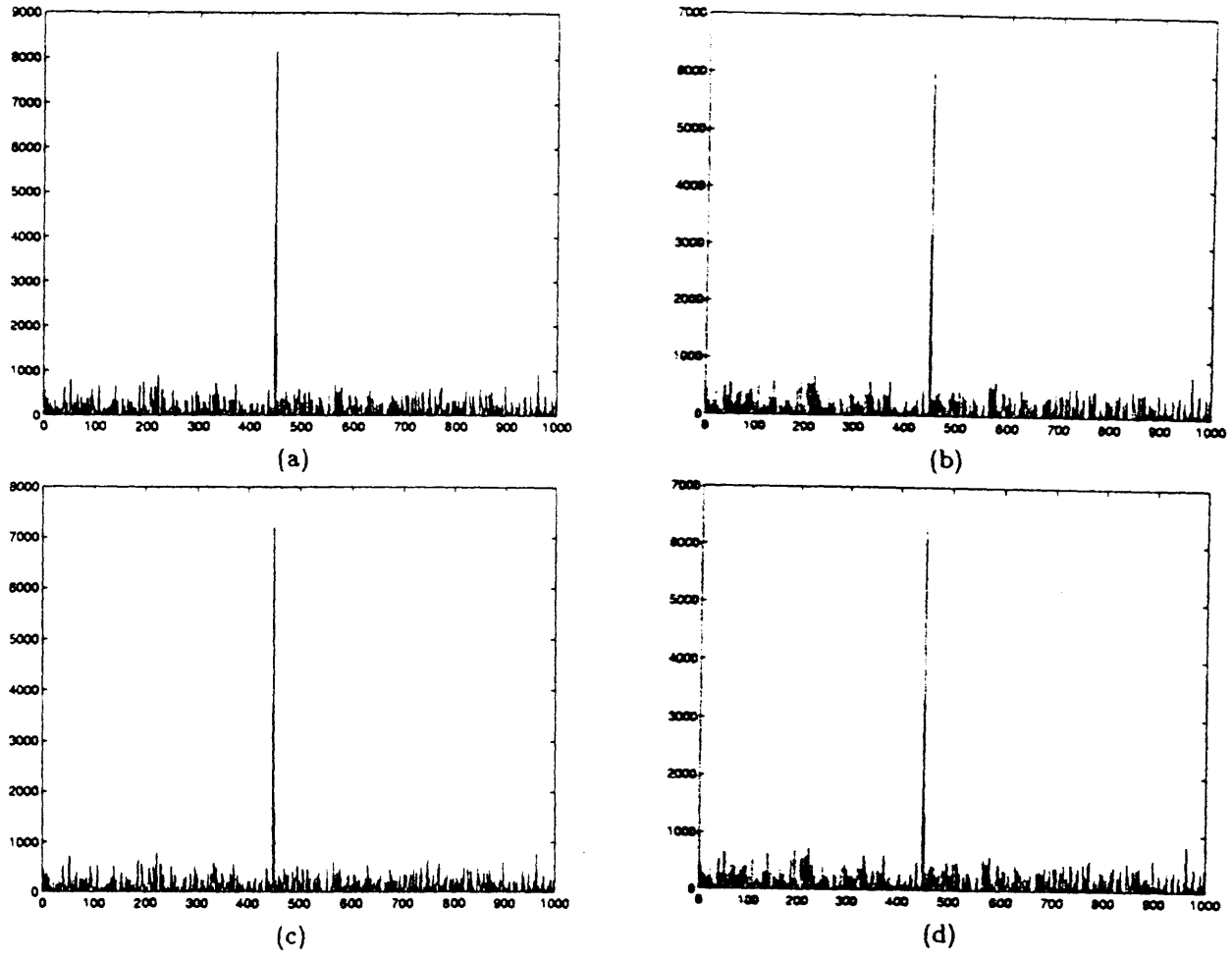


Fig. 2.20 Watermark retrieval from the protected 512 x 512 gray-level Lena image after (a) no attack, (b) the soften filtering attack, (c) the sharpen filtering attack, and (d) the medium filtering attack, where the y-axis represents the similarity measure and the x-axis denotes the ID number of different watermark sequences. The ID of the inserted watermark sequence is 450.

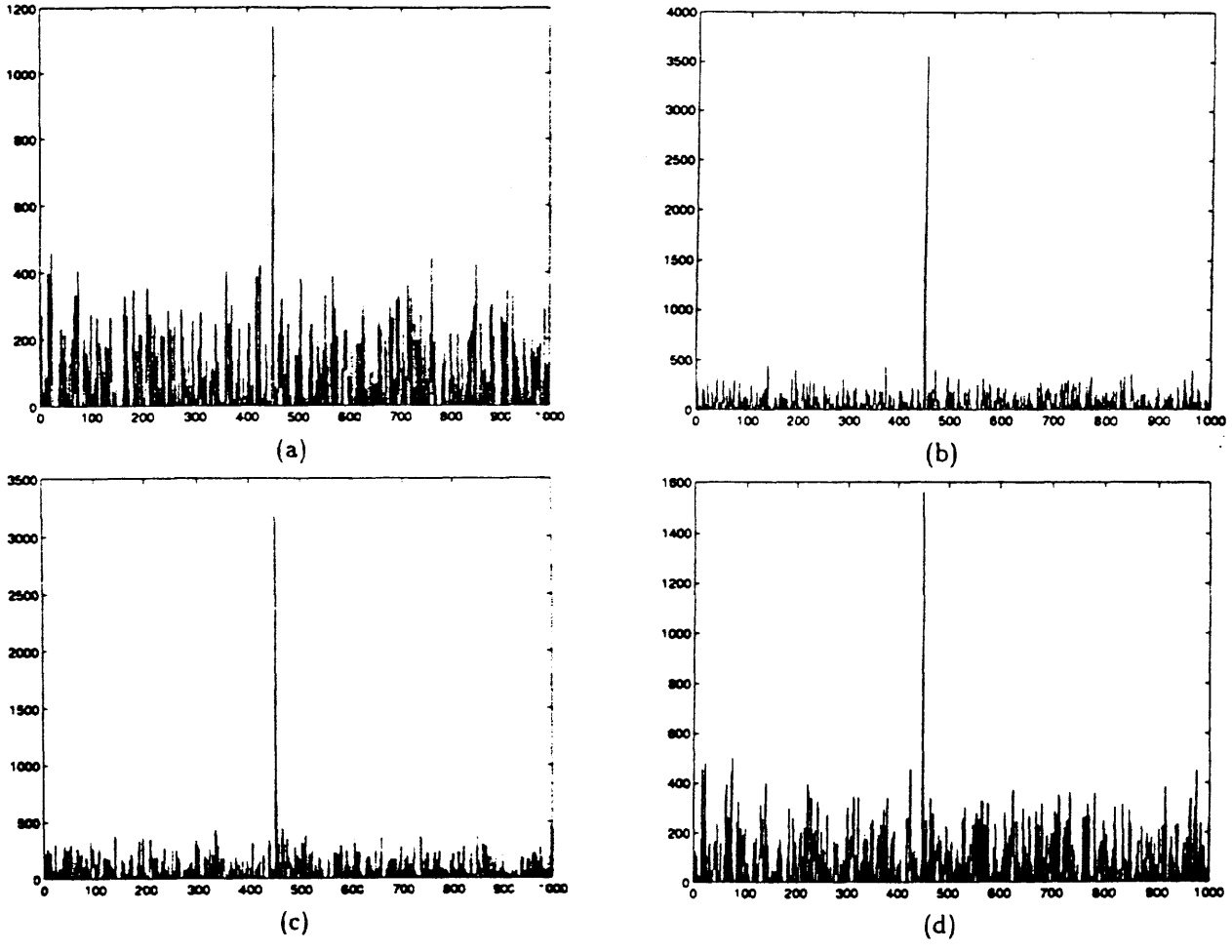


Fig. 2.21 Watermark retrieval from the protected 512 x 512 gray-level Lena image after (a) the 6 x 6 block mosaic attack, (b) the 50% uniform random noise attack, (c) the JPEG compression attack with 5 % quality factor setting, and (d) the 512:1 compression attack with SPIHT.

2.5 Fractal Compression Watermarking Scheme

The rapid growth of digital network has required the needs for copyright protection tool from piracy. We have discussed the binary encoding scheme and multilevel encoding scheme applied on DCT domain and DWT domain of still image. In this section, we will introduce a new scheme based on fractal coding and decoding.

A modified and summarized version of the paper "Using fractal compression scheme to embed a digital signature into an image " proposed by Joan Puate, and Fred Jordan. [11] is described as follows.

In general terms, a fractal coder exploits the spatial redundancy within the image by establishing a relationship between its different part. They describe a way to use this relationship as a means of embedding a watermark.

The fractal theory has proved to be suitable in many fields and particularly interesting in various applications of image compression. M. F. Barnsley introduces for the first time the term of Iterated Function Systems (IFS) [12] [13][14][15], based on the self-similarity of fractal sets. Barnsley's work assumes that many objects can be closely approximated by self-similarity objects that might be generated by use of IFS similar transformations. From this assumption, the IFS can be seen as relationship between the whole image and its parts, thus exploiting the similarities that exist between an image and its smaller parts.

2.5.1 Fractal Image Algorithm

The main idea of a fractal based image coder is to determine a set of contractive transformations to approximate each block of the image (or a segment, in a more general

sense), with a larger block. Some basic aspects of the theory are given in the lines below (a clear and brief explanation can be found in [16][17]).

Consider a metric space (£ , d) where d is a given metric and £ might be the space of the digital images. An IFS consists of a complete metric space (£ , d) and a number of contractive mappings β_i defined on £. The fractal transformation associated with an IFS is defined as:

$$A = B(A) = \cup_i \beta_i (A)$$

And

$$\lim B^n(E) = A$$

A is called the attractor of IFS and the transformations are usually chosen to be affine. Once B is determined, it is easy to get the decoded image by making use of the Contraction Mapping Theorem: the transformation B is applied iteratively on any initial image until the succession of image does not vary significantly.

However, given a set M, how to find a contractive transformation B such that its attractor A is close to M? That is to say that they can guarantee that M and A will be sufficiently close if we can make M and B(M) close enough.

In terms of β_i , and combining the two following expressions:

$$B(M) = \cup_i \beta_i (M)$$

They get $\cup_i \beta_i (M) \sim M$

So, they make a partition of M:

$$M = \cup_i m_i$$

Then, m_i can be closely approximated by applying a contractive affine transformation on the whole M:

$$m_i = \beta_i (M)$$

The theory of IFS was extended to Local IFS where each part of the image is approximated by applying a contractive affine transformation on another part of the image:

$$m_i = \beta_i (D_i)$$

where D_i is the bigger part from which m_i is approximated.

2.5.2 Fractal Image Coder

The main idea to automate the searching of a Local IFS relies on a partition of the image in blocks of a fixed size, called Rang Blocks. These blocks are then approximated from larger blocks, called Domain Blocks. The transformations normally applied on the Domain Blocks are contracting and luminance scaling and shifting. Some other isometric transformations are sometimes used.

Let O denote the image we want to encode. Let also O_r denote a partition of O in

n x n blocks referred to as Range Blocks (Rb). Similarly, O_d will denote another partition of O, this time in 2n x 2n blocks or Domain Blocks (Db) in steps of n x n pixels. The goal of the encoding algorithm is to establish a relationship between O_r and O_d in such a way that any Rb can be expressed as a set of transformations to be applied on a particular Db.

The transformations that have been considered are Contraction, Isometric transformation, Luminance Scaling and Luminance Shifting. For each Rb in O, denoted as Rb_j , the code will consist of a vector V_j and the appropriate transformations T_j , in such a way that:

- V_j has its origin in Rb and points to the correspondent Db_j which now becomes its Matching block (Mb_j).
- T_j If applied on Mb_j , minimizes the Mean Square Error (MSE) with respect to Rb_j .
- The couple { V_j, T_j } is the best solution (in the sense of the MSE) within a area surrounding Rb_j in which we search for Mb_j.

The region of O_d where the search of Mb_j is performed is commonly taken as a square region surrounding the Rb_j. They name this region LSR (Local Searching Region). The use of such in the Matching Block determination might be justified from spatial redundancies considerations and that is essentially true. But that does not mean that other shapes can not give more than acceptable results on the Ranges Blocks approximation. Figure 2.22 shows the square surrounding region and a possible alternative:

2.5.3 Fractal Image Decoder

Consider an initial image S with the only constraint that it has to be of the same size as O. As before, they consider a n x n partition S_r in Rb, and a 2n x 2n partition S_d in Db. The

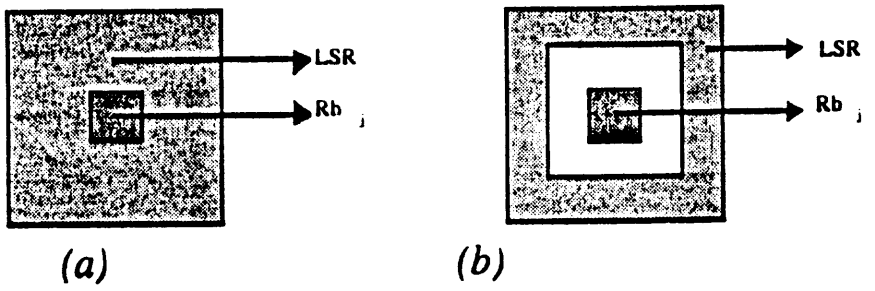


Fig. 2.22 (a) A square LSR and (b) an alternative solution

decoding algorithm takes for each Rb_j its Matching Block (pointed by V_j), applies on it the transformations (defined by T_j) and places the result back on Rb_j . These operations are performed for every Rb_j of S_r and through some iterations (typically four). After each iteration, a new image Q_i is obtained and it turns out that Q_i converges to O. An important point is that the solution $\{V_j , T_j\}$ obtained for O remains exactly the same for Qi . That is to say that for every Range Block the same Matching Block as before is found. They also take advantage of this point to embed the signature by a properly choice of the vectors V_j.

Figure 2.23 shows some iterations for image Lena, when S_0 has been taken a black image, n being equal to 4. Figure 2.24 shows some iterations for image Lena, S_0 being a black image and n equal to 8.

I t can be observed a better quality when n=4, above all in those parts of great detail. However, for n=4 the compression rate is much lower than for the case n=8. Therefore, there is choice to be made between quality and rate of compression.

Signing an image consists of a coding-decoding process with variable searching region. Consider two different LSR, A and B (Fig. 2.25), and a third one, C defined as their union (a different choice of the regions could have been made, perhaps as a function of the characteristics of the image). Let also $S=\{s_0, \ldots, s_{31}\}$ be a 32 signature. They embed every bit with a redundancy U. The coding process is as follows:

- For each bit s_i , U Range Blocks are randomly chosen and denoted by $\{Rb\}_i$. The random function used to get the blocks makes use of a "seed" that should only be known by the user.
- If $s_i = 1$, $\{Rb\}_i$ is coded by searching for $\{Mb\}_i$ in regions $\{A\}_i$.

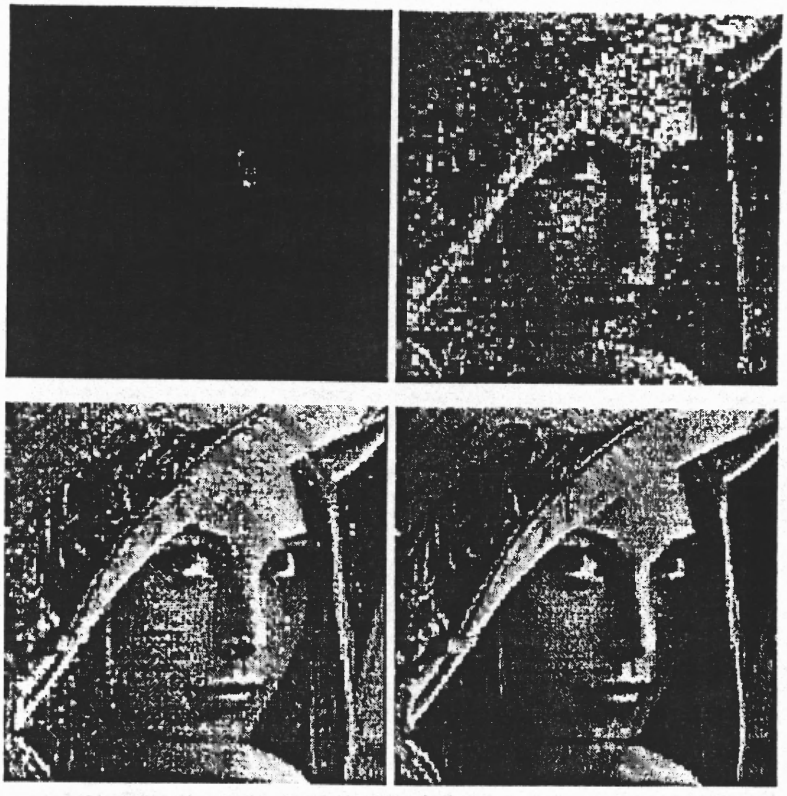


Fig. 2.23 Iterations 1,2,4 of a code for "Lena" applied on a gray image (n=4).

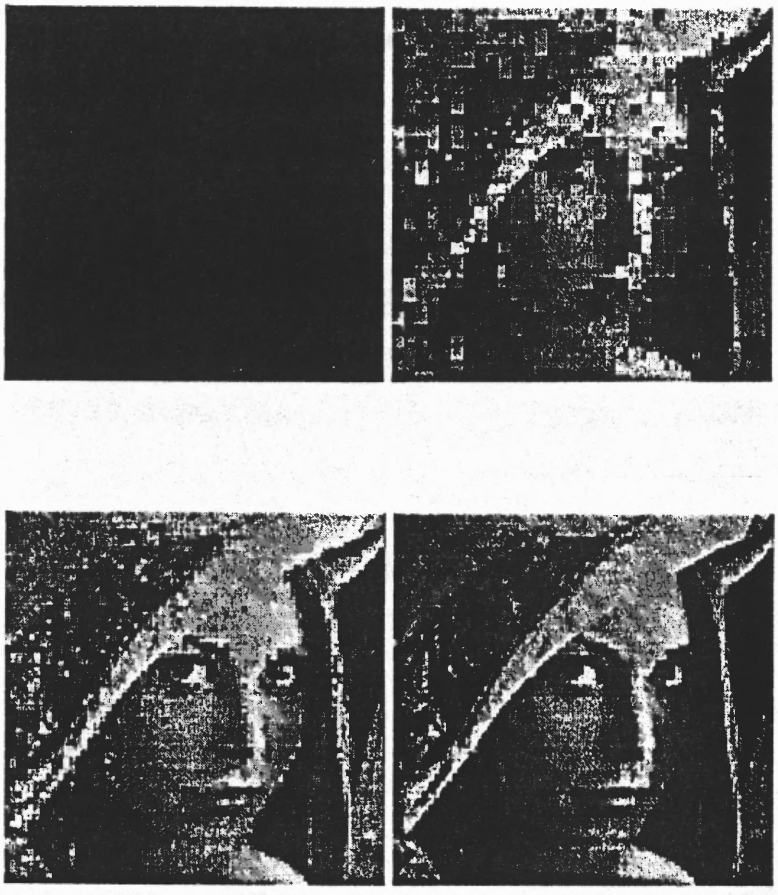


Fig. 2.24 Iterations 1,2,4 of a code for "Lena" applied on a gray image (n=8).

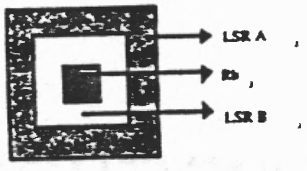


Fig. 2.25 A range Block, its LSR_A and LSR_B is defined as their union.

- If s_i =0, $\{Rb\}_i$ is coded by searching for $\{Mb\}_i$ in regions $\{B\}_i$.
- The rest of Rb_j are coded by searching for Mb_j in $\{C\}_i$. Note that this would be the case for all Range Blocks in a non-signed image.

The decoding is performed as described above. The resulting image contains the signature.

2.5.4 Retriever

The whole fractal code of an image can be expressed as the union of every Range Block single code:

$$M = \cup_j \{V_j, T_j\}$$

Likewise, for an image Q obtained after an iterative application of μ on any initial image, they consider its fractal code π. Since, in Q, every Range Block is not just an approximation to a transformed Domain Block but it is exactly a transformed Domain Block, it turns out that $\pi = \mu$.

Thus, they are able to identify the signature by simply accessing the Range Blocks of Q defined by the "seed" used when signing, recoding them and checking the values of V_j.

The rule to decide if a Range Block has signed with a zero or a one, is the following one:

- If V_j belongs to region A_j, then a one has been embedded.
- If V_j belongs to region B_j, then a zero has been embedded.

In normal conditions, there ought to be a number of U recognition of bit one for those bits one in the signature, and of U recognition of bit zero for those bits zero in the signature.

To make the final decision there is a need of a threshold. It is going to define as the mean of results obtained for bits two and three of the signature. Thus, they are always being embedded as a one and a zero, respectively.

2.5.5 Experimental Results

First one concerns the case for n=4, and second one the case for n=8. In both, the robustness to JPEG compression and to low pass filtering (3 x 3 –kernel blurring) is discussed, as well as the quality of the signed image against the original and non signed but fractal encoded.

All tests have been performed by embedding a 32-bits signature in "Lena" image (256 x 256), then applying the retriever. The chosen signature has a value of one in even bits and zero in odd bits. The redundancy U is equal to 50 for the case n =4, and equal to 25 for the case n=8.

Figure 2.26 shows, for a 'n' equal to 4, the original image 'Lena', the decoded image of 'Lena' with no signature, and the decoded image of 'Lena' with the signature. Both, the peak to Signal Noise Ratio (PSNR) between original and signed image and that between original and non-signed image present a value higher than 31.5 db.

They have tested the robustness against JPEG compression qualities of 90, 75 and 50 %. Figure 2.27 shows the original image 'Lena', the decoded image of 'Lena' with no

(a) (b) (c)

Fig. 2.26 (a) Original "Lena" image; (b) decoded image of "Lena" with no signature; (c) decoded image of "Lena" with the signature (n=4)

(a) (b) (c)

Fig. 2.27 (a) Original "Lena" image; (b) decoded image of "Lena" with no signature; (c) decoded image of "Lena" with the signature (n=8)

signature, and the decoded image of 'Lena' with the signature. The PSNR between the original image one and the signed decoded is equal to 25.40 dB (eighth iteration).

2.6 Video Watermarking

A digital watermark is a signal carrying information that is embedded into another transport signal, for example into a video signal. A watermarking scheme for video broadcast applications should comply more requirements:

1. A digital watermarking should be invisible and robust.
2. For broadcast application, it can be assumed that the broadcaster will usually store the video in compressed format. Therefore, it must be possible to incorporate the watermark into the encoded, i.e., into the bitstream.
3. Watermarking in the bit stream domain may not increase the bit-rate. This requirement is not obeyed by previous publications dealing with watermarking of still images during JPEG compression.

It is in practice that incorporating a watermark into compressed video has to obey much more constraints than incorporating a watermark into uncompressed video. Therefore, it is advantageous to do so in the domain of uncompressed video wherever possible. A modified and summarized version of the paper "Digital Watermarking of Raw and Compressed Video " proposed by Frank Hartung and Bernd Girod. [18] is introduced as follows.

2.6.1 Digital Watermarking of Raw Video

It is practical that incorporating a watermark into compressed video has to obey much more constraints than incorporating a watermark into uncompressed video. Hence, the watermarking algorithm should work interoperable for compressed and uncompressed video with the same type of decoder, that is, watermark detector (see Fig. 2.28).

The basic idea of watermarking for video is addition of a pseudo-random signal to the video that is below the threshold of perception and that cannot be identified and thus removed without knowledge of the parameters of the watermarking algorithm.

Their approach to accomplish this is a direct extension of ideas from direct-sequence spread spectrum communication. The approach is similar and was developed independently. They denote

$$a_j \ , \ a_j \in \{-1,1\} \tag{1}$$

a sequence of information bits they want to hide in video stream. They spread this discrete signal by a large factor cr, called the chip-rate, and obtain the spread sequence

$$b_i = a_j \ , \qquad j \cdot cr \leq i < (j+1) \cdot cr \tag{2}$$

The spread sequence b_i is amplified with an amplitude factor α and modulated with a binary pseudo-noise sequence

$$p_i \ , \qquad p_i \in \{-1, 1\} \tag{3}$$

The modulated signal, i. e., the watermark $w = \alpha \cdot b_i \cdot p_i$ is added to the line-scanned digital video signal v_i yielding a watermarked video signal

$$v_i' = v_i + \alpha \cdot b_i \cdot p_i \tag{4}$$

Due to the noisy nature of p_i , w_i is also a noise-like signal and thus difficult to detect, locate and manipulate. The recovery of the hidden information is easily accomplished by multiplying the watermarked video signal with the same pseudo-noise sequence p_i that was in the coder:

$$s_j = \sum_i p_i \cdot v_i' = \sum_i p_i \cdot v_i + \sum_i p^2_i \cdot \alpha \cdot b_i \tag{5}$$

The first term on the right-hand side of (5) vanishes, if

$$\sum_i p_i = 0 \tag{6}$$

(i. e., the pseudo-noise sequence contains as many – 1's in 1's in the interval $(j \cdot cr \ldots (j+1) \cdot cr)$), and p_i are v_i uncorrelated and therefore $\sum_i p_i \cdot v_i = 0$. In practice however, the sum in (6) is not zero, and a correction term

$$\Delta = - (\sum_i p_i) \cdot \text{mean } (v_i'), \tag{7}$$

which account for the different number of -1's and 1's in the pseudo-noise sequence, has to be added. s_j then ideally becomes

$$s_j = \sum_i p_i \cdot v'_i + \Delta \approx cr \cdot \alpha \cdot a_j \quad (8)$$

and the recovered information bit a'_j is

$$a'_j = \text{sign}(s_j). \quad (9)$$

A condition for the scheme to work is that for demodulation the same pseudo-noise sequence p_i is used that was used for modulation. Thus, even if the receiver knows the basic scheme, it can not recover the information without knowledge of the pseudo-noise sequence and its possible shift, For simplicity, they have assumed a binary pseudo-noise sequence in (3). Non-binary PN sequences are also possible without modifications of the scheme, and are in fact favorable in terms of security. Given several sequences with different watermarks, it is easier to figure out the unwatermarked pixel values if the watermark consists only of –1's and 1's. The amplitude factor can be varied according to local properties of the image and can be used to exploit spatial and temporal masking effects of the human visual system (HVS).

2.6.2 Digital Watermarking of Compressed Video

Consider a block of 8 x 8 samples, originating from a frame of the sequence for I-frames or from a prediction error signal for P- and B- frames, originating from a frame of the

sequence for I-frames or from a prediction error signal for P- and B- frames, respectively. The block is transformed with DCT, quantized, zig-zag-scanned and run-level-encoded with VLC codewords for the (run, level) –pairs. Thus, the block of 8 x 8 samples translates into a codeword representing the DC coefficient followed by a number of VLC codewords representing (run, level) pairs and specifying position and value of one DCT coefficient each. The (run, level) –codewords in MPEG-2 are fixed. Fig. 2.29 shows the number of bits for the (run, level)-codewords specified in the MPEG-2 VLC tables. (run, level) –combinations that not specifically represented in the VLC tables are coded with a codeword of 24 bits. In order to add a watermark, they process the encoded video signal block by block. For each signal block, the watermarking procedure consists of the following steps:

1. Calculate the DCT of the watermark (of the spread information bits modulated by the pseudo-noise sequence) for the 8 x 8 block. Do a zig-zag-scan, yielding a 1 x 64 vector of re-scanned DCT coefficients. Denote the DCT coefficients by W_n with W_0 being the DC coefficient and W_{63} being the highest-frequent AC-coefficient. Denote the DCT coefficients of the unwatermarked signal V and of the watermarked signal V'_n.
2. DC-coefficient: For the DC-coefficient, $V'_0 = V_0 + W_0$, that is, the mean value of the watermark-block is added to the mean value of the signal-block.
3. AC-coefficients: Search the bitstream of the coded signal for the next VLC codeword, identify the (run, level)-pair (r_m , l_m) belonging to that codeword and , thus, the position and amplitude of the AC DCT coefficient v represented by the VLC codeword.

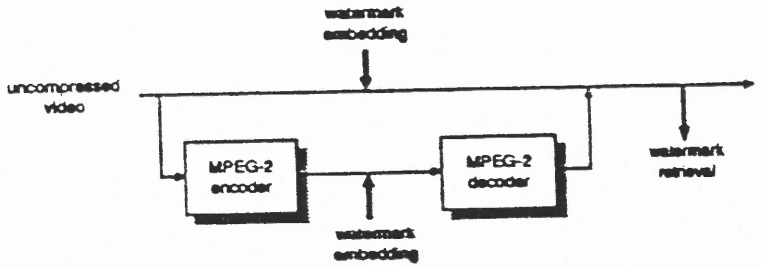


Fig. 2.28 Interoperability of watermarking in the uncoded and coded domain.

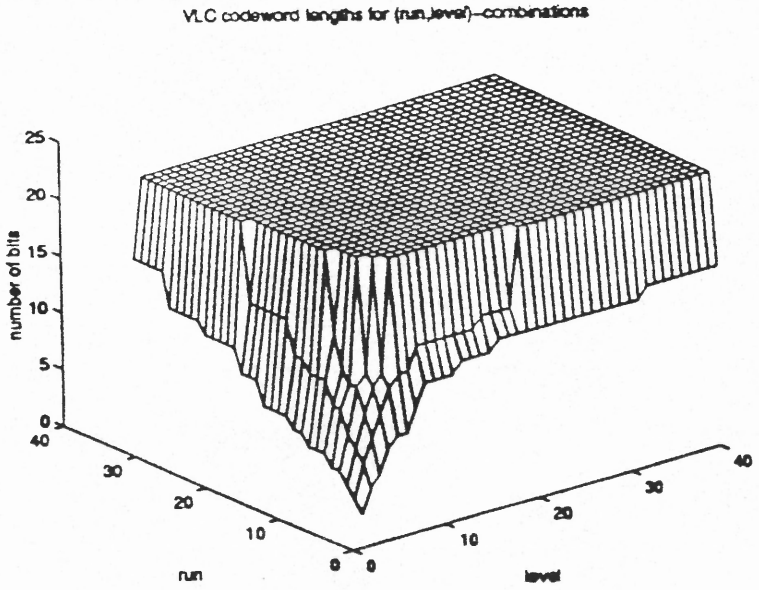


Fig. 2.29 MPEG-2 VLC codeword lengths for (run, level) codewords.

4. $V'_m = V_m + W_m$ is the candidate DCT coefficient for the watermarked signal. However, they do also have the constraint of not increasing the bit-rate. Thus, they have to check the number of bits they have to transmit for the watermarked DCT coefficient V'_m versus the bit-rate they have to transmit for the unwatermarked DCT coefficient V_m :
5. Let R be the number of bits used for transmitting the codeword for (r_m , l_m) (i.e., for V_m) and R″ be the number of bits used for transmitting the codeword for (r_m , l''_m) (i.e., for V''_m) . (R and R″ are determined by the VLC-tables defined in MPEG-2[10]).
6. If the bit-rate shall not be increased and $R \geq R''$ (or if the bit-rate of the video may be increased, unconditionally), transmit the codeword for (r_m , l''_m) . Else, transmit the codeword for (r_m , l_m) .
7. Repeat steps 3 to 6 until an end_of_block (EOB) codeword is encountered.

Due to the bit-rate constraint, usually only few DCT coefficients of the watermark can be incorporated per 8 x 8 block, in a lot of cases (especially for coarse quantization) it might be only the DC coefficient as outlined in step 2. As a result, the watermarking scheme in the bitstream domain is less robust than its counterpart in the pixel domain. In other words: in the bitstream domain, only a fraction of the signal energy of the watermark can successfully be embedded.

2.6.3 Implementation and Simulation Results

They have implemented the outlined as a C program which takes an MPEG-2 bitstream as its input. The program decodes the video simultaneously parses the bitstream and

writes it to a new file. Only those parts of the bitstream containing VLC codewords representing DC- and AC- coefficients of DCT blocks are located and replaced by VLC codewords representing DC- AND ac- coefficients of the same block plus watermark. Typical parameter are α =1...5 and cr =10,000...1,000,000, yielding data rates for the watermark of 1.25...125 bytes/second for NTSC TV resolution. The complexity, as shown in Fig. 2.30, is much lower than the complexity of a decoding process followed by watermarking in the pixel domain and re-encoding. For comparison, the complexity of decoding alone is also given. Fig. 2.31-2.33 show an example frame from a video sequence. Fig. 2.31 shows the original frame without compression and a detail from the hand of the hand of the table tennis play. Fig. 2.32 shows the same frame after MPEG-2 encoding and decoding and without an embedded watermark. Fig. 2.33 finally shows the compressed frame with an embedded watermark. As can be seen, the watermark results in slightly changed pixel amplitudes which are however not visible except in direct comparison to the unwatermarked image. The degradation can directly be influenced by varying the amplitude of the watermark. A higher amplitude leads to better robustness, but possibly results in visually annoying distortions.

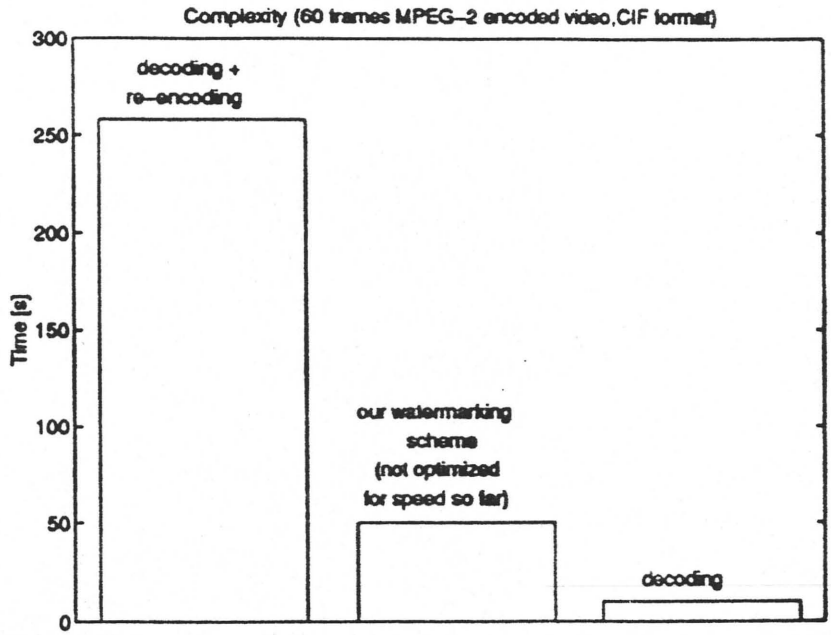


Fig. 2.30 Complexity of our watermarking scheme compared to encoding and decoding.

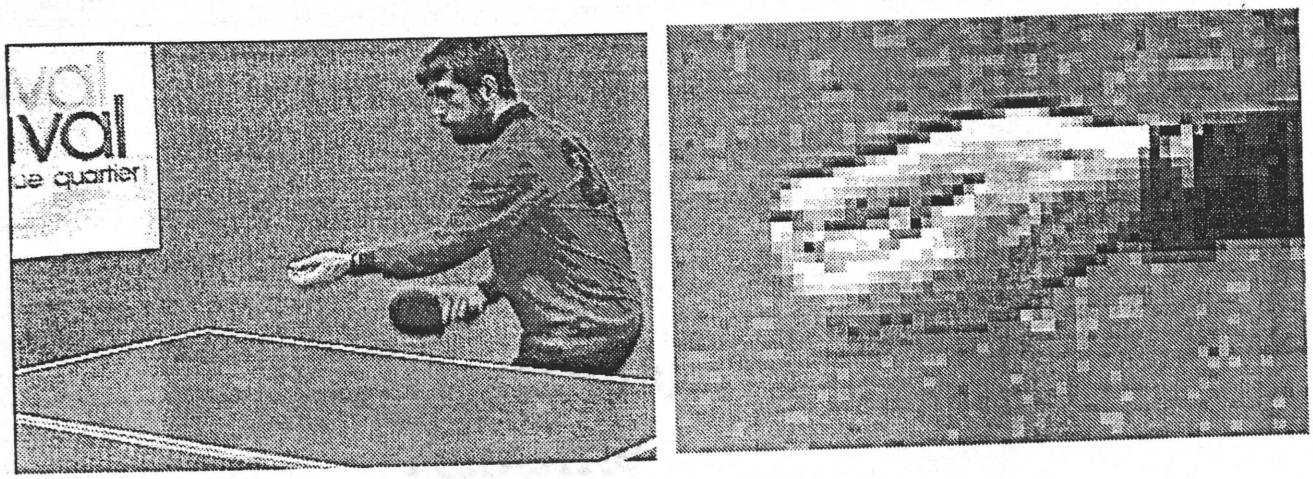


Fig. 2.31 Original

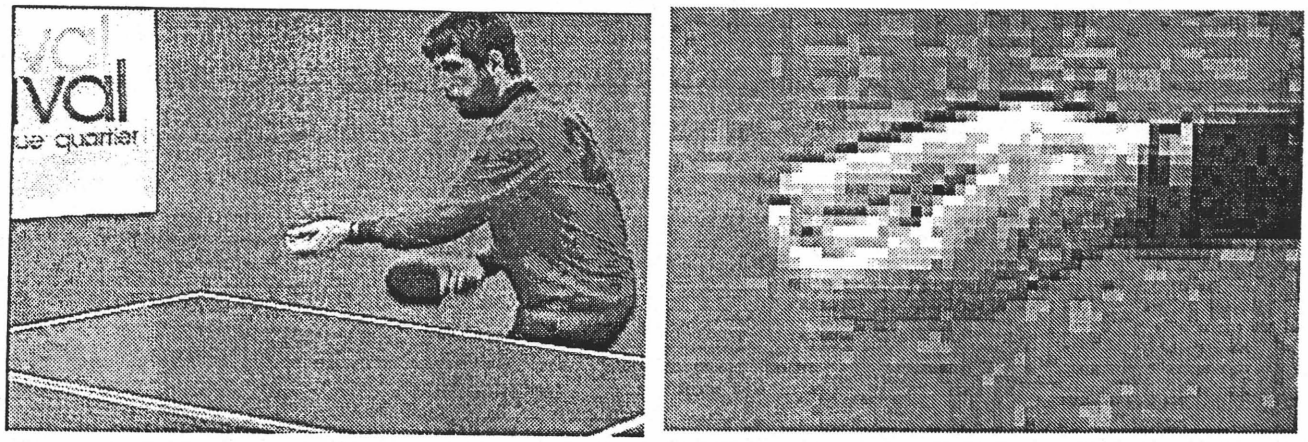


Fig. 2.32 MPEG-2 coded, without watermark

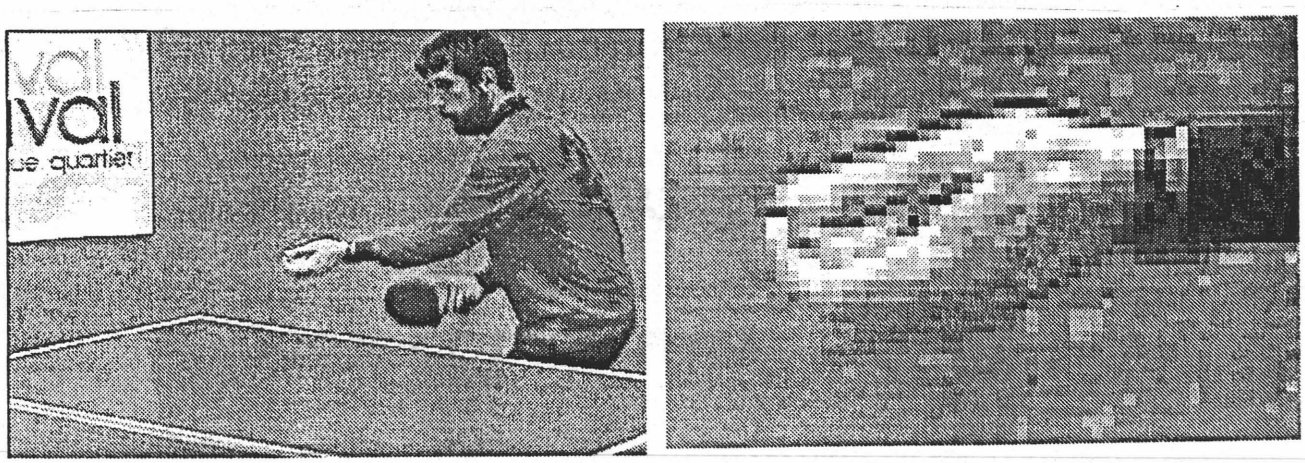


Fig. 2.33 MPEG-2 coded, with embedded watermark

CHAPTER 3

THE DISPUTE OF RIGHTFUL OWNERSHIP FOR INVISIBLE WATERMARK

In general, there are two types of digital watermarks (signatures) addressed in existing literature: visible and invisible watermarks. These watermarks are developed mainly for two purposes: copyright protection and data authentication. We will discuss and summarize the version of the IBM Research Report proposed by S. Craver, N. Memon, B.-L. Yeo and M. Yeung. [19]. In this paper, they focus on the large class of invisible watermarks developed for copyright protection – to identify the rightful owner. In this case the ownership labels which are embedded in an image have to be recoverable despite intentional or unintentional modification of the image. They show that current invisible watermarking schemes can not resolve rightful ownership of any image watermarked with multiple signatures (labels).

3.1 Watermarking of Images: Definitions and Formulations

They defined a generalized formulation of invisible watermarking schemes as follows. They denote an image by I, a signature S =$\{ s_1, s_2, \ldots\}$ and the watermarked image by I′. *E* is an encoder function if it takes an image I and a signature S, and generates a new image which is called the watermarked image I′, i.e.,

$$E(I,S) = I'. \qquad (1)$$

A decoder function D takes an image J (J can be a watermarked or un-watermarked image, and possibly corrupted) whose ownership is to be determined, and recovers a

signature S′ from the image. In this process, an additional image I can also be included into the original (and un-watermarked) version of J.

$$D(J, I) = S'. \qquad (2)$$

The extracted signature S′ will then compared with the owner signature sequence by a comparator function C_δ, and a binary output decision is generated. It is a 1 if there is a match and 0 otherwise:

$$C_\delta(S', S) = 1, \quad c \geq \delta;$$

$$C_\delta(S', S) = 0, \quad \text{otherwise.}$$

Here, c is the correlation of the two signatures.

The above framework describes what an invisible watermark is and how it can potentially be used to determine ownership. This is a generalized formulation. It does not give any insight into how exactly a watermarking scheme works. In view of this, here they specify some watermarking schemes. In particular, they describe the formulation of a class of invisible watermarking schemes, which they call feature-based watermarking schemes, that embed a signature $S = \{ s_1, s_2, \ldots \}$ into some set of derived features $D(I) = \{ f_1(I), f_2(I), \ldots \}$. The embedding process is achieved by an insertion operation which they denote by the symbol $\oplus$. That is,

$$f'_i = f_i \oplus s_i$$

The insertion operation has an inverse operation, namely the extraction operation, which they denote by θ. That is,

$$f'_i \; \theta \; f_i = s_i$$

Note that, for notational simplicity they take the insertion (and extraction) process to be binary operators, although in general they could be arbitrary functions of f_i and S_i.

Usually, the features set $\{ f_1 (I), f_2 (I),\ldots\}$ is chosen such that slight modification of individual feature does not perceptually degrade image I. In addition it is also desirable that each element in this set of features will not be changed significantly when the image is not perceptually degraded. An example of such a set features would be transformed domain (e.g., DCT, wavelet) coefficients which contain significant energy content. The labels s_i that compose the watermark in this case could be real numbers drawn from a specific distribution and the insertion operation could simply be the addition of s_i to these coefficients.

Example 1 *An invisible watermarking scheme as proposed by Cox et al.* [4].

In this scheme, the 2D DCT is taken of the image I and the set D(I) corresponds to the n AC DCT coefficients of highest magnitude. Such coefficients will typically correspond to low frequency ones.

The encoder E takes a signature S and place in the n AC DCT coefficients. An inverse 2D DCT is taken of this modified matrix, yielding the watermarked image I'. To

determine if a given image J contains the signature S, the decoder D first extract $T = \{ t_1, t_2, \ldots \}$ from J as follows:

$$t_i = f_i(J) - f_i(I) \tag{4}$$

The confidence measure c is then taken to be the

$$C = \Sigma_i \, t_i \cdot s_i \; / \; (\Sigma_i \, t_i^2 \;)^{1/2} \tag{5}$$

Alternatively, the normalized correlation is as follows.

$$C = \Sigma_i \, t_i \cdot s_i \; / \; (\Sigma_i \, t_i^2 \; \Sigma_i \, s_i^2 \;)^{1/2} \tag{6}$$

In this case, if $J = I'$, then $c = 1$. If J is a modified version of I', and the changes are not perceptually significant, c will be large value but smaller than 1.

3.2 Resolving Rightful Ownership by Invisible Watermarks

It is a common view that invisible watermarking schemes may used to protect the rights of copyright owners of images: the labels (the digital signatures) extracted from the watermarked images can be used to identify the rightful owners. Does it mean straight-forwardly that the one whose signature matches the embedded signature extracted from an image will automatically be the rightful owner of the image?

Suppose Alice and Bob use the same digital watermarking technique to watermark their images. This means that there is one unique decoding scheme to extract the labels embedded in the images. If a label extracted from a watermarked image matches the particular signature label of Alice, then the image is believe to belong to her. Similarly, if the label matches Bob's signature, then it must be his image. If a watermarked image contains both Alice and Bob's signatures, whose image is it?

Suppose now that Alice and Bob use different watermarking techniques. Given a watermarked image, Alice can take this image and decode the label using her decoding scheme. Similarly Bob can perform the label extraction process with his decoding scheme. If Alice's decoder indicates that the image belongs to her while Bob's decoder indicates that it is his image, whose image is it?

The question of how to determine or resolve rightful ownership of an image in the face of multiple copyright ownership claims has never been explicitly raised, or answered. But the scenario is valid, given that an image can be generated and modified digitally, and any image that is watermarked by Alice and in circulation can be watermarked again by Bob. In this cases Alice and Bob can use the same watermarking techniques, or apply different ones.

Without proper copyright registration and traditional protection of copyright laws, (after all, why are digital signatures necessary if copyright laws can fully protect the interests of the copyright owners ?) one can always look to these original images for an answer.

Now there is one watermarked image from which the digital signatures of both Alice and Bob have been extracted and both of them are claiming to be its rightful owner. Alice

can ask Bob for his original image and check if it contains her signature. Similarly, Bob can ask Alice for her original image and check for his signature. If Bob took Alice's watermarked image and introduced his own watermark into it, then both Bob's "original" and watermarked images contain Alice's mark. Alice's original does not contain Bob's.

The question is if Alice original image contains Bob's signature and vice versa, who owns this image: Alice or Bob?

In such a case rightful ownership can not be resolved by invisible watermarks alone. The following section will shows that this scenario is not hypothetical, but can be engineered with current watermarking schemes. They present in detail a counterfeit watermarking scheme that allows multiple claims of ownerships.

3.3 Invalidating Claims of Ownership

To invalidate claims of ownerships of an image, it is necessary to generate the confusion illustrated in the case of Alice and Bob – that there are two original images, each contains the watermark of the other party. But in reality there is one and only one original. It will be shown in this section how to create another "original" image I″ (the counterfeit original) from a watermarked image I′, without the access to the true original image I. More formally, given I′ which is watermarked by a watermarking scheme (E, D, C_δ), we have in procession I″, S′ and a decoding function D′ such that the following properties are satisfied:

1. $C_\delta(D(I'', I), S) = 1$.
2. $C_{\delta'}(D'(I, I''), S') = 1$.

δ' and δ are sufficiently large thresholds. D′ can be the same as, or different from, the decoding function D. The two parties can use different watermarking schemes in the latter.

Alice has an image I. She watermarks it with her watermarking scheme to generate a watermarked image I′ which is then made accessible to the public. Bob takes this watermarked image, and creates a counterfeit original I″ using their scheme which he then claims to be his original. The first property states that Bob's fabricated "original" I″ contains Alice's signature S. This is to be expected if the watermarking technique employed by Alice is robust. However, the second property implies that Alice's original image I contains Bob's signature S′!

Bob can claim by virtue of property (2) that the image I (Alice's original) actually contains his watermark S′. Of course, Alice, by virtue of property (1), also claims that Bob's "original" I″, contains her watermark S. Thus, it is not possible to determine the rightful owner of the image.

Given only I′, we want to construct $C_{\delta'}$, D′, I″ and S′ such that both property (1) and (2) are satisfied. This is achieved by removing a randomly selected watermark S′ instead of embedding the watermark in. The process is illustrated as follows.

Bob identified some features already presented in the watermarked image, and developed a scheme that removes these features which in return become his signatures S′, and creates a fake original image I″ which he then locks away his original along with S′, in the same way that Alice would lock away her original I and S.

More precisely, in context of the feature based watermarking schemes described in section 2, the attacker (in this case, Bob) constructs his counterfeit "original" image as

follows. He extracts a chosen (possibly random) watermark S' from the set $D'(I') = \{f'_i(I')\}$ to generate an image I″ such that

$$f'_i(I'') = f'_i(I') \,\theta\, S'_i. \tag{7}$$

The decoding scheme, operating on the counterfeit "original" I″ and the true original I, first extracts $T' = \{t'_1, t'_2, \ldots\}$ as follows:

$$t'_i = f'_i(I) \;\theta\; f'_i(I''). \tag{8}$$

The confidence measure, taken to be the normalized correlation between T′ and S′, defined in (6), is then compared to the threshold δ′. Because of the robustness of the set D′ (I′) against perceptually insignificant modification, we expect that

$$f'_i(I) \approx f'_i(I'). \tag{9}$$

Combining (7), (8) and (9), we have

$$t'_i = s'_i, \tag{10}$$

so that the correlation between T′ and S′ is large and implies that $C_{\delta'}(T',S')$ will most likely be equal to 1. The attacker (Bob) can thus claim that the true original I contains his signature S′ and that I is a modified version of I″. Conversely, the robustness of

watermarking scheme used to embed S onto I allows the true owner (Alice) to also argue that I″ contains the watermark S. We now have a scenario whereby rightful ownership cannot be resolved through invisible watermarking scheme.

The counterfeiting scheme works by inverting the watermarking process. The key step is (7). By "subtracting off" a watermark S′ in I′, we are essentially causing a watermark to be present in I, even when we do not have access to I.

Example 2 A successful implementation of the proposed attack on the watermarking scheme proposed by Cox et. al. [4].

In order to provide a more concrete example of counterfeiting an original image they wrote a program to implement the algorithm described in [19], and then modified it to perform the inverse operation as describe above. They used the same formula that Cox and et al. used to insert randomly generated watermark vector elements into an image's 1000 highest AC DCT coefficients v_i, yielding updated coefficients v_i'. To perform the inverse operation of identifying and removing a random watermark, this insertion formula was inverted to compute v_i as a function v_i', rather than the other way around.

This was an already watermarked image I′, using a watermark vector S′ to yield a new "original" image I″ (in reality a fake original) without any visible degradation of image quality. Using (5) as a measure of confidence of a watermark's presence in an image, the fabricated watermark S′ is present in the original image I with a confidence value of 23.52, while the original signature S is present in the fake original I″ with a confidence value of 23.02.

Figure 3.1 shows the true original, the watermarked image, and a fake original (from the "IBM Research Report"). There are no visible artifacts observed from looking at these three images. They are virtually identical.

Based on the test results, who is the rightful owner of this image? Which image is the true original, and which is the fake original? Under such circumstances, what can invisible watermarks achieve? There is no additional evidence available to support any answers to these questions, and consequently, invalidating the claims of rightful ownership.

No matter which scheme Alice uses to invisibly watermark her image, Bob can always use an invertible watermarking scheme (E,D,Cs) (such as the one in Example 2) to create a counterfeit original (that is, he uses E' to create a fake otiginal I", and can then show that this image, when watermarked with S' using E, will give the watermarked image I' as in circulation) and proceed to argue that the unique ownership can not be determined – thus Alice's claim of ownership is not validated based solely on the test of the presence of her invisible watermarks.

3.4 Non-Invertible Watermarking of Images

In the previous section they have shown how one can fabricate an "original" image from a watermarking image such that rightful ownership can not be resolved. They call the class of watermarking scheme that can be attacked by creating a "counterfeit original" as the invertible watermarking schemes. A more precise definition of the class of invertible watermarking schemes is as follows:

Definition: Invertible Watermarking Schemes

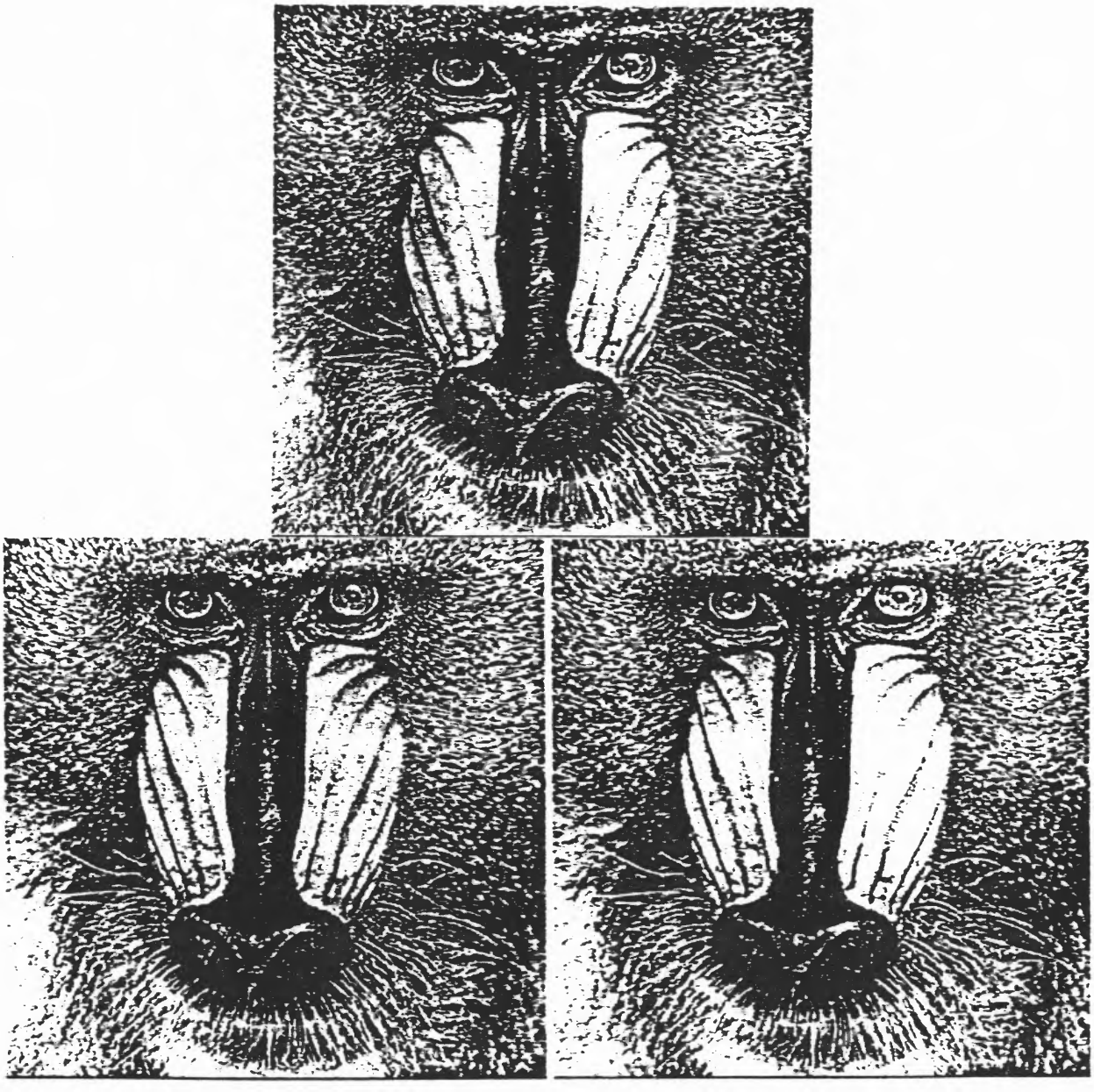


Fig.3.1 Three "Baboon" images (from database). (TOP CENTER) the watermarked image (I') of the original with 1000-element watermark sequence inserted. (BOTTOM LEFT) The original image I. (BOTTOM RIGHT) The fabricated "original" image I''. Confidence measure of the original watermark S in image I'' is 23.02. Confidence measure of the fabricated watermark S'' in image I is 23.52.

Let V be the class of invertible (invisible) watermarking schemes. Let S be a digital watermark, and S′ be another digital watermark, such that both S and S′ belongs to the set of allowable watermarks, then given an image I, (E, D, C_δ) included in V set, if for any S and S′, there exists a (E′, D, C_δ) with

$$E'(I',S') = I''$$

And

$$E(I'',S') = I'$$

Where I′ = E (I, S), such that E′ is a computationally feasible encoding function, and there is no perceptual quality degradation in the derivative images I′ and I″ from I. Otherwise, (E, D, C_δ) is non-invertible.

Note that this definition does not put any requirements on the decoding function D.

They require that from legal point of view, to establish rightful ownership through invisible watermarks, the watermark schemes applied to the images have to satisfy certain requirements- among them, the watermarking schemes can not be invertible.

There are a number of ways to enforce this new requirement. One could develop a method that can be inverted, but in such a way that image quality is degraded to a high enough degree that the fabricated " original" image is clearly not the real original.

Another approach would be to use one-way function in the watermarking process, i. e., it would not possible to extract the watermark once it is inserted. With reference to the general formulation for watermarking scheme described in Section 2, making the insertion operation non-invertible is an undesirable move, even though that insertion

operation's invertibility is the source of our problems. This is because the confidence measure relies on the fact that the watermark vector can be later extracted, and using a non-invertible insertion formula may make this difficult.

A third approach follows from noting that in order to fabricate a counterfeit original I", the attacker (Bob) first chooses a watermark S′ that he proceeds to "remove" from the watermarked image I′ that belongs to the owner (Alice). Now, if we enforce the requirement that any watermark S that is inserted into an image I be dependent on I, then we make it difficult for Bob to select S' as it depends on I" which has not yet been determined. The example is as follows.

Example 4 A modified version of the scheme described by Cox et. al. [4].

They first produce a 1000-bit $\{b_1,b_2,\ldots b_{1000}\}$ one-way hash of the original image before computing the 2D DCT of the image. They then use two slightly different equations for inserting the watermark vector elements. For each frequency bin v_i to be modified, we choose one of the two formulas depending on the value of the hash bit b_i. But using a different 1000-bit hash string, will not show up in the watermarked image.

Specifically, they used two versions of the second update formula as follows:

$$V'_i = V_i (1 + \alpha s_i), \quad b_i = 0;$$
$$V'_i = V_i (1 - \alpha s_i), \quad b_i = 1; \qquad (13)$$

Where in both cases α was chosen to be 0.1. A 1000-bit hash of the image was computed, and for each of the 1000 highest AC DCT matrix elements, one of the formula

was used depending on the value of the hash bit b_i. For convenience, these hash bits were stored in the watermark vector file.

Anticipating a possible attack involving rearranging watermark elements to match the required hash values, their scheme requires that the elements be embedded in the high-magnitude matrix elements in a left-to-right, top-to-bottom order. An attacker chooses an arbitrary binary watermarked image I′ to create an "original" I″:

$$V''_i = V'_i (1 - \alpha s'_i), \quad a_i = 0;$$
$$V''_i = V'_i (1 + \alpha s'_i), \quad a_i = 1; \qquad (14)$$

He computes the hash of his "original" I″: { b_1'',b_2'',...}. His watermark is given as follows:

$$s''_i = s'_i, \qquad a_i = b_i''.$$
$$s''_i = - s'_i, \qquad \text{otherwise.} \qquad (15)$$

They apply this watermarking scheme on a test image. The original watermark S is applied 1000 times, once with an original 1000-bit hash string, and the other 999 times using randomly selected bit strings. The 1000 different watermarked images are tested for the presence of the signature S. The results are displayed in Figure 3.2. As illustrated in the figure, if the 1000-bit hash of the "original" hash string can not be anticipated, the resulting watermark cannot be expected to have a high presence and is thus useless.

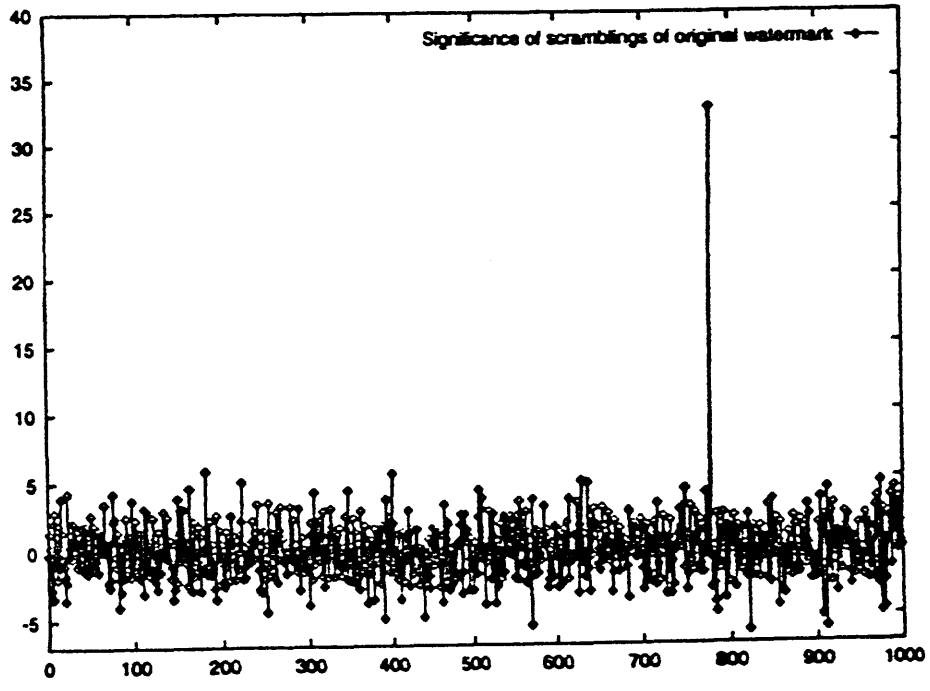


Fig. 3.2 Results of scrambling hash in watermark vectors. The original (as seen by the spike) and 999 copies with scrambled hashes

The above example illustrates the difficulty in "inverting" the watermarking scheme. Because the fake "original", I", has not been recreated, one cannot know the associated hash string $\{b_1'', b_2'', \ldots\}$. This is turn implies that it is not possible to decide which formula to use to "remove" a watermark element from I'.

3.5 Conclusions and Discussion of the IBM Research Report

Based on IBM report, the conclusion is requiring a non-invertibility of a watermarking scheme may be necessary to prevent fabrication attacks such as those they just illustrated. We think a non-invertible watermarking scheme may be necessary to prevent fabrication attacks but it still can not solve rightful ownerships if it would not possible to extract the watermark once it is inserted.

Even thought a invisible watermark can not really solve the rightful ownerships so far, a watermark is still very much needed for the following purpose that can be used to (1) identify source of the data or uniquely establish ownership, (2) to identify its intended recipient, and (3) to check if the data has been tampered with. Within each class of applications, there could be variations on the requirement of the watermarking scheme. Different applications would require different types of watermarking schemes with different requirements. For example, non-invertibility may not be an issue when there is a centralized authority with whom copyrighted image are registered. However in such an application it would be useful for an user to query an image for copyright protection.

That a invertible watermark is enough robust not to be extracted by attacker but can be extracted only by owner is still useful scheme to solve the ownerships.

CHAPTER 4

ATTACKS ON COPYRIGHT MARKING SYSTEM

Many watermarking schemes have been proposed for hiding copyright marks in digital image, video, audio, and other multimedia objects. Some applications that can remove the hidden information or destroy the mark have appeared in the research literature. We introduce the StirMark attack algorithm from the paper "Attacks on copyright marking systems " proposed by Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn [20] and do the experiment by using the StirMark robustness testing software.

4.1 Copyright Marks

There are two basic kinds of mark: fingerprints and watermarks. One may think of fingerprint as an embedded serial number while a watermark is an embedded copyright message. The fingerprint enables us to trace offenders, and the watermarks can provide some of the evidence needed to prosecute them.

They discuss simple hiding methods and the obvious attacks on them. The simplest schemes replace all the bits in one or more of the less significant bit planes of an image or audio sample with the 'hidden' information. This is particularly easy with pictures: even when the four least significant bits of the cover image are replaced with the four most significant bits of the embedded image, the eye cannot usually tell the difference. Audio is slightly harder, as the randomization of even the least significant bits of 8-bit audio add noise that is audible during quiet passages of music or pauses in speech. However, in many applications, a copyright pirate may destroy any watermark, fingerprint or other

message hidden by simple bit tweaking. So we introduce a robustness test tool – StirMark attack algorithm.

4.2 The StirMark Attack

StirMark is a generic tool developed for simple robustness testing of image marking algorithms and other steganographic techniques. In its simplest version, StirMark simulates a resampling process, i. e. it introduces the same kind of errors into an image as printing it on a high quality printer and then scanning it again with a high quality scanner. It applies a minor geometric distortion: the image is slightly stretched, sheared, shifted and/or rotated by an unnoticeable random amount (Fig. 4.1 – middle drawing) and then resampled using either bi-linear or Nyquist interpolation. In addition, a transfer function that introduces a small and smoothly distributed error into all sample values is applied. This emulates the small non-linear analog/digital converter imperfection typically found in scanners and display devices. StirMark introduces a practically unnoticeable quality loss in the image if it is applied only once. However after a few iterated applications, the image degradation becomes noticeable.

With those simple geometrical distortions they could confuse most marking systems available on the market. More distortions – still unnoticeable – can be applied to a picture. They applied a global 'bending' to the image: in addition to the general bi-linear property explained previously a slight deviation is applied to each pixel. On top of this a higher frequency displacement of the form $\lambda \sin(w_x x) \sin(w_y y) + \eta(x, y)$ – where is a random number – is added. In order for these distortions to be most effective, a medium JPEG compression should be applied after StirMark.

We now summarize briefly the main computation steps. Apart from a few simple operations such as rotations by 90 or 180 degrees, reflection and mirroring, image manipulation usually requires resampling when destination pixels do not line up with source pixels. In theory, one first generates a continuous image from the digital one, then modifies the continuous image; finally samples this to create a new pixel and evaluate the reconstruction function at that point.

There are numerous reconstruction filters. They implemented the sinc function as a reconstruction filter, which gives theoretically perfect reconstruction for photo images and can be described as follows. If (x, y) are the coordinates of the inverse transform – which, in their case is a distortion of the picture – of a point in the new image and f the function to be reconstructed, then, an estimate of f at (x, y) is given by $f'(x, y) = \Sigma\Sigma \text{ sinc}(x-i) \text{ sinc}(y-i) f_{i,j}$. This gives very much better results than the simple filter; an example of the removal of an NEC watermark is given in Fig. 4.2.

4.3 Experimental Results

In order to understand StirMark algorithm very well, we do an experiment. For this example, we watermarked a picture with NEC's algorithm. A watermark length of 1000 was used. We added the watermark to the image by modifying 1000 of the more perceptually significant component of the image spectrum. More specifically, the 1000 larger coefficients of DCT (excluding the DC term) were used. A fixed scale factor of 0.1 was used. We then applied StirMark software and tested the presence of the watermark.

A watermarked image with NEC's algorithm is shown on Fig. 4.3(a). We applied StirMark software with the default parameter and test the presence of the watermark. Fig.

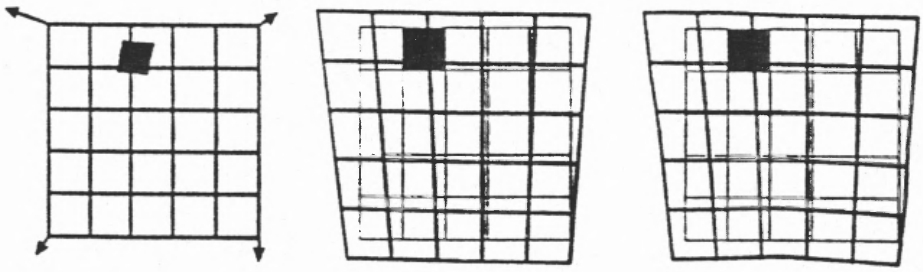


Fig. 4.1 We exaggerate here the distortion applied by StirMark to still picture. The first drawing corresponds to the original picture; the others show the picture after StirMark has been applied – without and with bending and randomisation.

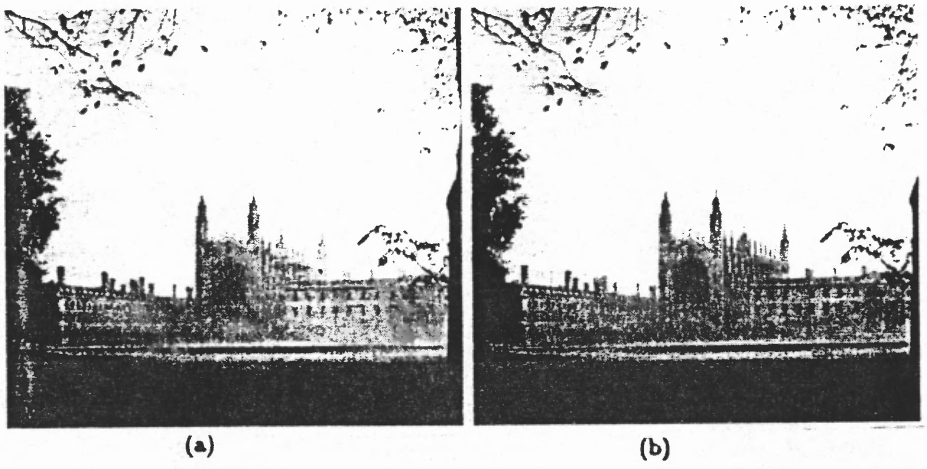


Fig. 4.2 Kings' College Chapel, courtesy of John Thompson, JetPhotographic, Cambridge. For this example we watermarked a picture with NEC'S algorithm. We used the default parameters suggested by their paper (N = 1000 and α = 0.1). (a) is the watermarked image. We then applied StirMark (b) and tested the presence of the watermark. The similarity between the original watermark and the extracted watermark was 3.74 instead of 21.08. This is well below the decision threshold.

4.3(b) is the attacked image after applying the StirMark software. The PSNR of the watermarked image comparing to the original file is 25.80103. The PSNR of the attacked watermarked image comparing to the original file is down to 16.7755.

4.4 Conclusions

Their experience in attacking the existing marking schemes has convinced them that any system which attempted to meet all the accepted requirements for marking (such as those set out by IFPI) would fail: if it met the robustness requirements then its bandwidth would be quite insufficient. This is hardly surprising when one considers that the information content of many music recording is only a few bits per second, so to expect to embed 20 bits per second against an opponent who can introduce arbitrary distortions is very ambitious.

Their more general conclusion from this work is that the 'marking problem' has been over-abstracted; there is not one 'marking problem' but a whole constellation of them. They do not believe that any general solution will be found. The trade-offs and in particular the critical one between bandwidth and robustness, will be critical to designing a specific system.

Our experiment result has shown the NEC's watermark algorithm could not survive in StirMark test algorithm. How to get the robust watermark scheme to survive in the StirMark test are the further efforts.

(a)

(b)

Fig. 4.3 The StirMark robustness testing experiment. (a) a watermarked image with NEC'S algorithm (b) an attacked watermarked image by using StirMark testing software.

CHAPTER 5

CONCLUSIONS AND DISCUSSIONS

We have introduced the binary and multilevel encoding schemes of the still image watermarking and the video watermarking in the chapter 2. For the binary encoding scheme, we propose an image watermarking encoding scheme and introduced the other encoding scheme from AT&T Bell Laboratories. For multilevel encoding scheme, we have explained the watermarking schemes applied on DCT (Discrete Cosine Transform) domain, DWT (Discrete Wavelet Transform) and Fractal coding and video coding. We will make a conclusion and discussion on every encoding scheme in this chapter.

5.1 Binary Image Watermarking

The binary watermarking scheme of still image from the AT&T paper "Electronic Marking and Identification Techniques to Discourage Document Copying" proposed by J. Brassil, S. Low, N. Maxemchuk, L. O'Gorman [3] was applied in document or the binary image by altering the text formatting, or by altering certain characteristics of textual elements (e.q., characters).

They proposed three encoding techniques: line-shift coding, word-shift coding and feature coding. In their experiments, the lines in the document were moved up or down by as little as 1/300 inch, the document was copied as many as ten times, then the document was scanned into a computer and decoded.

Based on their experimental results the centroid detection method successfully detected all line shifts for each generation of photocopy, even thought for baseline decoding of line-shift scheme, it could not be detected correctly for different repeated

copies; their proposed encoding techniques can make paper copies of documents traceable.

Actual spaces between words could be measured and compared to the formatter's (authors) expected spacing. Spacing differences resulting from this comparison would reveal the location and size of text displacement; but an attacker can apply some random horizontal shift to all words in the document to get back the original document. At least an attacker can use any word office tool to eliminate or destroy the watermark (signature).

Using the feature coding, an attacker should meet that the character endline lengths would be randomly lengthened or shorted and where the locations of encoded feature are. But an attacker can still apply random particular feature coding to all words in the document to eliminate the marks.

Invisible watermark let your document or binary image has signature there, but it still could not solve the rightful ownership. An attacker can scan the document to the computer to reformat the document or simply retype the document just paying for the time, even thought you combine the line-shifting coding, word-shifting coding and feature coding to the document.

Our proposed binary image watermark scheme embedded the watermark on the DC terms of DCT. In the proposed binary watermarking encoding scheme, the original binary image is processed first; then the preprocessed image is split into non-overlapped blocks of 8×8 and each block is DCT transformed. The watermark, which is a random number sequence and obeys Gaussian distribution, is inserted into the DC coefficients of each

block. After the modification of the DC component, an inverse DCT transform is performed for each block. Finally the binarization is made to the inverse DCT image.

Our proposed algorithm takes the advantage of the DCT transform. It is simple in implementation and yet it achieves some degree of robustness. However, further efforts are necessary in order to make it suitable for practical usage.

5.2 Image Watermarking in DCT Domain

The watermark should not be placed in perceptually insignificant regions of the image because any common signal processes affect these components. A watermark placed in high frequency spectrum of an image can not be degraded easily by any signal processes. We have introduced two different image watermarking scheme in DCT domain in section 2.2 and section 2.3. We will do some discussions and conclusions for them.

Based on the paper "Secure Spread Spectrum Watermarking for Multimedia", proposed by Ingemar J. Cox, Joe Killant, Tom Leighton and Talal Shamoon [4], we know that Cox's watermark was composed of n random numbers drawn from a Gaussian (N(0,1)) distribution and placed in the perceptually most significant components of the data in DCT domain. Their formula for computing the watermark is follows.

$$v'_i = v_i (1 + \alpha x_i)$$

In Huang's adaptive watermarking algorithm, they embed a part of the watermark in the first three low frequency AC components of each 8 x 8 block in DCT domain based on the following algorithm.

If $F_k(0,0) < T_1$ and number$\{\text{int}(F_k(u, v) / Q(u, v)) \neq 0\} < T_2$, then B_k is classified into Class 1 (dark and weak texture). If $F_k(0, 0) > T_1$ and number $\{ \text{int}(F_k(u, v) / Q(u, v)) \neq 0\}$

$> T_2$, then B_k is classified into class 3 (bright and strong texture). Otherwise, classified into Class 2 (remains).

We watermarked the Lena image with NEC's algorithm and used the default parameters suggested by their paper (N = 1000 and $\alpha = 0.1$). Based on our experimental results, NEC's image watermarking scheme is not robust. For example, the number of watermark (n) can be increased at least 2000 more depending on the size of image. The scaling parameter (α) can be increased based on the most significant coefficients.

Huang's encoding scheme embeds watermarks in first three low-frequency AC coefficients based on a classified block scheme, in order that the watermark can be robust. Their experiments indicate that if more DCT coefficients are modified for watermark embedding, it may have negative effect on either robustness or invisibility.

In Chapter 4, our experiments show NEC's algorithm fails in the StirMark robustness test tool. The future work is to test if Huang's watermarking algorithm is robust to survive in the StirMark robustness test tool.

5.3 Wavelet-Based Watermarking Scheme

Embedding watermark on DWT (Discrete Wavelet transform) [7] is described on Section 2.4. They cast the watermark in significant coefficients after transform. They find out which subband contains the most significant coefficients by successive subband quantization and search significant coefficients in the selected subband only.

An attacker who knows their algorithm very well but without the watermark information could damage the watermark with little probability by destroying selected

significant coefficients directly. They can protect the location of significant wavelet coefficients by keeping the following three types of data confidential:

1. the selected wavelet transform structure (e.g. pyramid, wavelet Packet etc.) and the transform level L;
2. the selected wavelet filter set;
3. the seed to generate the random sequence for significant coefficient skipping.

Based on their experimental results, Their wavelet image watermark algorithm is robust but their experimental result do not show their wavelet image watermark scheme is invisible.

5.4 Fractal Compression Watermarking Scheme

Standard methods of image compression come in several varieties. The currently most popular method relies on discarding high-frequency components of the signal by storing only the low-frequency Fourier coefficients. The JPEG standard uses a discrete cosine transform (DCT).

A fractal image compression technology is based on a set of contractive transformation. It means that a given transformation applied to any two points in input image must bring them closer together in the copy one. Storing images as collections of transformations could lead to image compression, so that we think that the fractal image compression technology is suitable in system which requires a bit rate of image compression.

A fractal compression watermark scheme consists of a coding-decoding process and retrieving the signature. The advantage of the fractal compression watermarking scheme

is that attackers can not obtain the information (where is the object and what kinds of iterated contractive transformation used) without the appropriate key. We think the fractal compression watermark scheme is most robust than any other one in our all discussed watermarking schemes.

5.5 Video Watermarking

Hartung's digital watermarking scheme embeds the information into the raw video or compressed video. The basic idea of watermarking for the raw video is a addition of a pseudo-random signal to the video. The receiver can not recover the information without knowledge of the pseudo-noise sequence, but attackers can destroy the watermarking by adding random- noise sequence anywhere within the video. This scheme has the problem with any other's invertible image watermark.

The basic idea of watermarking on the compression video encodes the watermarking after doing a zig-zag-scan. You transmit the watermarked AC DCT coefficient, if the bit rate of watermarked AC DCT coefficient represented by VLC codeword shall not increase; otherwise, you transmit the non-watermarked AC DCT codeword. Based on the bit-rate constraint, it is possible that only few DCT coefficients can embed watermark. It is not enough robust.

REFERENCES

1. Neil F. Johnson and Sushil Jajodia, "Exploring Steganography: Seeing the Unseen", IEEE Computer, February 1998: 26-34.

2. D. Gruhl, Bender, and A. Lu, "Echo Hiding", In[1], pp 295-315, 1996.

3. J. Brassil, S. Low, N. Maxemchuk, L. O'Gorman, "Electronic Marking and Identification Techniques to Discourage Document Copying", AT&T Bell Laboratories, September 1998.

4. Ingemar J. Cox, Joe Kilian, Tom Leighton and Talal Schamoon, "Secure Spread Spectrum Watermarking for Multimedia", IEEE Trans. On Image Processing, 6, 12, 1673-1687, (1997).

5. Jiwu Huang, Yun Q. Shi, "Adaptive Image Watermarking Scheme Based on Visual Masking", Electronic Letters, Vol. 34, no. 8, 16th April 1998.

6. N. Jayant, J. Johnston and R. Safranek, "Signal Compression Based on Model of Human Perception", Proceedings of the IEEE, Vol. 81, No. 10, Oct., 1993.

7. H.-J. Wang and C.-C. J. Kuo, "Watermark Design for Embedded Wavelet Image Codec", SPIE SD63 Application of Digital Processing XXI 1998, SPIE'S 43rd Annual Meeting San Diego, CA, July 1998.

8. H.-J. Wang and C.-C. J. Kuo, "High Fidelity Image Compression with Multithreshold Wavelet Coding (MTWC)," in SPIE's Annual Meeting – Application of Digital Image Processing XX, (San Diego, CA), SPIE, July 27 – August 1 1997.

9. H.-J. M. Wang, Y.-L. Bao, C.-C. Jay Kuo, "Novel Wavelet Coder Image Compression", in International Symposium on Voice, Audio and Data Compression, (ICIP 97), (Santa Barbara,CA) IEEE Signal Processing Society, July 1997.

10 H.-J. M. Wang, Y.-L. Bao, C.-C. Jay Kuo and H. Chen, "Multi-Threshold Wavelet Codec (MTWC)", JPEG 2000 Contribution, ISO/IEC JTCI/SC29/WGI N 819, Convergence Phrase Meeting, Geneva Swiss, March 1998.

11. Joan Puate and Fred Jordan, "Using Fractal Compression Scheme to Embed a Digital Signature into an Image", Signal Processing Laboratory, Swiss Federal Institute of Technology.

12. M.F. Barnsley, "Iterated Functions Systems", In R. L. Devaney, L. Keen, K.T. Alligood, J. A. York, M. F. Barnsley, B. Branner, J. Harrison, and P.J. Holmes, editors, Chaos and Fractals: The Mathematics Behind the Computer Graphics, American Mathematical Society, 1989.

REFERENCES
(Continued)

13. M.F. Barnsley, "Methods and Apparatus for Image Compression by Iterated Function Systems", United States Patent Number 4, 941, 193, 1990.

14. M.F. Barnsley and S. Demko, "Iterated Function Systems and the Global Construction of Fractals", Proceedings of the Royal Society of London. A 399: 243- 275, 1985.

15. M.F. Barnsley and L.P. Hud, *Fractals Image Compression*, AK Peters, Ltd., Wellesley, Massachusetts, 1993.

16. L. Torres, M. Kunt, "Video Coding", The Second Generation Approach.

17. Fisher Y., *Fractal Image Compression: Theory and Applications*, Spinger Verlag Edition, New York, 1995.

18. Frank Hartung and Bernd Girod, "Digital Watermarking of Raw and Compressed Video", Proceeding SPIE 2952: Digital Compression Technologies and Systems for video Communication, pp. 205-213, October 1996.

19. S. Craver, N Memon, B.-L. Yeo and M. Yeung, "Can Invisible Watermarks Resolve Rightful Ownerships?", Computer Science/Mathematics, RC 20509 (July 25, 1996).

20. Fabien A-P, Petitcolas, Ross J. Anderson, and Markus G. Kuhn, "Attacks on Copyright Marking Systems", Second Workshop on Information Hiding, Portland, Oregon, USA, April 15-17, 1998.