行政院國家科學委員會補助專題研究計畫成果報告

數位博物館影像版權資訊植入技術與軟體之開發

Copyright and Annotation Protection for Visual Data in Digital Museum by Using Image Watermarking, Hiding, and Authentication Techniques

計畫類別：個別型計畫
計畫編號：NSC89-2750-P009-00
執行期間：89 年 4 月 1 日至 91 年 3 月 31 日

計畫主持人：交通大學蔡文祥教授兼副校長
共同主持人：銘傳大學吳大鈞副教授

執行單位：國立交通大學資訊科學系

中 華 民 國 91 年 6 月 25 日

CONTENTS

ABSTRACT 1
Chapter 1 Introduction 2
1.1 Motivation 4
1.2 Application Requirements 4
1.2.1 Requirements for Image Data Hiding 4
1.2.2 Requirements for Image Watermarking 5
1.2.3 Requirements for Image Authentication 6
1.3 Overview of Proposed Method 8
1.3.1 Terminologies 8
1.3.2 Image Formats in Digital Museums 8
1.3.3 Overview of Proposed Methods 9
1.4 Contributions 13
1.5 Report Organization 15
Chapter 2 State-of-Art: A Survey 16
2.1 Techniques for Image Data Hiding 16
2.2 Techniques for Image Watermarking 17
2.3 Techniques for Image Authentication 19
Chapter 3 Copyright and Annotation Protection Schemes for Archive Images 21
3.1 Introduction 21
3.1.1 Uses of Archive Images 21
3.1.2 Properties of BMP Images 22
3.2 Proposed Annotation Hiding Scheme by Replacement of LSB Bits 23
3.2.1 Annotation Hiding Process 23
3.2.2 Annotation Extraction Process 26
3.2.3 Experimental Results 28
3.3 Proposed Watermarking Scheme by Replacement of LSB Bits 31
3.3.1 Proposed Watermarking Embedding Method 32
3.3.2 Watermark Extraction Process 32
3.3.3 Experimental Results 34
3.4 Proposed Authentication Scheme by A Human Visual Model 36
3.4.1 Fragile Watermarking and Boundary Line Embedding Process 36

3.4.2 Boundary Line Searching Process 40
3.4.3 Image Authentication Process 41
3.4.4 Experimental Results 42
3.5 Discussions 45
Chapter 4 Copyright and Annotation Protection Schemes for Reference Images 47
4.1 Introduction 47
4.1.1 Uses of Reference Images 47
4.1.2 Properties of JPEG Images 48
4.2 DCT-Domain Annotation Hiding by A Voting Scheme 51
4.2.1 Annotation Embedding Process 52
4.2.2 Annotation Extraction Process 55
4.2.3 Experimental Results 58
4.3 Watermarking by Spread Spectrum Method 60
4.3.1 Watermark Embedding Process 60
4.3.2 Watermark Detection Process 65
4.3.3 Experimental Results 66
4.4 Authentication Scheme by DC-signature 71
4.4.1 Idea and Preliminary Experiments 72
4.4.2 Proposed Signature Extraction Method 73
4.4.3 Authentication Process by DC-signature 74
4.4.4 Experimental Results 75
4.5 Discussions 78
Chapter 5 Copyright and Annotation Protection Schemes for Thumbnail Images 80
5.1 Introduction 80
5.1.1 Uses of Thumbnail Images 80
5.1.2 Characteristics of GIF Images 81
5.2 Annotation Hiding Scheme by Palette Index Replacement 82
5.2.1 Proposed Method 82
5.2.2 Annotation Extraction Process 85
5.2.3 Experimental Results 86
5.3 Watermarking Scheme by Palette Index Replacement 88
5.3.1 Proposed Method 88
5.3.2 Watermark Extraction Process 89
5.3.3 Experimental Results 90
5.4 Authentication Scheme by Nearest Palette Color Replacement Method 93
5.4.1 Proposed Fragile Watermarking Method 93

5.4.2 Authentication Process 94
5.4.3 Experimental Results 95
5.5 Discussions 96
Chapter 6 Conclusions and Suggestions 100
6.1 Conclusions 100
6.2 Suggestions for Future Works 101
References 103

Copyright and Annotation Protection for Visual Data in Digital Museum by Using Image Watermarking, Hiding, and Authentication Techniques

ABSTRACT

After artworks preserved in museums are digitized, the copyright and annotation data of them need to be protected. Proposed in this report are methods for embedding annotations within images of three types of formats to facilitate the association between images and their commentary data. To embed annotation data within BMP images, a novel method for embedding boundary line signals, which can be used for localizing starting points of annotations within stego-images, is proposed in this report. A new data hiding method is also proposed for embedding annotation data or image data within GIF images whose formats include color palettes. Also proposed are methods for embedding watermark signals imperceptibly within the three types of images to prove its copyright owner. To embed binary data like logos in GIF images, a practical method for image watermarking is proposed. Finally, methods for embedding fragile signals within the three types of images for image authentication to verify image integrity and fidelity is also proposed. For verification of a BMP image, a new anti-cropping fragile watermarking method is proposed. Another fragile watermarking method based on a sorted palette for GIF images is also proposed for detecting and localizing image tampering. Good experimental results prove the feasibility of the proposed methods.

Chapter 1 Introduction

1.1 Motivation

With the rapid growth of digital techniques, the life-style of human beings, the social structures, and the contents of civilization, have changed hugely. It is just started, and is unstoppable and unavoidable. In the transition from the old age to the digital world, the digitization of a civilization's asset has become a very important issue. Digitization does provide many advantages. First, digital data could be preserved permanently without any degradation. And data distribution via the Internet is easy and fast. The management of digital data is easy, too, which facilitates people to quickly find what they want.

A museum is a place where the human civilization from ancient times to the present, from primitive to modern, from science to life, and from art to engineering, are preserved. Almost every thing is included. The digitization of the museum content hence has become an urgent work for the academia and the public. A digital museum, different from a traditional museum, basically should provide two functions:

1. Digital preservation: The targets of digital preservation include antiques, specimens, and related documents and annotations. The targets should be digitized by high-resolution scanning, high quality shooting, or visualized by 3-D models, and then archived in digital format. In cooperation with the techniques of digital books, intellectual image indexing, and Internet searching, a digital museum provides the public a virtual space for sharing knowledge.
2. Digital exhibition: By applying the techniques of digital image processing, 3D computer vision, and virtual reality (VR), a visitor can browse a virtual gallery, where digitized arts are exhibited, from the Internet. In this way, a

digital museum may also extend its educational function via interactive and online distance learning.

But digitization, on the contrary, does reveal some new problems, too. Generally speaking, it is difficult to determine the fidelity or integrity of digital visual data because the development of digital processing technologies has made production of forgery an easy job. The problem originates from the intrinsic features of digital information: (1) making copies is easy and cheap; (2) each copy is exactly the same as the original one; (3) distribution via the Internet is easy and fast. The ease of copying and editing facilitates unauthorized use, misappropriation, and misrepresentation, which have brought un-estimated intellectual property losses. Thus, there is a great interest in developing technologies that help to protect the integrity of a digital media and the copyright of its owner.

Many techniques have been developed for resolving the problem. Data hiding in images is one of the techniques that embed information inside an image, called the cover image in this report, to provide data security. The goal of data hiding is to embed a secret message into a cover image without making perceptual changes to the cover image under human observation. An authorized user can extract the secret message from the processed cover image, called the stego-image in this report. An unauthorized user cannot even sense the existence of the secret message in the stego-image, and message protection is hence achieved.

In the applications of digital museums, almost every digitized image has some related annotations, which are descriptions or documents about the art. Embedding the annotation inside the image will greatly reduce the work about matching the image with its description, especially when images become plenty.

On the other hand, digital watermarking is a technique for protecting the copyright of image owners. It embeds a signal (called a watermark) into a cover image in a way

that yields imperceptible results under normal observation. The watermark is usually in the form of a logo, a serial number, or any other signal. In some applications, even if the existence of the hidden information is known, it is hard for an attacker to destroy the embedded watermark without destroying the image itself. This kind of watermark is hence called robust watermark. In applications of digital museums, the watermark is embedded into an image to uphold the copyright or for content tracking.

Image authentication, on the contrary, is a technique for verifying the integrity and fidelity of an image. The uses of fragile watermarks and semi-fragile watermarks are two of the techniques developed for image authentication. A fragile watermark is a kind of watermark that is designed to be easily destroyed if the watermarked image is manipulated in the slightest manner. Image authentication can be achieved by inspecting whether the embedded signal is destroyed. But in some applications, on the other hand, "information preserving" operations (such as JPEG lossy compression) should be considered as legal operations. A semi-fragile watermark, which tells the difference between an information preserving operation and an information altering one (such as feature replacement), is hence designed to point out real tampering and ignore legal operations in the authentication process.

1.2 Application Requirements

Image data hiding, image watermarking, and image authentication, essentially all hide some kinds of signals into images but for different applications. Different applications make different requirements. The requirements for data hiding, watermarking, and authentication are described in this section, respectively.

1.2.1 Requirements for Image Data Hiding

Because the purpose of image data hiding is to cheat the illicit user from

knowing the secret message embedded inside the cover image, several objectives as described in the following should be satisfied.

1. Imperceptibility: Even after the secret message is embedded, under normal observation, the human eyes cannot differentiate between the cover image and the stego-image.
2. Security: The environment to conduct image data hiding must be safe, and unauthorized users should be unable to detect the secret information. The cryptographic strategy (such as the use of random numbers, or the use of the Rivest, Shamir, and Adleman (RSA) Cryptosystem) can also be used to make the image data hiding system capable of preventing illicit access.
3. Unambiguousness: Authorized users should be able to extract the original secret messages from the stego-images.
4. Need of original images: Depending on different image hiding mechanisms, authorized users may or may not need the cover image as the information to extract the secret message.
5. Robustness: For some applications, the embedded message should be preserved even when the stego-image is compressed or manipulated.

1.2.2 Requirements for Image Watermarking

Watermarks are designed to prove the copyrights of image owners. It should reside in the protected object forever. Some requirements of a watermarking system are listed below:

1. Imperceptibility: The modifications caused by watermark embedding should be below a perceptibility threshold, which means that some sort of perceptibility criteria should be used not only to design the watermark, but also to quantify the distortion.

2. Undetectability: An illicit user should not be able to detect the watermark by comparing several watermarked signals belonging to the same owner.
3. Unambiguousness: A Retrieved watermark should unambiguously identify the copyright owner, and the accuracy of identification should degrade gracefully in the face of attacks.
4. Robustness: The watermark should be recoverable even after modifications or attacks like image processing operations (such as adding noises and cropping), signal manipulations (such as sharpening and blurring) or lossy data compressions (such as JPEG compression) are applied to it.
5. Need of the original image: Depending on the application, the original data are or are not required by the watermark recovery system.
6. Undeletability: The watermark must be difficult to remove. If an attacker just knows partial knowledge and wants to destroy the watermark, the watermarked image will have severe degradation before the watermark is lost.

1.2.3 Requirements for Image Authentication

In this report, a fragile watermark is embedded inside an image to verify the image's integrity and fidelity. Some requirements of fragile watermark design are described as follows.

1. Detection of tampering: A fragile watermarking system should be able to detect with high probability any tampering in a watermarked image. This is the most fundamental property of a fragile watermark and is a requirement to reliably test image authenticity.
2. Imperceptibility: An embedded watermark should not be visible under normal observation.

3. Need of the original image: In most image authentication applications, the detection of tempering should not require the original image. The original image may not exist or the owner might have a good reason not to trust a third party with the original image.
4. Location of tampering: The tampering detector should be able to locate and characterize alterations made to a watermarked image. This includes the ability to locate spatial regions within an altered image that is authentic or corrupted.
5. Robustness to lossy compression: For some applications, information preserving operations should not be judged as tampering. Consideration in the design of watermarks hence should include the ability for detecting information altering operations (such as feature replacement) even after the watermarked content is subjected to information preserving alterations. The fragile watermark for this kind of application is therefore called the semi-fragile watermark, as mentioned before.

1.3 Overview of Proposed Method

1.3.1 Terminologies

The definitions of some related terminologies in this report are described as follows.

1. Color image: A color image is composed by three data channels, namely, the Red (R) channel, the Green (G) channel, and the Blue (B) channel.
2. Image data hiding: Image data hiding is embedding of digital information into a cover image by changing the property of the cover image systematically and imperceptibly.
3. Digital watermark: A digital watermark is a set of visible, or preferably invisible, identification codes that are embedded in the data and remains present within the data after a decryption process.
4. Image authentication: Image authentication is a process for verifying the integrity and fidelity of suspicious images.
5. Fragile watermarking and semi-fragile watermarking: Fragile watermarking and semi-fragile watermarking are techniques developed for image authentication. In a fragile watermarking system, a signal (watermark) is embedded within an image such that any subsequent alterations to the watermarked image can be detected with high probability. A semi-fragile watermarking system, on the contrary, has the ability of differentiating real tampering from information preserving operations.

1.3.2 Image Formats in Digital Museums

In the applications of digital museums, three primary image formats are usually used.

1. Archive image: An archive image is an original copy of an image that is used for preservation and reproduction. Such images have higher resolution and better quality than other types of images. Because archive images are not exposed on the Internet environment, the probability of being unauthorizedly copied or illicitly tampered is very low.
2. Reference image: A reference image is one that is used for online content display. To decrease viewers' waiting time, a reference image is normally compressed to reduce its size.
3. Thumbnail image: A thumbnail image is one that is used for content preview before showing its larger version, namely, its reference image. The dimensions of thumbnail images are relatively smaller than those of the other types of images, and they are usually compressed, too.

In this report, we propose a system to protect the copyright and the annotation data for images of each of the three formats mentioned above by data hiding, digital watermarking, and image authentication techniques, simultaneously. That is, the proposed system is designed to embed into a cover image the annotation data of the image, a digital watermark, as well as some authentication information to create a stego-image. From the stego-image, the annotation data can be extracted, the copyright owner can be proved, and the integrity and fidelity of the image content can also be verified. Different image formats will have different characteristics and properties. A single method is usually not suitable for all kinds of image formats. Therefore, different methods will be proposed for archive images, reference images, and thumbnail images according to the characteristics of different image formats.

1.3.3 Overview of Proposed Methods

A. Proposed Method for Processing Archive Images

Fig. 1.1 shows a flowchart of the proposed method for processing archive images. First, the logo of a digital museum, taken as a watermark, is converted into a binary stream and then embedded into the G channel of an archive image by replacing the least-significant bits (LSB's) of the image pixels. Second, the B channel of the image is divided into non-overlapping 3×3 blocks. The annotation data is first converted into a binary stream and then embedded in the LSB's of the eight surrounding pixels in each 3×3 block. The annotation is embedded as many copies as possible in the image. For this, the number of 3×3 blocks needed to embed a copy of the annotation is first calculated. And some boundary lines together with a fragile watermark are also embedded in the image to separate every copy of the annotation. They are embedded in the central pixels of the 3×3 blocks, based on a human vision model. The boundary lines help the annotation extractor to find the start position of the embedded annotation even when the stego-image is cropped.

B. Proposed Method for Processing Reference Images

A flowchart of the proposed method for processing reference images is shown in Fig. 1.2. First, the watermark embedded here is a serial number, which is an index in a key space. With an owner-defined serial number, a corresponding key series is embedded into a reference image by a spread spectrum method, which is robust to many operations. Second, to make the embedded annotation data achieve some degree of robustness to compression, the annotation data is embedded in the frequency domain. The reference image is first divided into non-overlapping 8×8 blocks and the image data of every block are transformed into the frequency domain by the discrete cosine transform (DCT). The annotation is then converted into a binary stream and embedded by controlling the relative magnitudes of two selected DCT

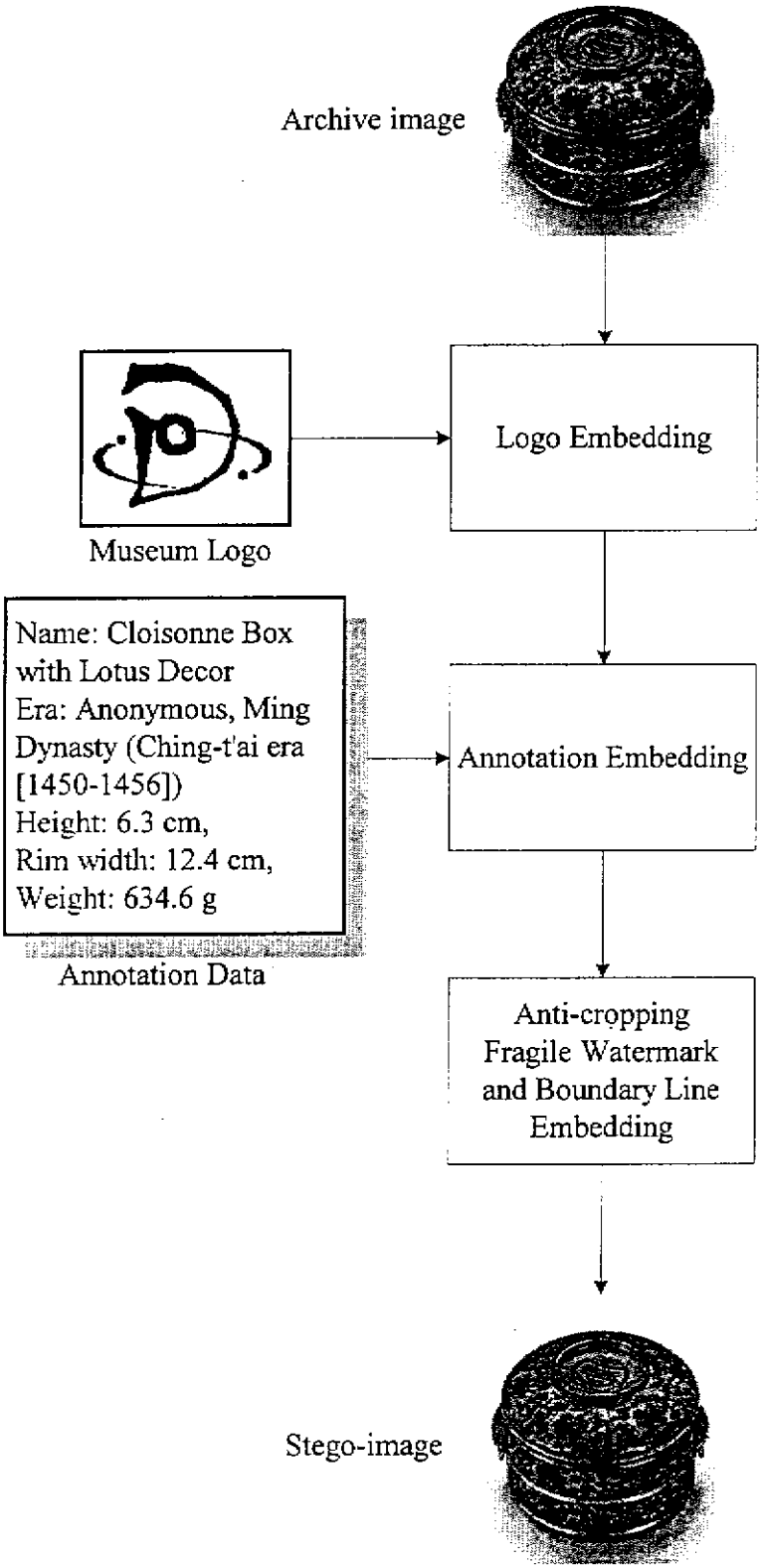


Figure 1.1 Flowchart of proposed method for processing archive images.

coefficients. Simultaneously, the direct current (DC) coefficient of every 8×8 block is extracted and recorded as a signature, called a DC-signature, for authenticating the image content. A copy of the extracted DC-signature is then transferred to an Authentication Center (AC).

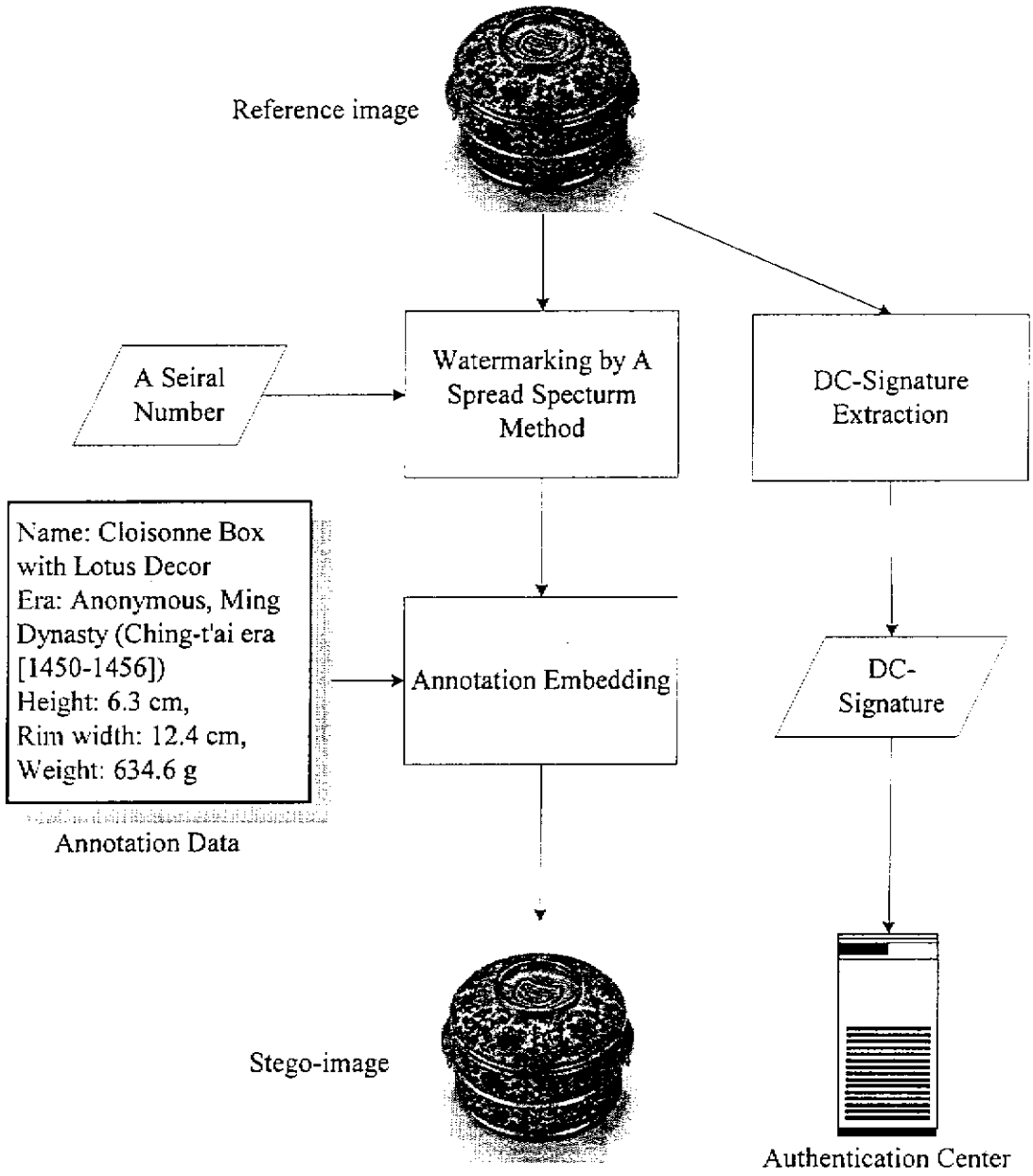


Figure 1.2 Flowchart of proposed method for processing reference images.

C. Proposed Method for Processing Thumbnail Images

Fig. 1.3 shows a flowchart of the proposed method for processing the thumbnail image. Each thumbnail image used in the digital museum is assumed to have an index color palette. Before the embedding process is started, the color palette must be re-sorted first from dark to bright. And the input image is divided into 3×3 non-overlapping blocks. The eight surrounding pixels are divided to form four two-pixel blocks. To embed a museum logo and some annotation data, they are converted into a binary stream and then embedded by controlling the even-odd relationship of every two-pixel block's color index according to the re-sorted color palette. Finally, a fragile watermark is embedded by replacing the color data of the central pixel of every 3×3 block with a color whose index is a multiple of a specified integer and is nearest to the mean of the color values of all the pixels of that 3×3 block.

1.4 Contributions

Several contributions are made in this report, as described in the following.

1. A system is proposed to simultaneously protect the copyright and the annotation data of the archive image in the applications of digital museums.
2. A system is proposed to simultaneously protect the copyright and the annotation data of the reference image in the applications of digital museums.
3. A system is proposed to simultaneously protect the copyright and the annotation data of the thumbnail image in the applications of digital museums.
4. An anti-cropping fragile watermarking method is proposed for verifying the integrity and fidelity of an image even when it has been cropped.

5. A data hiding method is proposed for embedding annotation data or image data within an image with a color palette.

6. A fragile watermarking method based on a sorted palette is proposed for detecting and localizing tampering of an image with a color palette.

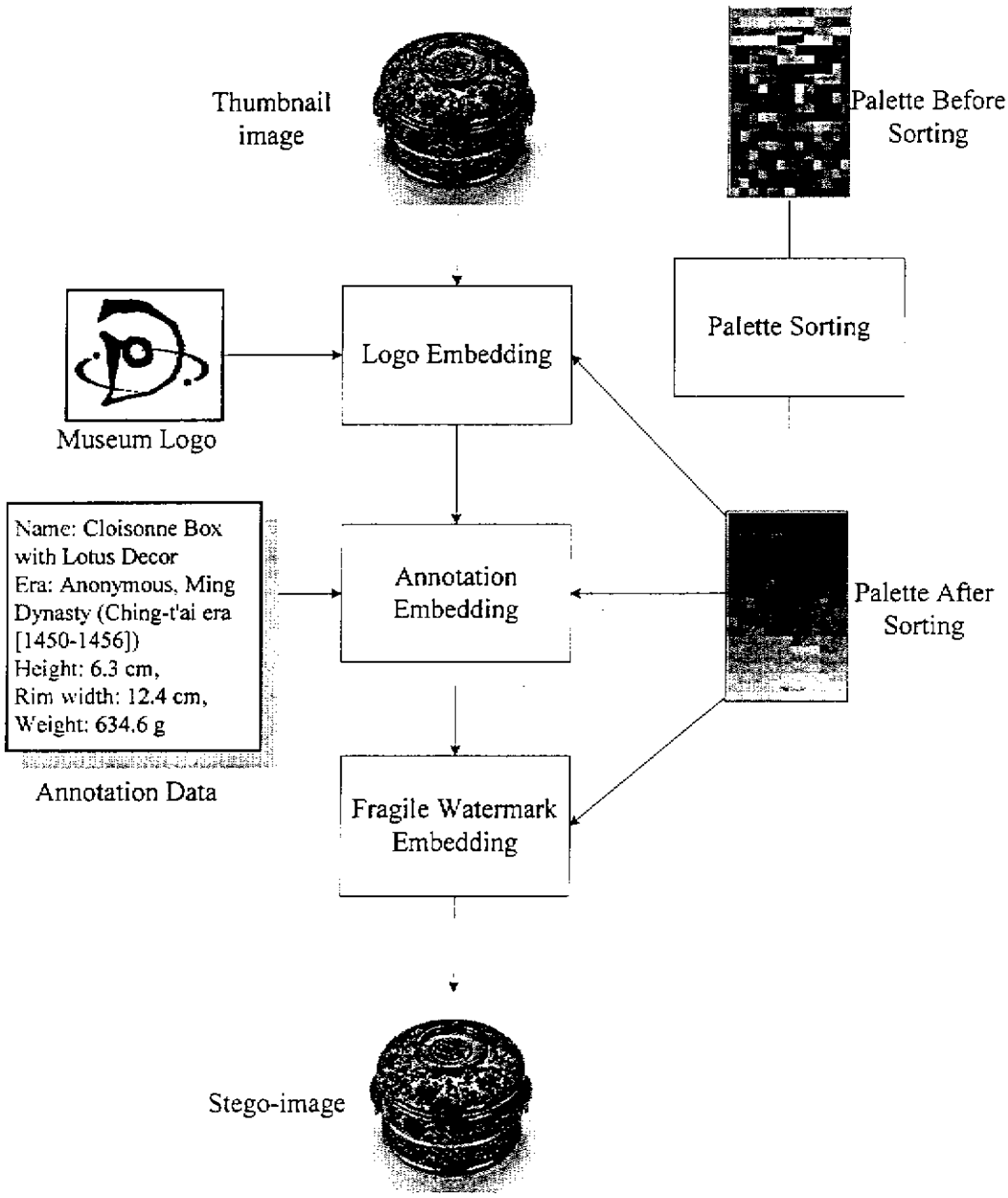


Figure 1.3 Flowchart of proposed method for processing thumbnail images.

7. A novel method for embedding the boundary line signal, which can be used

for localizing within the stego-image, based on a human visual model is proposed in this report.

1.5 Report Organization

In the remainder of this report, a survey of the related works about image data hiding, image watermarking, and image authentication are described in Chapter 2. The methods for embedding museum logos, annotation data, and anti-cropping fragile watermarks in archive images are described in Chapter 3. In Chapter 4, a spread spectrum method for embedding a watermark, an annotation hiding method in the frequency domain, and the proposed process for extraction of a DC-signature for authentication, all in reference images are described. In Chapter 5, the proposed methods for embedding museum logos, annotation data, and fragile watermarks according to re-sorted color palettes in thumbnail images are described. Finally, conclusions and some suggestions for future researches appear in Chapter 6.

Chapter 2 State-of-Art: A Survey

Image data hiding is a technique developed for embedding imperceptibly secret information, which may be texts, images, or any other data of the binary form, inside the so-called cover image. Because the target being protected is the secret message, the quality of the stego-image is the most important requirement. Image watermarking, on the other hand, is a technique for protecting the rightful ownership and copyright of a digital image by embedding a watermark signal in the protected image. The robustness of the embedded watermark is the major consideration. For image authentication, the fragile watermark is developed to embed a signal within an image such that subsequent alterations to the watermarked image can be detected with high probability. Although the techniques of image data hiding, image watermarking, and image authentication all embed information within images, different methods are adopted for different applications. In this chapter, a survey of image data hiding, image watermarking, and image authentication techniques developed in recent years will be described briefly.

2.1 Techniques for Image Data Hiding

Many different image data hiding methods has been proposed during the last few years and most of them can be seen as substitution systems. Such methods try to substitute redundant parts of the image with the secret message. The main disadvantage is the relative weakness against modifications. Recently, the development of new robust watermarking techniques led to advances in the construction of robust and secure image data hiding systems.

About the methods that embed the secret message in the spatial domain, the LSB (least-significant bit) method [1] proposed by Adelson in 1990 embedded secret data by replacing the least-significant bits of image pixels. Since only minor modifications

are made in the embedding process, the sender assumes that a passive attacker will not notice the change. The LSB method is easy and fast in implementation and a surprising amount of information can be hidden with little perceptible distortion to the image. But the LSB method is rather brittle and vulnerable to corruption due to small changes to the image.

Liaw and Chen [2] proposed an approach which is based on gray value replacement. Each pixel in a secret image is embedded in a cover image by replacing a pixel in the cover image with a similar pixel value. Chen, Chang, and Hwang [3] proposed a virtual image cryptosystem for encrypting an image into another based on a vector quantization technique. Wu and Tsai [4] proposed an image data hiding method based on image differencing. A difference value is calculated from every non-overlapping pixel pair of the cover image. All possible difference values are quantized into a number of ranges. The selection of the range intervals is based on the characteristic of human vision's sensitivity to gray value variations from smoothness to contractiveness. The difference value is then replaced by a new value to embed the value of a sub-stream of the secret message. This method provides an easy way to produce a more imperceptive result than those yielded by LSB methods.

About the methods that embed the secret data in the transformed domain, Chang and Tsai [6] proposed an image data hiding method based on the wavelet transform. The secret message is embedded within the cover image by replacing the middle and high frequency wavelet coefficients. In the method proposed by Yen and Tsai [5], both the cover image and the secret image are transformed into the frequency domain first, and a DCT coefficient replacement method is then utilized to accomplish the hiding process.

2.2 Techniques for Image Watermarking

Many techniques about embedding robust watermarks in images for copyright protection have been proposed in recent years. A watermark could be a serial number, a copyright logo, a random signal, or any image-adaptive value created by the watermarking procedure. Image watermarking techniques proposed so far can be categorized into two main types: the spatial-domain watermark and the frequency-domain watermark.

The most straightforward way to add a watermark to an image in the spatial domain is to add a pseudorandom noise pattern to the luminance values of its pixels. Many methods are based on this principle [15, 16]. The patchwork method [17] changes the gray values of pixels by adding a value to the gray values of one set of pixels while subtracting the same value from another set. The image watermark may be embedded in the frequency domain, too. In the method proposed by Koch and Zhao [18], a random sequence pulse position codes are used to embed the watermark. Chang and Tsai [6] embedded a watermark logo in the wavelet domain by changing the relationship of low frequency coefficients. A DCT-based method for embedding the watermark signal in the DCT coefficients was proposed by Hsu and Wu [19]. In the method proposed by Barni, Bartolini, Cappellini, and Piva [20], and Cox, Kilian, Leighton, and Shamoon [21], secure spread spectrum methods were applied in the watermarking procedure. Methods that embed the watermark in the frequency domain tend to be more complicated than methods used in the spatial domain, but they are more robust.

Some robust watermarking techniques exploit the characteristics of the human perceptive capability to guarantee that the modification made to the given image is imperceptible. Wu and Tsai [22] proposed a method to embed watermark logos based on a human visual model. The given image is first partitioned into subimages. The watermark information is embedded by properly adjusting the gray values of the

pixels in the central region of each subimage so that the mean gray value of them is equal to some chosen extreme values.

2.3 Techniques for Image Authentication

In the recent years, proposed systems that are used for verifying the authenticity of a digital image may be categorized, according to the nature of the employed approaches, into two types, the signature system [7] and the fragile watermark system. In a signature system, a digest of the data to be authenticated is obtained by the use of cryptographic hash functions. The recipient verifies the signature by examining the digest of the data and using a verification algorithm to determine if the data is authentic. A disadvantage of the signature system is that the additional signature must be stored and transmitted separately from the protected image. A fragile watermark system, on the contrary, embeds the authentication information inside the image and provides the ability to localize the altered areas within the image. Fragile watermark systems can be classified into two types, spatial-domain fragile watermarking system or transformed-domain watermarking one.

The method proposed by Walton [8] embeds a watermark in the least-significant bit plane for perceptual transparency. Another method proposed by van Schyndel [9] used check-sums information as the watermark message which is also embedded into the LSB plane. Wong [10] partitioned images into blocks and used the LSB plane of each block for embedding watermark information. The information is generated by a cryptographic hash function which uses the pixel values of all pixels in a block and the dimension information of the image as the parameters. Wu and Tsai [11] proposed a method for embedding perception-based fragile watermarks. A human visual model is employed to guarantee that modifications in images are imperceptible. And the watermark value is embedded in the image by replacing the gray value of the central

pixel of every 3×3 image block.

About the methods that embed fragile watermarks in the transformed domain, Fridrich [12] divided an image into 64×64 blocks, and inserted a watermark value into each block by modifying the middle third of the DCT coefficients of each block. Wu and Liu [13] described a technique based on a modified JPEG encoder. The watermark is inserted by changing the quantized DCT coefficients before entropy coding. One wavelet-based technique based on Harr wavelets was proposed Kundur and Hanzinzkos [14], which was shown to be tolerant with high quality JPEG compression. A wavelet decomposition of an image contains both frequency and spatial information about the image, and hence watermarks embedded in the wavelet domain have the advantage of being easy to locate and being effective for use to characterize illicit tampering.

Chapter 3 Copyright and Annotation Protection Schemes for Archive Images

3.1 Introduction

In this chapter, the proposed techniques for protecting the copyright and the annotation data of archive images are described. Archive images are digital images that are used in digital museums for preservation and reproduction. In Section 3.1.1, the usage and significance of archive images are discussed. In our implementation, the BMP image file format is adopted for the archive image. Therefore, In Section 3.1.2, some properties and characteristics of the BMP image are introduced. In Section 3.2, an annotation-hiding scheme for archive images is proposed. The archive image, as implied by the name, is used for archiving. The amount of annotation data that can be embedded within the image is much more than that for other image formats. As a result, the embedding method should be fast in embedding speed and large in the embedding amounts. In Section 3.3, the scheme for embedding a museum logo, as the watermark, is described. In Section 3.4, a fragile watermarking method is proposed. A human visual model is utilized to ensure perceptual invisibility of the embedded mark. And the proposed system also embeds boundary line signals within the image to assist locating the starting point of annotation data in the authentication process. In Section 3.5, some discussions about processing the archive images are made.

3.1.1 Uses of Archive Images

In applications of digital museums, an archive image is defined as a digital image that is primarily used for preservation in the museum database and for reproduction of reference and thumbnail images. Basically, a digital museum will

usually create and maintain a database that is used for archiving the digitized arts and antiques of the museum. The digitized works should be archived in a digital file format that stores the image in full colors, high quality, and without any loss or distortion. Because archive images are assumed not to be exposed on the Internet environment, the file size of an archive image is not an important consideration in picking up a suitable file format.

Under these considerations, the BMP (Bitmap) image file format announced by Microsoft hence becomes a good choice as the archive image. In the system implementation of this report, the general BMP file format, which describes an uncompressed image in full color, is adopted as the archive image format.

3.1.2 Properties of BMP Images

The BMP image file format is formulated by the Microsoft Corporation. It is mostly used in the Windows system at the start, but it can also be used in almost all systems now because of its simplicity. The "BMP" is an abbreviation of "Bitmap". As implied by its name, the BMP is an image file format that stores the pixels of the image in a bit-mapped way. That is, the pixel data of the image is stored one by one and sequentially. In the full-color version of the BMP format, every pixel of the image is stored in the (R, G, B) manner, where R,G, and B represents the red, green, and blue channels of the image pixel, respectively. Each component is an integer number ranging from 0 to 255, and can be expressed and stored in a byte. Every pixel requires three bytes of storage space. For example, an image of size 512×512 will need $512 \times 512 \times 3$ bytes of storage space, which is approximately 768k bytes.

Because the BMP file format describes an image in full color without any loss, the gorgeous image quality is a primary advantage, compared with other image

formats, especially for images that include sophisticated contents. On the contrary, because of the BMP's uncompressed nature and because it stores every pixel in full color, the size of a BMP image file is usually relatively larger than those of other image file formats.

Since the file size is not a consideration for the archive image and the digitization works should be preserved in best quality, the BMP image file format is adopted for the archive image in this report.

3.2 Proposed Annotation Hiding Scheme by Replacement of LSB Bits

In this section, the processes of embedding the annotation data into the BMP file and extraction of them will be described. The LSB method is adopted here in consideration of its simplicity, capacity, and embedding speed. In Section 3.2.1, the process of embedding the annotation data within the image will be described. In Section 3.2.2, the process of extracting the annotation data will be described. Finally, some experimental results will be shown in Section 3.2.3.

3.2.1 Annotation Hiding Process

Annotation data are embedded within the green channels of cover images in the proposed system. Two least-significant bits of a pixel are utilized to carry the annotation signal. In the hiding process, an input cover image is first divided into non-overlapping 3×3 sub-image blocks. And for every 3×3 block, the annotation data will be embedded in the two LSB's of the eight surrounding pixels as shown in Fig. 3.1. The central pixel of the 3×3 block is left unchanged to embed the fragile watermark signal and boundary lines, the use of which will be described in Section 3.4.

1	2	3
4		5
6	7	8

Figure 3.1 An example of 3x3 block.

In the embedding process, it is desired to embed as many copies of the annotation data as possible for best space utilization and retrieval reliability. Every copy of the annotation data will be embedded within a square area composed of 3×3 blocks. In corporate with the boundary lines that are embedded along with the fragile watermark, which will be described in Section 3.4, the annotation data can be extracted with high probability even when the stego-image is cropped. Fig. 3.2 shows a diagram that illustrates the hiding position of annotation data and the boundary lines within the image. In the diagram, every cell represents a 3×3 block of the image. The blocks filled with green color is the boundary lines that will be used to locate the start position of the embedded annotation. The start position can be decided to be the cross point of horizontal and vertical boundary lines. The blocks filled with colors other than green are used to embed the annotation data. Because a 3×3 block can be used to embed two bytes of annotation data, if the annotation data has L characters in length, $\left\lceil \frac{L}{2} \right\rceil$ 3×3 blocks are needed to embed a copy of annotation. And the border width B of the square area used to embed a full copy of annotation data can be computed as follows:

$$B = \left\lceil \sqrt{\left\lceil \frac{L}{2} \right\rceil} \right\rceil. \tag{3.1}$$

Therefore, B^2 blocks that comprise a square area will be used to embed a copy of the annotation data. And the number α of annotation copies that can be embedded within the image can be calculated as follows:

$$\alpha = \left\lfloor \frac{M}{B+1} \right\rfloor \times \left\lfloor \frac{N}{B+1} \right\rfloor, \tag{3.2}$$

where M and N are the width and the height of the cover image, respectively.

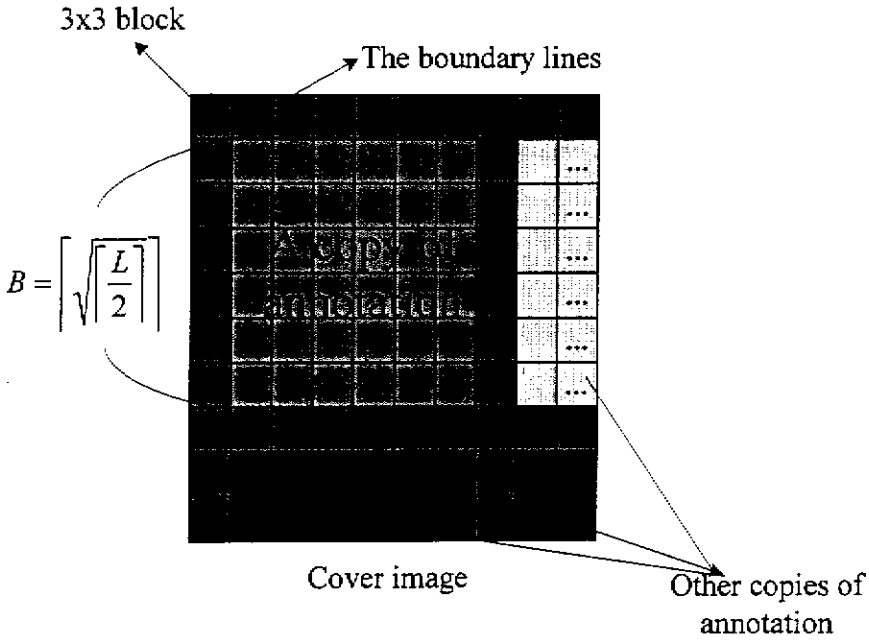


Figure 3.2 The diagram of annotation hiding position.

Let C be the cover image of size $M \times N$. Let S be the annotation we want to hide into C, which has L characters in length. The entire embedding algorithm can be briefly expressed as follows.

Step 1: Divide the cover image C into non-overlapping 3×3 blocks. The annotation data will be embedded in the eight surrounding pixels of every 3×3 block by replacing two LSB's of the pixels.

Step 2: Compute the border width B of the square area that will be used to

really embed the annotation data as follows:

$$B = \left\lceil \sqrt{\left\lceil \frac{L}{2} \right\rceil} \right\rceil.$$

Step 3: Compute the total number of annotation copies α that can be embedded within the cover image C by

$$\alpha = \left\lfloor \frac{M}{B+1} \right\rfloor \times \left\lfloor \frac{N}{B+1} \right\rfloor.$$

Step 4: Convert the annotation data S into binary form $S = (s_1 s_2 \ldots s_7 s_8 \ldots s_{(8\times L)})_2$

Step 5: Replace two LSB's of the pixels, which are within the square area, by two bits of the annotation data S repeatedly, until all of the binary data in S are embedded.

Step 6: Repeat Step 5 for α times to embed the annotation data S within the pixels of different square areas.

The boundary lines designed to separate the square areas are embedded later within the cover image C together with the fragile watermark, which will be described in Section 3.4. Fig. 3.3 shows a block diagram of the embedding process.

3.2.2 Annotation Extraction Process

In the annotation extraction process, no other information but the embedded image is needed. Because many copies of annotations are embedded within the stego-image and every copy of them is separated by the boundary line signal, the first thing to do in the extraction process is to find out where the boundary lines are. The process of searching the boundary lines within the stego-image will be described in Section 3.4. Two consecutive horizontal and vertical boundary lines

must be found to decide the border length of a square area, where the annotation data is embedded. Fig. 3.4 shows that a square area that can be located when two consecutive horizontal and vertical lines are found.

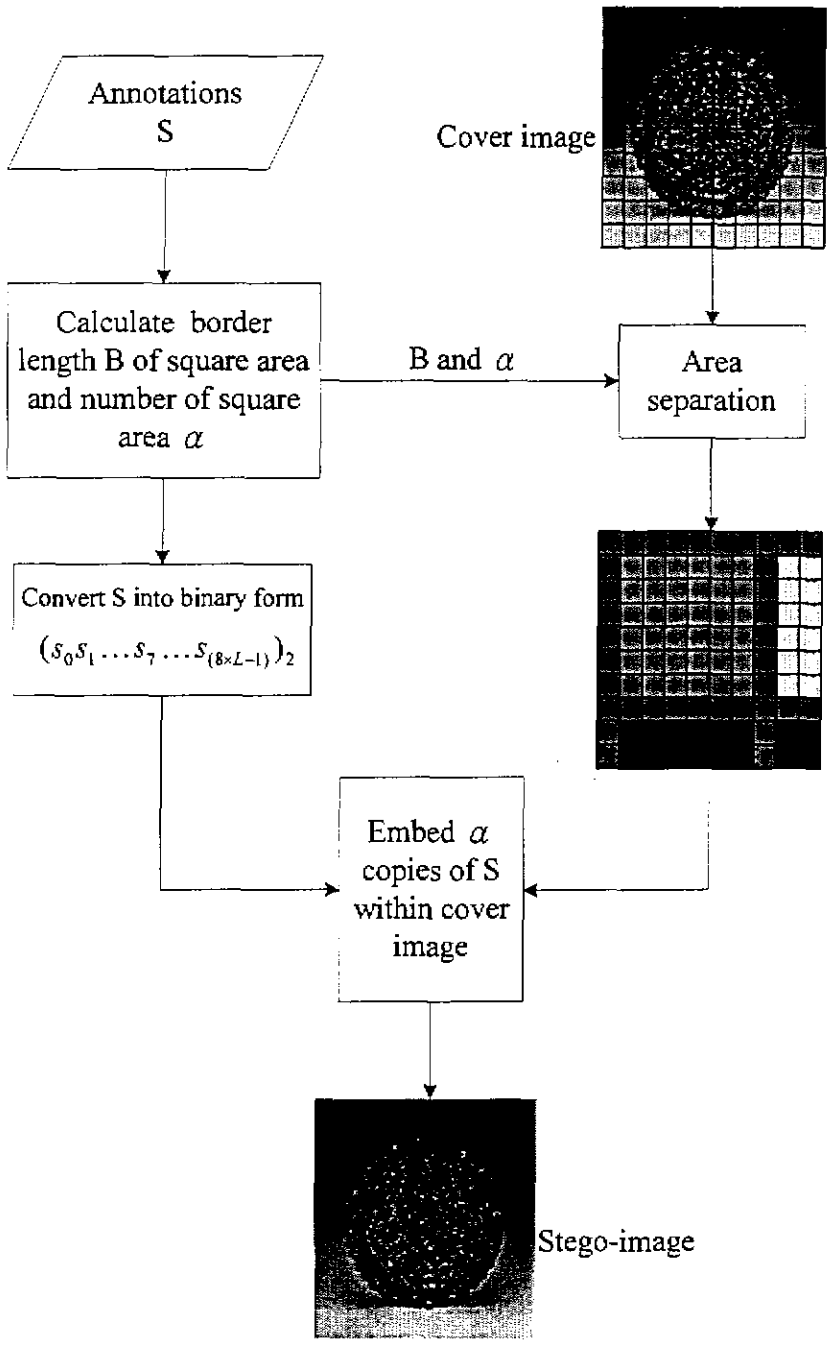


Figure 3.3 The block diagram of the annotation hiding process.

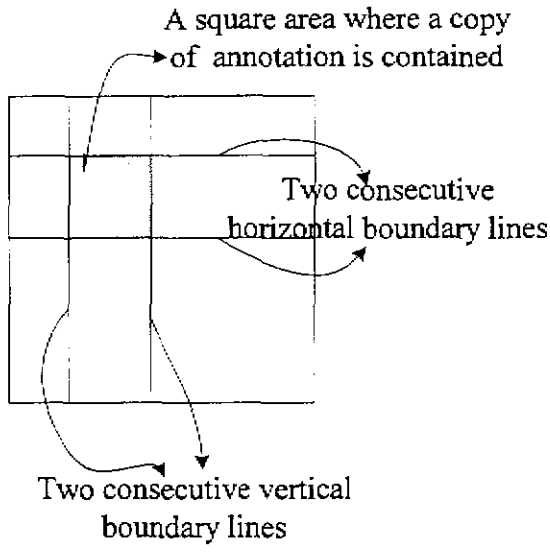


Figure 3.4 A diagram that illustrates the searching process of a square area.

Since the location of a square area can be determined, the annotation embedded within the square can be easily extracted. The square area is first divided into non-overlapping 3×3 blocks. And the annotation data are extracted block by block. For every 3×3 block, the annotation can be extracted from two LSB's of the surrounding eight pixels in the order shown in Fig. 3.1. For every 3×3 block, two bytes of data can be extracted. After all of the blocks are exhausted, convert the extracted binary-form annotation data into characters according to the ASCII codes. And now the embedded annotation data are obtained.

3.2.3 Experimental Results

In our experiments, the image "Lena" as shown in Fig. 3.5 with size 512x512 is used as a cover image. And the images that resulted from embedding 1000, 3000, and 5000 characters are shown in Figs. 3.6 (a), (b), and (c), respectively. The PSNR values of the stego-images are shown in Table 3.1. Figs. 3.6 (d), (e), and (f) show the boundary lines that are embedded within the stego-images. The lines

with gray color in the figures are the boundary line information that is used to separate each copy of the embedded annotation and the white block are the square areas where the annotation data are embedded. The shorter the annotation is, the more copies of them can be embedded. 49 copies of annotation can be embedded within the cover image when the data are 1000 characters in length. And 9 copies can be embedded when the data are 5000 characters in length.

Figure 3.5 The cover image "Lena".

Table 3.1 The PSNR values of the stego-images with different annotation lengths and numbers of copies.

	1000 characters 49 copies	3000 characters 16 copies	5000 characters 9 copies
PSNR	43.0	43.0	43.0

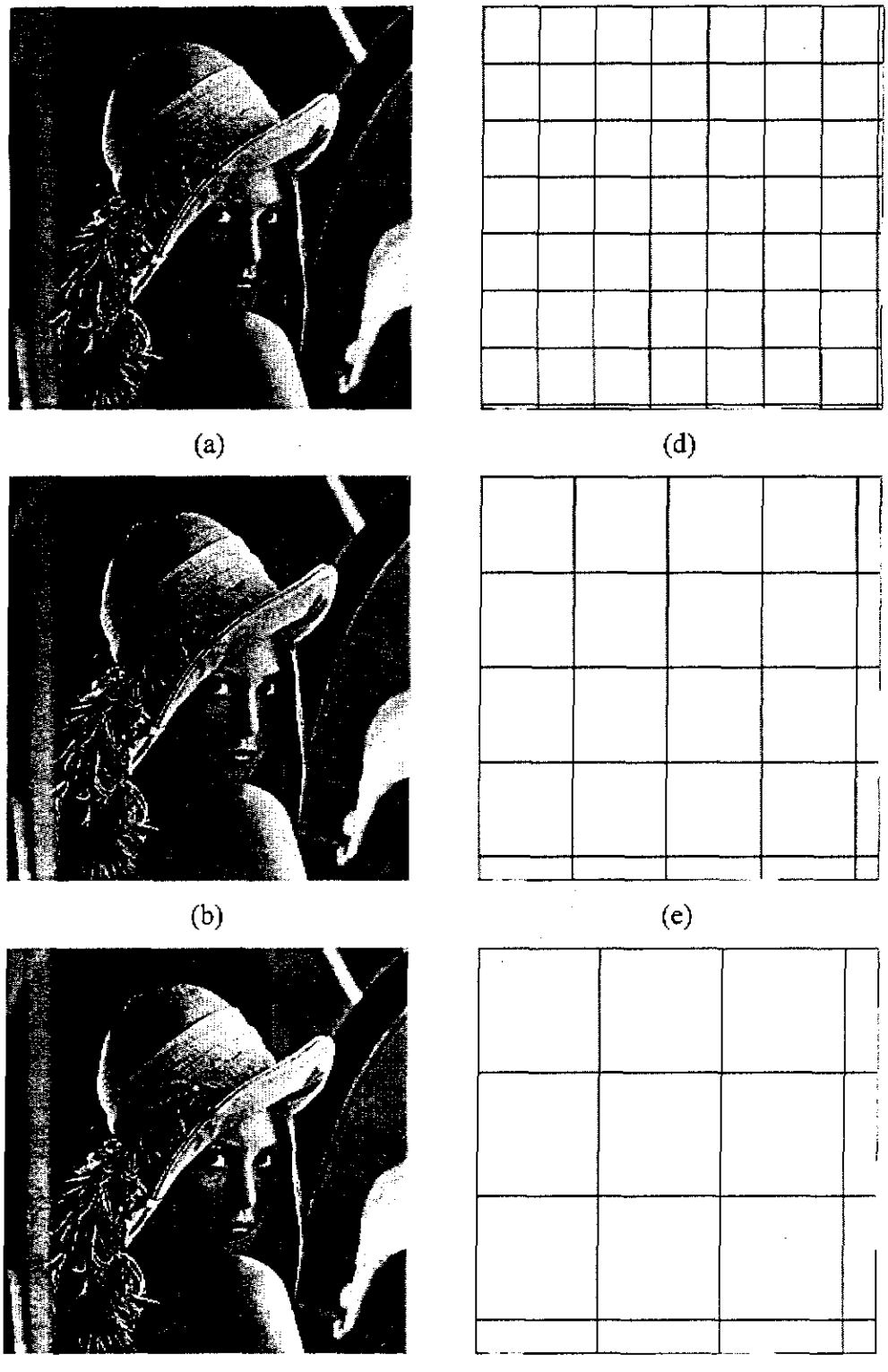
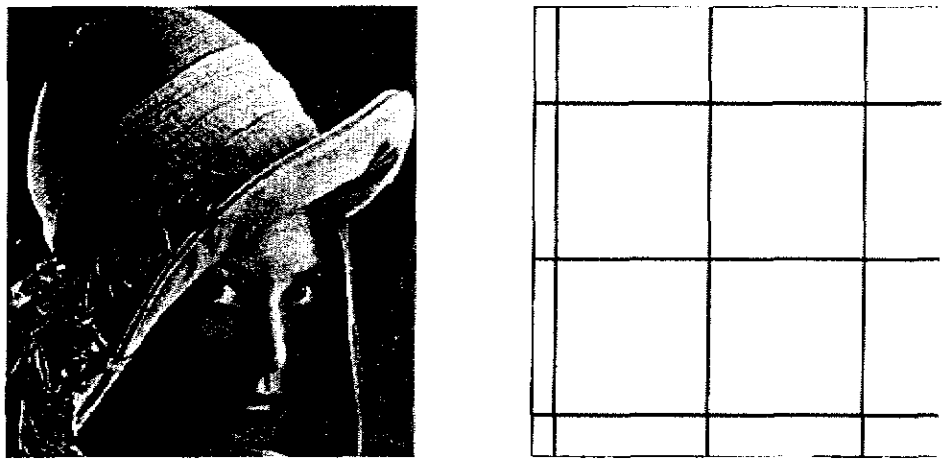


Figure 3.6 The stego-images after embedding different copies of annotation with different lengths and the boundary line information that is used to separate the square areas. (a) 49 copies of 1000 characters embedded. (b) 16 copies of 3000 characters embedded. (c) 9 copies of 5000 characters embedded. (d) - (f) Boundary lines that separate the square areas.

And Fig. 3.7 shows a cropped version of Fig. 3.5. Even the stego-image is cropped, the start position of a square area can be still found and the annotation is extracted correctly with the help of the boundary lines.



(a) (b)

Figure 3.7 The cropped stego-image and its corresponding boundary line information. (a) A cropped version of Figure 3.5. (b) Boundary lines and square areas.

3.3 Proposed Watermarking Scheme by Replacement of LSB Bits

In this section, the process of embedding a binary logo image within an archive image will be described. A logo image (for example, a museum logo) here is treated as a watermark to prove the ownership of the watermarked image. Since in the applications of digital museums, the archive image will not be exposed in the Internet environment, it is presumably impossible for an archived image to be stolen. The robustness of the watermark hence is not a major consideration. The quality of the watermarked image, on the contrary, is a more important issue. The insertion of the watermark should not cause much distortion to the archive image

since it will be used for preservation.

3.3.1 Proposed Watermark Embedding Method

Let C be a cover image of size $M \times N$, and L be a binary logo image of size $I \times J$ that will be embedded with C. Since L is a binary image, L can be transformed into binary form $L = (l_1 l_2 l_3 \ldots l_{I \times J})_2$ before embedding. In the proposed embedding process, the logo image L will be embedded by replacing one LSB of the pixel in the cover image. The width and height (I and J) of the logo image will be first converted into a binary stream and then embedded within the cover image since they are important information about how many pixels should be examined in the extraction process. After I and J are embedded, we start to embed L by replacing one LSB of each pixel in the cover image, one pixel a time and sequentially, until all bits in the logo image L are exhausted. Fig. 3.8 shows a flowchart of the entire embedding process.

3.3.2 Watermark Extraction Process

Since the LSB method is adopted in the embedding process, no other information but the stego-image is needed in the watermark extraction process. Before extracting the logo image data, the width and height information of the logo must first extracted from the stego-image so that we can know how many pixels should be extracted before starting the extraction process. The width and height information is extracted from one LSB's of each of the pixels in the left-up corner of the stego-image. Let I and J be the extracted width and height of the logo image, respectively. Then the logo binary data can then be extracted from the LSB's of the $I \times J$ pixels in the stego-image sequentially. Let $L = (l_1 l_2 l_3 \ldots l_{I \times J})_2$ be the extracted logo binary data. In cooperate with the width and height information I

and J, the embedded binary logo can be then reconstructed. Fig. 3.9 illustrates the process of logo extraction.

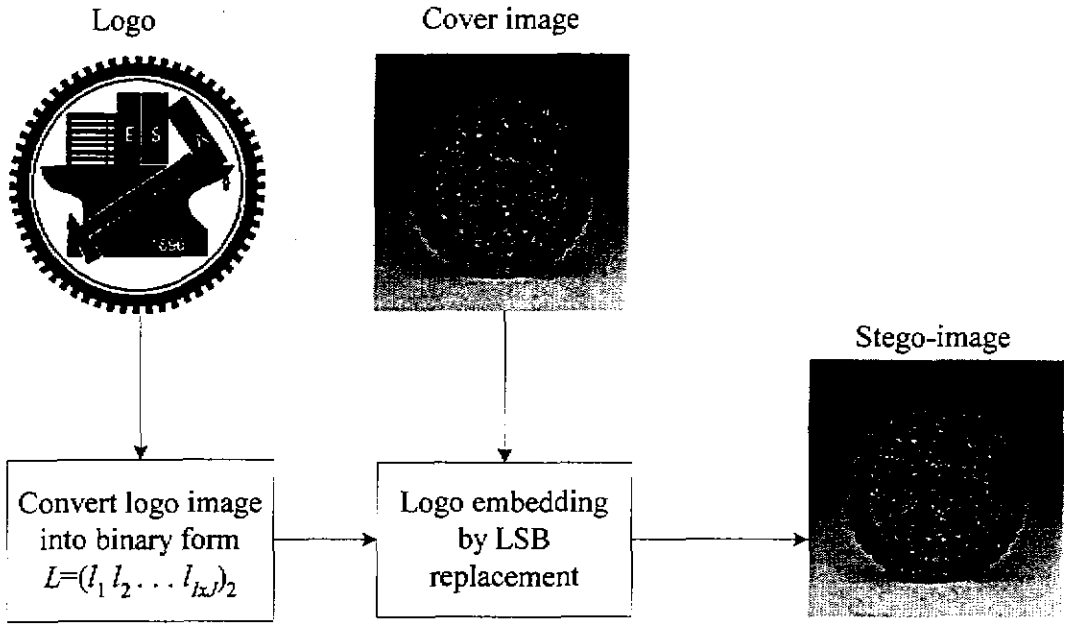


Figure 3.8 Logo embedding flowchart.

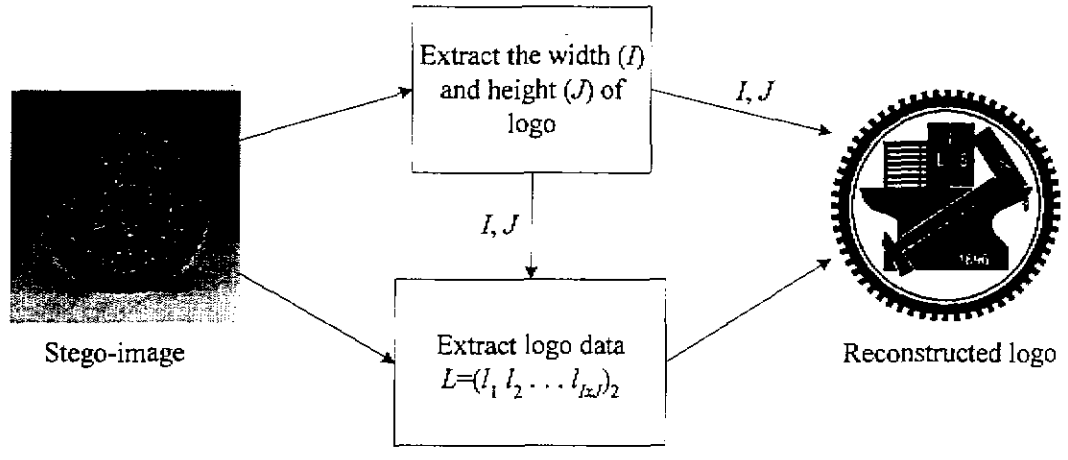


Figure 3.9 The logo extraction process.

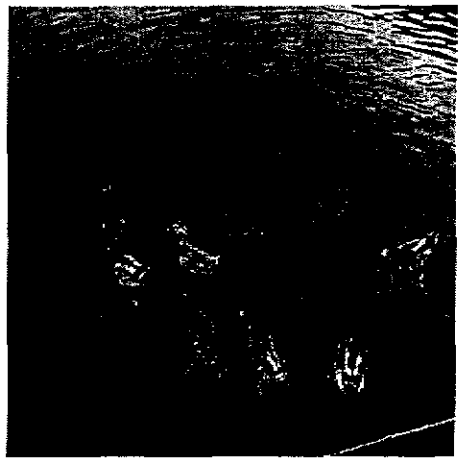
3.3.3 Experimental Results

In our experiments, the copyright logo of a digital museum of size 256×256, which is shown in Fig. 3.10, is embedded into the images as shown in Figs. 3.11 (a), (b), and (c) with size 512×512. And Figs. 3.11 (d), (e) and (f) show the embedding results, three stego-images. The PSNR values of the stego-image are shown in Table 3.2. No difference can be found visually between the cover and corresponding stego- images under normal observation and the PSNR values are high.

Figure 3.10 The copyright logo of a digital museum.

Table 3.2 The PSNR values of the images after embedding the copyright logo of the digital museum.

	Duck	Jet	Baboon
PSNR	47.0	45.4	43.0

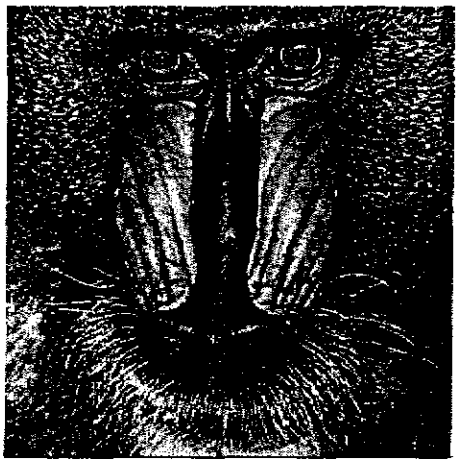


(a)

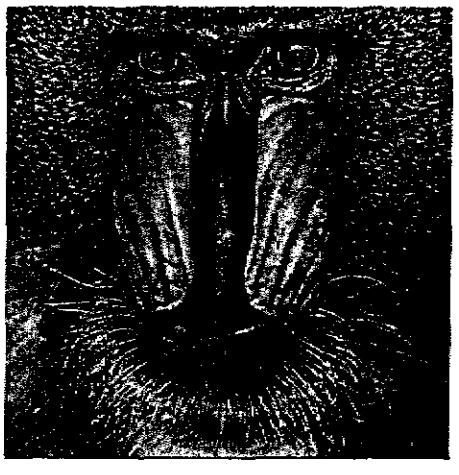
(d)

(b)

(e)



(c)



(f)

Figure 3.11 The cover images and the stego-images after embedding the logo image of Fig.3.10. (a) Cover image "Duck". (b) Cover image "Jet". (c) Cover image "Baboon". (d) - (f) Images after embedding Fig. 3.10.

3.4 Proposed Authentication Scheme by A Human Visual Model

In this section, a method for embedding fragile watermarks in archive images based on a human visual model, which is proposed in [23] and modified in [11], will be described. The human visual model is employed to guarantee that the modification of images in the fragile watermark embedding process is imperceptible. And another signal, called "the boundary line" proposed in this report, will also be embedded together with the fragile watermark. With the help of the boundary lines, every copy of the embedded annotation inside a square area, which is described in section 3.2, can be separated. Any alteration of the watermarked image can be verified and localized by examining the fragile watermark and the authentication can be accomplished without referencing the original image. In section 3.4.1, the human visual model adopted in the watermark embedding process will be first introduced. And the watermark embedding process will be described. In section 3.4.2, the process for searching the boundary lines will be described. In section 3.4.3, the process of authenticating a suspicious image will be described. And Finally, some experiments will be shown in section 3.4.3.

3.4.1 Fragile Watermarking and Boundary Line Embedding Process

The human visual system has been studied for years in the field of image coding and compression. Some of the proposed human visual systems have the ability to calculate some thresholds called JND (Just-noticeable distortion) or TEL (tolerable-error level) by the gray value of a pixel and its background intensity. Any change to the gray value between the threshold ranges is considered to be

imperceptible. In this section, a human visual model proposed in [24] and modified in [11] is utilized to embed the fragile watermark and the boundary line information.

Before the embedding process starts, the original image is first divided into non-overlapping 3×3 image blocks. The eight surrounding pixels of a 3×3 image block are considered as the background of the central pixel. In the embedding process, the standard deviation σ of the eight surrounding pixels is first calculated. The human visual model takes σ as a parameter and classified the 3×3 blocks into four classes, from smooth areas to edged areas. The contrast function of the central pixel is then decided by equally quantizing the gray values into n levels according to which class it is assigned. The criteria used to categorize the classes of the background and the quantization levels are shown in the following:

$$\text{the number of the quantization levels } n = \begin{cases} 32 & \text{when} \quad \sigma \le 2.4; \\ 24 & \text{when } 2.4 \le \sigma \le 3.6; \\ 16 & \text{when } 3.6 \le \sigma \le 4.8; \\ 12 & \text{when } 4.8 \le \sigma. \end{cases} \tag{3.3}$$

Let g be the gray value of the central pixel. g will definitely fall within one of the quantization levels, say L, defined by two visual thresholds, say g_{min} and g_{max}. From the viewpoint of the adopted human visual model, this means that any gray value in the range L will have the same sensitivity under the same background with standard deviation σ. That is, if we replace g with a gray value in range L, the modification will be imperceptible.

Take Fig. 3.12 as an example. The standard deviation of the eight surrounding pixels is about equal to 3.95. According to the classification criteria, the contrast function values of the central pixel will be equally quantized into 16

quantization levels. And the gray value 138 of the central pixel falls within the quantization level range from 128 to 143. This means that if we replace the gray value of the central pixel with any value from 128 to 143, the modification is imperceptible.

136	138	129
136	138	129
136	138	129

Figure 3.12 A 3×3 block and its gray values.

With the help of the human visual model, the fragile watermark and the boundary line information can be embedded in the central pixel of every 3×3 image block. The detail is described as follows.

Step 1: Let C be the original image with size $M\times N$. Divide C into non-overlapping 3×3 blocks, and allocate coordinates (X, Y) to every block according to its position within C, where $0\le X\le\left\lfloor\frac{M}{3}\right\rfloor-1$ and $0\le Y\le\left\lfloor\frac{N}{3}\right\rfloor-1$. Fig. 3.13 shows the 3×3 blocks with allocated coordinate (X, Y).

(0,0)	(1,0)	(2,0)	(3,0)	
(0,1)	(1,1)	(2,1)	(3,1)	
(0,2)	(1,2)	(2,2)	(3,2)	
(0,3)	(1,3)	(2,3)	(3,3)	

Figure 3.13 3x3 blocks with allocated coordinates (X,Y).

Step 2: For every 3×3 block, calculate the standard deviation $\sigma_{(X,Y)}$ of the background to determine to which class the block belongs. And the range $L=\{g_{\min(X,Y)},g_{\max(X,Y)}\}$ of the quantization level can be obtained according to the gray value $g_{(X,Y)}$ of the central pixel.

Step 3: The border length B of the square area, which is described in Section 3.2, is important information in this step to determine whether the fragile watermark or the boundary line signal should be embedded in the central pixel of 3×3 block. For every 3×3 block, modify the gray value of the central pixel by the following conditions:

$$g'_{(X,Y)}=\begin{cases} g_{\min(X,Y)}+\alpha, & \text{if } X \bmod (B+1)=0 \text{ or} \\ & \text{if } Y \bmod (B+1)=0; \\ g_{\min(X,Y)}+\beta, & \text{otherwise;} \end{cases} \tag{3.4}$$

where $g_{\min(X,Y)}+\alpha<g_{\max(X,Y)}$, $g_{\min(X,Y)}+\beta<g_{\max(X,Y)}$, $\alpha\neq\beta$, and α and β are constants that indicate whether the boundary line signal or the fragile watermark are embedded. That is, the central pixel of a 3×3 block is replaced by $g_{\min(X,Y)}+\alpha$ to indicate that the

boundary line signal is embedded. Otherwise, we replace the gray value of the central pixel by $g_{\min(X,Y)}+\beta$ to indicate that the fragile watermark signal is embedded. And selection of the values of α and β should ensure that the embedding result will not make any visible distortion to the watermarked image.

3.4.2 Boundary Line Searching Process

In the process of searching a vertical (or horizontal) boundary line, the adopted human visual model is utilized to examine whether the boundary line signal is present in the 3×3 image block. And a 3×3 block mask is used to determine whether a boundary line signal exists. The algorithm can be briefly expressed as follows.

Step 1: For every masked 3×3 image block, compute the standard deviation σ of the background. The quantization level range $L=\{g_{\min},g_{\max}\}$ of the central pixel g can also be obtained from the human visual model. The boundary line signal is decided to be present if the following condition holds:

$$g=g_{\min}+\alpha \tag{3.5}$$

where α is a constant and $g_{\min}+\alpha<g_{\max}$. That is, if the gray value of g satisfies the constraint defined in Eq. (3.5), then we judge that the boundary line signal is contained in that block. We start the searching process from the left-top 3×3 block of the image. If the boundary line signal does not exist, we move the mask one pixel to right (or down) in an overlapping manner until the first block that contains the boundary line is found.

Step 2: To ensure that the boundary line does really exist, we must examine some more 3×3 blocks to make sure of it. We examine γ 3×3 blocks under (or to the right of) the block we found in Step 1 to observe if the boundary line signal exists. If the boundary line signal does exist in all of the γ blocks, the position of the boundary line can then be determined. Otherwise, repeat Step 1 to find another block that contains the boundary line signal.

3.4.3 Image Authentication Process

In the image authentication process, no other information but the suspicious image is needed for verifying the integrity and fidelity of the image. With the help of the boundary line signal, the proposed fragile watermark has the ability to authenticate the image even when it is cropped. Since the left-top pixel (with coordinates (0,0)) of the suspicious image may not be a starting point of the 3×3 block if it has been cropped, there will be a false authentication if we cannot decide the starting point before proceeding the authentication process. To determine the starting point for authentication, a vertical boundary line and a horizontal one must be found first by the method described in Section 3.4.2. Let i_v be the x-coordinate of the vertical line, and j_h be the y-coordinate of the horizontal line. The starting point O can be then determined by

$$O = (i_v \bmod 3, j_h mod 3). \tag{3.6}$$

From the starting point O, the suspicious image is divided into non-overlapping 3×3 blocks. For every 3×3 block, the standard deviation σ of the background is calculated and the quantization level range $L = \{g_{\min}, g_{\max}\}$ of the central pixel g is also obtained from the human visual model. The block is

determined not being tampered with if $g = g_{min} + \alpha$ or $g = g_{min} + \beta$. That is, if the gray value of the central pixel is equals to $g_{min} + \alpha$ or $g_{min} + \beta$, it means that the boundary line signal or the fragile watermark is found to be present and the block is thus judged as not being tampered with.

In our experiments, a visual inspection tool for localizing any alteration in the watermarked image is provided. The blocks marked with black color are the blocks judged as being tampered with. The white and gray blocks are the blocks judged as not being tampered and as containing a boundary line, respectively.

3.4.4 Experimental Results

The images shown in Figs. 3.14 (a), (b), and (c) of size 512×512 are used in our experiments. And the images after embedding fragile watermarks and boundary lines are shown in Figs. 3.14 (d), (e), and (f). The embedding results show that the proposed method can embed the fragile watermark without noticeable changes. The PSNR values are shown in Table 3.3.

Table 3.3 The PSNR values of the images after embedding the fragile watermark and boundary line information.

	Lena	Baboon	Painting
PSNR	44.0	43.0	46.0

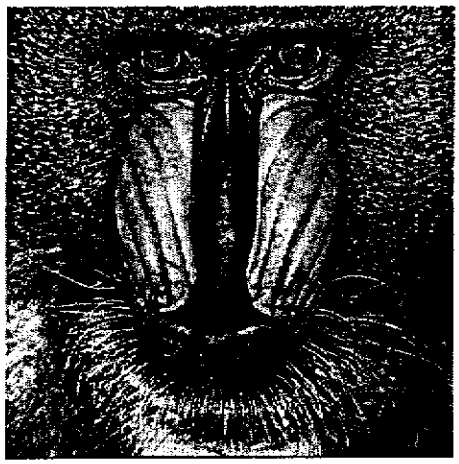
Figs. 3.15 (a), (b), and (c) shows some tampered images of Figs. 3.14 (a), (b), and (c), respectively. Fig. 3.15 (a) is tampered with by replacing the face of "Lena" with another one. Fig. 3.15 (b) is tampered with by sharpening the eyes and blurring the cheek. And Fig. 3.15 (c) is altered by drawing some extra lines and exchanging the places of the snake and the mouse in the image. The authentication results are shown in Figs. 3.15 (d), (e), and (f), respectively. The

alterations are detected with high probability and locate precisely, as shown.

Fig. 3.16 (a) shows a tampered and cropped image. With the help of the boundary line signal, the starting point can be still found and the authentication could be applied. Fig. 3.16 (b) shows the authentication result.

(a)

(d)



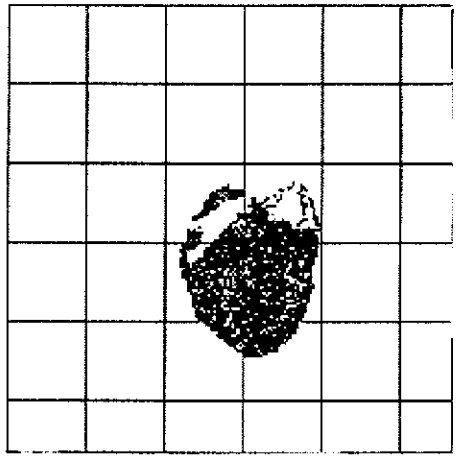
(b)

(e)

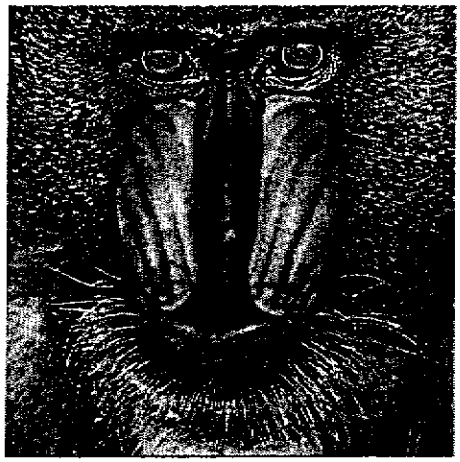
(c) (f)

Figure 3.14 The cover images and the watermarked images. (a) Cover image "Lena". (b) Cover image "Baboon". (c) Cover image "Painting". (d) Watermarked image "Lena". (e) Watermarked image "Baboon". (f) Watermarked image "Painting".

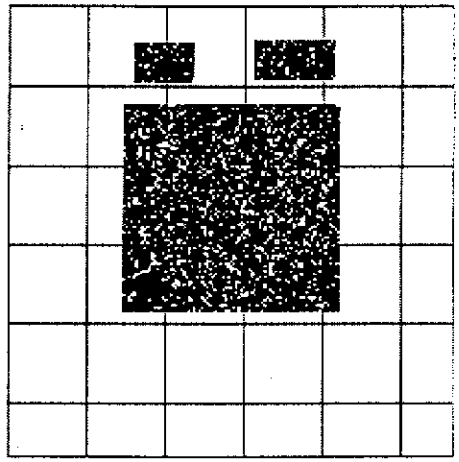
(a)



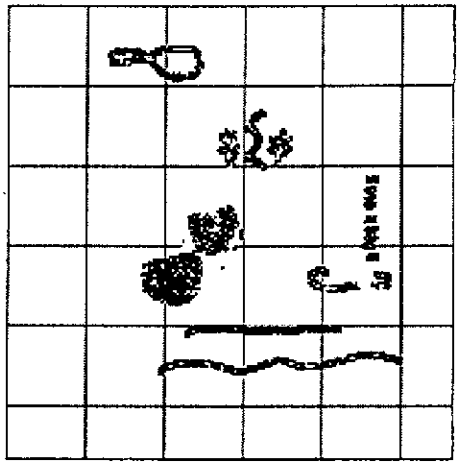
(d)



(b)

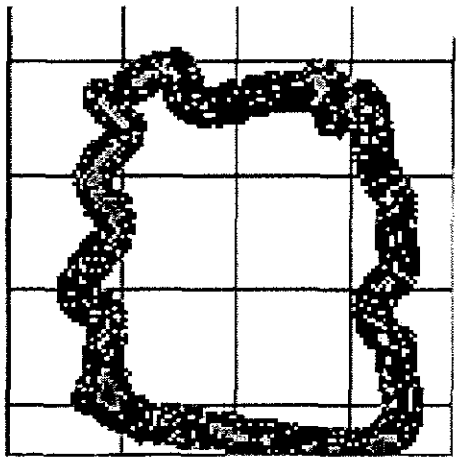


(e)



(c) (f)

Figure 3.15 Tampered images and its authentication results. (a) Tampered image "Lena". (b) Tampered image "Baboon". (c) Tampered image "Painting". (d) - (f) The authentication results.



(a) (b)

Figure 3.16 A cropped and tampered image and its authentication results. (a) Cropped and tampered image. (b) Authentication result.

3.5 Discussion

In this chapter, a system that can embed annotation data, museum copyright logos, and fragile watermarks simultaneously within an archive image is proposed. Annotation data are embedded within the eight surrounding pixels of each 3×3 image block by applying the LSB replacement method. Multiple copies of an annotation can be embedded. Every copy of them is separated by boundary line signals, which are embedded together with the fragile watermark. Even when the image is cropped, the annotation data can be still extracted if two consecutive vertical and horizontal boundary lines, which embrace a square area, can be found. A museum copyright logo can be embedded to prove the ownership of the archive image, too. Finally, a fragile watermark based on a human visual model can be embedded in central pixels of 3×3 blocks imperceptibly. Alterations to the

watermarked image can be detected and located with high probability and a visual inspection tool is provided to observe if an image has been tampered.

Chapter 4 Copyright and Annotation Protection Schemes for Reference Images

4.1 Introduction

In this chapter, three methods for protecting the copyright and the annotation data of reference images in the applications of digital museums will be described. In Section 4.1.1, the significance and usages of reference images will be introduced. In Section 4.1.2, some properties of JPEG images will be described. The JPEG image file format is adopted for reference images in this report for the applications of digital museums because of its high compression ratio and gorgeous quality even after lossy compression. In Section 4.2, a scheme for embedding an annotation, which is a description of an image, in the frequency domain will be described. Robustness of the embedded annotation can be achieved by means of embedding many copies of the annotation and extracting them out by a voting scheme. In Section 4.3, a watermarking scheme by the spread spectrum method will be described. The watermark used in the reference image is in the form of a serial number, which is a key for certificating the ownership of the image. People who claim having the copyright of an image must provide a correct watermark serial number for proving his or her ownership. In Section 4.4, a proposed authentication scheme for reference images will be described. The areas of alterations could be located if the image has been tampered with. In Section 4.5, some discussions about processing the reference image will be made.

4.1.1 Uses of Reference Images

In the applications of digital museums, the reference image is defined as the

image that is used essentially for online displays or data exchanges between museums. Basically, a digital museum will be revealed in the form of a web site. Every digitized work will be put on the web pages in the form of images. People can make a visit to the digital museum via the Internet by browsing the web site. Reference images, different from archive images, are images that are really exhibited in the digital museum for people browsing. For this purpose, the storage size of each reference image must be relatively smaller among all image file formats used in the digital museum, to accelerate the data transfer speed. In other words, the reference image can be viewed as a compressed version of the archive image.

In our experiments, the JPEG image file format is adopted as the reference images. JPEG is a standardized image compression mechanism. JPEG stands for Joint Photographic Experts Group. There are two good reasons for using the JPEG as the reference image format. First, it makes the image size smaller. This will speed up online displays. Second, even with a smaller size of storage, the quality of a JPEG image is still good. Because the reference image will be exposed on the Internet, and because of the JPEG's compression nature, the method designed to protect the copyright and annotation data must also be simultaneously robust to some attacks and to JPEG compression itself.

4.1.2 Properties of JPEG Images

The JPEG format is designed for compressing either full-color or gray-scale images of natural, real-world scenes. It works well on photographs, naturalistic artworks, and similar materials. And it does not work so well on characters, simple cartoons, or line drawings.

The JPEG compression technique is "lossy", which means that the

decompressed image will not be all the same as the one we start with. JPEG is designed to exploit known limitations of the human eye, notably the fact that small color changes are perceived less accurately than small changes in brightness. Thus, JPEG is intended for compressing images that will be looked at by humans.

A useful property of JPEG is that adjusting compression parameters can vary the degree of lossiness. This means that an image-maker can trade off the file size against the output image quality. You can make extremely small files if you do not mind poor quality. Conversely, if you are not happy with the output quality at the default compression setting, you can jack up the quality until you are satisfied, and accept less compression.

Fig. 4.1 and Fig. 4.2 show the JPEG's encoding and decoding framework. In the encoding process, the source image is first divided into non-overlapping 8×8 blocks and taken as input to the Forward DCT (FDCT). Each 8×8 block of the source image can be viewed as 64 discrete points, or a function of the two dimensions of x and y. The FDCT takes these 64 points as input and decomposed them into 64 orthogonal basis signals. The outputs of the FDCT are 64 integer values, called "DCT coefficients". The coefficient with zero frequency in both dimensions is called the "DC coefficient" and the other 63 coefficients are called the "AC coefficients". Let $f(x,y)$ be the gray value of the pixel at the spatial coordinates (x,y), and $F(u,v)$ be the value of DCT coefficient at the frequency coordinates (u,v), where $0\le x,y,u,v\le 7$. The following equation is the definition of the 8×8 FDCT:

$$F(u,v)=\frac{1}{4}C(u)C(v)\left\{\sum_{x=0}^{7}\sum_{y=7}^{7}f(x,y)\times\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)u\pi}{16}\right\}, \qquad (4.1)$$

where

$$\begin{cases} C(u), C(v) = 1/\sqrt{2} & \text{for } u, v = 0, \\ C(u), C(v) = 1 & \text{otherwise.} \end{cases}$$

After being output from the FDCT, each DCT coefficient is divided by its corresponding step size defined inside a quantization table. The purpose of this step is to discard information that is not visually significant. After being divided by the corresponding step size, each coefficient is rounded to the nearest integer. This causes the primary source of lossiness in the JPEG encoder. After the quantization, the DC coefficients of every block are treated separately from the AC coefficients. Before entropy encoding, the 63 AC coefficients are reordered into a "zig-zag" order as shown in Fig. 4.3 to improve the results of entropy coding. The entropy coding achieves additional compression losslessly based on their statistical characteristics. There are two entropy coding methods, the Huffman coding method and the arithmetic coding method [25]. The details are omitted here.

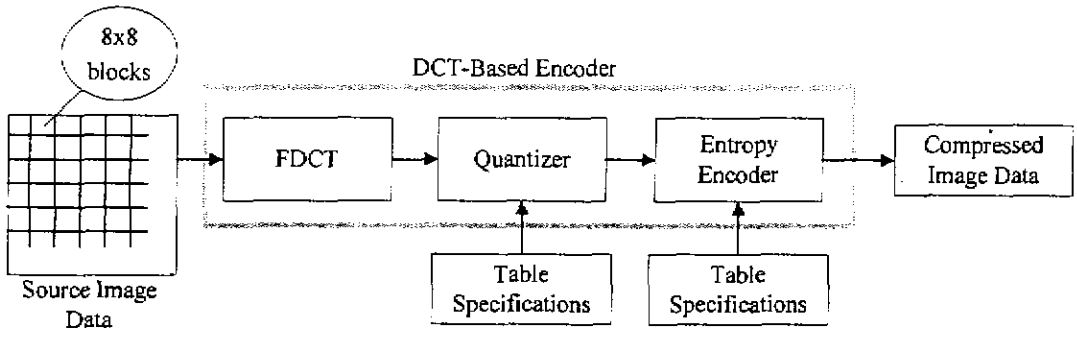


Figure 4.1 DCT-Based Encoder Processing Steps.

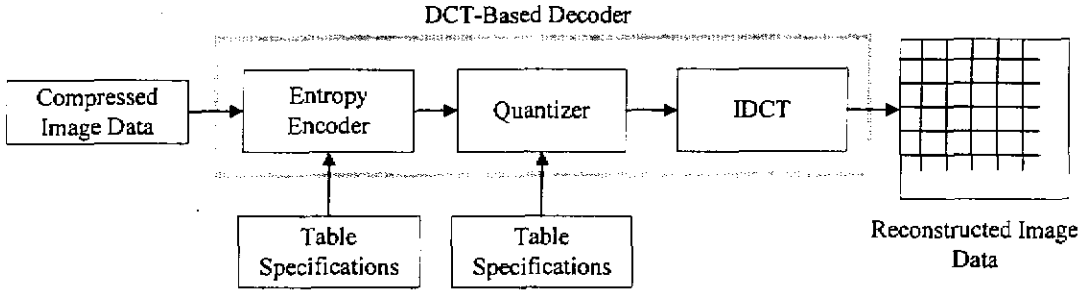


Figure 4.2 DCT-Based Decoder Processing Steps.

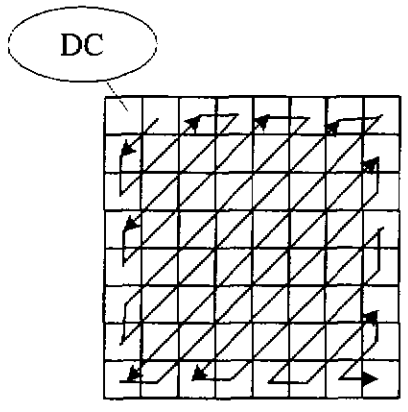


Figure 4.3 The zig-zag sequence.

4.2 DCT-Domain Annotation Hiding by A Voting Scheme

In the applications of digital museums, some amounts of annotations have to be embedded within reference images. Hiding annotation inside images will help associating one image with its description data. This is especially useful when we have enormous number of images. Since the JPEG image format is adopted for the reference image, the embedded annotation must have some robustness to the JPEG lossy compression. In this section, an annotation-hiding scheme for the reference image is proposed. Every 8×8 image block of the reference image is used to hide exactly one bit of the annotation data. And a voting scheme is utilized in the extraction process. No other information but the embedded reference image is needed when extracting the annotation. In Section 4.2.1, the proposed hiding process will be described. In Section 4.2.2, the annotation extraction process will

be stated. In Section 4.2.3, some experimental results will be shown.

4.2.1 Annotation Embedding Process

In the annotation embedding process, a cover image C is first divided into non-overlapping 8×8 blocks. The color values of every 8×8 block are transformed into the YC_bC_r color model to make the embedded annotation more robust to JPEG compression. And the Y channels are then transformed into the frequency domain by performing 8×8 forward DCT operations. Two DCT coefficients, which have the same quantization step size within the JPEG standard quantization table, are selected to embedded a bit of the annotation data by altering their relative values. Since the quantization step size of the two corresponding DCT coefficients are equal, the relative sizes between them will not be affected even they are quantized by the JPEG quantization table [23]. The altered block is then transformed back by performing an inverse 8×8 DCT operation. The processed Y channel is finally transformed back to the RGB color model, resulting in a stego-image.

In the annotation embedding process, the location of the two chosen DCT coefficients in a 8×8 block is an important step. The coefficients adopted for embedding the annotation data should be located in the middle band of the DCT frequency. This will ensure that the information is embedded in the significant part of the block, and hence it will not be completely damaged by JPEG compression. This also makes sure that the embedding of the annotation will not generate severe distortion to the cover reference image. In the JPEG encoding process, every DCT coefficient will be quantized according to a quantization table, as shown in Table 4.1.

Since the designed system should be robust against JPEG compression, we

choose the DCT coefficients in such a way that the quantization values associated with them in the standard quantization table are equal. Let (x,y) denote the location of a coefficient in a 8×8 block. According to Table 4.1, the coefficients located at (1,4) and (2,3) or (0,5) and (3,2) hence are good candidates under this consideration. In the applications of embedding annotations within an image, the extracted results should be exactly the same as the embedded one. It is possible for the extracted annotation to become misrepresentative if some bits of them are wrong, especially when the annotation contains numerical data. The proposed system hence duplicates the annotation data for many times before embedding to reduce the probability of misrepresentation. And a voting process will be proceeded in the extraction process to decide the final extracted annotation data.

Table 4.1 Standard quantization table in the JPEG compression standard (luminance component).

(u,v)	0	1	2	3	4	5	6	7
0	16	11	10	16	24	40	51	61
1	12	12	14	19	26	58	60	55
2	14	13	16	24	40	57	69	56
3	14	17	22	29	51	87	80	62
4	18	22	37	56	68	109	103	77
5	24	35	55	64	81	104	113	92
6	49	64	78	87	103	121	120	101
7	72	92	95	98	112	100	103	99

Let C be an original image of size $M\times N$, and $\alpha=\{\alpha_1,\alpha_2,.....\alpha_8,.....,\alpha_{8\times L}\}$ be the annotation data that will be embedded, with L characters in length. The entire embedding algorithms can be expressed as follows.

Step 1: Divide the original image C into non-overlapping 8×8 blocks. And transform every block into the YC_bC_r color model by the following equations:

$$\begin{cases} Y = 0.299R + 0.587G + 0.144B, \\ C_b = -0.169R - 0.331G + 0.500B, \\ C_r = 0.500R - 0.419G - 0.081B. \end{cases} \tag{4.1}$$

Also, transform the Y channel of every block into the frequency domain by performing the 8×8 forward DCT operation.

Step2: Expand the annotation α by duplicating it K times as follows:

$$\alpha'(i) = \overbrace{\alpha_1, \alpha_2, \ldots\ldots, \alpha_8, \ldots\ldots \alpha_{8\times L}}^{copy1} \overbrace{\alpha_1, \alpha_2, \ldots\ldots, \alpha_8, \ldots\ldots \alpha_{8\times L}}^{copy2} \ldots\ldots \overbrace{\alpha_1, \alpha_2, \ldots\ldots, \alpha_8, \ldots\ldots \alpha_{8\times L}}^{copyk},$$

where $1 \le i < 8 \times L \times K$.

Step 3: For every block, select the DCT coefficients S_1 and S_2, which are located at (1,4) and (2,3), respectively, to embed a bit of the annotation $\alpha'(i)$ by the following conditions.

$$For\ 1 \le i \le 8 \times L \times K$$
$$\begin{cases} if\ \alpha'(i) = 1\ \ and\ \ S_1 < S_2 \quad then\ \ Swap(S_1, S_2), \\ if\ \alpha'(i) = 0\ \ and\ \ S_1 \ge S_2 \quad then\ \ Swap(S_1, S_2), \end{cases} \tag{4.2}$$

where $Swap(S_1, S_2)$ means a process to swap the values of S_1 and S_2 when the relative values do not match bit to be encoded.

Step 4: Make $|S_1 - S_2| > T$, where $T > 0$. This can be done by gradually adding a random value to the larger one while subtracting a random value to the smaller one.

Step 5: Finally, transform each block with an embedded annotation bit into the spatial domain by performing an inverse 8×8 DCT operation. And

transform every block into the RGB color model from the YC_bC_r color model according to the following equation, which resulting in the stego-image:

$$\begin{cases} R = 0.9709Y - 0.0525Cb + 1.4017Cr, \\ G = 0.9709Y - 0.3953Cb - 0.7142Cr, \\ B = 0.9709Y + 1.7205Cb + 0.0010Cr. \end{cases} \tag{4.3}$$

In Step 4, the magnitudes, S_1 and S_2, are tuned up to make sure that their difference is large enough. This is because the JPEG compression can affect the relative values of the two selected coefficients in the quantization step. The selection of the magnitude of T is a tradeoff between the robustness and the quality of the stego-image. The higher T is, the more robust the stego-image will be against JPEG compression, however, at the expense of image quality. A flowchart for the embedding process is shown in Fig. 4.4.

4.2.2 Annotation Extraction Process

In the annotation extraction process, no other information but the embedded reference image is needed. The stego-image is first divided into non-overlapping 8×8 blocks. Every block is then transformed into the YC_bC_r color model according to Eq. (4.1). And the Y channel of every block is transformed into the frequency domain by performing an 8×8 DCT operation. The embedded annotation can then be extracted by comparing the relative values of the two selected DCT coefficients. A voting scheme is utilized to determine the extracted data since many copies of the annotation are embedded. Finally, the extracted binary-form annotations are converted into characters according to the ASCII codes.

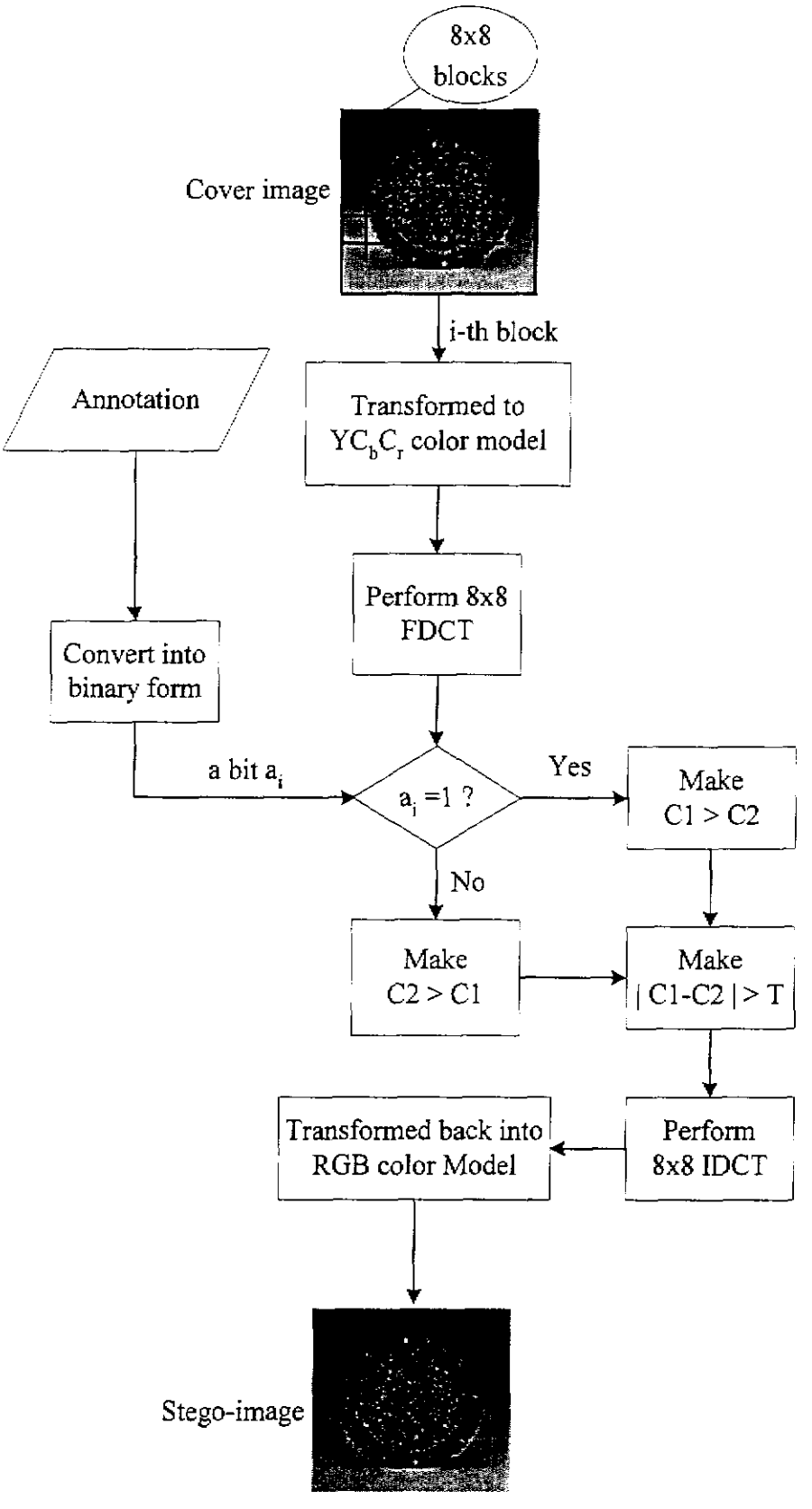


Figure 4.4 The annotation embedding flowchart for reference image.

Let E be the stego-image, and α' be the extracted annotation with L characters in length. And $\alpha'(i)$ is the *i-th* bit of the annotation, where $1 \le i \le 8 \times L \times K$, and K is the number of copies embedded. The detailed algorithm for extracting annotations is described as follows:

Step 1: The stego-image E is first divided into non-overlapping 8×8 blocks and every block is transformed into the YC_bC_r color model according to Eq. (4.1).

Step 2: For a 8×8 block B_i, where $1 \le i \le 8 \times L \times K$, the Y channel is transformed into the frequency domain by performing an 8×8 FDCT operation. And two DCT coefficients S_{i1} and S_{i2}, which are located at (1,4) and (2,3), are selected to determine the value of $\alpha'(i)$.

Step 3: The value of $\alpha'(i)$ can be determined by the following condition:

$$\alpha'(i) = \begin{cases} 0 & if\ S_{i1} < S_{i2}, \\ 1 & if\ S_{i1} \ge S_{i2}, \end{cases} \tag{4.4}$$

where $1 \le i \le 8 \times L \times K$.

Step 4: After all of the annotation values are extracted, the majority voting process is performed to derive the voting result in the following way. Let

$$V(m) = \sum_{j=0}^{k-1} \alpha'(j \times 8 \times L + m), \tag{4.5}$$

where $1 \le m \le 8 \times L$.

Reconstruct the annotation α by the following rule:

$$\alpha(m) = \begin{cases} 1 & \text{if } V(m) > \dfrac{k}{2}, \\ 0 & \text{if } V(m) < \dfrac{k}{2}, \end{cases} \tag{4.6}$$

where $1 \le m \le 8 \times L$.

Step5: Convert α into characters according to the ASCII codes, which result in the extracted annotation.

Fig 4.5 shows the flowchart for extracting the annotation from the stego-image.

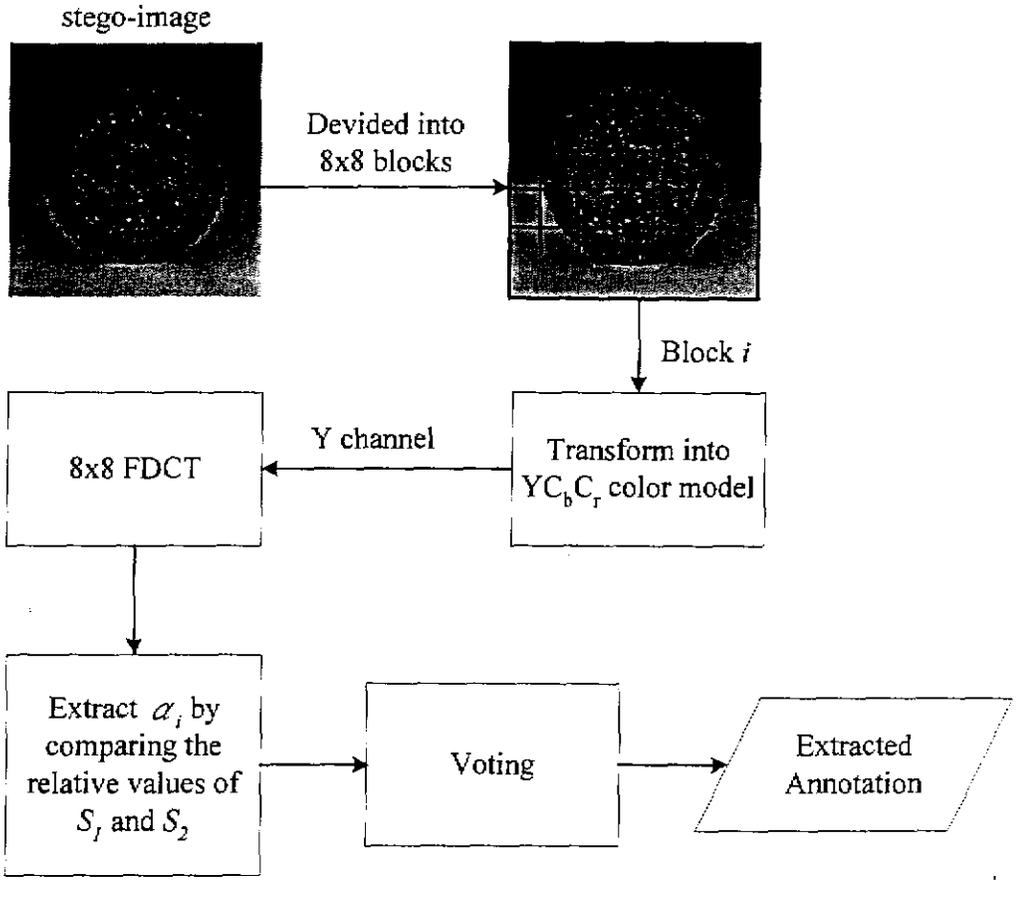


Figure 4.5 The flowchart of annotation extraction process.

4.2.3 Experimental Results

In our experiments, the images shown in Figs. 4.6 (a), (b), and (c) of size 512×512 are used as the cover images. And the images after being embedded with 5 copies of annotation, each copy having 100 characters in length, and recompressed by JPEG with quality factor 85 are shown in Figs. 4.6 (d), (e), and (f), respectively. In the embedding process, the difference threshold T between

the two DCT coefficients is set to 21. The PSNR values are shown in Table 4.2. The embedded annotation can be extracted without any error after JPEG compression with quality factor 85.

Table 4.2 The PSNR values of the stego-images with 100 characters.

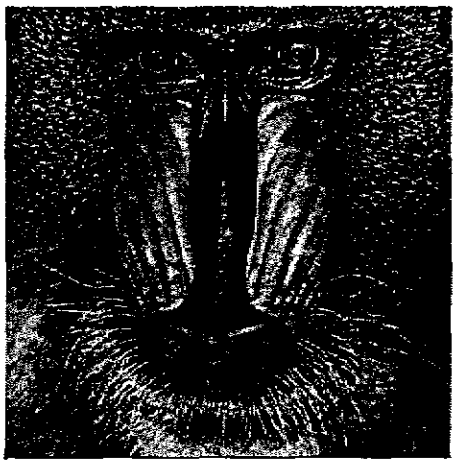
	Lena	Baboon	Painting
PSNR	34.3	31.0	38.2

(a)

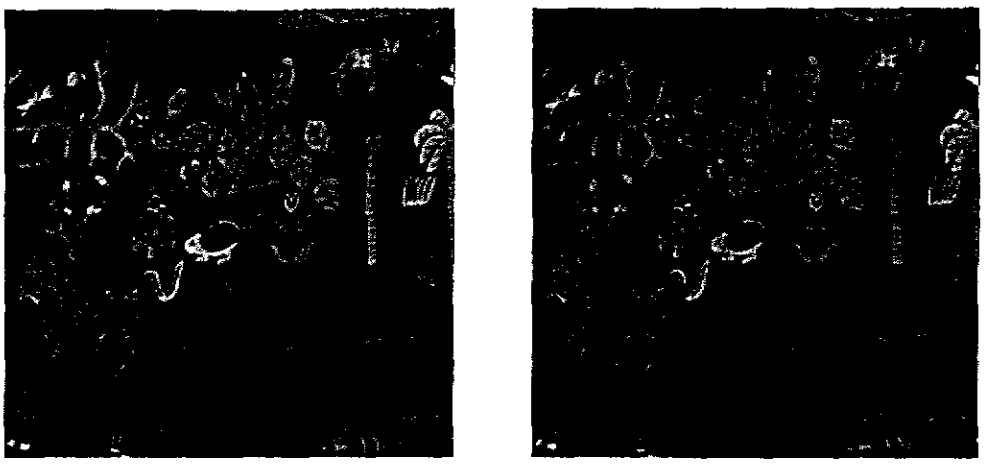
(d)

Figure 4.6 The cover images and the stego-images with 5 copies of 100 characters embedded. (a) Cover image "Lena". (b) Cover image "Jet". (c) Cover image "Baboon". (d)-(f) Stego-images.

(b)



(e)



(c) (f)

Figure 4.6 The cover images and the stego-images with 5 copies of 100 characters embedded. (a) Cover image "Lena". (b) Cover image "Jet". (c) Cover image "Baboon". (d)-(f) Stego-images (continued).

4.3 Watermarking by Spread Spectrum Method

In this section, a watermarking scheme based on a spread spectrum method [20]-[21] for protecting the copyright and ownership of reference images will be described. The watermark signal here is in the form of a long sequence consisting of integers 1 and –1, which are produced by a selected serial number. And the watermark signal is embedded in the frequency domain of the reference image by using the full-frame DCT transform. The embedded watermark is robust to many image operations. And watermark detection can be proceeded without referencing the original image, which is referred to as a blind watermarking technique. In Section 4.3.1, the process of producing the watermark sequence and that of watermark embedding will be described. In Section 4.3.2, the watermark detection process will be described. Finally, some experimental results will be shown in Section 4.3.3.

4.3.1 Watermark Embedding Process

In some conventional watermarking methods, the watermark signal is embedded

in the middle or high frequency band in the transformed frequency domain. But in these ways, many common signal processing or geometric transformation operations applied to the watermarked image will affect the embedded watermark component. The watermarking method proposed in [20] and [21] hence pointed out that the watermark signal should be placed in the most perceptually significant region of the spectrum in a fidelity preserving fashion. The original image is viewed as a communication channel used to transmit the watermark signal. Attacks are then viewed as noise that the proposed method should be immune to. The rationale is similar to the spread spectrum communication, which is a technique used in the data communication area. The spread spectrum communication method transmits a narrowband signal over a much larger bandwidth such that the signal energy present in any single frequency is undetectable. That is, the watermark signal is inserted and spread all over a perceptually significant area by slightly changing the energy of the frequency bins. The embedded result will not have severe distortion because only slight changes are made. And it is almost impossible to destroy the embedded watermark signal without degrading the image quality since the watermark signal is spread all over the image.

In the applications of digital museums, the JPEG image format is adopted for the reference image. The embedded watermark thus should basically have some degree of robustness to JPEG compression. For this reason, the watermark signal here is embedded in the DCT domains of the original image. In this report, a pool of watermark signals is maintained by the system. By providing a selected serial number (an integer number), a watermark sequence can be produced through a sequence generation procedure. Every watermark sequence is composed of integers 1 or −1 with zero mean and unit variance. And each pair of watermark

sequences produced by different serial numbers is mutually orthogonal.

The proposed system produces and maintains a pool of watermark sequences by expanding the watermark sequence matrix H as shown in the following:

$$
\begin{aligned}
H^0 &= [1], \\
H^1 &= \begin{bmatrix} H^0 & H^0 \\ H^0 & -H^0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \\
H^2 &= \begin{bmatrix} H^1 & H^1 \\ H^1 & -H^1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, \\
&\vdots \\
&\vdots \\
H^K &= \begin{bmatrix} H^{K-1} & H^{K-1} \\ H^{K-1} & -H^{K-1} \end{bmatrix}.
\end{aligned}
\tag{4.7}
$$

Let H_i^K be the *i-th* row of H^K, where $0 \le i \le 2^K - 1$. Every row of the H^K except H_0^K is a sequence with zero mean and unit variance. And the *I*-th row of the matrix will be the watermark sequence corresponding to the serial number I in our watermarking system. To get a watermark sequence that have ℓ elements from a given serial number I, we must expand the watermark sequence matrix for $\lceil \log_2 \ell \rceil$ times to obtain $H^{\lceil \log_2 \ell \rceil}$, and the *I*-th row of $H^{\lceil \log_2 \ell \rceil}$ will be the watermark sequence corresponding to the serial number *I*.

To embed the watermark sequence, the original image is first transformed into the frequency domain by performing the full-image DCT operation. And all of the DCT coefficients are reordered by a zigzag scan as shown in Fig. 4.3. A sequence of DCT coefficients in the zigzag order is then selected to embed the watermark sequence. After the embedding is done, the altered coefficients are put back and the full-image inverse DCT operation is performed to get the

watermarked image. The embedding flowchart is shown in Fig 4.7 and the detailed algorithm for embedding the watermark sequence is described in the following.

Step 1: Pick up a watermark serial number I and produce the corresponding watermark sequence W_I according to Eq. (4.7).

Step 2: Transform the reference image into the YC_bC_r color model before embedding, and use the Y channel (denote as C_y hereafter) to embed the watermark sequence.

Step 3: Let C be the original reference image of size $M \times N$. Transform the C_y channel into the frequency domain by performing an $M \times N$ DCT operation, and reorder the coefficients into an zigzag order.

Step 4: To embed the watermark sequence W_I, select the *(L+1)-th* to *(L+ℓ)-th* coefficients from the zigzagged DCT coefficients, and ignore the first L coefficients to achieve the perceptual invisibility of the mark. And let $T = \{t_{L+1}, t_{L+2}, t_{L+3}, \cdots\cdots, t_{L+(\ell-1)}, t_{L+\ell}\}$ be the selected coefficients.

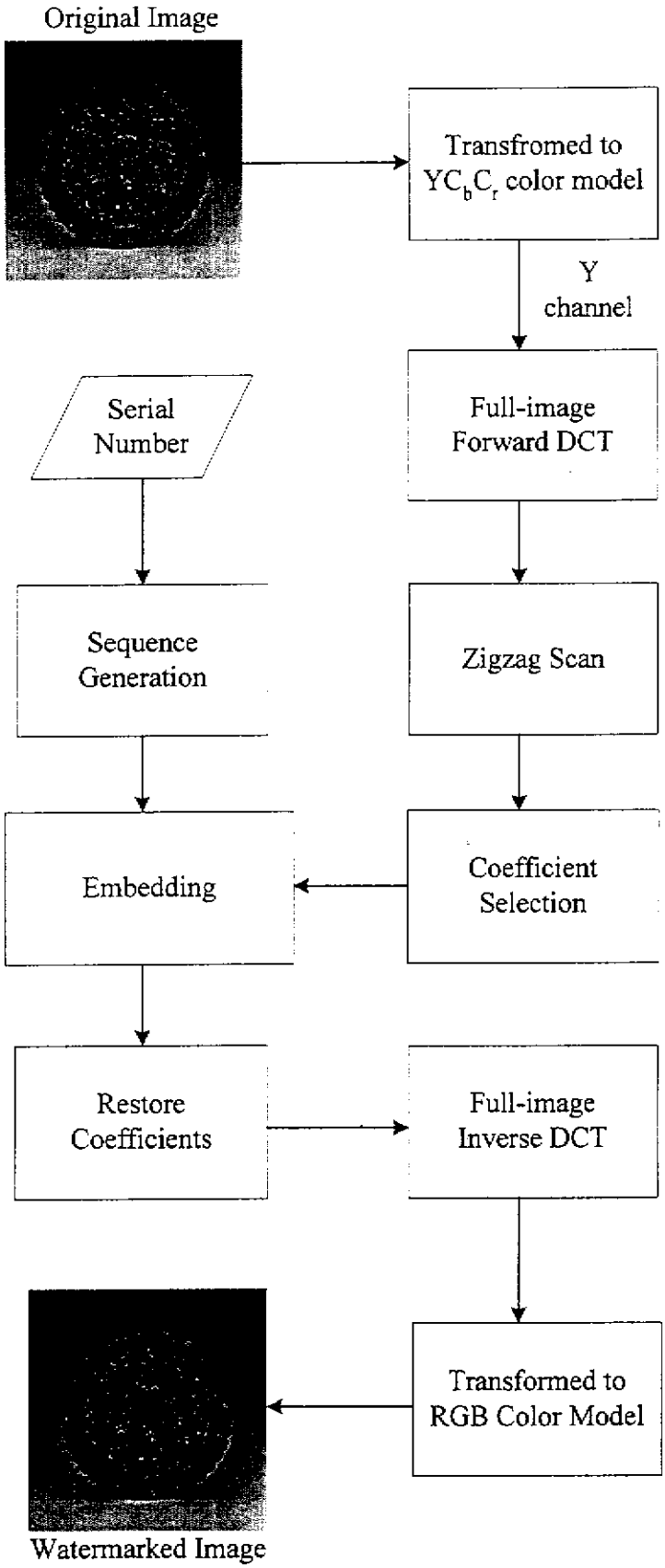


Figure 4.7 The watermark embedding flowchart.

Step 5: Embed the watermark sequence $W_I = \{w_{I1}, w_{I2}, w_{I3}, \cdots\cdots, w_{I(\ell-1)}, w_{I\ell}\}$, where w_{Ii} is the *i-th* element of W_I, in T by changing the coefficients by the following rule:

$$t'_{L+i} = t_{L+i} + \varepsilon |t_{L+i}| \, w_{Ii}, \tag{4.8}$$

where $1 \le i \le \ell$ and ε is a watermark strength factor.

Step 6: Reinsert the altered coefficients $T' = \{t'_{L+1}, t'_{L+2}, t'_{L+3}, \cdots, t'_{L+(\ell-1)}, t'_{L+\ell}\}$ into the DCT coefficients of C_y, and perform the inverse $M \times N$ DCT operation to get the watermarked C_y channel, and transform the image back into the RGB color model according to Eq. (4.3). And this completes the generation of the watermarked image.

4.3.2 Watermark Detection Process

In the watermark detection process, people who claim to be the copyright owner should provide a serial number to compute a correlation value. And the magnitude of this computed correlation can be used to determine if the watermark sequence is present in the image, and the ownership hence can be proved. A detailed algorithm is described as follows.

Step 1: Let C^* be the possibly corrupted image of size $M \times N$. Transform C^* into the YC_bC_r color model to get the Y channel C_y^*, and perform the $M \times N$ forward DCT operation on C_y^*.

Step 2: Reorder the DCT coefficients by a zigzag scan, and select the *(L+1)-th* to *(L+ℓ)-th* coefficients to form a vector

$$T^* = \{t^*_{L+1}, t^*_{L+2}, t^*_{L+3}, \cdots\cdots, t^*_{L+(\ell-1)}, t^*_{L+\ell},\}.$$

Step 3: Compute the correlation value z as follow:

$$z = \frac{W_J \cdot T^*}{\ell} = \frac{1}{\ell}\sum_{i=1}^{\ell} w_{J,i} t^*_{L+i}, \tag{4.9}$$

where W_J is the watermark sequence produced with serial number J.

In our implementation, z is compared with a predefined threshold to determine whether the watermark sequence is present (that is, whether the serial number is valid). In other applications, z is computed for each of the marks and the one with the largest correlation is assumed to be that really present in the image.

4.3.3 Experimental Results

In our experiments, three images with size 512×512 as shown in Figs. 4.8 (a), (b), and (c) are used as the original images. We chose I=500 to be the serial number to produce a watermark and embed the watermark sequence from 10000-th DCT coefficients. Every watermark sequence is 16384 in length and the watermark strength factor defined in Eq. (4.8) is selected to be $\varepsilon = 0.1$. The watermarked images are shown in Figs. 4.8 (d), (e), and (f), respectively. And the PSNR values are shown in Table 4.3. The results show that the watermark embedding method can embed watermark sequence without noticeable changes.

Table 4.3 The PSNR values of watermarked image with I=500, L=10000, $\ell = 16384, \varepsilon = 0.1$.

	Lena	Jet	Baboon
PSNR	41.3	41.0	39.1

To show the robustness of the watermark, a graph that shows the magnitudes of the correlation values from serial number 1 to 1000 is adopted in our system. Fig. 4.9 (a) shows the watermarked image after compressing by JPEG with quality factor 85, and the graph in (b) shows that the watermark sequence with serial

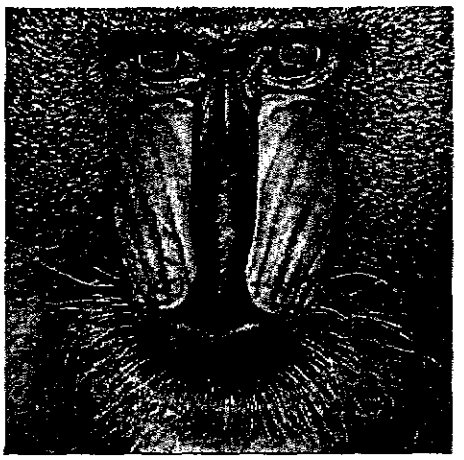
number 500 have the highest correlation magnitude among 1000 watermark sequence. Similarly, Fig. 4.10 (a) shows the watermarked image after compressing by JPEG with quality factor 30. The watermark sequence is still detectable with a high correlation. The images shown in Fig. 4.11 (a), Fig. 4.12 (a), Fig. 4.13 (a), and Fig. 4.14 (a) are the watermarked images after the operations of "histogram equalization", "invertion", "cropping", and "addition of 15% Gaussian noise" (processed by the Photoshop application software) were applied, respectively. And the graph shown to the right of them indicate that the watermark sequence with serial number 500 still have the highest correlation values.

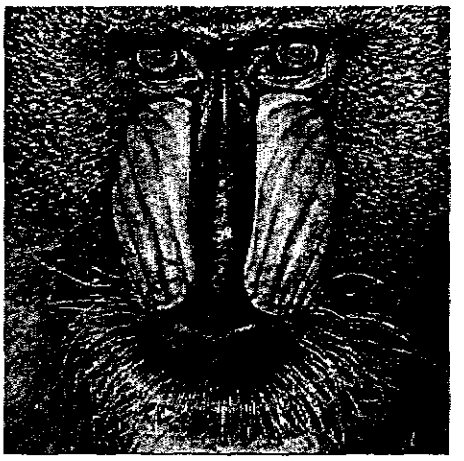
(a)

(d)

Figure 4.8 The original images and the watermarked images. (a) Original image "Lena". (b) Original image "Jet". (c) Original image "Baboon". (d)-(f) Watermarked images.

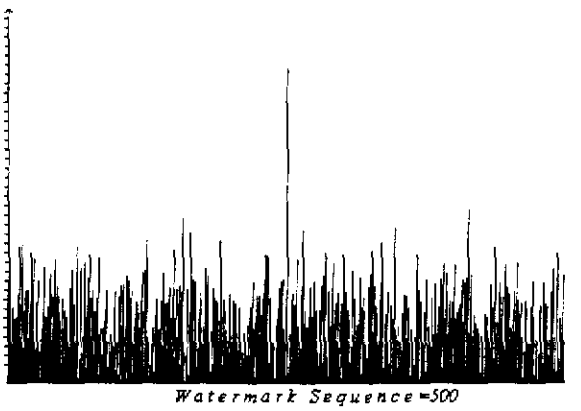
(b) (e)





(c) (f)

Figure 4.8 The original images and the watermarked images. (a) Original image "Lena". (b) Original image "Jet". (c) Original image "Baboon". (d)-(f) Watermarked images (continued).

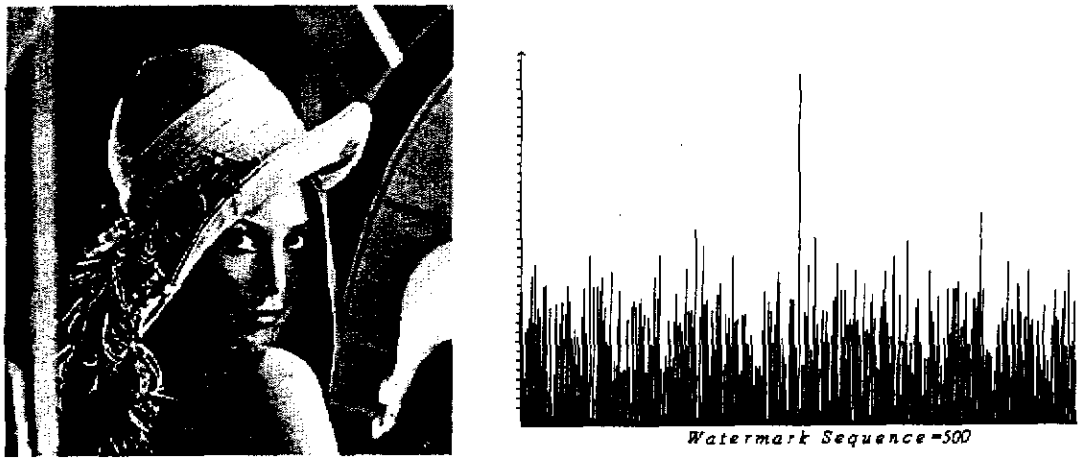


(a) (b)

Figure 4.9 The JPEG compressed image and the watermark detection result. (a) Image after JPEG compression with quality factor 85. (b) Watermark detection result.

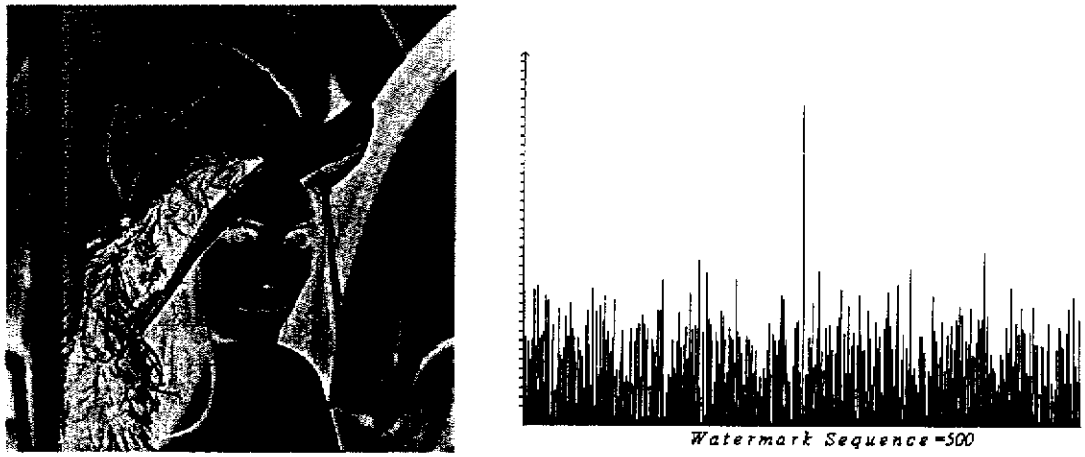
(a) (b)

Figure 4.10 The JPEG compressed image and the watermark detection result. (a) Image after JPEG compression with quality factor 30. (b) Watermark detection result.



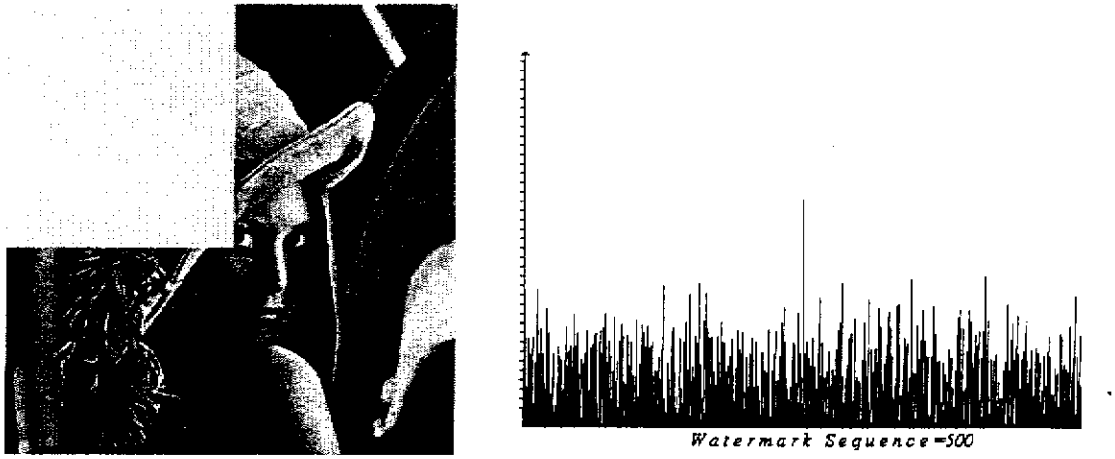
(a) (b)

Figure 4.11 The image after histogram equalization and the watermark detection result. (a) Image after histogram equalization. (b) Watermark detection result.



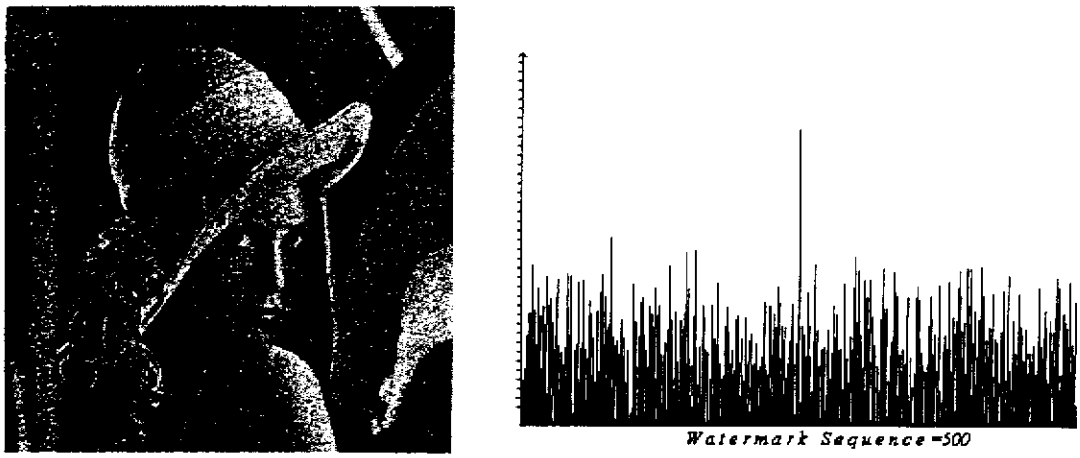
(a) (b)

Figure 4.12 The image after invert operation and the watermark detection result.(a) Image after invert operation. (b) Watermark detection result.



(a) (b)

Figure 4.13 The image after cropped and the watermark detection result. (a) Image after cropped, (b) Watermark detection result.



(a) (b)

Figure 4.14 The image after adding 15% Gaussian noise and the watermark detection result. (a) Image after adding 15% Gaussian noise. (b) Watermark detection result.

4.4 Authentication Scheme by DC-signature

The digitization of the antiques and arts in the museum has brought many advantages in the continuation and distribution of human cultures. But since reference images are exposed in the Internet environment, it is easy for someone to copy and edit them for unauthorized use, misappropriation, and misrepresentation. To devise a way for authenticating the fidelity of reference images now became an urgent issue for the content providers of digital museums. In this section, an authentication scheme for the reference image will be described. The DC values of every 8×8 block are utilized as the signature for verifying whether the image is tampered with. The proposed method also includes the ability to differentiate between "information preserving" and "information altering" transformations. In Section 4.4.1, some preliminary experimental results are shown to support the idea of adopting the DC value as a feature for tamper proving. In Section 4.4.2, the signature extraction processes will be described. In Section 4.4.3, the authentication processes with the extracted signature will be described. In Section 4.4.4, some experimental results will be shown.

4.4.1 Idea and Preliminary Experiments

Since the JPEG image format is adopted for reference images in the applications of digital museums, there is a good chance for content providers to re-compress the image with different JPEG quality factors, and the authentication method used here should not consider it as illicit tampering. That is, the features used for verifying image contents should have the ability to differentiate between "information preserving" transformations (such as JPEG compression, blurring, or sharpening operations) and "information altering" transformations (such as local feature replacement). In our preliminary experiments, we found that the DC coefficient of every 8×8 image sub-block is competent for this kind of job. In our experiments, we tried to observe the relationship between the variation degree of DC coefficients and some information preserving transformations. Table 4.4 shows some results of our preliminary experiments. Each image used is of size 512×512. And each number in the table entry is calculated by the following equation:

$$\frac{\frac{1}{M\times N}\sum_{i=1}^{M\times N}\left|DC_i - DC'_i\right|}{\frac{1}{M\times N}\sum_{i=1}^{M\times N}DC_i}\times 100\%, \tag{4.10}$$

where DC_i and DC'_i are the DC coefficients of the i-th 8×8 block in the original image and the transformed image, respectively; M and N are the numbers of 8×8 blocks in the horizontal and vertical orientations, respectively. For example, let the size of an image be 512×512, then $M=\left\lfloor\frac{512}{8}\right\rfloor$ and $N=\left\lfloor\frac{512}{8}\right\rfloor$. The results of our preliminary experiments show that the DC coefficient did not change too much after information preserving transformations (all below 5 %, even with JPEG compression factor 20). On the contrary, there is a

good chance for the information altering transformation to make larger modification on the DC coefficient. That is, we can differentiate information preserving transformations and information altering transformations by comparing the DC coefficients of the same image block.

4.4.2 Proposed Signature Extraction Method

In the proposed method, the DC value of every 8×8 image block is extracted out as the signature of the image. Fig. 4.15 shows the flowchart of this process. A source image is first divided into non-overlapping 8×8 blocks. The DC coefficient of every block is then calculated and added into the DC-signature. The entire process ends when blocks are exhausted. And the extracted DC-signature is then reordered by a pseudo-random mechanism. This will prevent illicit extraction of the signature without a correct random seed. The image cannot be correctly authenticated without the correct random seed in the authentication process, either. Finally, one copy of the DC-signature is dispatched to the AC (Authentication Center) and the copyright owner of the image keeps another copy.

Table 4.4 The variation degree of DC by applying some information preserving transformations on different images.

Image / Operation	Jet	Lena	Baboon	Painting
JPEG 80	0.20 %	0.26 %	0.48 %	0.94 %
JPEG 60	0.48 %	0.56 %	0.92 %	1.83 %
JPEG 40	0.71 %	0.91 %	1.33 %	2.84 %
JPEG 20	1.52 %	1.56 %	2.23 %	4.81 %
Blur + JPEG 85	0.62 %	0.53 %	1.02 %	2.15 %
Sharpen + JPEG 85	0.74 %	0.62 %	1.24 %	2.00 %

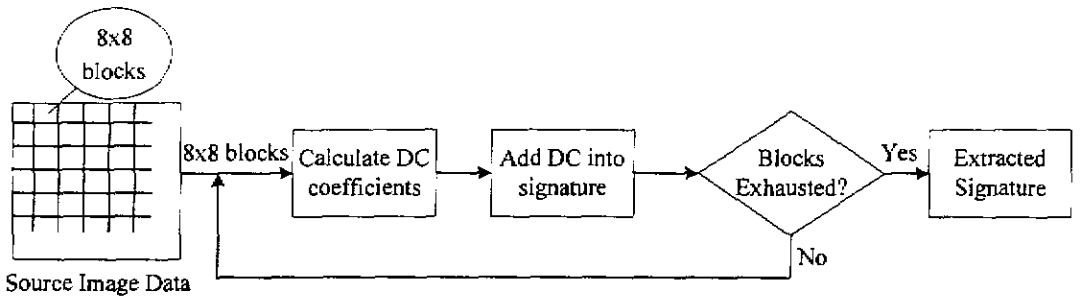


Figure 4.15 The DC-signature extraction flowchart.

4.4.3 Authentication Process by DC-signature

In the authentication process, the suspicious image and the DC-signature are needed. Fig. 4.16 shows the flowchart of the proposed authentication process. First, the DC-signature is obtained from the AC (Authentication Center) or the person who claims to be the copyright owner of the suspicious image. Before authentication, the content of the DC-signature is first reordered by the pseudo-random mechanism with a correct seed number. Then the suspicious image is divided into non-overlapping 8×8 blocks. The DC coefficients are calculated and then compared with the corresponding DC value in the DC-signature. Let DC_i be the DC coefficient of the *i-th* block in the suspicious image and DC'_i be the corresponding DC value in the DC-signature. If the difference between DC_i and DC'_i is larger than a threshold t, then the 8×8 image block is considered as being tampered with by some information altering transformation. On the contrary, if the difference is smaller than t, the image block is determined not being tampered with or just being operated by some information preserving transformation (such as JPEG compression).

In our implementation, a visual inspection tool for localizing the altered blocks is provided in our system. The black color is used to replace the block that is judged as being tampered with, and the block that is judged as not being

tampered with is unchanged.

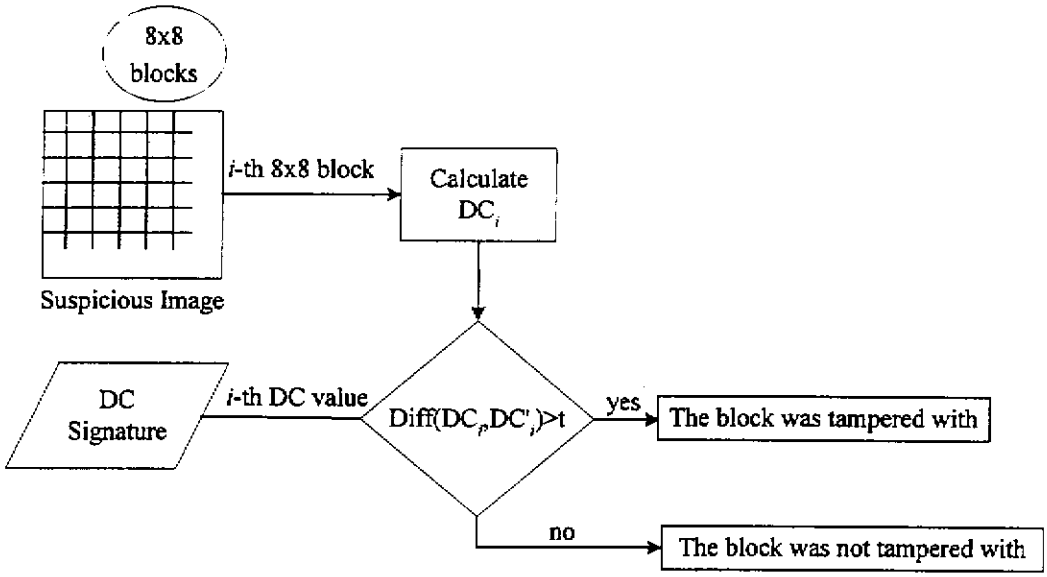
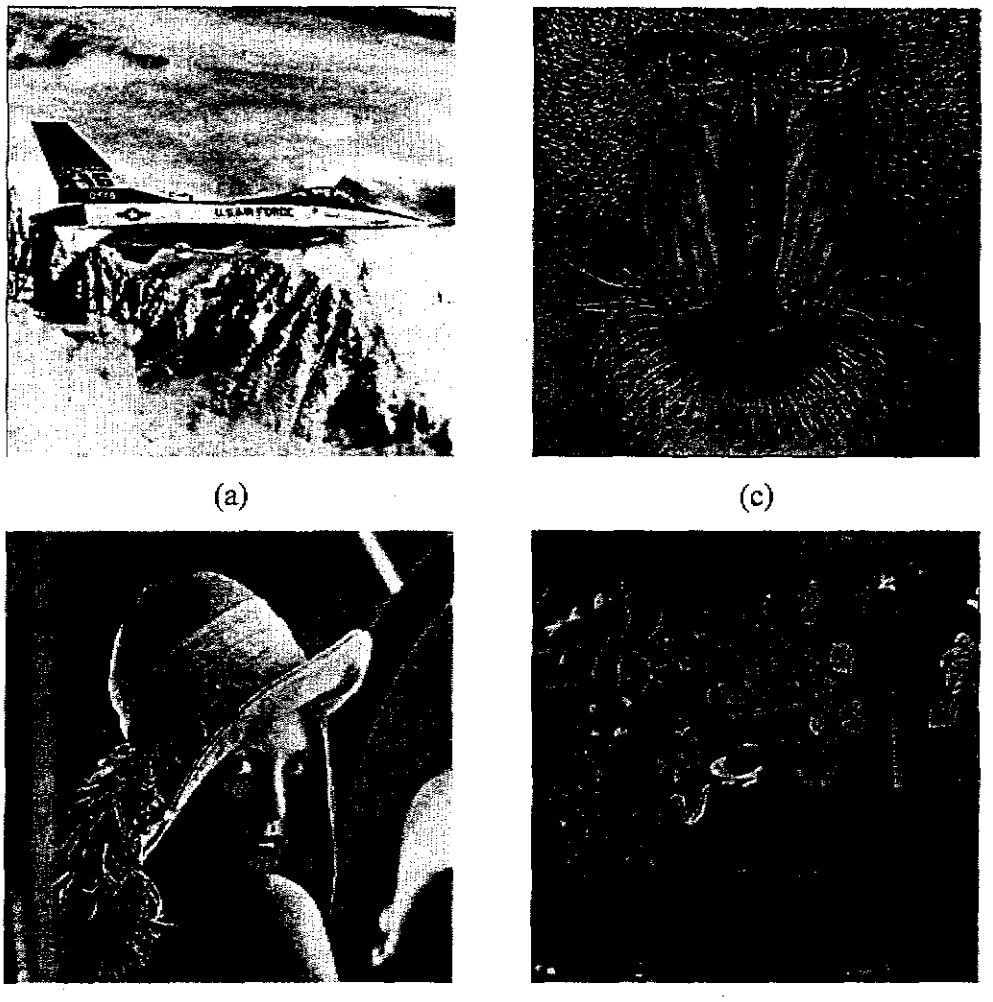


Figure 4.16 The flowchart of authentication with DC-signature.

4.4.4 Experimental Results

Four images as shown in Figs. 4.17(a), (b), (c), and (d), each with size 512×512, are used as the original images in our experiments. A DC-signature is first extracted out from each of these images. Some tampered images are shown in Figs. 4.18(a), (b), (c), and (d). Each image is first altered by some kinds of feature replacement operations and then re-compressed by JPEG with quality factor 85. All of the image operations were performed by Photoshop, which is an application software for image processing. In Fig. 4.18(a), the characters on the body and rear fin of the jet were replaced. In Fig. 4.18(b), the entire face of Lena was replaced by someone else's face. In Fig. 4.18(c), three colors were added on the cheek of Baboon and two lines were drawn on both sides of her face. In Fig. 4.18(d), three additional color lines and two leaves were drawn. The authentication results with the use of the DC-signature are shown in Fig. 4.18(e), (f), (g), and (h), with threshold t = 0.05. The experimental results show that the areas affected by

information altering transformations can be marked out with high probabilities. And the areas only affected by information preserving transformations are judged as not being tampered with. That is, the DC-signature is appropriate for differentiating information preserving and information altering operations.



(a) (c)

(b) (d)

Figure 4.17 The original images with size 512×512 (a) Original image "Jet". (b) Original image "Lena". (c) Original image "Baboon". (d) Original image "Painting".

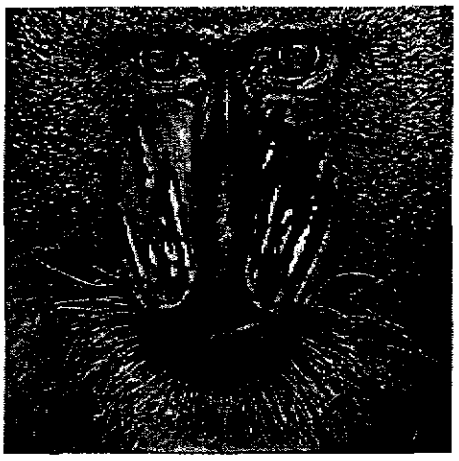
(a)

(e)

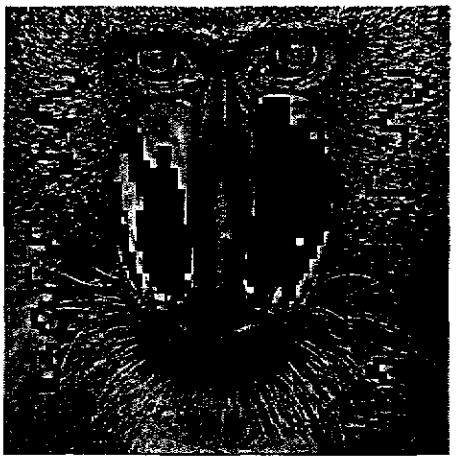
(b)

(f)

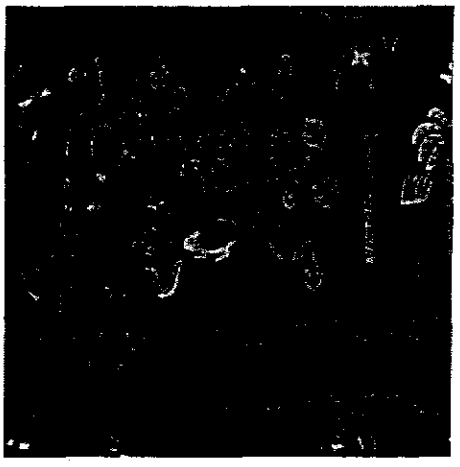
Figure 4.18 The tampered images and the authentication results. (a) Tampered image of "Jet". (b) Tampered image of "Lena". (c) Tampered image of "Baboon". (d) Tampered image of "Painting". (e) Authentication result of (a). (f) Authentication result of (b). (g) Authentication result of (c). (h) Authentication result of (d).



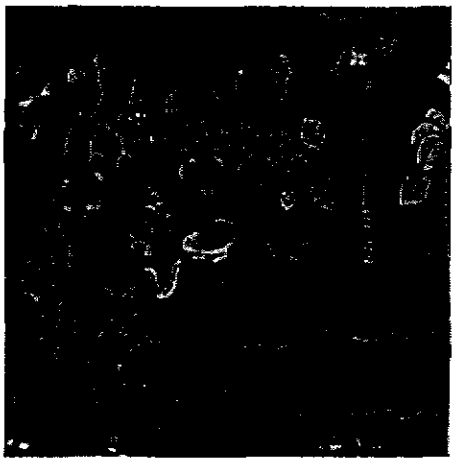
(c)



(g)



(d)



(h)

Figure 4.18 The tampered images and the authentication results. (a) Tampered image of "Jet". (b) Tampered image of "Lena". (c) Tampered image of "Baboon". (d) Tampered image of "Painting". (e) Authentication result of (a). (f) Authentication result of (b). (g) Authentication result of (c). (h) Authentication result of (d) (continued).

4.5 Discussion

In this chapter, a system that can embed annotation data and a serial number simultaneously within reference images is proposed. And the proposed system can also extract a signature from a reference image for verifying its integrity. In the applications of digital museums, the JPEG image is adopted for the reference image. The proposed system embeds annotation data in the DCT-domain by

changing the relative sizes of two DCT coefficients within every 8×8 block. The embedded annotation can still be extracted after the stego-image is compressed by JPEG. The proposed system can also protect the copyright and prove the ownership of a reference image by embedding a watermark sequence, which is produced according to a serial number, in the DCT-domain of the reference image. The watermark can still be correctly detected even after the watermarked image is compressed by JPEG with low quality factor, or modified by some signal processing operations (such as Gaussian noise addition, invertion, histogram equalization), etc. The proposed system extracts the DC value of every 8×8 block as a signature to verify the integrity and fidelity of the reference image. The extracted signature has the ability to differentiate information altering operations from information preserving operations. Tampering on a reference image can be detected and localized. And the system also provides a visual inspection tool that shows the block judged as being tampered in black color.

Chapter 5 Copyright and Annotation Protection Schemes for Thumbnail Images

5.1 Introduction

In this chapter, the techniques for protecting the copyright and annotation data of the thumbnail image will be described. Because of the limited Internet bandwidth, there is always a tradeoff between the image file size and the image quality for the content provider. To effectively utilize the Internet resource, a small-sized image that shows only the outline of an object is widely adopted on the WWW environment and is referred to as the thumbnail image. In Section 5.1.1, the use of the thumbnail image in the applications of digital museums will be introduced. In this report, the GIF (Graphics Interchange Format) image file format is adopted for the thumbnail image. Therefore, some characteristics of the GIF image and a palette-sorting algorithm will be described in Section 5.1.2. An annotation-hiding scheme by a palette index replacing method will be proposed in Section 5.2. A logo-hiding scheme by using the palette index replacement method will be proposed in Section 5.3. And an authentication scheme using sorted color palettes will be proposed in Section 5.4. Finally, a discussion on the thumbnail image will be made in Section 5.5.

5.1.1 Uses of Thumbnail Images

In the applications of digital museums, the thumbnail image is defined as an image that shows only the outline of an art to the visitor who browses the web pages of the museum. If the visitor is interested in a certain thumbnail image, then the reference version of that thumbnail image will be brought in front of the visitor after clicking on it. For this purpose, the dimensions of thumbnail images are usually smaller than those of other kinds of image formats. And the thumbnail

image need not be of true color because it is just a preview version of the reference image. The GIF image format announced by CompuServe Co. in 1987 and 1989 suits these characteristics very well. As a result, we use the GIF image file format for thumbnail images in digital museums and develop techniques to hide logo and annotation information inside GIF images. And an authentication method is proposed to detect and locate possible tampering on GIF images. A museum logo, the annotation associated with a given image, and some fragile watermark information will all be embedded in the image simultaneously.

5.1.2 Characteristics of GIF Images

In order to hide information within the GIF image, some characteristics must be identified:

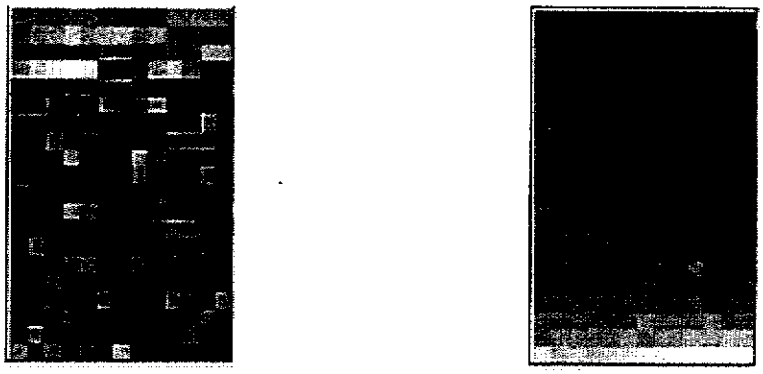
1. every GIF image contains at most 256 colors;
2. a color palette is used for storing 256 colors used in the image;
3. every pixel can be thought as an index number, which is a reference to the color palette, and so every pixel needs one byte of storage.

In this report, the proposed method is closely dependent on the pixel's index number within the color palette. But almost every image processing software has a different method to store the palette data. Even when we just re-save the same image by different software, the palette data may be different from the original one. As a result, we will first sort the color palette before the hiding process to make it in the dark-to-bright order. After sorting, we may reference the same palette data. This will facilitate the subsequent hiding process.

Let $p_i = (R_i, G_i, B_i)$ be the *i-th* color within the color palette, where $0 \le i \le 255$. To sort the palette colors, we first calculate

$$\mu(p_i) = \omega_1 \times R_i^2 + \omega_2 \times G_i^2 + \omega_3 \times B_i^2 \quad (5.1)$$

for every color in the palette, where ω_1, ω_2, *and* ω_3 are weighting values that make color such as (140,130,120) to be different from (130,140,120),and $\omega_1 \neq \omega_2 \neq \omega_3$. Then we sort $\mu(p_i)$ in a descending order to obtain a sorted palette. Fig 5.1 (a) shows a color palette before sorting and Fig 5.1 (b) shows the sorted result of Fig5.1 (a).



(a) (b)

Figure 5.1 The color palette before and after sorting. (a) The color palette before sorting. (b) The color palette after sorting.

5.2 Annotation Hiding Scheme by Palette Index Replacement

In this section, the proposed method for embedding and extracting the annotation information will be described. An even-odd relationship between two pixels is utilized to embed or extract a binary bit. The embedding process will be described in Section 5.2.1. And the extracting process will be described in Section 5.2.2. Some experimental results will be shown in Section 5.2.3.

5.2.1 Proposed Method

Let C be a cover image of size $M \times N$. Let S be the annotation that will be

hidden into C, which has L characters in length. And let β be the sorted palette, with β_i $(0 \le i \le 255)$ being a color inside β that has index i. Before the embedding process, S must first be converted into a binary form $S = s_1 s_2 \ldots\ldots s_8 \ldots\ldots s_{L*8-1} s_{L*8}$ according to the ASCII codes of the characters in S. As Fig. 5.2 shows, we hide the binary data into the four marked area (I, II, III, and IV) of every 3x3 block in C, and left the central pixel unchanged, which will be used for hiding fragile information as described in Section 5.4.

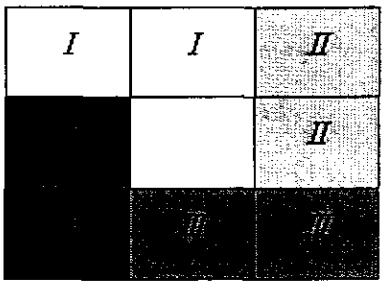


Figure 5.2 A 3×3 block.

Let the Euclidean distance between the two colors be

$$\mu(\beta_x, \beta_y) = \sqrt{(R_x - R_y)^2 + (G_x - G_y)^2 + (B_x - B_y)^2}, \qquad (5.2)$$

where $0 \le x, y \le 255$. The algorithm for embedding a bit of annotation, say s_i, can be expressed briefly as follows.

Step 1: Select a two-pixel block (I, II, III or IV) first from a 3x3 block. Name the two involved pixels as α_1 and α_2.

Step2: By viewing every pixel as an index number in the GIF image format, get the color indices i_1 and i_2 of the two selected pixels α_1 and α_2 from the sorted palette β, respectively.

Step3: Use the following rule to judge whether i_1 and i_2 are even or odd numbers:

$$isEven(\iota) = \begin{cases} true & \text{if } \iota \bmod 2 = 0; \\ false & \text{if } \iota \bmod 2 = 1. \end{cases} \tag{5.3}$$

Step4: In the case that $s_i = 1$ and $isEven(\iota_1) \neq isEven(\iota_2)$, modify one of the pixels by another color in β to make the two indices either both even or both odd in a quality preserving fashion. Let β_{r_1} and β_{r_2} be two candidate colors that will be used to replace α_1 or α_2, respectively. Choose the color β_{r_1} to be the one among β that has minimal Euclidean distance to α_1 and satisfy the condition $isEven(\iota_1) \neq isEven(r_1)$. On the other hand, choose the candidate color β_{r_2} to be the one among β that has minimal Euclidean distance to α_2 and satisfy the condition $isEven(\iota_2) \neq isEven(r_2)$. And then modify one of the pixels by following condition:

$$\begin{cases} if \ \ \mu(\alpha_1, \beta_{r_1}) > \mu(\alpha_2, \beta_{r_2}) & \text{replace } \alpha_2 \text{ by } \beta_{r_2}, \\ if \ \ \mu(\alpha_1, \beta_{r_1}) < \mu(\alpha_2, \beta_{r_2}) & \text{replace } \alpha_1 \text{ by } \beta_{r_1}. \end{cases} \tag{5.4}$$

Otherwise, in the case that $s_i = 1$ and $isEven(\iota_1) = isEven(\iota_2)$, just leave the two pixels unchanged.

Step5: In another case that $s_i = 0$ and $isEven(\iota_1) = isEven(\iota_2)$, modify one of the pixels to make one of the index be even and the other be odd. Perform similar operations in Step 4 to obtain two candidate colors β_{r_1} and β_{r2} from β and use them to replace α_1 or α_2 according to the condition listed in Eq. (5.4). Otherwise, if $s_i = 0$ and $isEven(\iota_1) \neq isEven(\iota_2)$, just leave the two pixels unchanged.

In the proposed method, the even-odd relationship of the two chosen pixels is utilized to embed a binary bit of annotation information. The indices ι_1 and ι_2 are

tuned up to become both even or both odd to represent the fact that a "1" is embedded. On the contrary, a "0" is embedded by adjusting the indices of the two chosen pixels to become different in the even-odd relation, that is, for one index to become even and the other to become is odd. There is a good chance that the original indices just fit these two constraints. In this kind of situation, what we have to do is just to leave the two pixels unchanged. On the other hand, when the indices of the two pixels do not meet the constraints, a color with its index satisfying to the constraints described in Step 4 and Step 5 must be found to replace the corresponding pixel. In order to preserve the quality of the stego-image, the color with the minimal Euclidean distance to the corresponding pixel is the best candidate for replacement.

5.2.2 Annotation Extraction Process

In some conventional methods, besides the embedding result, an extraction process needs either a lookup table or the source image to obtain the secret message. This is inconvenient for the receiver to get the secret message because the user has to keep the source image or a table. In our proposed method, only the stego-image is needed to extract the annotation message.

Because the annotation embedded in the image can be any kind of description with an unfixed length, in our implementation the length of the annotation is first embedded before the annotation content is treated. As a result, the length information must first be extracted in the extraction process so that we can know how many blocks should be examined before ending the process.

Let E be the stego-image to be processed, β be the sorted palette of E, and s_i be the i-th extracted bit of the annotation data. The process of extracting a bit from the stego-image is described below:

Step1: Choose a two-pixel block (see Figure 5.3) from a 3×3 block of E.

And name the two involved pixels α_1 and α_2.

Step2: Get the indices ι_1 and ι_2 of α_1 and α_2, respectively, from β according to their RGB values.

Step3: The annotation s_i can be determined by the following rule:

$$s_i = \begin{cases} 1 & \text{if } \iota_1 \text{ and } \iota_2 \text{ are both even or both odd;} \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

In the entire extraction process, we first extract the first ℓ bits of the data that compose a number that is the value of the length of the annotation embedded in E. Then we can know how many two-pixel blocks we should examine and after repeating the above-mentioned process by $\ell \times 8$ times, we can get the binary form of the extracted annotation $S = s_1 s_2 \ldots\ldots s_8 \ldots\ldots s_{\ell*8-1} s_{\ell*8}$. According to the ASCII codes, S is then converted into character form that finally results in the embedded annotation.

5.2.3 Experimental Results

In our experiments, the images "Lena", "Baboon", and "Pepper" with size 512×512, which are shown in Fig. 5.3 (a), (b), and (c), are used as the cover images. And Fig. 5.3 (d), (e), and (f) are the stego-images after embedding 10,000 characters. The PSNR values of the stego-images are listed in Table 5.1. The PSNR values are high, which mean that only little perceptual effect is created during the annotation embedding process. And the embedded annotation can be extracted without any error, according to the experimental results.

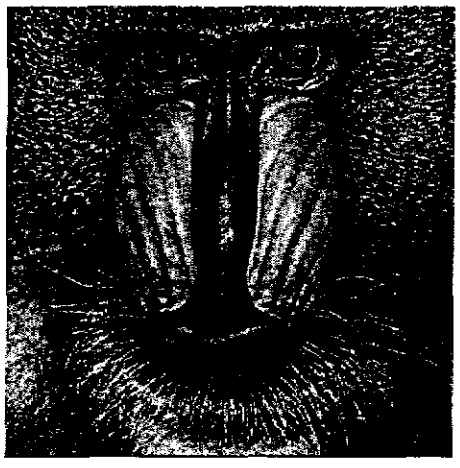
Table 5.1 The PSNR values after embedding 10,000 characters.

	Lena	Baboon	Pepper
PSNR	39.0	35.0	37.0

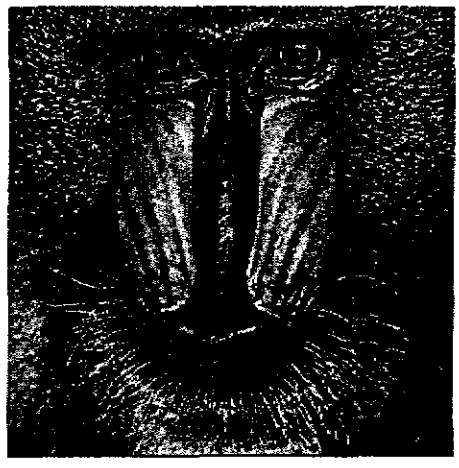
(a)

(d)

Figure 5.3 The cover images and stego-image with 10,000 characters embedded. (a) Cover image "Lena". (b) Cover image "Baboon". (c) Cover image "Pepper". (d) Stego-image "Lena" with 10,000 characters embedded. (e) Stego-image "Baboon" with 10,000 characters embedded. (f) Stego-image "Pepper" with 10,000 characters embedded.



(b)



(e)

(c) (f)

Figure 5.3 The cover images and stego-image with 10,000 characters embedded. (a) Cover image "Lena". (b) Cover image "Baboon". (c) Cover image "Pepper". (d) Stego-image "Lena" with 10,000 characters embedded. (e) Stego-image "Baboon" with 10,000 characters embedded. (f) Stego-image "Pepper" with 10,000 characters embedded (continued).

5.3 Watermarking Scheme by Palette Index Replacement

In this section, a method used for embedding a logo image within a thumbnail image will be described. In the applications of digital museums, the museum logo is a persuasive symbol that can be used to claim the ownership of the image. Based on the method described in Section 5.2, the museum logo can be embedded within the thumbnail image for protecting the copyright of that image. In Section 5.3.1, the process of embedding a museum logo within a thumbnail image will be described. In Section 5.3.2, a logo extraction process will be described. Finally, some experiments will be shown.

5.3.1 Proposed Method

Let C be a cover image of size $M \times N$, and L be a binary logo image of size $I \times J$ that will be embedded within C. Since L is a binary image, L can be converted into binary form $L = (l_1 l_2 l_3 \ldots l_{I \times J})_2$ before embedding. In the embedding process, every bit of L will be embedded by changing the even-odd relation between two pixels in a 3×3 block, which is described in Section 5.2.1. In the implementation, the width and height of logo image (I and J) will be first converted into binary stream and then embedded within the cover image since they are important information about how much two-pixel blocks should be examined in the logo extraction process. After I and J are embedded, the logo data L are embedded by turning up the even-odd relation of the corresponding two-pixel block sequentially until all the bits in the logo image L is exhausted. Fig

5.4 shows the flowchart of the entire embedding process.

5.3.2 Watermark Extraction Process

In the watermark extraction process, no other information but the stego-image is needed for extracting the museum logo. Before extracting the logo data, the color palette of the stego-image must be resorted to make it in the dark-to-bright order. The logo data then can be extracted by examining the even-odd relation of a two-pixel block according to Eq. (5.4). That is, if the two indices are both even or both odd, the extracted bit is "1". Otherwise, if one of the indices is even and the other one is odd, the extracted bit is taken to be "0". The width and height information about the embedded logo will be first extracted. Let I and J be the extracted width and height, respectively. That means $I \times J$ two-pixel blocks should be examined before stop. Let $L = (l_1 l_2 l_3 \ldots l_{I \times J})_2$ be the extracted logo data. In cooperate with the width I and height J, the embedded copyright logo can be reconstructed. Fig. 5.5 shows the logo-extraction flowchart.

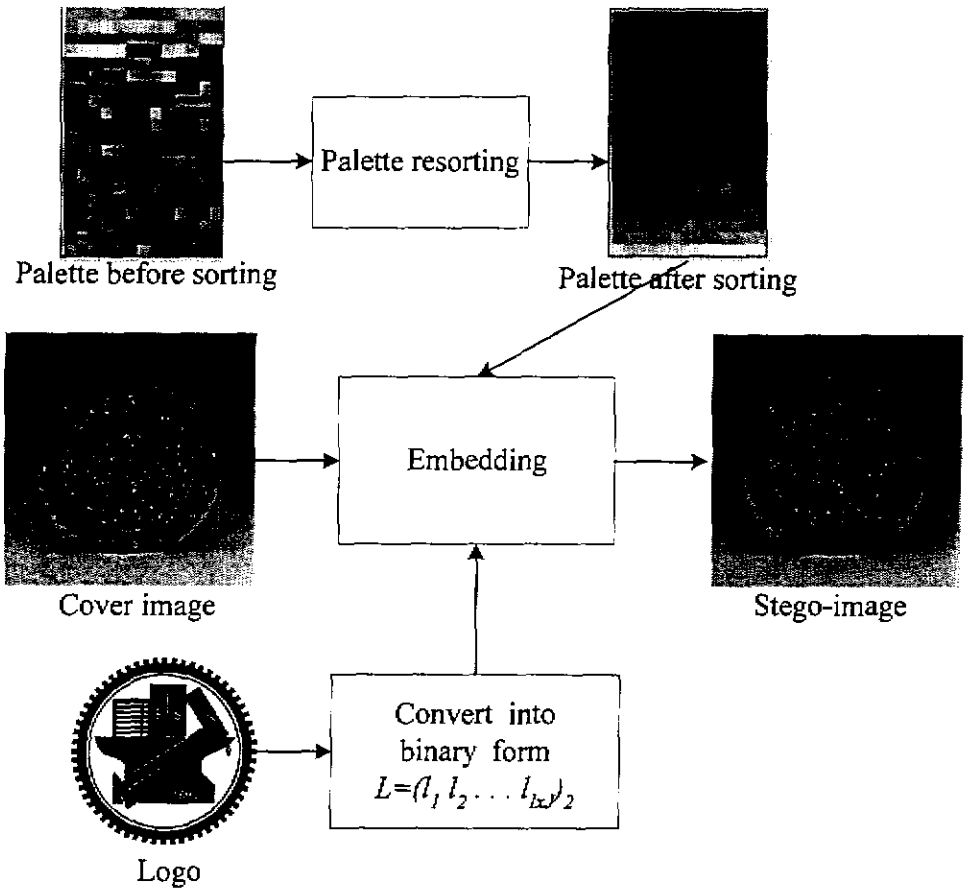


Figure 5.4 The logo embedding flowchart.

5.3.3 Experimental Results

In our experiments, the image shown in Fig 5.6 with size 256×256 is used as the logo image. And the three images as shown in Fig. 5.7 (a), (b), and (c) are used as the cover images. The images with the logo being embedded are shown in Fig. 5.7 (d), (e), and (f). The PSNR values are shown in Table 5.2.

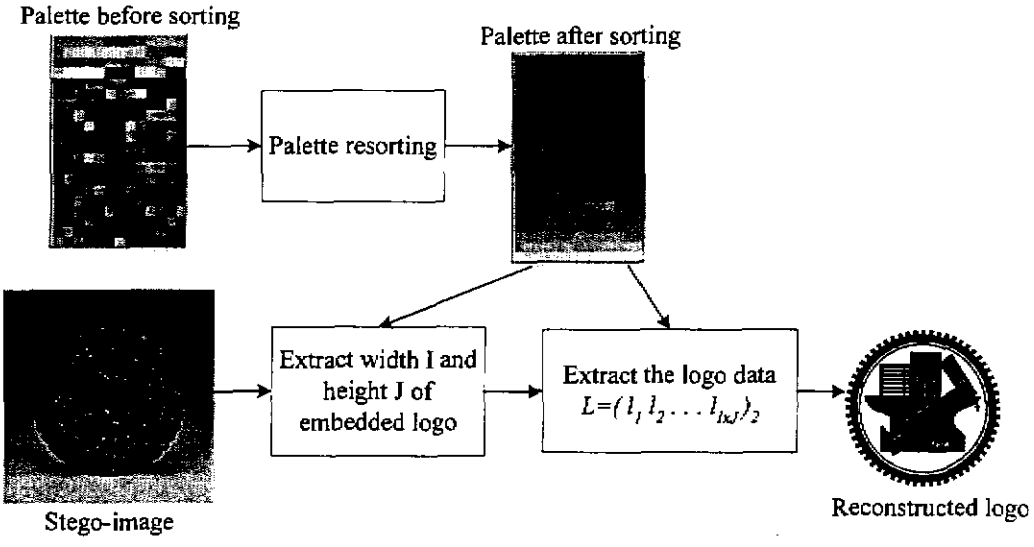


Figure 5.5 The logo extration flowchart.

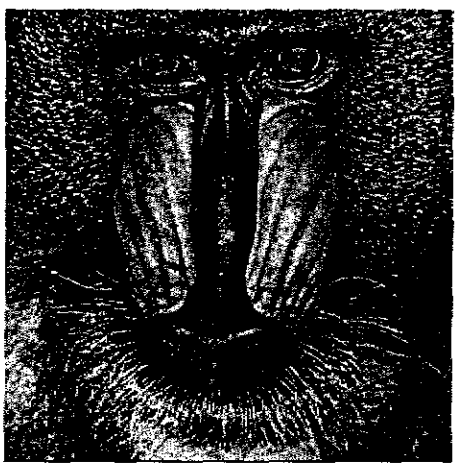
Figure 5.6 The image logo with size 256×256.

Table 5.2 The PSNR values of the stego-images after embedding Fig. 5.6.

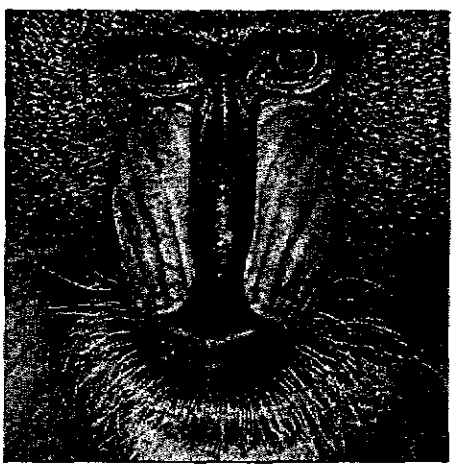
	Lena	Baboon	Pepper
PSNR	34.0	29.0	33.0

(a)

(d)



(b)



(e)

(c)

(f)

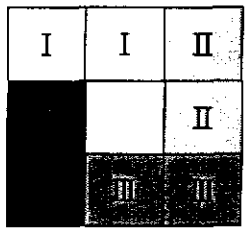
Figure 5.7 The cover images and the stego-images after embedding the image of Fig. 5.6. (a) Cover image "Lena". (b) Cover image "Baboon". (c) Cover image "Pepper". (d) Stego-image "Lena" after embedding Fig 5.6. (e) Stego-image "Baboon" after embedding Fig 5.6. (f) Stego-image "Pepper" after embedding Fig 5.6.

5.4 Authentication Scheme by Nearest Palette Color Replacement Method

In this section, a novel method for authenticating the integrity and fidelity of the thumbnail image is proposed. Alternations to the watermarked image can be detected and localized. And the verification processes can be proceeded without referencing the original image. In Section 5.2 and Section 5.3, the annotation data and the museum logo are embedded in the surrounding eight pixels of a 3×3 block and the central pixel is left unchanged. In the proposed method, the central pixel of every 3x3 block is utilized to embed the fragile watermark. The fragile watermark embedding process will be shown in Section 5.4.1. The authenticating process will be described in Section 5.4.2. And several experimental results are shown in Section 5.4.3.

5.4.1 Proposed Fragile Watermark Embedding Method

Let C be a cover image of size $M\times N$, β be the sorted palette of C, and β_i be the *i-th* color in β $(0\le i\le 255)$. The cover image C is first divided into non-overlapping 3x3 image blocks, like the one shown below.



We utilize the surrounding four two-pixel blocks (I, II, III and IV) to embed the annotation and the logo information by using the method described in Section 5.2 and Section 5.3. After embedding, every block's central pixel is still unchanged. This pixel is hence used for embedding the fragile watermark. To embed the fragile watermark

into a 3×3 block B, the algorithm can be shown as the following steps:

Step 1: Calculate the mean value μ of B, which can generally represent the color feature of a block. And the central pixel of the block is supposed to be close to μ in color. Since μ is the mean color of a block, there is a good chance that it is not contained in β.

Step2: Find a color β_k from β that is closest to μ according to the Euclidean distance of the RGB values, and $k_{\bmod Q}=0$. Here Q is a watermark strength factor.

Step3: Replace the color value of central pixel of the block by β_k.

The magnitude of Q in Step 2 is a tradeoff between the probability to prove the tampering and the quality of the stego-image. When Q is a small value, the quality of stego-image is relatively higher because the candidate colors could be more so that we can find a closer color for replacing the central pixel of every 3x3 block. But the opportunity of proving and localizing the tampering will became relatively lower. On the contrary, if Q is large, the image quality will become poorer but the probability of detecting tampering will become higher. The detailed data about the influence of Q and some discussions will be shown in Section 5.4.

5.4.2 Authentication Process

The authentication process can proceed without referencing the original image in our proposed method. Let T be the image that is suspicious of being tampered. And let β be the sorted palette of T, and β_i be the i-th color in β $(0\le i\le 255)$. To authenticate whether T has been tampered, we first divide T into 3x3 non-overlapping blocks. For every block, we focus on the central pixel p and get the palette index k of

p from β, where the RGB values of β_k are equal to those of p. In the embedding process described in Section 5.4.1, the central pixel of every block is replaced by the color whose index is a multiple of Q. Therefore, k should be a multiple of Q if this block has not been tampered with. That is, if k is a multiple of Q, the block is judged as not being tampered with. On the other hand, the block is decided as being tampered with if k is not a multiple of Q.

For visualization, we replace the block by its inverse color if the authentication result indicates the block has been altered. This will help us to localize the tampering area in the suspicious image.

5.4.3 Experimental Results

In our experiments, we use three cover images as shown in Fig. 5.8 (a), (b), and (c) with size 512×512. And Fig. 5.8 (d), (e), (f) are the images after embedding a fragile watermark with Q = 8. We almost cannot tell the difference between the cover and the stego-images. The PSNR values are shown in Table 5.3.

Table 5.3 PSNR values of watermarked images.

	Jet	Lena	Pepper
PSNR	36.0	35.4	34.2

Figs. 5.9 (a), (b), and (c) are the images that have been tampered. And the authentication results are shown in Figs. 5.9 (d), (e), and (f). In Fig. 5.8 (a), we tampered the stego-image by rubbing the characters and the flag on the body of the airplane. In Fig. 5.9 (b), we altered Lena's hair color and paste some wig on her hat. In Fig. 5.9 (c), we cropped two green peppers and pasted them onto different parts of the stego-image. The experimental results showed that the altered areas are accurately marked out.

Fig. 5.10 are the embedded results of "Lena" by applying different Q values. When Q becomes larger, the quality of the stego-image goes down. In Fig. 5.10 (c) and Fig. 5.10 (d), we can find some false contouring effect on the girl's shoulder and cheek because there are only 4 or 2 colors can be used for replacing the central pixel of every 3x3 block. The relations between the size of Q and the PSNR values of the stego-image is shown in Table 5.4.

Table 5.4 The PSNR value of stego-image by applying different Q value.

	Q=16	Q=32	Q=64	Q=128
PSNR	33.2	32.2	29.0	25.0

5.5 Discussions

In this chapter, a system that embeds annotation data, an image logo, and a fragile watermark simultaneously within the thumbnail image has been proposed. For the applications of digital museums, the GIF image file format is adopted for the thumbnail image for this report. The GIF image contains a color palette with all the colors used in that image. In order to reference the palette in the same order, the color palette is first reordered from dark to bright. In the embedding process, the annotation and the logo image are embedded in the eight surrounding pixels of a 3×3 block. To embed a bit of data, two pixels are first selected and the indices of these two pixels are obtained from the color palette. The indices are tuned up to become both even or both

(a)

(d)

(b)

(e)

(c)

(f)

Figure 5.8 The cover images and the fragile watermarked images (a) Cover image "Jet". (b) Cover image "Lena". (c) Cover image "Pepper". (d) Watermarked image "Jet". (e) Watermarked image "Lena". (f) Watermarked image "Pepper".

(a)

(d)

(b)

(e)

(c)

(f)

Figure 5.9 The suspicious images and their verification results. (a) Tampered "Jet". (b) Tampered "Lena". (c) Tampered "Pepper". (d) Verification result of tampered "Jet". (e) Verification result of tampered "Lena". (f) Verification result of tampered "Pepper".

(a) Q = 16

(b) Q = 32

(c) Q = 64

(d) Q = 128

Figure 5.10 The watermarked image with different Q values. (a) Watermarked image with Q = 16. (b) Watermarked image with Q = 32. (c) Watermarked image with Q = 64. (d) Watermarked image with Q = 128.

odd if a "1" is to be embedded. Otherwise, the indices are tuned up to become different in the even-odd relation, that is, with one even and the other odd. To embed the fragile watermark, the mean value of every 3×3 block is calculated. And the central pixel is replaced by a color that is closest to the mean value of the block. And the magnitude of the index of this color must be a multiple times of the value of a predefined watermark strength factor.

Chapter 6 Conclusions and Suggestions

6.1 Conclusions

In this report, a comprehensive system for protecting the copyright and annotation data of various image formats in the applications of digital museums is proposed. In the proposed system, annotation data are embedded within cover images imperceptibly to facilitate the association between images and commentary data. Watermarks, which can be adopted as courtside evidence, are embedded within images to prove the copyright and ownership of the watermarked image. And a fragile watermark or a signature is used in our system to verify the integrity and fidelity of a suspicious image. The proposed system has the ability to detect whether an image has been tampered with, and localizes the exact position of the tampered area within the tampered image. A visual inspection tool is also provided in the proposed system, which let a user easily point out the areas that have been tampered in the authentication result.

In the proposed system, annotation data are embedded within the archive image by replacing two LSB's of an image pixel. And as many copies of the annotation as possible are embedded for best space utilization. Each copy of the annotation is separated by boundary line signals. With the help of the boundary line signal, the annotation data hence can still be extracted even when the stego-image is cropped. To protect the copyright of an archive image, a museum logo is embedded by replacing one LSB of the image pixel to declare its ownership. And a fragile watermark based on a human visual model is proposed in this report to verify the integrity and fidelity of an archive image. With the help of the boundary line signal, which is embedded together with the fragile watermark, the suspicious image is still authenticable even

when it is cropped.

To protect the copyright and annotation data of a reference image, the proposed system embed the annotation data in the DCT-domain by controlling the relative sizes of two DCT coefficients of every 8×8 block. And a watermark sequence, which is produced according to a serial number, is embedded in the full-frame DCT-domain of a reference image to protect its copyright and ownership. The embedded watermark is still detectable even after being attacked by many signal operations and the JPEG lossy compression. To verify a reference image, a signature composed of the DC value of every 8×8 block is first extracted from the image. By comparing the DC value of the suspicious image and that of the extracted signature, the altered area can be detected and localized with high probability.

To protect the copyright and annotation data of a thumbnail image, the proposed system first sorts the color palette of the image by the Euclidean distance of the RGB values, which makes unique references to the palette data later in the remainding process. The proposed method embeds the annotation data and a museum logo by controlling the even-odd relationship between two pixels' indices in the sorted color palette. And a fragile watermark method is proposed to verify the integrity and fidelity of the thumbnail image, too. The fragile watermark is embedded imperceptibly by replacing the central pixel of every 3×3 block by the color whose index is a multiple of a watermark strength factor and is closest to the mean of the 3×3 block in the Euclidean distance.

6.2 Suggestions for Future Works

Several suggestions for future research works are as follows.

1. In the applications of digital museums, the copyright and annotation data of other kinds of digital medias (such as video, audio and other image formats) need to be protected. As a result, how to extend the methods proposed in this study to other kinds of digital data are worth for further research.
2. How to embed data with better image quality?
3. How to authenticate an image without extra data?
4. How to use the different characteristics of the three channels of the color image to embed more bits of data?

References

[1] E. H. Adelson, "Digital signal encoding and decoding apparatus," U.S. Patent 4939515,1990.

[2] M. S. Liaw and L. H. Chen, "An effective data hiding method," in *Proc. IPPR Conf. on Computer Vision, Graphics, and Image Processing*, Taiwan, R.O.C., 1997, pp.146-153.

[3] T. S. Chen, C. C. Chang, and M. S. Hwang, "A virtual image cryptosystem based upon vector quantization," *IEEE Transactions on Image Processing*, vol. 7, no. 10, pp. 1485-1488, 1998.

[4] Da-Chun Wu and Wen-Hsiang Tsai, "Embedding of Any Type of Data in Images Based on A Human Visual Model And Multiple-Based Number Conversion," accepted and to appear in *Pattern Recognition Letters*.

[5] T. K. Yen, "Image hiding by random bit replacement and frequency transformations," *Master report, Department of Computer and Information Science, National Chiao Tung University, Taiwan, Republic of China*, 1998.

[6] H. Y. Chang, "Data hiding and watermarking in color images by wavelet transforms," *Master report, Department of Computer and Information Science, National Chiao Tung University, Taiwan, Republic of China*, 1999.

[7] D. Stinson, Cryptography Theory and Practice, CRC Press, Boca Raton, 1995.

[8] S. Walton, "Information authentication for a slippery new age," Dr. Dobbs Journal, vol. 20, no. 4, pp. 18-26, April 1995.

[9] R. van Schyndel, A. Tirkel, and C. Osborne, "A Digital Watermark," *Proceeding of the IEEE International Conference on Image Processing*, vol 2, pp. 86-90, Austion, Texas, November 1994.

[10] P. W. Wong, "A public key watermark for image verification and authentication,"

in *Proc. IEEE International Conference on Image Processing*, vol. II, 1998, pp. 455-459.

[11] D. C. Wu and W. H. Tsai, "A Method for Creating Perceptually Based Fragile Watermarks for Digital Image Verification," submitted to *IEEE Transactions On Multimedia.*

[12] J. Fridrich, "Image watermarking for tamper detection," in *Proc. IEEE International Conference on Image Processing*, vol. II, 1998, pp. 404-408.

[13] M. Wu and B. Liu, "Watermarking for image authentication," *in Proc. IEEE International Conference on Image Processing*, vol. II, pp. 437-441, Chicago, Illinois, October 1998.

[14] D. Kundur and D. Hanzinakos, "Towards a telltale watermarking technique for tamper-proofing," *in Proc. IEEE International Conference on Image Processing*, vol. 2, pp. 409-413, Chicago, Illinois, October 1998.

[15] G. Voyatzis, I. Pitas, Applications of toral automorphisms in image watermarking. Proc. IEEE Internet. Conf. on Image Processing (ICIP'96), Vol. II, Lausanne, Switzerland, 16-19 September 1996, pp. 237-240.

[16] J. Fridrich, "Robust bit extraction form images," *in Proc. IEEE ICMCS'99 Conf.* Florence, Italy, June 7-11, 1999.

[17] W. Bender, N. Morimoto, and D. Gruhl, "Method and apparatus for data hiding in images," U. S. Patent, No. 5689587, 1997.

[18] E. Koch and J. Zhao, "Towards robust and hidden image copyright labeling," *in Proc. IEEE Nonlinear Signal and Image Processing Workshop*, Thessaloniki, Greece, 1995, pp. 452-455.

[19] C. T. Hsu and J. L. Wu, "DCT-Based watermarking for video," *IEEE Transactions on Image Processing*, vol. 8, pp. 58-68, 1999.

[20] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT-domain system for

robust image watermarking," *Signal Processing*, vol. 66, pp. 357-372, 1998.

[21] I. J. Cox, J. Kilian, F. T. Leighton and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, pp. 1673-1687, 1997.

[22] C. H. Kuo and C. F. Chen, "A prequantizer with the human visual effect for the DPCM," *Signal Processing: Image Communication*, vol. 8, pp. 433-442, 1996.

[23] Stefan Katzenbeisser and Fabien A. P. Petitcolas, "Information hiding techniques for stagonography and digital watermarking," Artech House, Boston London, 2000.

[24] G. K. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Transactions on Consumer Electronics*, Vol. 38, No. 1, February 1992.