國立交通大學

資訊科學系

碩 士 論 文

利用數位浮水印與影像驗證作影像版權保護及竄改偵測之研究

A Study on Digital Watermarking and Authentication of Images for Copyright Protection And Tampering Detection

研 究 生：邱彥中

指導教授：蔡文祥　教授

中 華 民 國　九 十 三　年 六 月

利用數位浮水印與影像驗證作影像版權保護及竄改偵測之研究
A Study on Digital Watermarking and Authentication of Images for Copyright Protection And Tampering Detection

研 究 生：邱彥中　　Student：Yen-Chung Chiu

指導教授：蔡文祥　　Advisor：Wen-Hsiang Tsai

國 立 交 通 大 學

資 訊 科 學 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer and Information Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

利用數位浮水印與影像驗證作影像版權保護及竄改偵測之研究

研究生：邱彥中　　指導教授：蔡文祥 博士

國立交通大學資訊科學研究所

摘要

由於數位科技的進步，數位影像可能會被非法複製，甚至是竄改。因此，發展數位影像版權保護的方法是很重要的課題。在本論文中，針對這個目的，我們提出了數種數位浮水印的技術。數位浮水印必須具有強韌性來抵抗非法使用者對影像的各種操作。針對彩色影像，我們提出了四種方法分別來抵抗不同的攻擊。首先，利用在離散富利葉轉換頻域上產生環狀和對稱的峰值來抵抗旋轉和縮放的攻擊。第二，應用離散富利葉轉換和離散餘弦轉換兩個頻域來抵抗旋轉和裁切的攻擊。第三，我們提出了一種在離散富利葉轉換頻域上，利用一個組合函數來對峰值位置作編碼，此方法可用來抵抗影像列印和再掃描成數位影像的操作。第四，是利用一種重新縮放影像的技術來抵抗縮放和行列移除的攻擊。最後，我們也提出一種彩色影像驗證的方法，來證明影像的真實性和完整性。藉由檢查藏入影像中的驗證信號，我們可以指出影像被竄改之處。利用藏入驗證信號來作影像驗證的方法，可以不用另外存有一個影像特徵資料。相關的實驗結果證明所提方法是可行性的。

A Study on Digital Watermarking and Authentication of Images for Copyright Protection And Tampering Detection

Student: Yen-Chung Chiu Advisor: Dr. Wen-Hsiang Tsai

Institute of Computer and Information Science
National Chiao Tung University

ABSTRACT

Due to the advance of digital techniques, digital images may be copied or tampered with illegally. Therefore, it is important to develop methods to protect the copyright of digital images and verify their integrity. In this study, digital watermarking techniques for such purposes are proposed. In addition, the watermark must have an ability of robustness because users may apply various operations on a watermarked image. For color images, four methods are proposed to handle different attacks. First, a method based on creating peaks circularly and symmetrically in the DFT domain is proposed to resist rotation and scaling attacks. Second, a watermarking method utilizing the DFT and DCT domains is proposed to survive rotation and cropping attacks. Third, a method based on coding peaks by a combinatorial function in the DFT domain is proposed to work against print-and-scan operations. Fourth, a method based on an image rescaling technique is proposed to resist scaling and line-removal attacks. Finally, an authentication method for verifying the fidelity and integrity of color images is also proposed, which uses a key to generate authentication signals randomly. By checking authentication signals in an image, tampered parts can be pointed out. Such authentication signals can be used to conduct the authentication work without using other signature data. Good experimental results prove the feasibility of the proposed methods.

ACKNOWLEDGEMENTS

The author is in hearty appreciation of the continuous guidance, discussions, support, and encouragement received from his advisor, Dr. Wen-Hsiang Tsai, not only in the development of this thesis, but also in every aspect of his personal growth.

Thanks are due to Mr. Chih-Hsuan Tzeng, Mr. Chang-Chou Lin, Mr. Chih-Jen Wu, Mr. Tsung-Yuan Liu, Mr. Wei-Liang Lin, Mr. Yi-Chieh Chen, Mr. Kuei-Li Huang, Mr. Cheng-Jyun Lai, Miss Yen-Lin Chen, and Mr. Nan-Kun Lo for their valuable discussions, suggestions, and encouragement. Appreciation is also given to the colleagues of the Computer Vision Laboratory in the Department of Computer and Information Science at National Chiao Tung University for their suggestions and help during his thesis study.

Finally, the author also extends his profound thanks to his family for their lasting love, care, and encouragement. He dedicates this dissertation to his parents.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Motivation

Because of the rapid development of the Internet, aplenty digital multimedia are spread fast and widely on the Internet, such as digital images, videos, audios, texts, and so on. It is convenient to get and exchange digital multimedia through the Internet. Furthermore, people can easily use application programs to edit and copy these digital data. As a consequence, many unauthorized uses and illegal tampering activities appear in today's digital world. Therefore, it is important to develop methods to protect the copyright of digital multimedia and verify their integrity. In this study, we will focus on copyright protection and authentication of digital images (or simply, images).

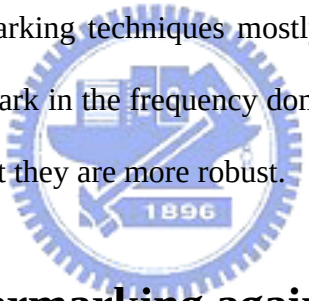
Many researches have been conducted to achieve the goal of image copyright protection and authentication. Digital watermarking is one of the ways to solve the problems of infringement acts on digital images. It has recently become a very active area of researches. For image copyright protection, a digital watermark is embedded into an image and the result is imperceptible under normal observations. Later, the watermark is extracted to prove the image copyright. For image authentication, authentication signals are embedded into images and detected later to decide whether the images are tampered. By theses ways, we can verify the integrity and fidelity of images.

After a watermark is embedded into an image, the result is called a *stego-image*.

Robustness of stego-images is desirable. That means that, even if a stego-image is suffered from lossy image compression, like JPEG compression, or standard image processing operations, such as rotation, scaling and cropping, the watermark can still be extracted.

1.2 Review of Related Works

Many different watermarking techniques for copyright protection have been proposed in recent years. These watermarking techniques can be classified into three approaches. One is the spatial-domain approach [1-6]. The second is the frequency-domain approach [7-15]. And the third is a combination of the first two [16, 17]. Earlier proposed watermarking techniques mostly belong to the first approach. Methods to embed the watermark in the frequency domain are more complicated than those in the spatial domain, but they are more robust.



1.2.1 Digital Watermarking against Geometric Attacks

Watermarking techniques that are robust to common geometric transformations of rotation, scaling, and translation (RST) are mostly performed in the frequency domain. O'Ruanaidh and Pun [11] proposed the use of Fourier-Mellin transform-based invariants for digital image watermarking. A public watermarking method based on the Fourier-Mellin transform and an extension of it based on the Radon transform was proposed by Wu, et al. [12]. In Lin, et al. [13] a watermark is embedded into a one-dimensional (1-D) signal obtained by taking the Fourier transform of the image, re-sampling the Fourier magnitudes into log-polar coordinates, and then summing a

function of those magnitudes along the log-radius axis. Su and Kuo [16] proposed a spatial-frequency composite digital image watermarking scheme to make the embedded watermark survive generalized geometrical transformations. The frequency-domain watermark was embedded in the discrete Fourier transform (DFT) coefficients. The spatial-domain watermarking is used to help recover the image to its original orientation and scale.

1.2.2 Digital Watermarking against Print-And-Scan Attacks

Some researches about watermarking techniques for copyright protection against print and scan attacks have been proposed in recent years. A print and scan attack means destruction of the embedded watermark after a digital stego-image is printed and rescanned into another digital version. Fleet and Heeger [14] described a model of human color vision to ensure that the embedded signal is invisible and proposed a method for embedding sinusoidal signals, which act as a grid and provide a coordinate frame on the image. In Solachidis and Pitas [15], a private key, which allows a very large number of watermarks, determined the watermark, which was embedded on a ring in the DFT domain. And the measure of correlation was used for watermark detection. Lefebvre et al. [17] proposed a method, which combined an additive watermarking algorithm in the spatial domain and a synchronization template in the Fourier domain. In the method proposed by Chotikakamthorn and Pholsomboon [6], a watermark constructed with a ring-shaped constraint was embedded in the spatial domain and a sinusoidal function with random phases was used for generating each watermark ring.

In fact, these methods for digital image watermarking, which are resistant to

print-and-scan attacks, are also resistant to geometric transformations.

1.2.3 Digital Image Authentication

Some researches about image authentication have been proposed. Yeung and Mintzer [18] embedded a binary image, taken as authentication signals, into images. In Fridrich [19], an image was divided into 64×64 blocks, and a watermark value was inserted into each block by modulating the middle thirty percent of the DCT coefficients of each block. Wong [20] divided an image into blocks and used the LSB plane of each block for embedding watermark information. In Wu and Tsai [21], a human visual model was used to embed perception-based fragile watermarks. Wu and Liu [22] embedded a watermark in an image by changing quantized DCT coefficients before entropy coding. In Yin and Tsai [5], an image was partitioned into 8×8 blocks, and the DC values of the DCT coefficients were saved as a signature file.

1.3 Overview of Proposed Methods

1.3.1 Definitions of Terms

Before describing the proposed methods, some definitions of terms used in this study are introduced as follows.

1. *Authentication signal*: An authentication signal is a fragile signal embedded into an image such that any alteration to the watermarked image can be detected.
2. *Cover image*: A cover image is an image into which a watermark is embedded

3. *Stego-image*: A stego-image is an image that is produced by embedding a watermark into a cover image.
4. *Authentication image*: An authentication image is an image that is obtained by checking the embedded authentication signals.
5. *Embedding process*: An embedding process is a process to embed data in an image.
6. *Extraction process*: An extraction process is a process to extract data from an image.
7. *Authentication process*: An authentication process is a process to detect whether a stego-image is tampered with or not.

1.3.2 Brief Descriptions of Proposed Methods

In this study, we focus on dealing with full-color images. And for them, different watermarking methods will be proposed to handle different attacks.

A. Copyright Protection by Watermarking for Color Images against Rotation and Scaling Attacks Using Coding and Synchronization of Peak Locations in DFT Domain

A method for copyright protection by watermarking for color images against rotation and scaling attacks using coding and synchronization of peak locations in the DFT domain is proposed, in which a watermark is embedded in a color image to create a stego-image. First, a serial number is embedded as a watermark into a cover image by adjusting the magnitude values of the coefficients in the middle band of the DFT domain to create *peaks* and coding their locations in certain concentric circles in the band as watermark signals. A peak for synchronization is also embedded in the

middle band. The embedded watermark can be extracted from the stego-image by coefficient-value peak detection in the DFT domain. By this method, the embedded watermark becomes robust to survive rotation and scaling attacks. Figure 1.1 shows a flowchart of the proposed embedding method.

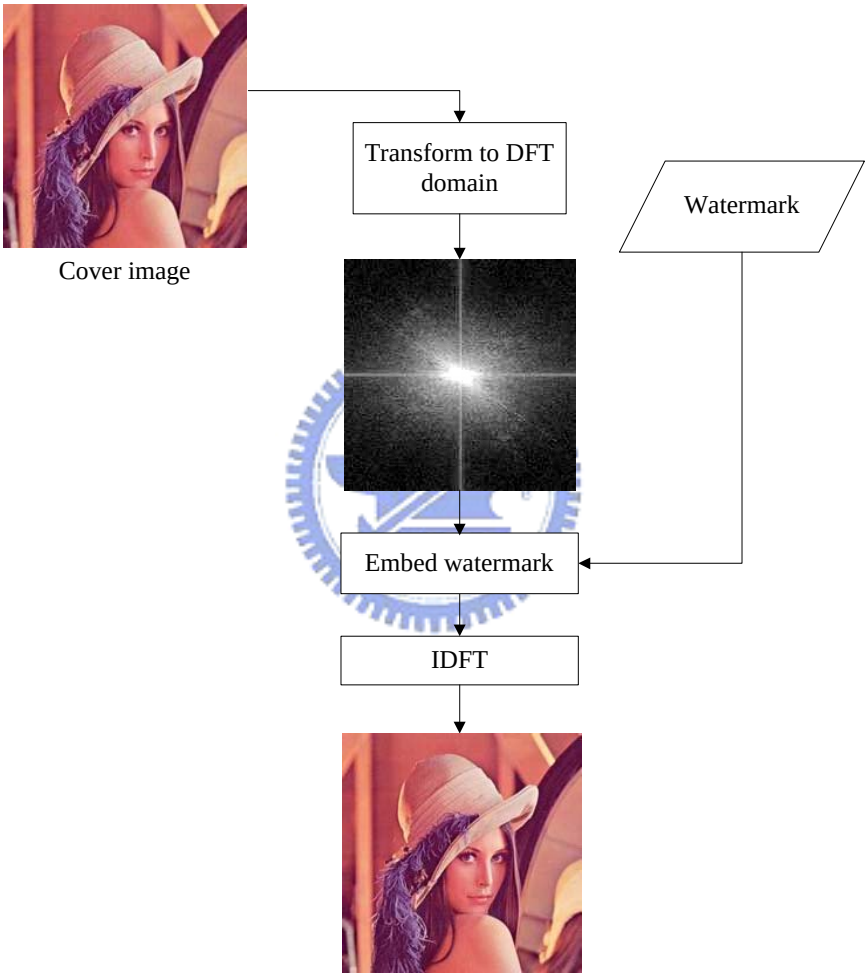


Figure 1.1 Flowchart of first proposed method of watermarking for copyright protection.

B. Copyright Protection by Watermarking for Color Images against Rotation And Cropping Attacks Using Synchronization of Peak Locations in DFT Domain And Coefficient Relationship Comparison in DCT Domain

A method for copyright protection by watermarking for color images against rotation and cropping attacks using synchronization of peak locations in the DFT domain and DCT-coefficient relationship comparison is proposed. A synchronization peak is embedded into a cover image in the DFT domain and a binary image as a watermark is multiply duplicated and embedded into the 8×8 blocks of the cover image in the DCT domain by adjusting the magnitude relation of certain selected DCT coefficients. The synchronization peak is detected later to check whether the stego-image has been rotated or not. Finally, the embedded watermark can be extracted from the DCT domain by checking the relationship of the DCT coefficients. Using this method, the embedded watermark becomes resistant to rotation and cropping attacks. Figure 1.2 shows a flowchart of the embedding method.

C. Copyright Protection by Watermarking for Color Images against Print-and-Scan Operations Using Coding and Synchronization of Peak Locations in DFT Domain

A method for copyright protection by watermarking for color images against print and scan operations using coding and synchronization of peak locations in the DFT domain is proposed. A rescanned image always has pixel-value distortion and geometric transformations, like scaling, slight rotation, and a little zero padding. The pixel-value distortion means that the color and luminance of the image pixels have been altered.

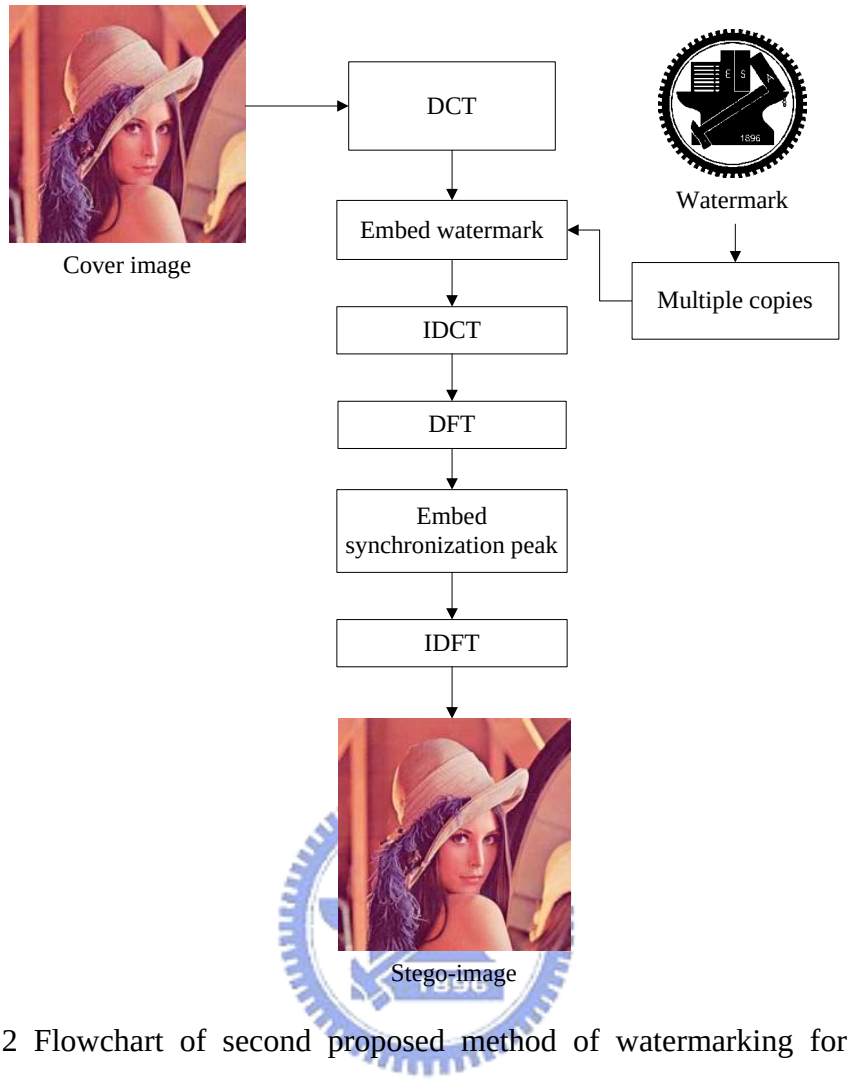


Figure 1.2 Flowchart of second proposed method of watermarking for copyright protection.

Therefore, a watermark embedded in a rescanned image must be provided with robustness against pixel-value distortion and geometric operation attacks. In the proposed method, watermark signals are embedded in certain concentric circles in the DFT domain by adding coefficient-value peaks in the middle band and using a combinatorial function to code the peak locations. In the extraction process, the positions of the coefficient-value peaks are detected and mapped into a combinatorial function to get a watermark signal. In addition, the synchronization peak is detected, too. By the proposed method, the watermark can survive print and scan operations. Figure 1.3 shows a flowchart of the proposed method.

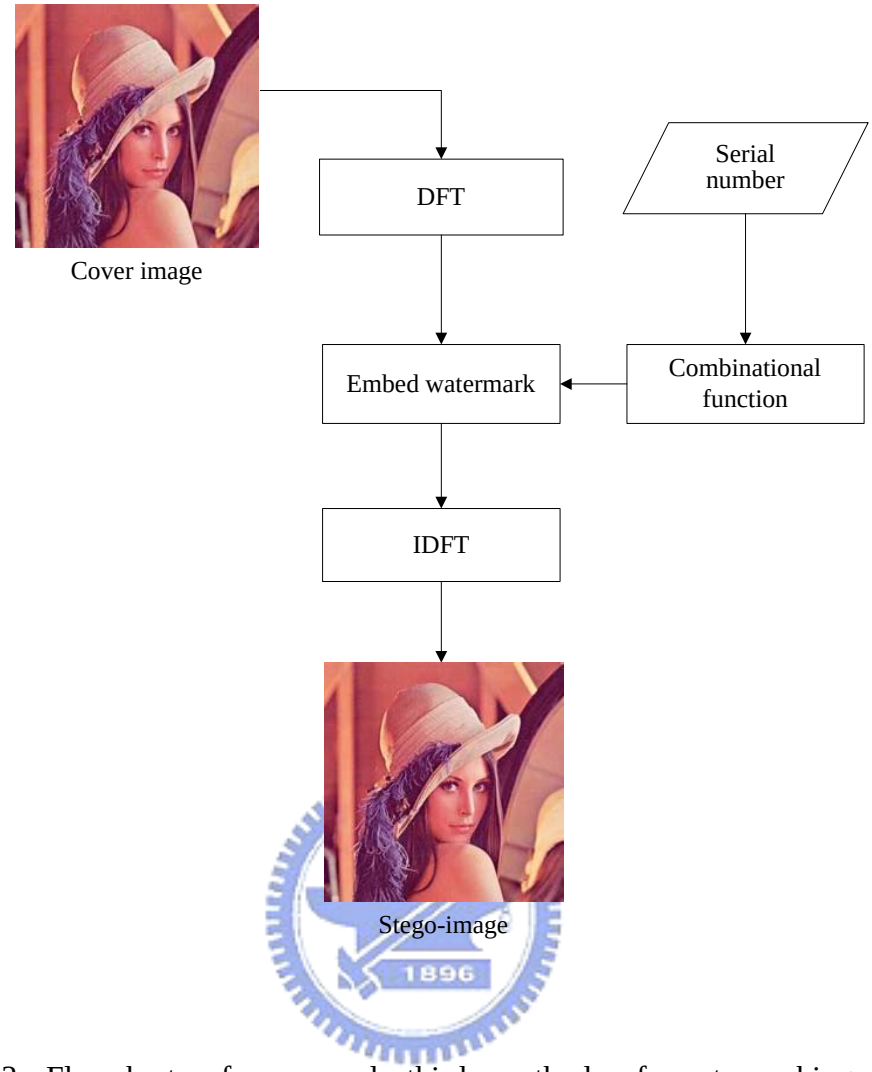


Figure 1.3 Flowchart of proposed third method of watermarking against print-and-scan operations.

D. Copyright Protection by Watermarking for Color Images against Scaling And Line-Removal Attacks Using An Image Rescaling Technique

In this topic, we focus on watermarking for color images against scaling and line-removal attacks. The proposed methods are based on those proposed by Cheng and Tsai [23] and Yin and Tsai [5], which embed invisible watermarks into images in the DCT domain. We improve it by rescaling an image to a pre-determined normal size before the embedding and extraction processes, and inserting certain verification

codes to check whether a watermark is embedded in the extraction processes. Figure 1.4 shows a flowchart of the proposed method for this topic.

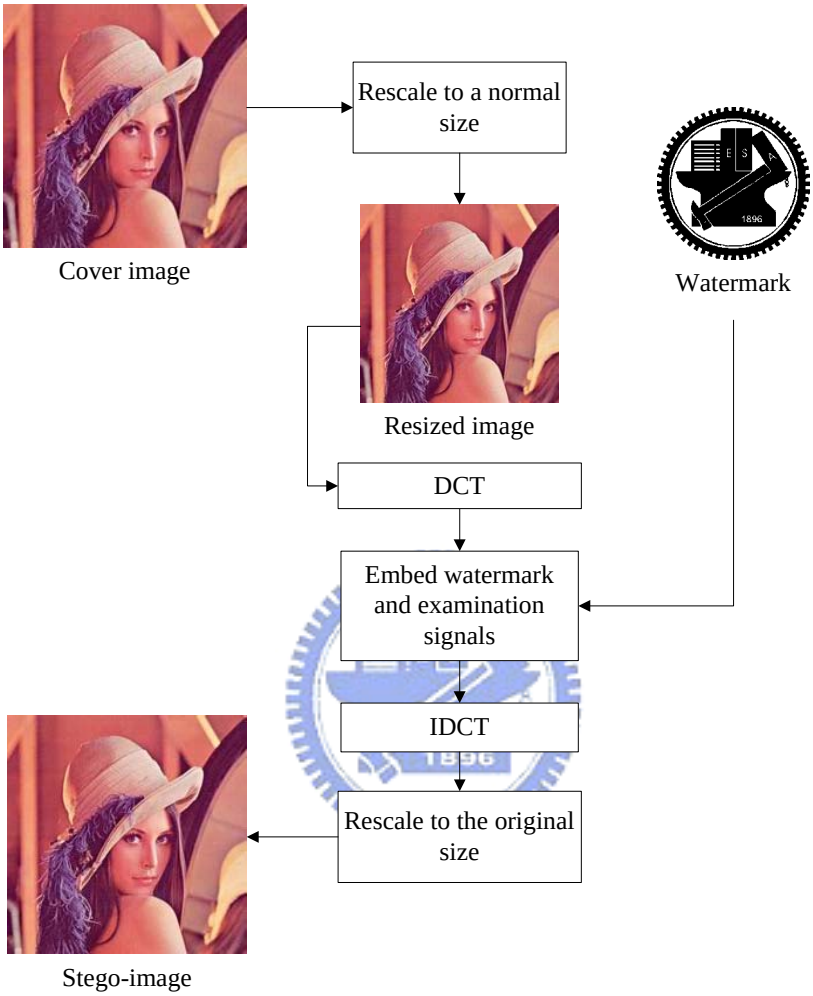


Figure 1.4 Flowchart of fourth proposed method of watermarking against scaling and line removal.

E. Tampering Detection in Color Images by Signature-Free Authentication Using DCT-Coefficient Relationship Comparison

A method for image authentication in color images without the use of signatures

is proposed. Authentication signals, randomly generated by a key and some information of an image, are embedded into each 8×8 image block in the DCT domain by adjusting the magnitude relationship of the coefficients. There is no need to save an extra signature file, which contains image features, when authenticating a suspicious image later. The signature file is a waste of storage space and makes the file management more complicated. By checking the relationships of the selected DCT coefficients of the image block, the fidelity and integrity of the image can be verified. Figure 1.5 shows a flowchart of the proposed method.

1.4 Thesis Organization

In the remainder of this thesis, the proposed method for copyright protection by watermarking for color images against rotation and scaling attacks is described in Chapter 2. In Chapter 3, the proposed method for copyright protection by watermarking for color images against rotation and cropping attacks is described. And in Chapter 4, the proposed method for copyright protection by watermarking for color images against print-and-scan operations is described. In Chapter 5, the proposed method for copyright protection by watermarking for color images against scaling and line-removal attacks using an image rescaling technique is described. In Chapter 6, the proposed method for tampering detection in color images by signature-free authentication via DCT-coefficient relationship comparison is described. Finally, in Chapter 7, we give some conclusions and briefly point out some possible directions for future research works.

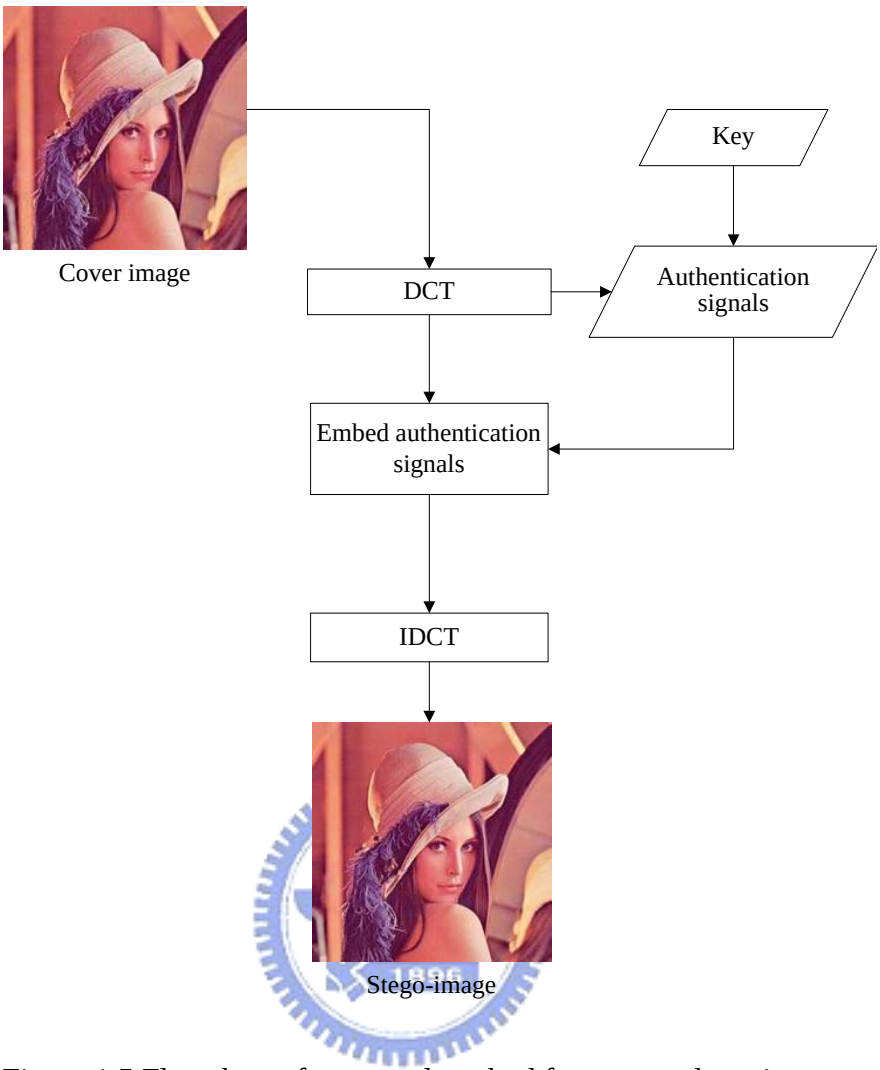


Figure 1.5 Flowchart of proposed method for temper detection.

Chapter 2
Copyright Protection by Watermarking for Color Images against Rotation and Scaling Attacks Using Peak Detection and Synchronization in DFT Domain

The proposed method for copyright protection of color images against rotation and scaling attacks is described in this chapter. The main idea is to embed a watermark as coefficient-value peaks in the DFT domain of an input image. Then, by detecting the peaks in the DFT domain, the embedded watermark can be extracted.

2.1 Introduction

Digital watermarking is a technique for embedding a watermark into an image to protect the owner's copyright of the image. The embedded watermark must be robust. The stego-image may be rotated or scaled by illicit users. It is desirable that after applying these operations on the stego-image, the watermark is not fully destroyed and can be extracted to verify the copyright of the image.

The remainder of this chapter is organized as follows. In Section 2.2, the idea of the proposed method will be described. By certain properties of the DFT coefficients, we can embed a watermark in the DFT domain with robustness against rotation and

scaling attacks. In Section 2.3, the proposed watermark embedding process is presented. In Section 2.4, the proposed watermark extraction process is described. In Section 2.5, some experimental results are illustrated. Finally, in Section 2.6 some discussions and a summary are given.

2.2 Idea of Proposed Method

2.2.1 Properties of Coefficients in DFT Domain

After applying a discrete Fourier transformation (DFT) to an input image, the DFT coefficients in the frequency domain can be obtained. The DFT of an image $f(x,y)$ of size $M \times N$ can be described by the equation described below:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M+vy/N)} . \quad (2.1)$$

The Fourier transform is a complex function of the real frequency variables. It has several properties, and some of them are described in the following.

A. Symmetry property

If a 2D signal is real, then the Fourier transform has a symmetry property, as shown by the following equation [27]:

$$F(u,v) = F^*(-u,-v) . \quad (2.2)$$

The symbol (*) indicates complex conjugation. Because the Fourier transform of an image can be complex, we can divide it into two functions. One is the *magnitude* function or spectrum $|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{\frac{1}{2}}$, and the other the *phase* function $\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$, where $R(u,v)$ and $I(u,v)$ are the real and

imaginary parts of $F(u,v)$. And for real signals, Equation (2.2) leads to:

$$|F(u,v)| = |F(-u,-v)|. \tag{2.3}$$

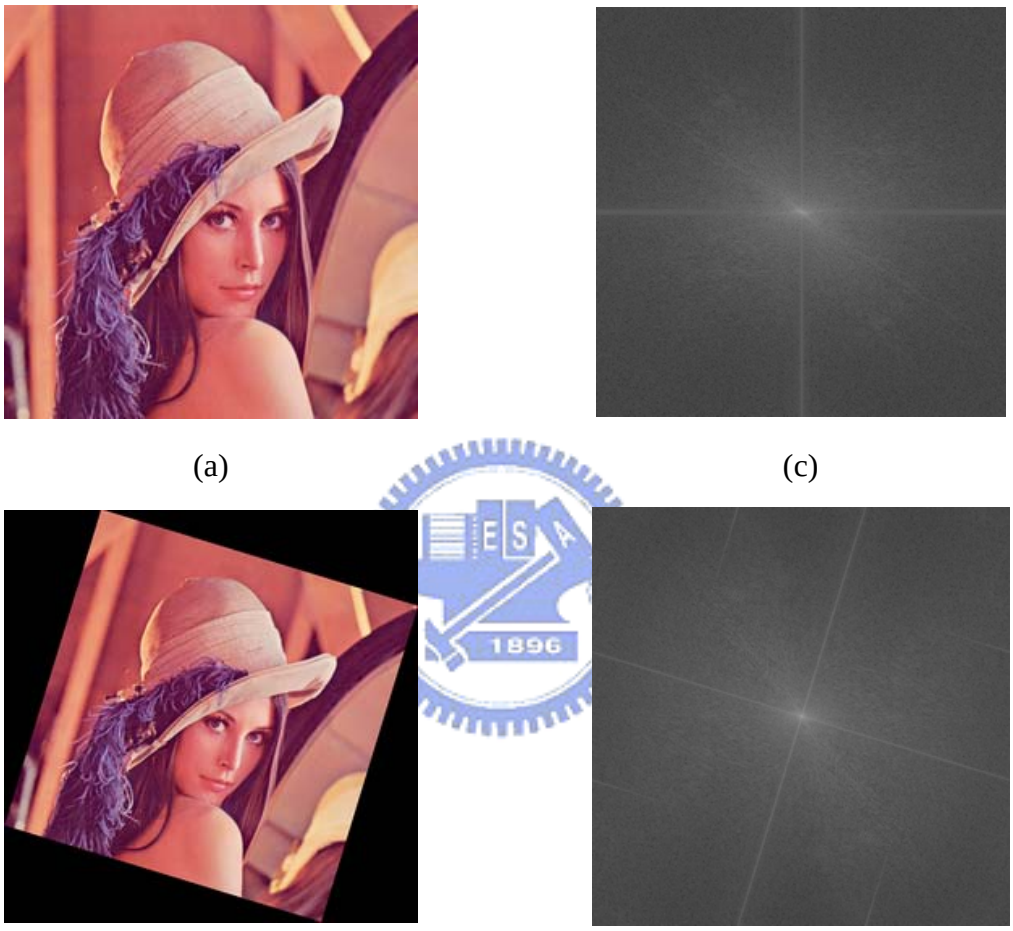
It means that the magnitude value of a coefficient (or simply a coefficient value) and its symmetric version are equal. In addition, both the magnitude and the phase functions are necessary for complete reconstruction of an image from its Fourier transform. But the magnitude part is less important than the phase part. The magnitude-only image is unrecognizable. On the contrary, the phase-only image is barely recognizable [24]. Therefore, we may calculate and adjust the magnitude values of the DFT coefficients to embed information without causing significant loss of image quality.

B. Invariant properties of rotation and scaling

After we apply some image processing operations like rotation and scaling to an image, the coordinates and magnitude values of the DFT coefficients of the image will be altered, too. Changes of the DFT coefficients after scaling and rotation operations in the discrete image domain are listed in Table 2.1 [28]. The scaling operation has almost no effect on the DFT coefficients. It means that when an image is scaled, each DFT coefficient is the same as the original one except only with some noise. On the other hand, after rotating an image in the spatial domain, the locations of the DFT coefficient values will have the same rotation in the DFT domain. Figures 2.1(a) and (b) show an original image and a rotated version of it. And the corresponding Fourier spectrum images, in which each pixel value is equal to the magnitude value of the DFT coefficient, are shown in Figures 2.1(c) and (d), respectively. Note that the Fourier spectrum image in Figure 2.1(d) has the same rotation like Figure 2.1(b).

Table 2.1 Changes of DFT coefficients after operations in discrete spatial domain.

Operations	**Scaling**	**Rotation**
Changes of DFT coefficients	Almost no effect	Rotation



(a) (c)

(b) (d)

Figure 2.1 Input images, and Fourier spectrum images of G channel. (a) Image "Lena". (b) Image "Lena" after rotation. (c) Fourier spectrum image of image "Lena" (d) Fourier spectrum image with the same rotation angle of (b).

2.2.2 Properties of Color Channels

A full-color image has three color channels, namely, red (R), green (G), and blue (B). Generally speaking, we can embed watermark information into all of these three

channels. However, human eyes are less sensitive to the frequency of blue color. And its greatest sensitivity is distributed over the region of the yellow/green frequency [25]. In addition, according to experiments, a watermark can be embedded into both red and blue channels in the DFT domain without creating perceivable effects. On the contrary, hiding information in the green channel is too sensitive to human vision. If we embed the watermark in the DFT domain of the green channel, the stego-image will appear to include obvious reticular effects.

2.2.3 Proposed Technique of Using DFT Peaks for Watermarking

In the proposed watermarking method, after the zero frequency point $F(0,0)$ is shifted to the center of the DFT domain, a watermark is embedded in a ring region which covers a middle band, denoted as B subsequently, in the frequency domain between two circles with two pre-selected radii R_1 and R_2 where $R_1 < R_2$, as shown in Fig. 2.2. The middle band of the frequency domain is divided into n equally-spaced concentric circles with radii $r_1, r_2, \cdots, r_n$, and into m angle ranges with starting angles $\theta_1, \theta_2, \cdots, \theta_m$, as seen in Fig. 2.3. Then, *$n \times m$ embeddable positions* $p_1, p_2, \cdots, p_{n \times m}$ are selected in this study to be located at (u_k, v_k) in the frequency domain described by:

$$p_k = (u_k, v_k) = (r_i \cos\theta_j, r_i \sin\theta_j), \tag{2.4}$$

where $1 \le i \le n$, $1 \le j \le m$ and $1 \le k \le n \times m$, and at each embeddable position p_k, the coefficient value is adjusted to be a *local peak* in the frequency domain.

More specifically, let W be a watermark to be embedded, which is taken to be a serial number in this study in the form of a bit stream, and let $M(u_k, v_k)$ be the DFT coefficient value at an embeddable position $p_k = (u_k, v_k)$. Then, we embed a *watermark*

bit w_i at p_k in the frequency domain in this study by modifying $M(u_k, v_k)$ to be *a local peak* $M'(u_k, v_k)$ by the following equation:

$$M'(u_k, v_k) = M(u_k, v_k) + c \times w_i \tag{2.5}$$

where c is a pre-selected parameter that determines the strength of the embedded watermark signal.

It is noted that, when conducting watermarking in the above way of changing the DFT coefficient value at an embeddable position $p_k = (u_k, v_k)$ for the amount of $\delta = c \times w_i$, we must preserve the *positive symmetry* property of the DFT [26] by changing the corresponding coefficient value at $p_k' = (-u_k, -v_k)$ for the same amount δ. Otherwise, the peak created at p_k will be counteracted by the symmetric coefficient value at p_k' after applying the inverse DFT. That is, we must perform, as is done in this study, the following operation

$$M'(-u_k, -v_k) = M(-u_k, -v_k) + \delta \tag{2.6}$$

in addition to Eq.(2.5) each time we embed a watermark bit w_i at an embeddable position $p_k = (u_k, v_k)$.

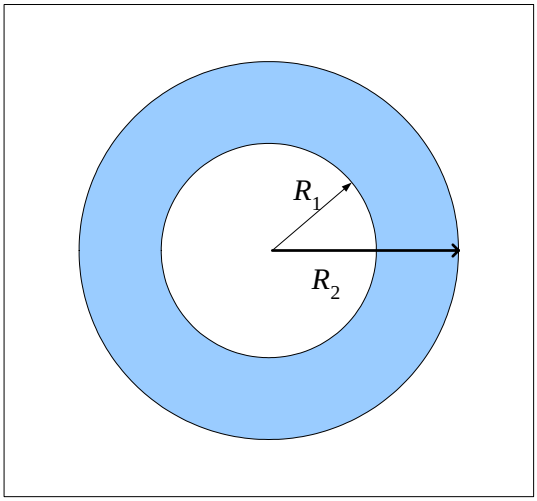


Figure 2.2 A ring region of middle frequency band.

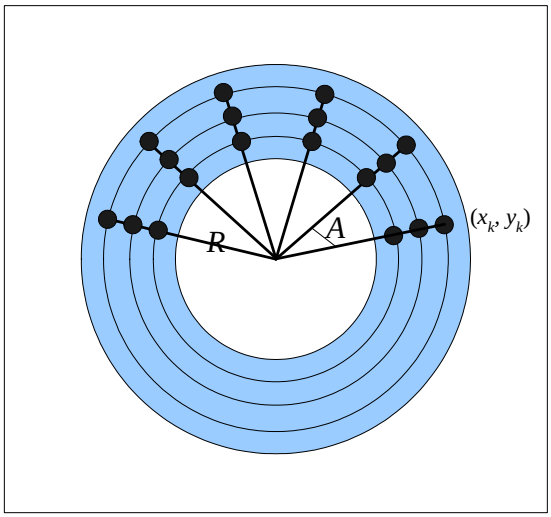


Figure 2.3 The ring region divided into concentric circles and into angular sectors.

2.2.4 Proposed Technique for Synchronizing Peak Locations for Protection against Rotation and Scaling Attacks

In order to deal with rotation and scaling attacks, an extra local peak P_{sync}, called *synchronization peak*, is created in the DFT domain to serve as a signal for *synchronizing* the peak locations p_1, p_2, …, $p_{n\times m}$ mentioned previously in a way described later. P_{sync} is embedded into the previously-mentioned middle frequency band B at a location p_{sync} described by:

$$p_{sync} = (u_{sync}, v_{sync})$$

$$= (r_{sync}\cos\theta_{sync}, r_{sync}\sin\theta_{sync}) \quad (2.7)$$

where r_{sync} is selected to be larger than R_2 and θ_{sync} is a pre-selected angle value. We adjust the DCT value M of P_{syn} to be a peak value $M' = M + c$ where c is the constant value mentioned previously.

We now describe how we use the synchronization peak P_{sync} in the proposed watermark extraction process to calculate the rotation angle of a tampered stego-image which suffered possibly from a rotation attack. Because of the DFT properties mentioned previously and illustrated by Fig. 1, if a stego-image is rotated, the location of P_{sync} will also be changed with the same rotation angle. We may calculate first the new angle θ'_{sync} of P_{sync} and take the difference $\Delta\theta$ between θ'_{sync} and θ_{sync} to decide whether the stego-image has been rotated: if $\Delta\theta \neq 0$, then rotated; else, not. If rotated, then we find the angles θ'_k of the remaining local peaks, and compute their original angles θ''_k by

$$\theta''_k = \theta'_k - \Delta\theta. \tag{2.8}$$

On the other hand, as mentioned previously, if a stego-image is rescaled, the DFT coefficient values are almost unaffected. It means that the radii of the local peaks will not be changed.

2.3 Watermark Embedding Process

As mentioned previously, a watermark used for image ownership protection is assumed to be a serial number in this study, and the watermark is transformed into a watermark bit stream. In this section, the process of embedding a watermark bit stream in a color image will be described.

2.3.1 Embedding of Watermarks

In the proposed watermark embedding process, we use the two channels of red and blue to embed a watermark bit stream in the DFT domain according to the idea described in Section 2.2.2. And the middle band area of the Fourier spectrum is

divided into several concentric circles. Then, the watermark bit stream is embedded in the region of the concentric circles.

Furthermore, the watermark bit stream is divided into two halves to be embedded in the red and blue color channels, respectively. For either channel, the spatial domain is transformed into the frequency domain by the DFT. In the middle band of the DFT domain, locations that can be use to create peaks are decided according to the scheme described in Section 2.2.3. Then, we can get pairs of locations (u_k, v_k) and $(-u_k, -v_k)$. Using the watermark bit stream W, if a bit w_k of W equals "1," coefficient values of the corresponding embeddable positions (u_k, v_k) and $(-u_k, -v_k)$ are adjusted to be peaks by Eqs. (2.5) and (2.6) to embed a watermark bit. On the contrary, if w_k equals "0", the corresponding coefficient values are not changed. In addition, a synchronization peak is also embedded into the middle frequencies according to the scheme described in Section 2.2.4.

2.3.2 Detailed Algorithm

The inputs to the proposed watermark embedding process are a color image C and a watermark W. The output is a stego-image S. The process can be briefly expressed as an algorithm as follows. Figure 2.4 shows a flowchart of the process.

Algorithm 1: *Watermark embedding process*.

Input: A given color image C and a watermark W.

Output: A stego-image S.

Steps.

1. Transform the red and blue channels of C into the frequency domain by the DFT to get C'_{red} and C'_{blue}.
2. Divide W into two parts $W_{\text{red}} = w_1 w_2 \cdots w_\ell$ and $W_{\text{blue}} = w_{\ell+1} w_{\ell+2} \cdots w_{2\ell}$.

3. Embed W_{red} and W_{blue} into C'_{red} and C'_{blue}, respectively, by performing the following operations.

   3.1 Decide n radiuses R = $\{r_1, r_2, \cdots, r_n\}$ of equally-spaced concentric circles in the middle band between two circles with radiuses R_1 and R_2, with $R_1 < R_2$.

   3.2 Decide m angles $\Theta = \{\theta_1, \theta_2, \cdots, \theta_m\}$ equally distributed in the range from $0°$ to $180°$. Also, take ℓ to be $m \times n$.

   3.3 Obtain ℓ positions P = $\{p_1, p_2, \cdots, p_\ell\}$ with p_k (k = 1, 2, …, ℓ) located at $(r_i \cos\theta_j, r_i \sin\theta_j)$ with $k = (i - 1) \times m + j$, and their ℓ symmetric positions Q = $\{q_1, q_2, \cdots, q_\ell\}$ with q_k located at the symmetric location of p_k, where $1 \le i \le n$, and $1 \le j \le m$.

   3.4 If watermark bit w_k equals 1, then adjust the pair of the coefficient values located at p_k and q_k to be local peaks by Eqs. (2.5) and (2.6), where $1 \le k \le \ell$ for C'_{red} or $\ell + 1 \le k \le 2\ell$ for C'_{blue}.

   3.5 Add a synchronization peak P_{sync} according to the scheme described in Section 2.2.4.

4. Transform the C'_{red} and C'_{blue} back into the spacial domain by the inverse DFT.

5. Take the final result as the desired stego-image S.

Color image

Get R and G channels

Decide peak value K

DFT

Divide into concentric circles

A symmetric pair of coefficients (p_i, q_i)

Watermark

Transform into binary form

A bit w_i

Embed the watermark, $w_i = 1$?

Yes

Adjust coefficients to be peaks

No

Embed a synchronization peak

IDFT

Combine R, G, and B

Stego-image

Figure 2.4 Flowchart of the embedding process.

2.4 Watermark Extraction Process

In the proposed watermark extraction process, no other information but the stego-image is needed as the input. The watermark can be extracted to verify the copyright. The processes of applying this technique will be described in this section. And a detailed algorithm for the process will be given.

2.4.1 Extraction of Watermarks

In the proposed watermark extraction process, the red and blue channels of a stego-image are accessed. Each of these two channels is transformed into the DFT domain. Then, the local peaks in the middle frequency band of the DFT domain are detected using a pre-selected threshold value T: if any DFT coefficient value M is larger than T, it is judged to be a local peak. Because of the symmetry property of the DFT coefficient values specified in Eq. (2.3), we may only detect peaks within the range of the upper-half Fourier spectrum image. After collecting all the peaks, a detected peak with the largest radius r'_{sync} and angle θ'_{sync} is taken to be the synchronization peak, which is then used to synchronize all the remaining peaks in a way described by Eq. (2.8). The result is a set of local peaks $P' = \{p'_1, p'_2, \cdots, p'_h\}$.

Then, we calculate the new radius r'_{sync} of P_{sync} and take the ratio ρ between r'_{sync} and r_{sync} to decide $R'_1 = R_1 \times \rho$ and $R'_2 = R_2 \times \rho$. Also, we divide the ring area of the middle frequency band B between the two circles with radii R'_1 and R'_2 into n equally-spaced concentric circles and into m angle ranges to make B become a set of ℓ sectors $D = \{d_1, d_2, \cdots, d_\ell\}$ where $\ell = m \times n$, as seen in Figure 2.5. Then, we compare P' and D to decide the watermark bit stream $W = w_1 w_2 \cdots w_\ell$ by:

$$w_k = \begin{cases} 1 & \text{if certain } p_i' \text{ falls in } d_k, \\ 0 & \text{otherwise,} \end{cases} \tag{2.9}$$

where $1 \le k \le \ell$ and $1 \le i \le h$. This means that, if there is a peak within a sector d_k, the bit w_k is set to be "1;" otherwise, "0." Finally, we transform the bit stream into an integer number as the extracted watermark and complete the watermark extraction process. The detail of the process can be described as an algorithm as follows.

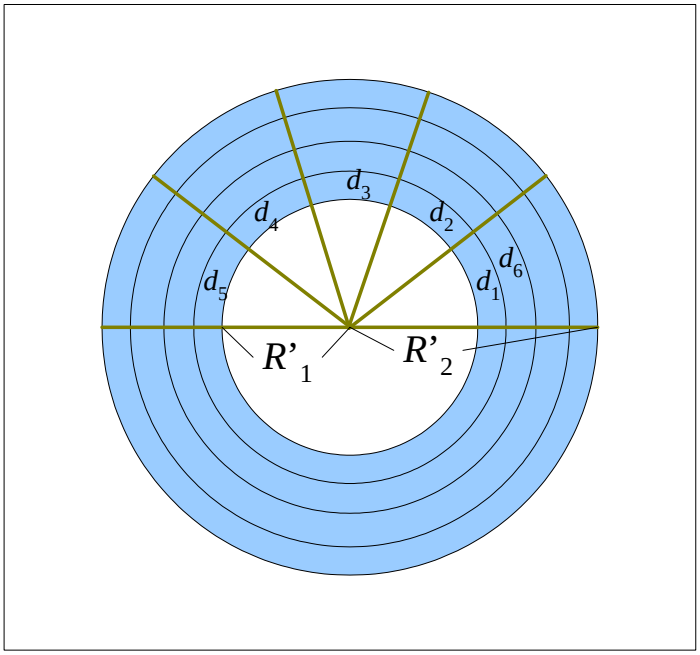


Figure 2.5 The middle frequency band is separated into concentric circles and into angular sectors.

2.4.2 Detailed Algorithm

The input to the proposed watermark extraction process includes just a stego-image *S*. The output is a watermark *W* that is a serial number embedded presumably in *S*. The extraction algorithm can be expressed as an algorithm as follows. Figure 2.6 illustrates the proposed process of watermark extraction.

Algorithm 2: *Watermark extraction process*.

Input: A stego-image S.

Output: A watermark W.

Steps.

1. Transform the red and blue color channels of S into the DFT domain to get Fourier spectra S'_{red} and S'_{blue}.
2. Detect peaks within the upper halves of S'_{red} and S'_{blue}, respectively, by performing the following operations.
   2.1 Use a threshold value T to detect peaks in the middle-frequency band. If a coefficient value is larger than T, it is considered as a peak.
   2.2 Select the peak with the largest radius as the synchronization peak P_{sync}, and calculate its angle change $\Delta\theta$ with respective to the original angle of the synchronization peak.
   2.3 Reconstruct the angles of the remaining peaks by Eq. (2.8) to get their new locations $P' = \{p'_1, p'_2, \cdots, p'_h\}$.
   2.4 Divide the middle frequency band between R'_1 and R'_2 into n equally-spaced concentric circles and into m sectors to make B become a set of ℓ sectors $D = \{d_1, d_2, \cdots, d_\ell\}$, where $\ell = m \times n$.
   2.5 Compare P' and D to decide the watermark bit stream according to the way specified by Eq. (2.9).
3. Concatenate the two watermark bit streams obtained from processing S'_{red} and S'_{blue} sequentially, and transform the result into a serial number as the desired watermark W.

Stego-image

Get R and G channels

Decide a threshold value T

DFT

Detect peaks

Find synchronization peak

Reconstruct angles of remaining peaks

Decode watermark bit stream

Watermark

Figure 2.6 Flowchart of the extraction process.

2.5 Experimental Results

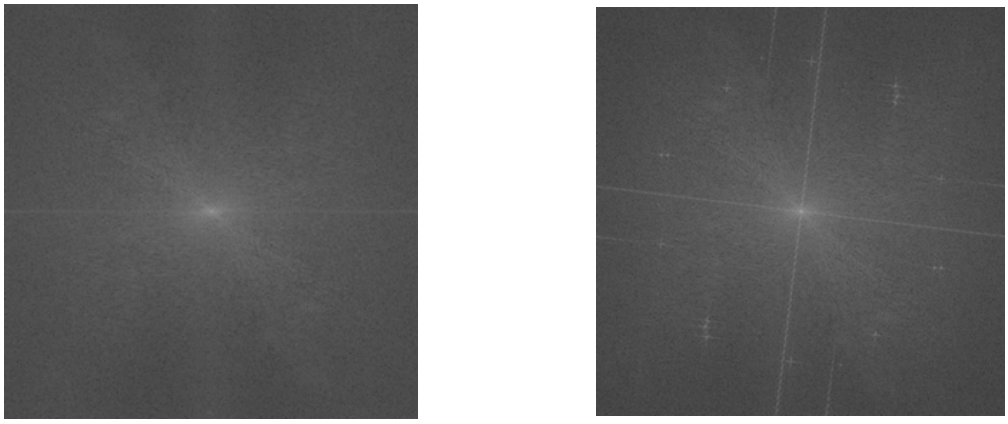
Some experimental results of applying the proposed method are shown here. A serial number 877 is transformed into binary form to be a watermark bit stream. The factor *c* that determines the embedded watermark strength is assigned to be 1.5. Figure 2.7 shows an input image with size 512×512. And Figure 2.8(a) shows the stego-image of Figure 2.7 after embedding the watermark. In addition, Figures 2.8(b) and (c) show the corresponding Fourier spectrum image and the detected locations of the peaks marked with red and green marks. The green mark is the synchronization peak. Figure 2.8(d) show a rotated image of Figure 2.8(a) and the corresponding Fourier spectrum image and the detected peak locations are shown in Figures 2.8(e) and (f), respectively. It shows that the Fourier spectrum image have the same angle of rotation with the tampered image. Figure 2.9(a) shows a scaled image of Figure 2.8(a) and the corresponding Fourier spectrum image with the detected peak locations are shown in Figure 2.9(b). The embedded peaks can be successfully detected in our experiments.

Figures 2.10(a) and (b) show two other color images both with size 512×512. And the corresponding stego-images after embedding the watermark are shown in Figures 2.10(c) and (d), respectively. The corresponding PSNR values are shown in Table 2.1, which show that the quality of each of the stego-images is still good. And the embedded watermark is imperceptible by human vision.

Finally, two rotated images are shown in Figures 2.11(a) and (b). And Figures 2.12(a) and (b) show two scaled images. The watermarks can be extracted successfully from each of these images by the proposed watermark extraction process in our experiments.

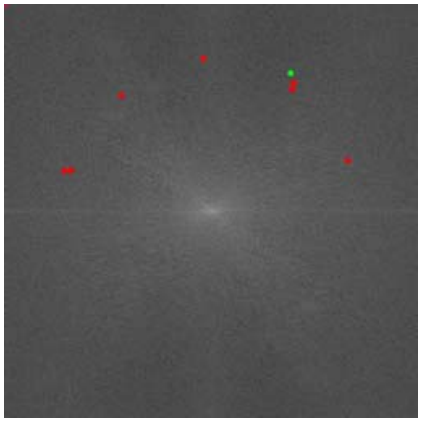
Figure 2.7 An input image “Lena”.

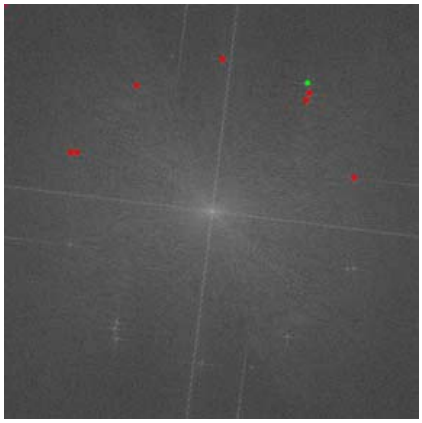
(a) (d)



(b) (e)

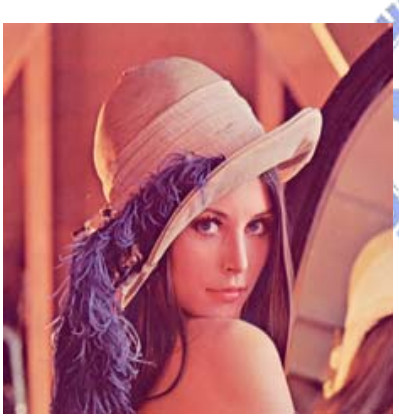
Figure 2.8 An output stego-images with the watermark, the tampered image and Fourier spectrum images. (a) Stego-Image “Lena”. (b) Fourier spectrum image of (a). (c) Peak locations of (c). (d) Tampered image after rotating 13 degree clockwise. (e) Fourier spectrum image of (d). (f) Peak locations of (e).

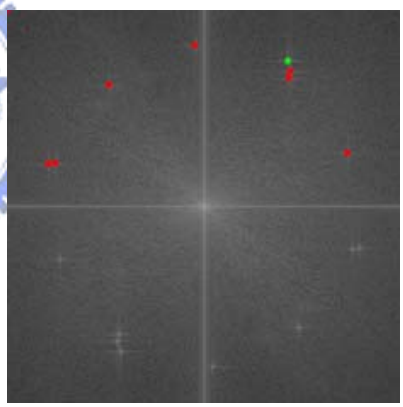




(c) (f)

Figure 2.8 An output stego-images with the watermark, the tampered image and Fourier spectrum images. (a) Stego-Image “Lena”. (b) Fourier spectrum image of (a). (c) Peak locations of (c). (d) Tampered image after rotating 13 degree clockwise. (e) Fourier spectrum image of (d). (f) Peak locations of (e) (continued).





(a) (b)

Figure 2.9 The tampered image and the Fourier spectrum image. (a) Tampered image after scaling to 90%. (b) Fourier spectrum image of (a) with peak locations.

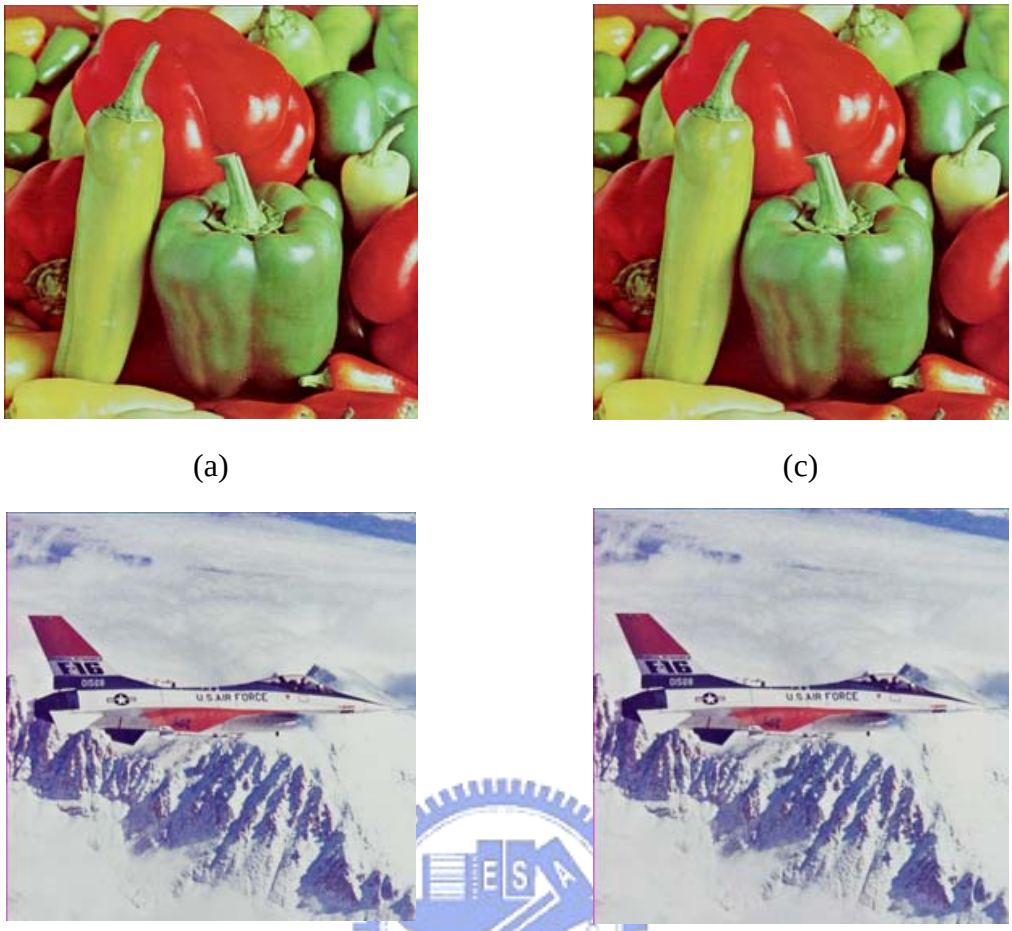


Figure 2.10 Input images, and output stego-images with the watermark. (a) Image "Pepper". (b) Image "Jet". (c) and (d) Stego-images after embedding the watermark, respectively.

Table 2.2 The PSNR values of recovered images after embedding watermarks.

	Lena	Pepper	Jet
PSNR	33.0	33.0	33.0

(a) (b)

Figure 2.11 Some tampered images with different rotations. (a) Tampered image after rotating 97 degree clockwise. (b) Tampered image after rotating 7 degree counterclockwise.

(a) (b)

Figure 2.12 Some tampered images with different scaling ratios. (a) Tampered image after scaling to 150%. (b) Tampered image after scaling to 90%.

2.6 Discussions and Summary

In this chapter, we have proposed a method for embedding a watermark into a color image by using peak detection and synchronization of coefficient-value peak locations in the DFT domain. Utilizing some properties of image coefficients in the

DFT domain, we can embed a watermark in the form of a binary stream by creating the peaks circularly and symmetrically in a middle frequency band in the transform domain. On the other hand, an extra synchronization peak is added to synchronize the peak locations. The embedded watermark was shown by the experimental results to be robust against rotation and scaling attacks, thus achieving the goal of image copyright protection.

However, the data hiding capability of the proposed watermark embedding method is not large and cannot accommodate a normal-sized logo image. It may be tried to solve this problem in the future.

Chapter 3
Copyright Protection by Watermarking for Color Images against Rotation And Cropping Attacks Using Synchronization of Peak Locations in DFT Domain And DCT-Coefficient Relationship Comparison

In this chapter, the proposed method for embedding a watermark in color images against rotation and cropping attacks is described. The idea is based on hiding information in two frequency domains, the DFT and DCT domains. In the DFT domain, a synchronization peak is embedded for the purpose of detecting if an image is suffered rotation attacks And in the DCT domain, multiple copies of watermarks and verification codes are embedded for the purpose of surviving cropping attacks.

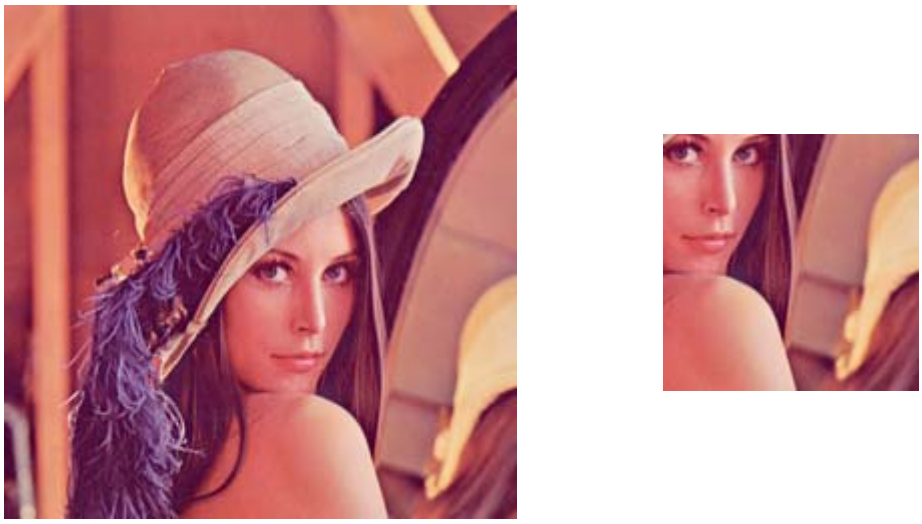
The remainder of this chapter is organized as follows. In Section 3.1, an introduction is given first. In Section 3.2, several ideas behind the proposed methods are described. In Section 3.3, the proposed watermark embedding process is presented. In Section 3.4, the proposed watermark extraction process is presented. Some experimental results are shown in Section 3.5. And finally, in Section 3.6 some discussions and a summary are made.

3.1 Introduction

The watermarking technique has been proposed for copyright protection of digital images. However, a watermarked image may suffer many kinds of attacks and image processing operations. In order to achieve the goal of image copyright protection, the watermark must be robust against these attacks. In this chapter, we focus on embedding a watermark to survive rotation and cropping attacks.

3.1.1 Problem Definition

In order to embed watermarks in color images for surviving rotation and cropping attacks, invariant features of images with respect to rotation and cropping operations should be adopted. After a cropping attack, a tampered image becomes just part of an original image, in which many pixel values are lost, as we can see in the example shown in Figure 3.1. In this case, the embedded watermark may be destroyed or lose too much information to complete the watermark extraction work. After a rotation attack, a stego-image is rotated for a certain angle. In the subsequent watermark extraction process, the location in the frequency domain where the watermark can be extracted may be shifted. We call this situation a watermark synchronization problem. The proposed method has the merit of integrating techniques applied in the two frequency domains to achieve resistance capabilities to rotation and cropping attacks. Another merit is that the watermark can be extracted without referencing the original image.



(a) (b)

Figure 3.1 A color image and a cropped image. (a) Color image “Lena”. (b) Cropped image of (a).

3.1.2 Review of Employed Techniques

In Yin and Tsai [5], annotation data, which are duplicated for many times before embedding, are embedded in the DCT domain by changing the magnitude relation of two DCT coefficients denoted as S_1 and S_2 and located at (1,4) and (2,3) in the standard quantization table within every 8×8 image blocks. And a voting process will be proceeded in the extraction process to decide the final extracted annotation data.

In Cheng and Tsai [23], an invisible watermark, which can survive JPEG compression, were proposed. A bit of the watermark was embedded into an 8×8 image block by adjusting the magnitude relation of the two DCT coefficients at (1,4) and (2,3) in the standard quantization table. By the magnitude relation of two DCT coefficients, the embedded invisible watermark can be extracted. Table 3.1 shows the standard quantization table. And the methods of adjusting the magnitude relation of

two selected DCT coefficients S_1 and S_2 are described as follows:

$$\begin{cases} if\ b=1 \quad and\ S_1 < S_2, \quad then \quad Swap(S_1, S_2), \\ if\ b=0 \quad and\ S_1 \geq S_2, \quad then \quad Swap(S_1, S_2). \end{cases} \tag{3.1}$$

The way of changing the magnitude relation of two DCT coefficients includes four situations as follows.

1. If a bit *b* of the embedding data is "1" and the relationship of a pair of AC coefficients S_1 and S_2 is $S_1 \geqq S_2$, then S_1 and S_2 are unchanged.
2. If b is "1" and $S_1 \leqq S_2$, then S_1 and S_2 are exchanged.
3. If b is "0" and $S_1 \geqq S_2$, then S_1 and S_2 are exchanged.
4. If b is "0" and $S_1 \leqq S_2$, then S_1 and S_2 are unchanged.

That is, if the bit to be embedded is "1", the relation $S_1 \geq S_2$ is created; otherwise, $S_1 < S_2$ created. Later, by checking the magnitude relation of the two selected DCT coefficients within every 8×8 image block as follows, a hidden bit *e* can be reconstructed:

$$e = \begin{cases} 1 & if \quad S_1 \geq S_2, \\ 0 & if \quad S_1 < S_2. \end{cases} \tag{3.2}$$

That is, if $S_1 \geq S_2$, the embedded bit is taken to be "1;" otherwise, "0".

3.2 Ideas of Proposed Method

The ideas involved in the proposed embedding process are described as follows.

Table 3.1 A 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

3.2.1 Proposed Technique for Multiply Embedding Watermark in DCT Domain for Preventing Cropping Attack

A watermark used for image ownership protection is assumed to be a binary image in this study, called a *watermark image*. First, the watermark image is rescaled to a pre-determined size $n \times m$. In the proposed method, a secret key and a random number generator are employed to randomize the locations of pixels of the rescaled watermark image to get a randomized watermark image W_r. In addition, W_r is multiply duplicated to get a large randomized watermark image W_l. Let C be the cover image of size $M \times N$. Because we only embed a bit of W_l in an 8×8 image block, the number d of copies of W_r is $\left\lfloor \frac{M}{8 \times m} \right\rfloor \times \left\lfloor \frac{N}{8 \times n} \right\rfloor$, where $\lfloor \bullet \rfloor$ means the integer floor function. Then, W_l is embedded into a cover image by changing the magnitude relation of two DCT coefficients within every 8×8 image blocks in the DCT domain.

The embedding process is based on the method described in Section 3.1.2.

3.2.2 Proposed Technique for Hiding Verification Code in DCT Domain for Watermark Existence Check

When a stego-image is suffered from cropping attacks, the synchronization of the 8×8 image blocks is missed. This means that the locations of the 8×8 image blocks are not the same as the original ones. In order to synchronize the 8×8 image blocks to ensure success of watermark extraction, we try to extract verification codes to make sure that the locations of image blocks are the same as the original ones. In the proposed method, the verification code is a pre-defined square pattern with the same size $n \times m$ as the randomized watermark image. The square pattern is shown in Figure 3.2 with a quarter of its pixels being white and three-fourths black. Then, we select two DCT coefficients at (S_3, S_4) and embed multiple copies of the verification patterns by Eq. (3.1) described previously.

In the watermark extraction process, we set a start location and get a sub-image of the stego-image with the range of embedding a square pattern. Then, we extract each bit from 8×8 image blocks of the sub-image by Eq (3.2) and search in it one copy of the verification pattern. If we cannot extract the verification pattern, the start location is shifted a pixel in zigzag order to extract again. If one copy of the verification patterns can be extracted successfully, the locations of the 8×8 image blocks can be synchronized. This means that the watermark can be extracted from the sub-image in sequence.

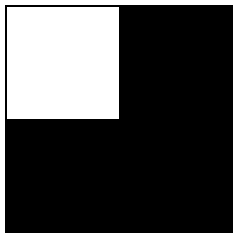


Figure 3.2 A square verification pattern.

3.2.3 Proposed Technique for Hiding Synchronization Peak in DFT Domain to Detect Rotated Angle

In order to synchronize the 8×8 image blocks in a rotated stego-image, we should know the angle of rotation, and rotate the stego-image reversely. Then, we use the verification codes to check the location where the watermark can be extracted.

As we described the properties of the DFT coefficients in Section 2.2.1, a synchronization peak can be embedded into the DFT domain of a cover image in the way described in Section 2.2.4. By detecting the change of the synchronization peak location, we can check whether the stego-image has been rotated or not. If the stego-image is rotated, the rotation degrees will be the same as the angle changes of the synchronization peak. Then, we can detect the angle of rotation and rotate the stego-image reversely. In this situation, there may be an opportunity for the stego-image to synchronize the locations of the image blocks with the original ones, and the watermark can be extracted from the image blocks.

3.3 Proposed Watermark Embedding Process

In this section, the method proposed to embed watermarks in color images and to extract the watermark from stego-images is described. In order to embed a watermark to survive rotation and cropping attacks, two frequency domains, DFT and DCT domains, are adopted in this study. And the watermark is assumed to be a binary image. The watermark embedding process is presented subsequently.

3.3.1 Embedding of Watermarks

In the proposed embedding watermark process, a watermark W is resized to $m \times n$ and the locations of the pixels are randomized with a key K and a random generator G to get a randomized watermark W_r. In addition, let C be the cover image of size $M \times N$. Then, W_r is duplicated into d copies to form a large-sized randomized watermark W_l, with $d = \left\lfloor \frac{M}{8 \times m} \right\rfloor \times \left\lfloor \frac{N}{8 \times n} \right\rfloor$. On the other hand, a verification code E is created according to a pre-defined pattern image with the size of $M \times N$. And E is also duplicated into d copies to form a multiple-copies pattern image E_l. Based on the employed method described in Section 3.1.2, C is divided into non-overlapping 8×8 image blocks. For each image block, the RGB color values of the block are transformed into the YC_bC_r color values. The 8×8 FDCT is performed on the Y channel of the resulting image. Two pairs of DCT coefficients (S_1, S_2) and (S_3, S_4), each coefficient pair having the same quantization step size within the JPEG standard quantization table, are selected. Then, we adjust the first pair of DCT coefficients to embed an image pixel of W_l and the second pair to embed a pixel of E_l by Eq. (3.1). The 8×8 inverse DCT are then performed on these DCT coefficients and then the YC_bC_r color values are transformed back into RGB color values. After all blocks are processed, we transform the image into the DFT domain. And a synchronization peak is embedded according to the method described in Section 3.2.3. Finally, we use the

inverse DFT to transform the DFT-domain image into the spatial domain. A stego-image is thus obtained. A detailed algorithm for the watermark embedding process is described next.

3.3.2 Detailed Algorithm

The inputs to the proposed watermark embedding process include a color image *C*, a watermark *W*, a square verification pattern *E*, a secret key *K*, and a random generator *G*. The output is a stego-image *S*. The process can be briefly expressed as an algorithm as follows. Figure 3.3 shows a flowchart of the embedding process.

Algorithm 1: *Watermark embedding process*.

Input: A given color image *C*, a watermark *W*, a verification pattern *E*, a secret key *K*, and a random generator *G*.

Output: A stego-image *S*.

Steps.

1. Divide *C* of size $M \times N$ into non-overlapping 8×8 image blocks. Let c_{ij} be an image block of *C*, where $0 \le i \le \left\lfloor \frac{M}{8} \right\rfloor$, and $0 \le j \le \left\lfloor \frac{N}{8} \right\rfloor$.
2. Resize *W* to a pre-selected size $n \times m$ and use *K* and *G* to randomize the locations of pixels of resized *W* to get a randomized watermark image W_r.
3. Duplicate *E* of size $n \times m$ and W_r $\left\lfloor \frac{M}{8 \times m} \right\rfloor \times \left\lfloor \frac{N}{8 \times n} \right\rfloor$ times, respectively, to form W_l and E_l. Let w_{ij} and e_{ij} be the pixel value of W_l and E_l at position (i, j), where $0 \le i \le \left\lfloor \frac{M}{8} \right\rfloor$, and $0 \le j \le \left\lfloor \frac{N}{8} \right\rfloor$.
4. Transform every image block into the YC_bC_r color model.
5. Transform the Y channel of each image block into the frequency domain by performing the 8×8 FDCT.

6. For each block, two pairs of the DCT coefficients (S_1, S_2) and (S_3, S_4) are selected to embed W_l and E_l. For each pixel w_{ij} and the corresponding image block c_{ij}, adjust the magnitude relation of the first pair of the coefficients by Eq. (3.1). For each pixel e_{ij} and the corresponding image block c_{ij}, adjust the magnitude relation of the second pair of the coefficients by Eq. (3.1).
7. Transform each image block back into the spatial domain by performing the 8×8 inverse DCT.
8. Transform each image block from the YC_bC_r color model into the RGB color model to get an image *H*.
9. Transform the blue color channel of *H* into the DFT domain to get a Fourier spectrum *F*.
10. Add a synchronization peak P_{syn} into *F* according to the scheme described in Section 2.2.4.
11. Transform F back into the spacial domain by the inverse DFT.
12. Take the final result as the desired stego-image *S*.

3.4 Watermark Extraction Process

In the proposed watermark extraction process, a watermark can be extracted to verify the copyright. The process of applying this technique will be described in this section. And a detailed algorithm for the process will be given.

3.4.1 Extraction of Watermarks

In the proposed watermark extraction process, the blue color channel of a stego-image is first accessed to extract the synchronization peak to check whether the

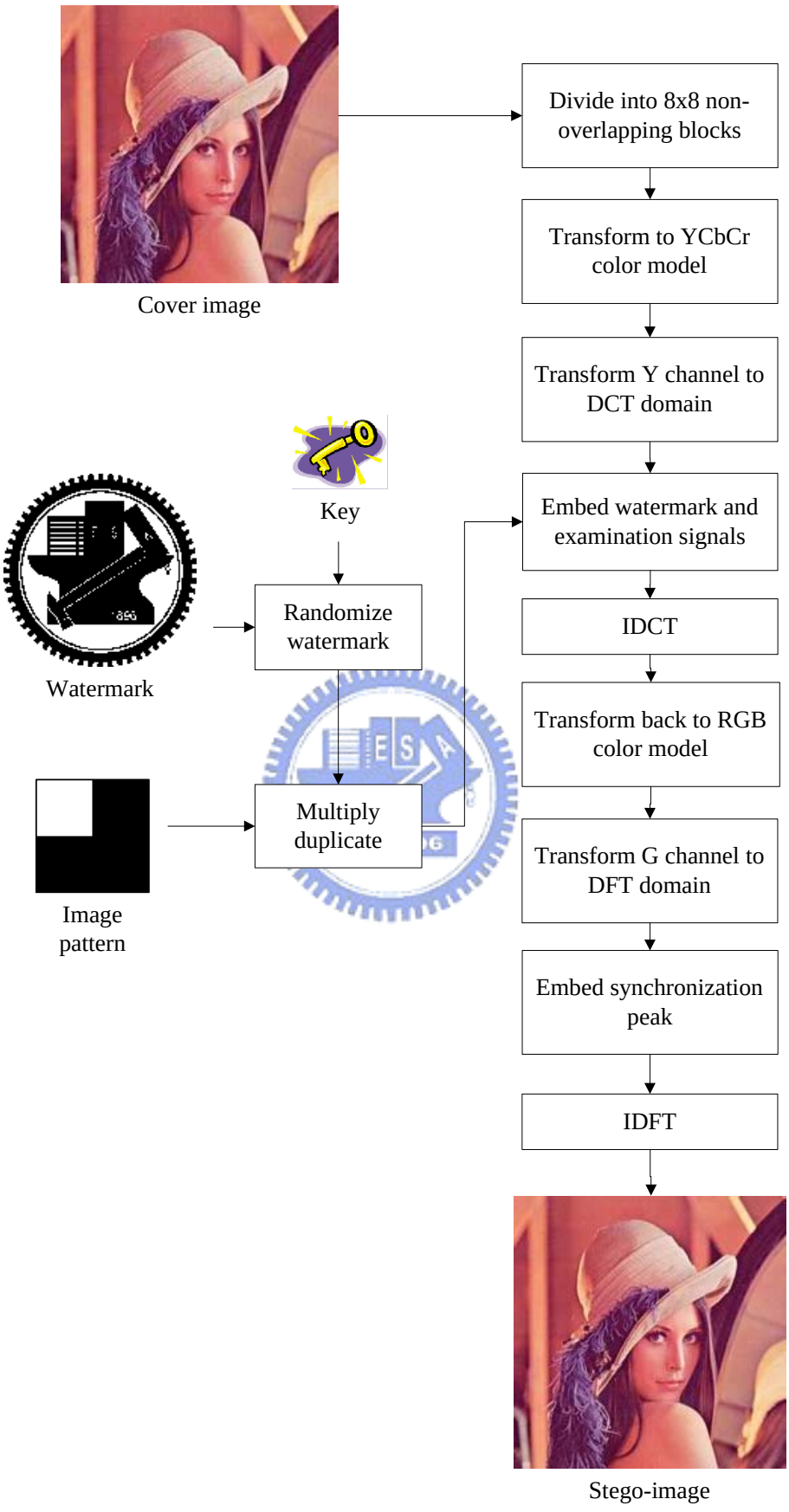


Figure 3.3 Flowchart of the proposed embedding process.

stego-image is rotated or not. By the way described in Section 3.2.3, we can know the angle of rotation from the change of synchronization peak location and rotate the stego-image reversely to get S_r. In addition, S_r is divided into non-overlapping image blocks, and the magnitude relation of a pair of coefficients at (S_3, S_4) in each image block of the range of a square pattern is checked to extract the verification code. If the extracted data do not equal the verification code, we shift the locations of image blocks in a zigzag order to check again. As long as the extracted data equal the verification code, we can extract watermark pixels from each image block of the same locations by comparing the magnitude relation of the DCT coefficients at (S_1, S_2).

Finally, we use a key and a random generator to recover the locations of the watermark image pixels. Then we can get the extracted watermark. This completes the extraction process of the watermark.

3.4.2 Detailed Algorithm

The input to the proposed watermark extraction process includes a stego-image *S*, a secret key *K*, and a random generator *G*. The output is a watermark *W* that is a binary image embedded presumably in *S*. The extraction algorithm can be expressed as an algorithm as follows. Figure 3.4 illustrates the proposed process of watermark extraction.

Algorithm 2: *Watermark extraction process*.

Input: A stego-image *S*, a secret key *K*, and a random generator *G*.

Output: A watermark *W*.

Steps.

1. Transform the blue color channel of *S* into the DFT domain to get a Fourier spectrum S'_{blue}.

2. Detect the synchronization peak within S'_{blue}, respectively, and calculate its angle change $\Delta\theta$ with respective to the original angle of the synchronization peak.
3. Rotate S reversely $\Delta\theta$ degrees to get a re-rotated image S_r.
4. Set a start location $P(x, y)$, and get a sub-image S_b of S_r with size $8m \times 8n$.
5. Divide S_b into non-overlapping 8×8 image blocks. Let s_{ij} be an image block of S_b, where $0 \le i \le m$, and $0 \le j \le n$.
6. For each 8×8 image block s_{ij}, transform its Y channel into the frequency domain by performing the 8×8 FDCT.
7. Select two DCT coefficients S_3 and S_4 to determine each embedded data in the following way:

$$e_{i,j}\begin{cases}1 & if\ \ S_3 \ge S_4,\\ 0 & if\ \ S_3 < S_4.\end{cases} \tag{3.3}$$

8. If the number k of bits with "0" of the extract data is larger than a threshold value T and smaller than $\frac{m \times n}{4}$, decide that a verification code is obtained, and extract the watermark in sequence. Otherwise, shift one pixel of the location of P in a zigzag order, and repeat Step 4 to Step 7.
9. For each image block in S_b, select two DCT coefficients S_1 and S_2 to determine the value of a watermark pixel by Eq. (3.2).
10. Continue detecting blocks one after another until all the pixel values of the embedded randomized watermark W_r are extracted.
11. Use K and G to recover the pixel locations of W_r.
12. Take the final result as the desired watermark W.

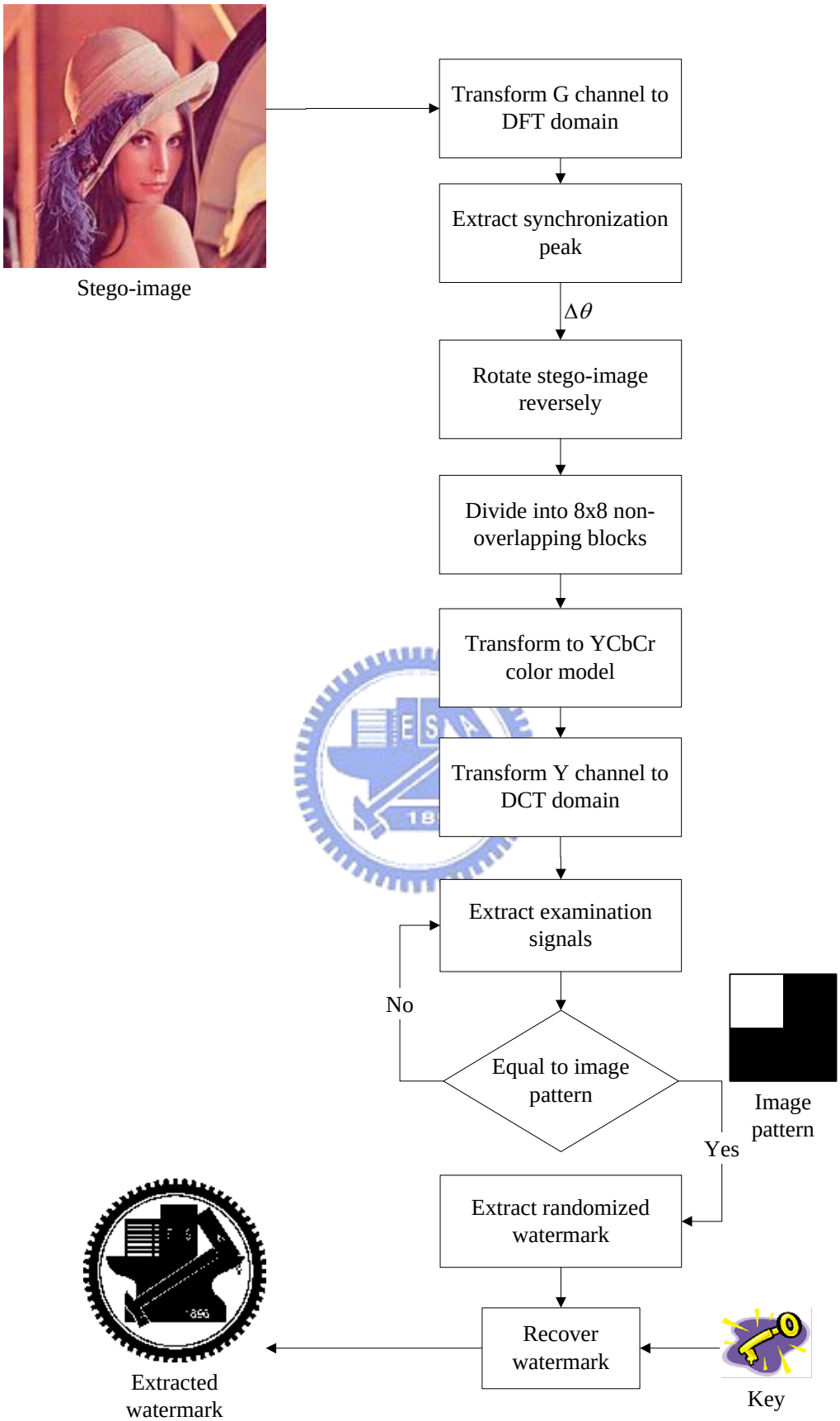


Figure 3.4 Flowchart of proposed extraction process.

3.5 Experimental Results

Some experimental results of applying the proposed watermark extraction method are shown here. The watermark used in the experiment is shown in Figure 3.5. Figure 3.6(a) shows a color image with size 512×512. And the stego-images after the watermark was embedded are shown in Figure 3.6(b). Figures 3.6(c) and (d) are the tampered images after being rotated 8 degrees counterclockwise and cropped partially, respectively. And Figures 3.6(e) and (f) show the extracted watermarks of the tampered images, respectively. Table 4.1 shows the error rates of the extracted watermarks, which are extracted from the tampered images subject to rotation and cropping attacks. The formula adopted for calculating the error rate (ER) is as follows:

$$Error\ rate = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|w_{ij} - w'_{ij}\right|}{M \times N} \tag{3.4}$$

where $M \times N$ is the size of the watermark, w_{ij} is the pixel value of the original watermark, and w'_{ij} is the pixel value of the extracted watermark. If the error rate is smaller, the embedded watermark and the extracted one are more similar.

(a)

(b)

Figure 3.5 Watermark images. (a) Binary image of size 256×256. (a) Binary image of size 32×32.

(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.6 Input binary images, output stego-images with secret data, and the differences. (a) Color image “Lena”. (b) Stego-images after embedding the watermark. (c) The tampered images after rotating 8 degrees counterclockwise. (d) Cropping image of (b). (e) and (f) The extracted watermark of (c) and (d) , respectively.

Table 3.2 The error rates of the extracted watermark of Figures 3.4(e) and (f).

	Fig. 3.6(e)	Fig. 3.6(f)
Error rate	0.246	0.247

Figures 3.7(a) and (b) show two color images both with size 512×512. And the stego-images after embedding the watermark are shown in Figures 3.7(c) and (d), respectively. And Figures 3.7(e) and (f) show the extracted watermark of Figures 3.7(c) and (d), respectively. The PSNR values of the stego-images are shown in Table 3.3, which show that the quality of each of the stego-images is still good. And the embedded watermark is imperceptible by human vision.

(a) (b)

Figure 3.7 Input color images, output stego-images with the watermarks, and the extracted watermarks. (a) Color image “Pepper”. (b) Color image “Jet”. (c) and (d) Stego-images after embedding watermarks, respectively. (e) and (f) The extracted watermarks, respectively.

(c)

(d)

(e)

(f)

Figure 3.7 Input color images, output stego-images with the watermarks, and the extracted watermarks. (a) Color image “Pepper”. (b) Color image “Jet”. (c) and (d) Stego-images after embedding watermarks, respectively. (e) and (f) The extracted watermarks, respectively (continued).

Table 3.3 The PSNR values of the stego-images after embedding the watermark.

	Lena	Pepper	Jet
PSNR	36.0	36.0	34.4

3.6 Discussions and Summary

In this chapter, we have proposed a method for embedding a watermark into a color image by using synchronization peak detection in the DFT domain and the DCT-coefficient relationship comparison. Two kinds of frequency domains are

utilized in this study. We embed the watermark in the DCT domain by adjusting the magnitude relation of two selected the DCT coefficients. On the other hand, an extra synchronization peak is added in the DFT domain to detect the angle of a rotation attack. The embedded watermark is shown to be robust and can survive rotation and cropping attacks by the experimental results. The proposed method can achieve the goal of protecting image copyright of the owner.

Chapter 4
Copyright Protection by Watermarking for Color Images against Print-and-Scan Operations Using Coding and Synchronization of Peak Locations in DFT Domain

The proposed method for copyright protection of color images against print-and-scan operations is described in this chapter. The main idea is to embed a watermark as coefficient-value peaks in the DFT domain of an input image. Then, by detecting the peaks in the DFT domain of a *reproduced image*, which is the image obtained from scanning a print version of a watermarked image, the embedded watermark can be extracted.

The remainder of this chapter is organized as follows. In Section 4.1, an introduction is given first. In Section 4.2, the ideas of the proposed method are described. In Section 4.3, the proposed watermark embedding process is presented. In Section 4.4, the proposed watermark extraction process is described. In Section 4.5, some experimental results are shown. Finally, in Section 4.6 some discussions and a summary are given.

4.1 Introduction

Because of the rapid development of electronic products, the printer and scanner are commonly used for publications and reproductions of documents. Digital images can be printed to spread around. And the reproduced image becomes a digital version similar to the original one, with some distortions sometimes. Digital watermarking is a technique for embedding a watermark into an image to protect an owner's copyright of the image. The embedded watermark must be robust against print-and-scan operations. It is desirable that after applying these operations on the stego-image, the watermark is not fully destroyed and can be extracted to verify the copyright of the image.

4.1.1 Properties of Images Applied Print-and-Scan Operations

In a reproduced image, there are two categories of distortions, called geometric transformations and pixel value distortions. The geometric transformations include rotation, scaling, and padding. On the other hand, the luminance, contrast, gamma correction, chrominance variations, and the blurring of neighboring pixels occur in the pixel value distortions [28]. The geometric transformations do not cause significant effects on the visual quality but the pixel value distortions do, as seen in the examples shown in Figure 4.1.

4.1.2 Problem Definitions

A reproduced image always has pixel-value distortions and geometric transformations. Therefore, a watermark embedded in a reproduced image must have a certain degree of robustness against pixel-value distortions and geometric operation

(a) (b)

Figure 4.1 A color image and a reproduced image. (a) Color image “Lena”. (b) Reproduced image of (a) with quality of 100dpi.

attacks. In order to embed watermarks in color images to survive geometric transformations, invariant features of images with respect to geometric transformations should be adopted. And the embedded watermark must be imperceptible, of course.

4.2 Ideas of Proposed Method

4.2.1 Proposed Technique for Coding Peak Locations for Watermarking

In the proposed watermarking method, first we shift the zero frequency point $F(0,0)$ to the center of the DFT domain and embed a given watermark in a ring region in a middle band, denoted as B subsequently, in the DFT domain between two circles with two pre-selected radii R_1 and R_2 where $R_1 < R_2$. Then, ℓ *embeddable positions* P $= \{p_1, p_2, \cdots, p_\ell\}$ are selected according to the methods described in Section 2.2.3. And

we adjust the coefficient values of some of these positions to be *local peaks* in the frequency domain to form a desired watermark in a way described next.

First, we select a number h of peaks, among the ℓ ones at the embeddable positions, for use to embed a watermark W which is a pre-selected series number with an integer value w. These peaks may be viewed to *code* the watermark value w.

To decide which peaks should be used, we apply a combinatorial operation to get all possible *codes* R = $\{r_1, r_2, \cdots, r_g\}$, with each code r_i specifying a set of h peak locations, where $g = C(\ell, h)$ with $C(\ell, h)$ being a *combinatorial number* which means the number of ways of picking h *unordered* outcomes from ℓ possibilities. In this study, we choose h to equal $\ell/2$ because $C(\ell, h)$ will then has the maximal value for a specific $\ell = m \times n$. For example, if ℓ is equal to four and h is equal to two, we have P = $\{p_1, p_2, p_3, p_4\}$ and $g = C(4, 2) = 6$ which means that we have 6 possible codes R = $\{r_1, r_2, \cdots, r_6\}$ for use as watermarks where $r_1 = \{p_1, p_2\}$, $r_2 = \{p_1, p_3\}$, $r_3 = \{p_1, p_4\}$, $r_4 = \{p_2, p_3\}$, $r_5 = \{p_2, p_4\}$, and $r_6 = \{p_3, p_4\}$.

Then, after choosing a watermark W with integer value w no larger than g, we get the w-th code r_w in R and modify the coefficient values $M(u_k, v_k)$ of the corresponding embeddable positions p_k specified by r_w to be local peaks $M'(u_k, v_k)$ by the following equation:

$$M'(u_k, v_k) = M(u_k, v_k) + c \tag{4.1}$$

where c is a pre-selected factor that determines the embedded watermark strength.

4.2.2 Proposed Technique for Automatically Adjusting Threshold Value for Extracting Watermark

To extract the embedded watermark in a reproduced image, we have to detect, using a threshold value *T*, local peaks in the DFT domain of the image to recover the code representing the watermark. Because the reproduced image has pixel-value changes which degrades the original image quality and counteracts the values of the embedded peaks, the threshold value *T* is difficult to determine. The way to solve this problem is to select first an initial value T_0 for *T* and adjust *T* to get a refined value in the *i*th iteration according to the following rule:

$$T_i = \begin{cases} T_{i-1} + \delta & if \quad e_i > h, \\ T_{i-1} - \delta & if \quad e_i < h, \end{cases} \tag{4.2}$$

where T_i is the value for *T* in the *i*th iteration, *h* is the previously-mentioned number of embedded peaks of each code, e_i is the number of the detected peaks using the threshold T_{i-1}, and δ is a pre-selected constant. This means that if the number of detected peaks is larger than the number of the embedded peaks, the threshold value is incremented for the amount of δ to make the detected peaks in the next iteration become fewer, and vice versa. The iterations stop at the moment when the number of the detected peaks equals *h*. The detected peaks are then *decoded* to recover the embedded watermark value *w*.

4.3 Watermark Embedding Process

As mentioned previously, a watermark used for image ownership protection is assumed to be a serial number in this study. In this section, the process of embedding a watermark in a color image will be described.

4.3.1 Embedding of Watermarks

In the proposed watermark embedding process, first we rescale an input image to a pre-selected $M \times M$ square image, where M is a radix-2 number. Next, we use radix-2 Fast Fourier Transform (FFT) to transform the input image to the DFT domain fast. Then, we use the DFT domains of the red and blue channels of the input image to embed a series-number watermark. The watermark is transformed into a bit stream which is then divided into two halves. Each half is transformed back to be an integer as a smaller watermark to be embedded in one of the red and blue color channels according to the idea described in the last section. In addition, a synchronization peak is also embedded into the middle frequencies according to the scheme described in Section 2.2.4. A detailed algorithm of this process is described as follows.

4.3.2 Detailed Algorithm

The inputs to the proposed watermark embedding process are a color image C and a watermark W. The output is a stego-image S. The process can be briefly expressed as an algorithm as follows. Figure 4.2 shows a flowchart of the process.

Algorithm 1: *Watermark embedding process*.

Input: A given color image C and a watermark W.

Output: A stego-image S.

Steps.

1. Rescale C to get an $M \times M$ square image C', where M is a radix-2 number.
2. Transform the red and blue channels of C' into the frequency domain by the DFT to get C_r' and C_b'.
3. Transform W into a binary stream, divide the result equally into two substreams, and transform them back into two integers W_r and W_b.
4. Embed W_r and W_b into C_r' and C_b', respectively, by performing the

following operations.

3.1 Decide a set of radiuses R = $\{r_1, r_2, \cdots, r_n\}$ for n equally-spaced concentric circular stripes in the middle band B of the frequency domain between two pre-selected circles with radiuses R_1 and R_2, with $R_1 < R_2$.

3.2 Decide m angles $\Theta = \{\theta_1, \theta_2, ..., \theta_m\}$ equally distributed in the range from 0° to 180°. Also, take ℓ to be $m \times n$.

3.3 Obtain ℓ embeddable positions P = $\{p_1, p_2, \cdots, p_\ell\}$ with p_k (k = 1, 2, …, ℓ) located at $(r_i \cos\theta_j, r_i \sin\theta_j)$ where i and j are such that $k = (i - 1) \times m + j$, and their symmetric positions Q = $\{q_1, q_2, \cdots, q_\ell\}$ with each q_k located at the symmetric location of p_k.

3.4 Apply the combinatorial operation mentioned previously to get g codes R = $\{r_1, r_2, \cdots, r_g\}$ with each code r_k (k = 1, 2, …, g) specifying a set of peak locations , where $g = C(\ell, h)$ with $h = \ell/2$.

3.5 According to the value w of W', take r_w out of R and adjust the coefficient value at each location within r_w and that of its symmetric location to be local peaks by Eq. (4.1).

3.6 Add a synchronization peak P_s according to the scheme described in Section 2.2.4.

5. Transform C_r' and C_b' back into the spatial domain by the inverse DFT.
6. Rescale C' to the original size of C.
7. Take the final result as the desired stego-image S.

4.4 Watermark Extraction Process

In the proposed watermark extraction process, no other information but the stego-image is needed as the input. The watermark can be extracted to verify the

copyright. The processes of applying this technique will be described in this section. And a detailed algorithm for the process will be given.

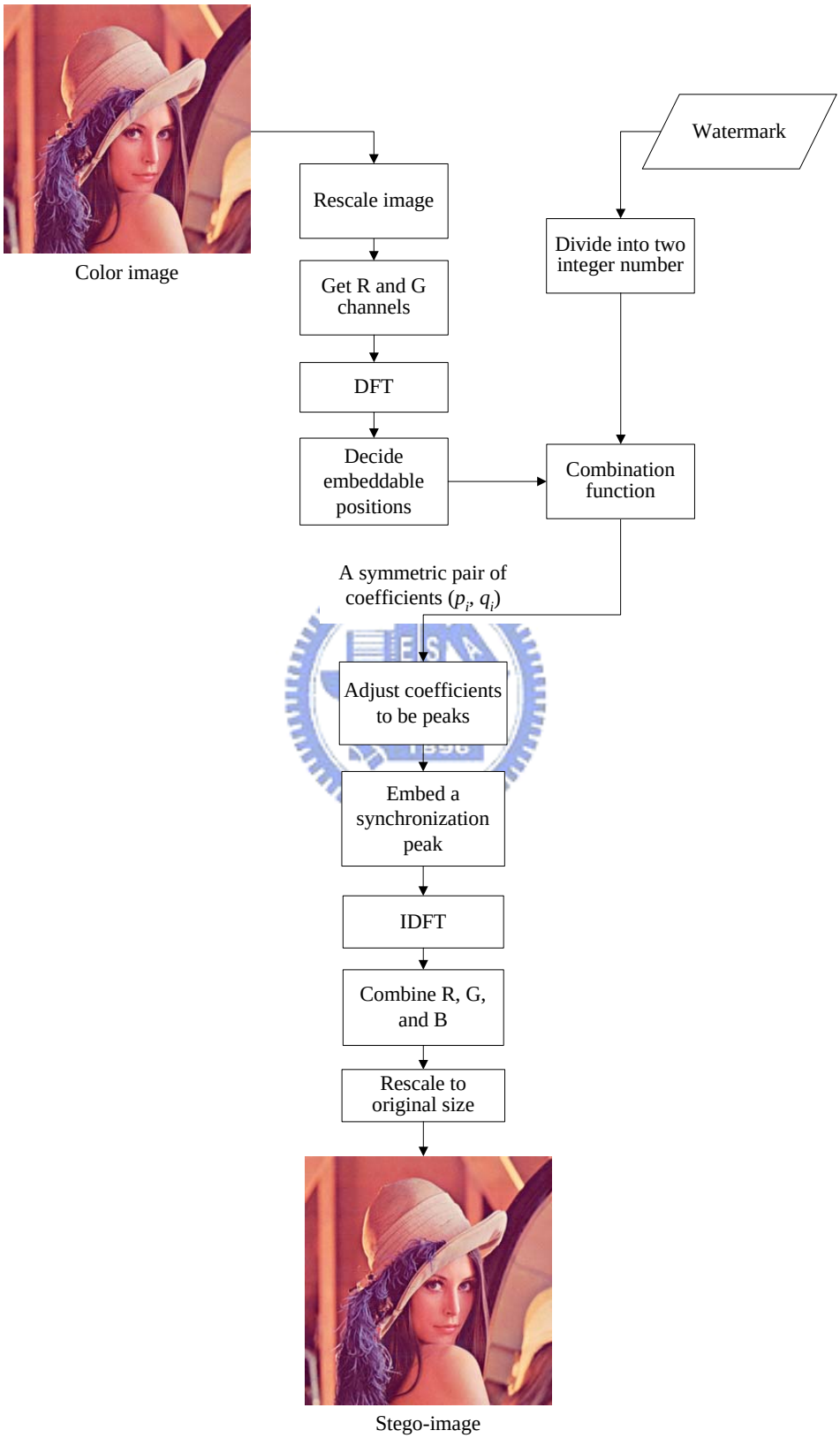


Figure 4.2 Flowchart of the embedding process.

4.4.1 Extraction of Watermarks

In the proposed watermark extraction process, the stego-image is rescaled to a square image of the pre-selected size $M \times M$ where M is a radix-2 number mentioned previously. The red and blue channels are transformed into the DFT domain by using FFT. Because of the symmetric property of the DFT coefficient values specified in Section 2.2.1, we only need to detect local peaks within the range of the upper-half Fourier spectrum image. After collecting all the peaks, a detected peak with the longest radius is taken to be the synchronization peak P_s, which is then used to synchronize the peak locations. Then, we reconstruct the angles of the remaining h peaks in $\mathrm{P} = \{p_1, p_2, \cdots, p_h\}$ by Eq. (2.8) to get their new locations $\mathrm{P}' = \{p'_1, p'_2, \cdots, p'_h\}$.

Also, we separate the ring area of the middle frequency band B between the two circles with the previously-mentioned radii R_1 and R_2 into n equally-spaced concentric circles and into m angle ranges to make B become a set of ℓ sectors $\mathrm{D} = \{d_1, d_2, \cdots, d_\ell\}$, where $\ell = m \times n$, as seen in Fig. 5. Then, P' and D are compared to collect h sectors to form a set A by the following way:

$$\begin{aligned} &\text{for all } k = 1, 2, \ldots, \ell \text{ and } i = 1, 2, \ldots, h, \\ &\textit{if } p_i' \textit{ falls in } d_k\textit{, then regard } d_k \textit{ to be in } \mathrm{A}. \end{aligned} \tag{4.3}$$

This means that, if there is a peak within an area d_k, d_k is taken to into A. Finally, we use a combinatorial operation with D and h as inputs to get g kinds of possible *codes* $\mathrm{R} = \{r_1, r_2, \cdots, r_g\}$, where $g = C(\ell, h)$ with $h = \ell/2$. Then, we check if there is any r'_j which is equal to A with $1 \le j \le g$. The integer number j is then taken as the extracted watermark value. This completes the extraction process of the watermark.

4.4.2 Detailed Algorithm

The input to the proposed watermark extraction process includes just a stego-image S. The output is a watermark W that is a serial number embedded presumably in S. The extraction algorithm can be expressed as an algorithm as follows. Figure 4.3 illustrates the proposed process of watermark extraction.

Algorithm 2: *Watermark extraction process*.

Input: A stego-image S.

Output: A watermark W.

Steps.

1. Rescale S to get an $M \times M$ square image S', where M is a radix-2 number.
2. Transform the red and blue color channels of S' into the DFT domain to get Fourier spectra S'_{red} and S'_{blue}.
3. Detect peaks within the upper-half areas of S'_{red} and S'_{blue}, respectively, by performing the following operations.
   - 2.1 Use an adjusted threshold value T to detect peaks in the middle-frequency band according to the method described in Section 4.2.2.
   - 2.2 Select a peak with the longest radius to be the synchronization peak, and calculate its angle change $\Delta\theta$ with respective to the original angle of the synchronization peak.
   - 2.3 Reconstruct the angles of the remaining h peaks by Eq. (2.8) to get their new locations $P' = \{p'_1, p'_2, \cdots, p'_h\}$.
   - 2.4 Divide the middle frequency band between R_1 and R_2 into n equally-spaced concentric circles and into m angle ranges to make the middle band become several ℓ sectors $D = \{d_1, d_2, \cdots, d_\ell\}$, where $\ell =$

$m \times n$.

2.5 Compare P′ and D to select h areas as a set A according to the way specified by Eq. (4.3), where $h = \ell/2$.

2.6 Apply a combinatorial operation to get g codes R' = $\{r'_1, r'_2, \cdots, r'_g\}$, with each code r'_j (j = 1, 2, …, g) specifying a set of h areas of D, where $g = C(\ell, h)$. Then, check if there is any r'_j equal to A with $1 \le j \le g$. And j is taken as the desired serial number.

4. Link two serial numbers in binary form from S'_{red} and S'_{blue} sequentially.
5. Transform the linked bit stream into a serial number.
6. Take the final result as the desired watermark W.

4.5 Experimental Results

Some experimental results of applying the proposed method are shown here. A serial number 888 is a watermark. The factor c that determines the embedded watermark strength is assigned to be 1.5. Figure 4.4 shows an input image with size 512×512. And Figure 4.5(a) shows the stego-image of Figure 4.4 after embedding the watermark. In addition, Figures 4.5(b) and (c) show the corresponding Fourier spectrum image and the detected locations of the peaks marked with red and green marks. The green mark is the synchronization peak. Figure 4.5(d) shows that Figure 4.5(a) was printed at 600 dpi on an HP Color LaserJet 5500 laser printer and scanned at 100 dpi using a MICROTEC Scanmaker9800XL flatbed scanner, and the corresponding Fourier spectrum image and the detected peak locations are shown in Figures 4.5(e) and (f), respectively. The embedded peaks can be successfully detected in our experiments.

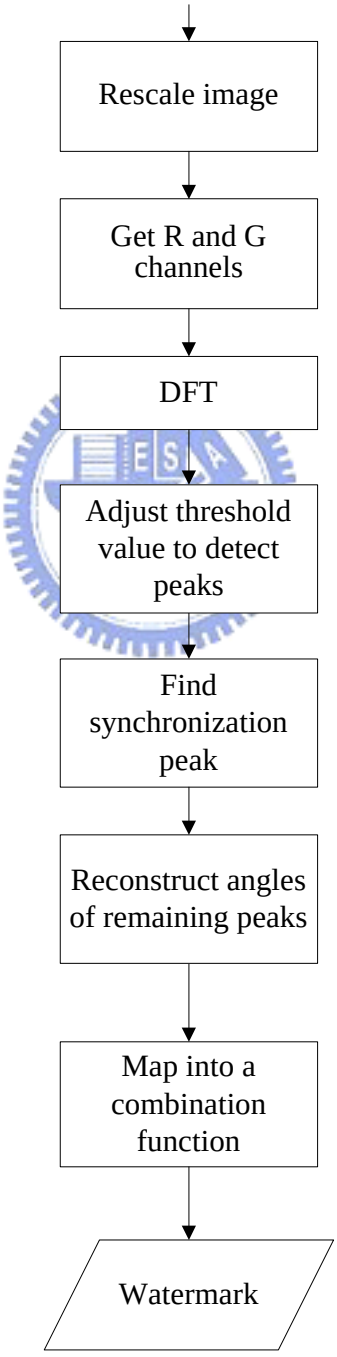


Figure 4.3 Flowchart of the extraction process.

Figures 4.6(a) and (b) show two other color images both with size 512×512. And the corresponding stego-images after embedding the watermark are shown in Figures 4.6(c) and (d), respectively. The corresponding PSNR values are shown in Table 4.1, which show that the quality of each of the stego-images is still good. And the embedded watermark is imperceptible by human vision.

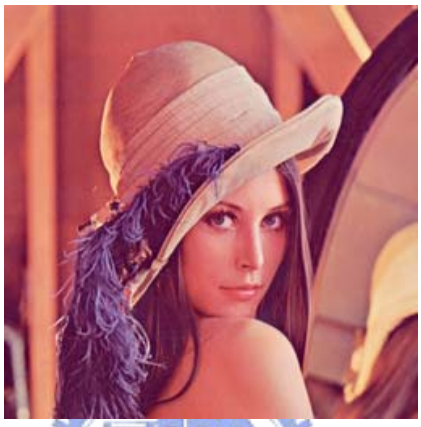
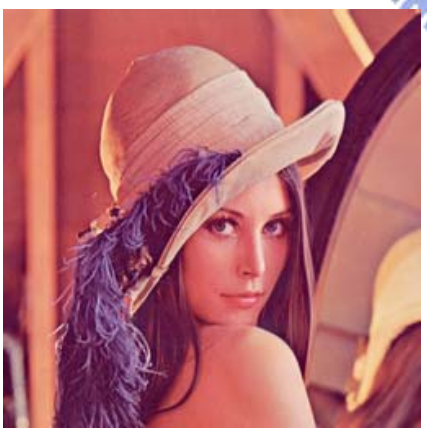
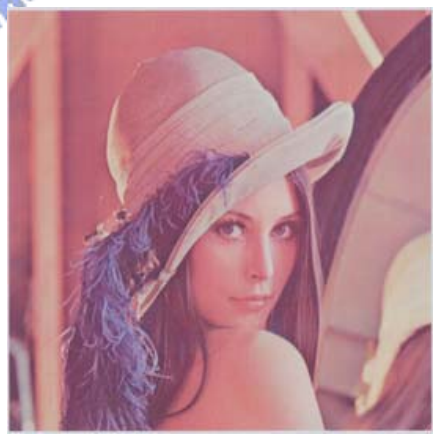


Figure 4.4 An input image “Lena”.

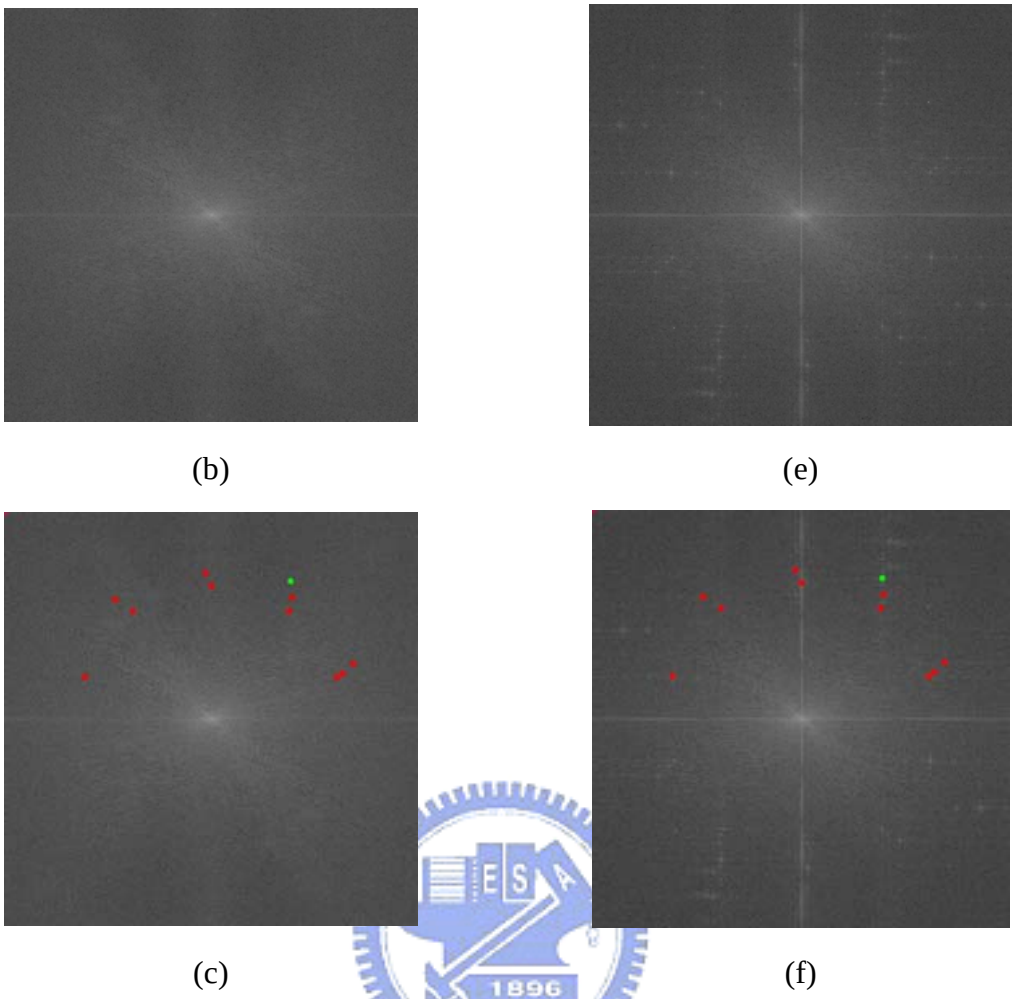


(a)



(d)

Figure 4.5 An output stego-images with the watermark, the reproduced image and Fourier spectrum images. (a) Stego-Image “Lena”. (b) Fourier spectrum image of (a). (c) Peak locations of (c). (d) Reproduced image with the resolution of 100dpi. (e) Fourier spectrum image of (d). (f) Peak locations of (e).



(b) (e)

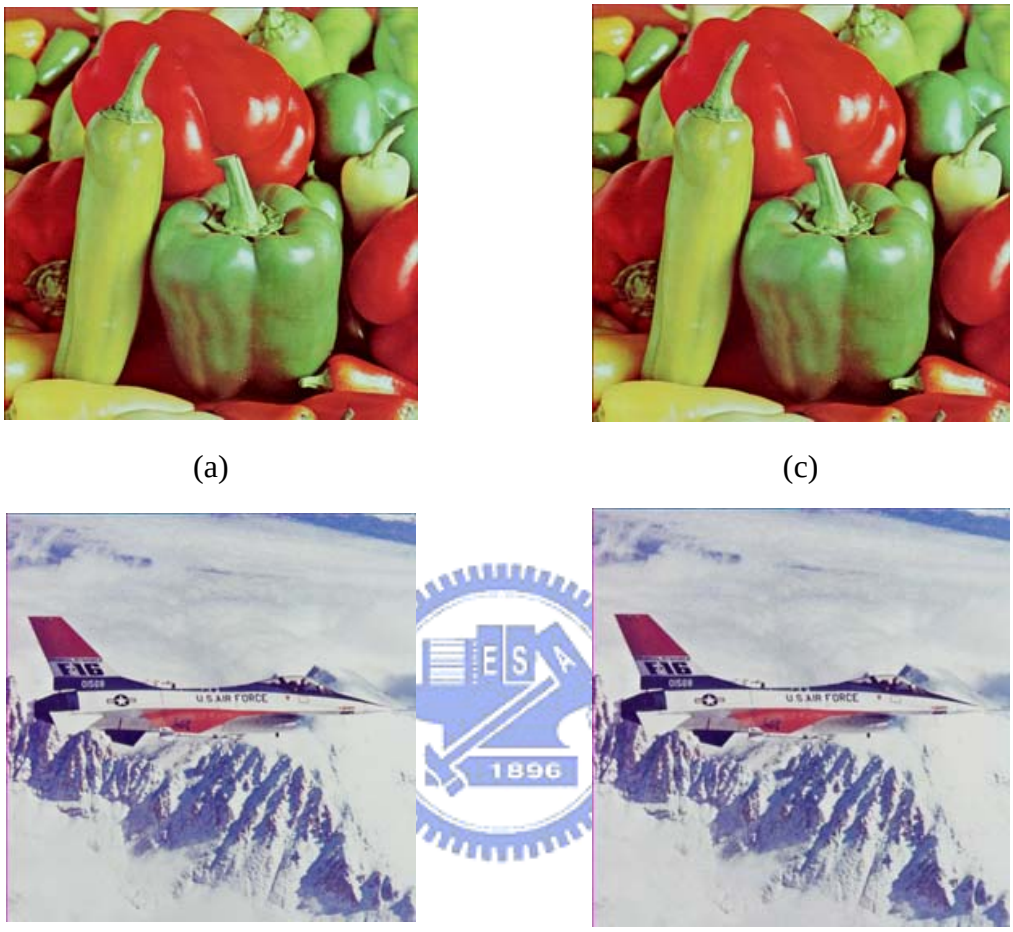
(c) (f)

Figure 4.5 An output stego-images with the watermark, the reproduced image and Fourier spectrum images. (a) Stego-Image “Lena”. (b) Fourier spectrum image of (a). (c) Peak locations of (c). (d) Reproduced image with the resolution of 100dpi. (e) Fourier spectrum image of (d). (f) Peak locations of (e) (continued).

In addition, two reproduced images are shown in Figures 4.7(a) and (b), with resolutions of 100dpi and 150dpi, respectively. The watermarks can be extracted successfully from each of these images by the proposed watermark extraction process in our experiments.

Finally, we test 120 reproduced images which are generated from twenty digital color images by printing at 600 dpi and scanning again at 85dpi, 100dpi, 150dpi, 200dpi, 250dpi and 300dpi, respectively. And the success probability of extracting the

watermarks is 91.67%. The errors came mainly from the use of improper image resolutions when rescanning the printed version of the original input images.



(a) (c)

(b) (d)

Figure 4.6 Input images, and output stego-images with the watermark. (a) Image "Pepper". (b) Image "Jet". (c) and (d) Stego-images after embedding the watermark, respectively.

Table 4.1 The PSNR values of recovered images after embedding watermarks.

	Lena	Pepper	Jet
PSNR	33.0	33.0	32.4

4.6 Discussions and Summary

In this chapter, we have proposed a method for embedding a watermark into a color image by coding and synchronization of coefficient-value peak locations in the DFT domain. According to the properties of image coefficients in the DFT domain, we embed the watermark by creating the peaks circularly and symmetrically in the middle frequencies. And we use a combinatorial function to code the peak locations. On the other hand, an extra synchronization peak is added to synchronize the peak locations. In the watermark extraction process, the positions of the coefficient-value peaks are detected and mapped into a combinatorial function to get a watermark. The embedded watermark is shown to be robust and can survive the print-and-scan operations by the experimental results. The proposed method can achieve the goal to protect the image copyright of the owner.

However, in the proposed watermark embedding method, the capacity of a regular-sized image for hiding data is not large. It is not enough to embed a logo image. In future works, it may be tried to solve this problem.

(a) (b)

Figure 4.7 Some reproduced images with different quality. (a) Reproduced image with the resolution of 100dpi. (b) Reproduced image with the resolution of 150dpi.

Chapter 5
Copyright Protection by Watermarking for Color Images against Scaling And Line-Removal Attacks Using An Image Rescaling Technique

In this chapter, the proposed watermarking method for embedding watermarks in color images against scaling and line-removal attacks is described. The proposed method is based on Yin and Tsai [5] and Cheng and Tsai [23]. By modifying their methods and adding a rescaling technique for color images, watermarks can be made robust to survive some geometric operations, like image-scaling and line-removal operations. The copyright of images can be proved by extracting the embedded watermarks.

The remainder of this chapter is organized as follows. In Section 5.1, an introduction is given. In Section 5.2, the proposed watermark embedding process is presented. In Section 5.3, the proposed watermark extraction process is presented. Some experimental results are shown in Section 5.4. Finally, in Section 5.5 some discussions and a summary are made.

5.1 Introduction

With the rapid growth of digital techniques, many images are digitized. It is an important issue to protect digital images. A common answer is to use digital watermarking techniques. Watermarking techniques utilize certain weaknesses of the human visual system to embed watermarks in images without arousing obvious artifacts. The stego-image usually is similar to the cover image.

5.1.1 Review of Employed Techniques

In this section, some techniques employed in the proposed methods are reviewed. These techniques are based on Yin and Tsai [5] and Cheng and Tsai [23]. They are introduced briefly as follows. In this study, watermarks are assumed to be binary images.

Yin and Tsai [5] proposed a method which can be employed to embed annotation data in JPEG images. Annotation data are embedded in the DCT domain by changing the magnitude relation of two DCT coefficients located at (1, 4) and (2, 3) in the standard quantization table within every 8×8 image block.

In Cheng and Tsai [23], a method was proposed to embed invisible watermarks in JPEG images. By adjusting the magnitude relation of the two DCT coefficients located at (1, 4) and (2, 3) in the standard quantization table, an invisible watermark can be embedded into a JPEG image. By checking the magnitude relation of the two DCT coefficients, the embedded invisible watermark can be extracted.

5.1.2 Problem Definition

To achieve the purpose of copyright protection, a watermark embedded in a color image should be robust. In this study, we investigate the problem of making a

watermark robust against scaling and line-removal attacks.

After these attacks, the size of a stego-image will be changed. If the employed watermark embedding method is block-based, like Yin and Tsai [5] and Cheng and Tsai [23], the coefficient locations in the frequency domain where the watermark can be extracted may be shifted or scaled. This causes a *watermark synchronization problem* in the watermark extraction process. In order to solve this problem, we propose a rescaling technique in this study to make the locations of the image blocks "synchronous" with the original ones in the sense that the image blocks with the embedded watermark are re-located to their original positions. In addition, the watermark can be extracted without referencing the original image.

5.2 Brief Description of Idea for Proposed Rescaling Technique

In the proposed method, we first rescale a cover image into a square image, called *canonical image*, with a certain size, called *the rescaling size*, which is the closest to that of the cover image, before watermark embedding. The rescaling size is limited to one of a pre-determined set of square image sizes, called *canonical scale set*. During the watermark embedding process, a *verification code*, in addition to a watermark, is also embedded in the canonical image. The code is embedded for the purpose of providing the information of the rescaling size mentioned previously.

Then, assume that the resulting stego-image might be subjected to a scaling attack by illegal users. The idea we propose in this study to solve this problem is to transform the image in suspicion into a series of rescaled square images, each with a size specified in the canonical scale set. Presumably, one of these images will be of the same size of the canonical image. To decide which of these images is the

canonical image, we check the existence of the verification code in each of the images. If the code can be extracted correctly, then the corresponding image must be of the same size as that of the canonical image because, otherwise, the verification code will not be in a proper form and cannot be extracted correctly.

More specifically, in the proposed method, we define n square images with the canonical size set H = $\{h_1, h_2, \cdots, h_n\}$, with each square image I_i with the size of $h_i \times h_i$. For a cover image I of size $M \times N$, before watermark embedding, we rescale it to be a canonical image with the rescale size of $h_r \times h_r$, where $h_r \in$ H such that $h_r \le Min(M, N) < h_{r+1}$ with $Min(M, N)$ being the minimum value of M and N. Let the canonical image be denoted as I_r. Then, we embed the input watermark and some verification codes into I_r by adjusting the magnitude relations of some DCT coefficients in I_r at [5, 23] to get an intermediate image I_r'. Finally, I_r' is rescaled to the original size of I to get a stego-image I_s. This completes the watermark embedding process.

In the watermark extraction process, a given image I_s, presumably a stego-image, is rescaled to be n canonical images I_1, I_2, …, I_n with I_i being with a size in the canonical size set H. Then, we check the existence of the verification codes in each of these canonical images one by one until the existence is confirmed in a certain canonical image I_k. At that time, the watermark in I_k is extracted. This completes the watermark extraction process.

5.3 Watermark Embedding Process

5.3.1 Embedding Watermarks

In this section, we explain the details of the proposed watermark embedding

process. Let I be a cover image of size $M \times N$. We first rescale I to get a canonical image I_r with size $h_r \times h_r$ according to the method described previously in Section 5.2. Then, based on the employed method described in Section 5.1.1, we divide I_r into non-overlapping 8×8 image blocks. A watermark W is resized to $\lfloor h_r/8 \rfloor \times \lfloor h_r/8 \rfloor$ and the verification code is a pre-defined symbol K which is transformed into binary form and duplicated k times to get a bit stream K' before being embedded. For each image block, the RGB color values are transformed into YC_bC_r color values. The 8×8 FDCT is performed in the Y channel. Two pairs of DCT coefficients (S_1, S_2) and (S_3, S_4), each coefficient pair having the same quantization step size within the JPEG standard quantization table, are selected. Then, we adjust the first pair of DCT coefficients to embed a pixel of the watermark and the second pair to embed a bit of K' by Eq. (3.1). The 8×8 inverse DCT is performed on these DCT coefficients and then the YC_bC_r color values are transformed back into RGB color values. When all blocks are checked, a stego-image is obtained. And a detailed algorithm for the watermark embedding process will be given later.

5.3.2 Detailed Algorithm

The inputs to the proposed watermark embedding process include a color image I and a watermark W. The output is a stego-image I_s. The process can be briefly expressed as an algorithm as follows. Figure 5.1 shows a flowchart of the embedding process.

Algorithm 1: *Watermark embedding process*.

Input: A given color image I and a watermark W.

Output: A stego-image I_s.

Steps.

1. Define n kinds of square image sizes with the canonical size set H = {h_1, h_2,⋯, h_n}, with each square image I_i with the size of $h_i \times h_i$.
2. Rescale I with size $M \times N$ to be a canonical image I_r with the rescale size of $h_r \times h_r$, where $h_r \in$ H, according to the method described in Section 5.2.
3. Divide I_r of size $h_r \times h_r$ into non-overlapping 8×8 image blocks. Let b_{uv} be an image block of I_r, where $0 \le u \le \left\lfloor \frac{h_r}{8} \right\rfloor$, and $0 \le v \le \left\lfloor \frac{h_r}{8} \right\rfloor$.
4. Resize W to form W_s with size $\left\lfloor \frac{h_r}{8} \right\rfloor \times \left\lfloor \frac{h_r}{8} \right\rfloor$.
5. Transform every image block of I_r into the YC_bC_r color model.
6. Transform the Y channel of each image block into the frequency domain by performing the 8×8 FDCT.
7. For each block, select two DCT coefficients S_1 and S_2 to embed W_s. For each pixel w_{uv} of W_s and the corresponding image block b_{uv}, adjust the magnitude relation of the coefficients by the following way.

$$\begin{cases} if\ w_{uv} = 1 \quad and\ S_1 < S_2, \quad then \quad Swap(S_1, S_2), \\ if\ w_{uv} = 0 \quad and\ S_1 \ge S_2, \quad then \quad Swap(S_1, S_2), \end{cases} \tag{5.1}$$

   where $Swap(S_1, S_2)$ means a process of swapping the values of S_1 and S_2 when the magnitude relation does not match the embedded bit.
8. Pre-define a verification code K and transform it into binary form. Then, duplicate it t times to get a bit stream K'.
9. For each block, two DCT coefficients S_3 and S_4 are selected to embed a bit of K' by adjusting the magnitude relation of the coefficients.
10. Transform each image block back into the spatial domain by performing the 8×8 inverse DCT.
11. Transform each image block from the YCbCr color model into the RGB

color model to get an intermediate image I_r'.

12. Rescale I_r' to the original size of I with size $M \times N$.
13. Take the final result as the desired stego-image I_s.

5.4 Watermark Extraction Process

In the proposed watermark extraction process, a watermark can be extracted to verify the copyright. The process of applying this technique will be described in this section. And a detailed algorithm for the process will be given.

5.4.1 Extraction of Watermarks

In the proposed watermark extraction process, a stego-image S is first rescaled to be n canonical images $I_1, I_2, \ldots, I_n$ with I_i being with a size in the canonical size set H. In addition, I_i is divided into non-overlapping image blocks, and the magnitude relation of a pair of coefficients at (S_3, S_4) in each image block is checked to extract the embedded data K'. Then, we use the voting scheme [5] to check the t copies of the extracted verification code within the bits of K' to determine a symbol K. We check if K equals the verification code in each of these canonical images one by one until the equalization is confirmed in a certain canonical image I_k. Then, we extract the watermark pixels from each 8×8 image blocks of I_k by comparing the magnitude relation of the DCT coefficients at (S_1, S_2). Finally, we can get the extracted watermark. This completes the extraction process of the watermark.

5.4.2 Detailed Algorithm

The input to the proposed watermark extraction process only includes a

stego-image I_s. The output is a watermark W that is a binary image embedded presumably in I_s. The extraction process can be expressed as an algorithm as follows. Figure 5.2 illustrates the proposed process of watermark extraction.

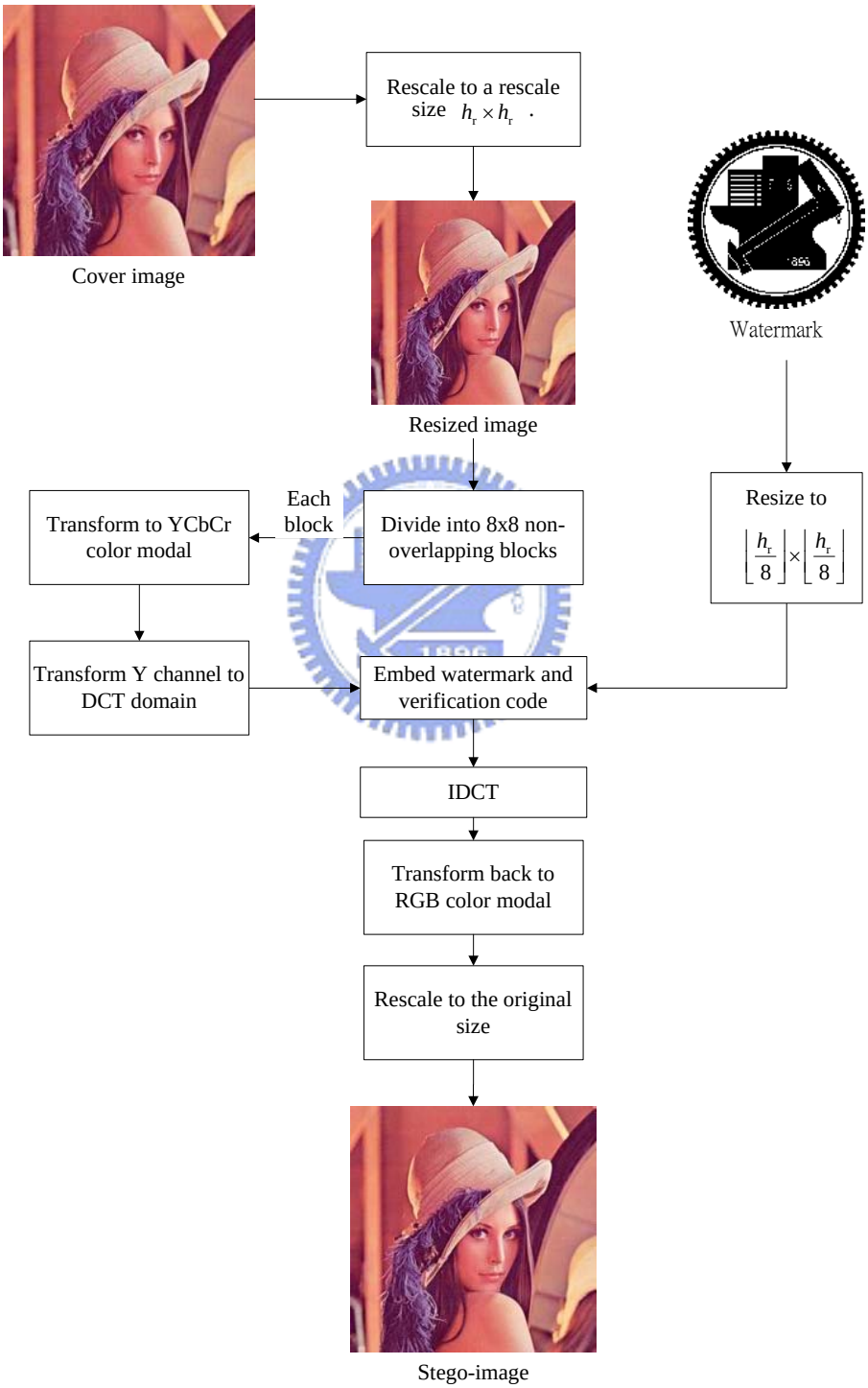


Figure 5.1 Flowchart of the proposed embedding process.

Algorithm 2: *Watermark extraction process*.

Input: A stego-image I_s.

Output: A watermark W.

Steps.

1. Define n distinct square image sizes as the canonical size set H = {h_1, h_2,⋯, h_n}, with each square image I_i with the size of $h_i \times h_i$.
2. Rescale I_s with size $M \times N$ to be n canonical images I_1, I_2, …, I_n with I_i being with a size in the canonical size set H, where $1 \le i \le n$.
3. Divide I_i with size $h_i \times h_i$ into non-overlapping 8×8 image blocks. Let b_{uv} be an image block of I_i, where $0 \le u \le \left\lfloor \frac{h_i}{8} \right\rfloor$, and $0 \le v \le \left\lfloor \frac{h_i}{8} \right\rfloor$.
4. For each 8×8 image block b_{uv}, transform the Y channel into the frequency domain by performing the 8×8 FDCT.
5. Select two DCT coefficients S_3 and S_4 to check the magnitude relation and determine the bit value of the verification codes by the voting scheme [5].
6. Detect the existence of the verification codes in each of these canonical images one by one until the existence is confirmed in a certain canonical image I_k.
7. For each 8×8 image block in I_k, select two DCT coefficients S_1 and S_2 to determine the value w_{uv} of a watermark pixel in the following way:

$$w_{uv} \begin{cases} 1 & if \quad S_1 \ge S_2, \\ 0 & if \quad S_1 < S_2. \end{cases} \tag{5.1}$$

8. Take the final result as the desired watermark W.

Stego-image

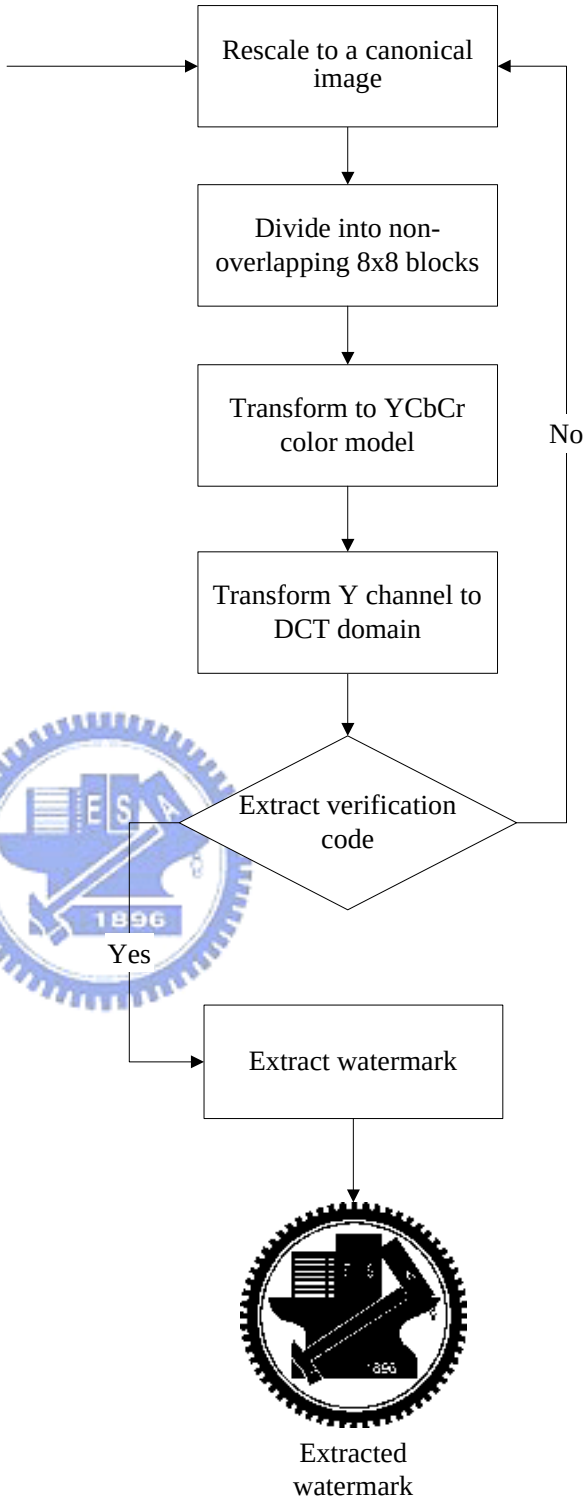


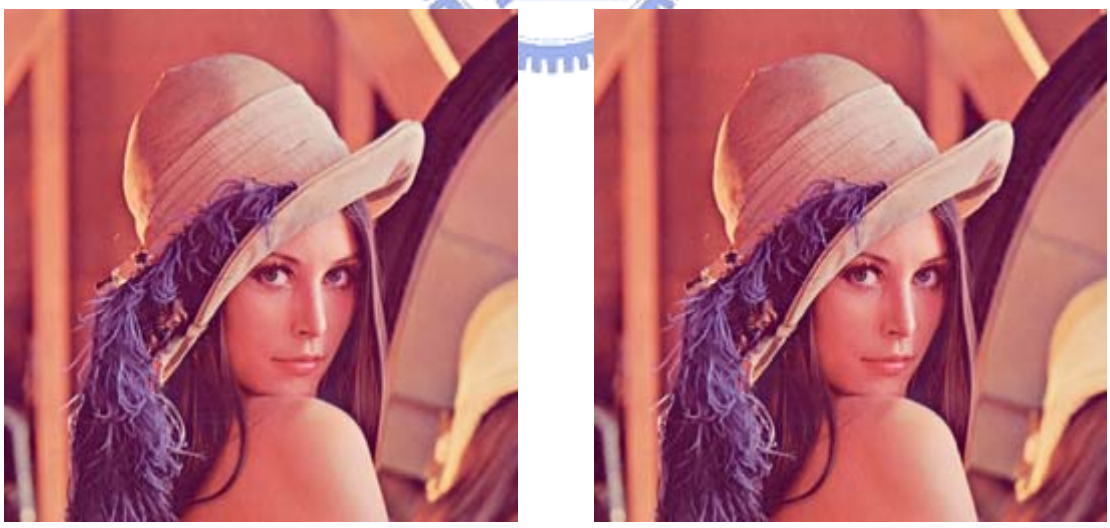
Figure 5.2 Flowchart of the proposed extraction process.

5.5 Experimental Results

Some experimental results of applying the proposed method are shown here. A watermark to be embedded is shown in Figure 5.3.

Figures 5.4(a), (b), and (c) show three color images all with size 512×512. And the resulting stego-images after the watermark was embedded are shown in Figures 5.4(d), (e), and (f), respectively.

Figure5.3 Watermark images of size 256×256.



(a) (d)

Figure 5.4 Input color images, and output stego-images with the watermark of Figure 5.3. (a) Color image “Lena”. (b) Color image “Pepper”. (d) Color image “Jet”. (d) – (e) and (f) Stego-images after embedding the watermark, respectively.

(b)

(e)

(c)

(f)

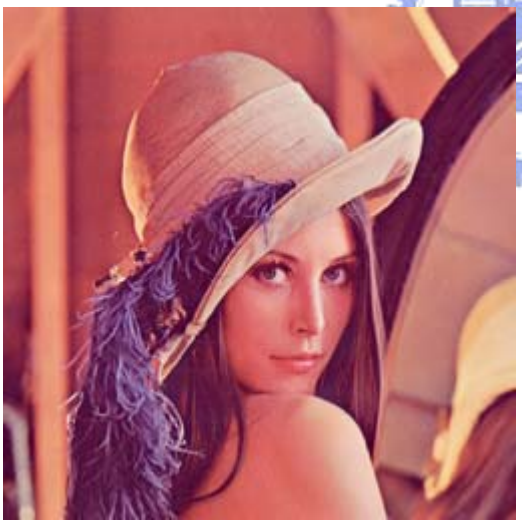
Figure 5.4 Input color images, and output stego-images with the watermark of Figure 5.3. (a) Color image “Lena”. (b) Color image “Pepper”. (d) Color image “Jet”. (d) – (e) and (f) Stego-images after embedding the watermark, respectively (continued).

The corresponding PSNR values are shown in Table 5.1, which show that the qualities of the stego-images are still good.

Table 5.1 The PSNR values of the stego-images after embedding watermark.

	Lena	Pepper	Jet
PSNR	37.0	36.0	35.0

Figure 5.5(a) shows a stego-image with watermark of Figure 5.3 and the tampered images after applying scaling and line-removal attacks are shown in Figures 5.5(b) – (d) and (e). Figures 5.5(f) – (i) and (j) show the corresponding extracted watermarks, respectively. The embedded watermarks can be successfully detected in our experiments. Table 5.2 shows the error rates of the extracted watermarks of the tampered images. We can see that the error rates are small. And that means that the watermarks are robust and less destroyed.



(a)

(f)

Figure 5.5 Stego-images, tampered images, and extracted watermarks. (a) Stego-image. (b) Tampered image after scaling 75%. (c) Tampered image after scaling 150%. (d) Tampered image after line-removal with two columns. (e) Tampered image after line-removal with two columns and one row. (f) – (i) and (j) Extracted watermarks, respectively.

(b)

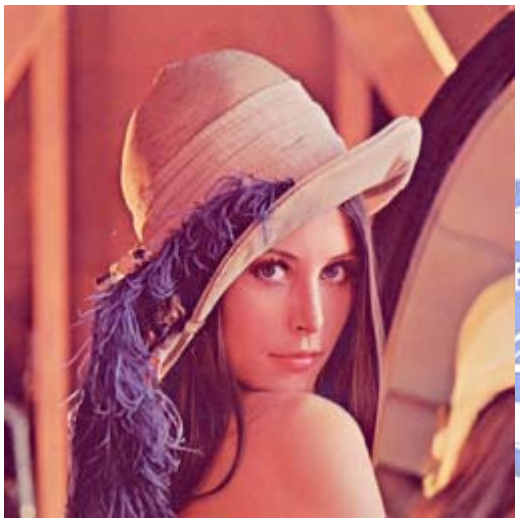
(g)

(c)

(h)

Figure 5.5 Stego-images, tampered images, and extracted watermarks. (a) Stego-image. (b) Tampered image after scaling 75%. (c) Tampered image after scaling 150%. (d) Tampered image after line-removal with two columns. (e) Tampered image after line-removal with two columns and one row. (f) – (i) and (j) Extracted watermarks, respectively (continued).

(d) (i)



(e) (j)

Figure 5.5 Stego-images, tampered images, and extracted watermarks. (a) Stego-image. (b) Tampered image after scaling 75%. (c) Tampered image after scaling 150%. (d) Tampered image after line-removal with two columns. (e) Tampered image after line-removal with two columns and one row. (f) – (i) and (j) Extracted watermarks, respectively (continued).

Table 5.2 The error rates of the extracted watermark of Figures 5.5(g) – (i) and (j).

	Fig. 5.5(g)	Fig. 5.5(h)	Fig. 5.5(i)	Fig. 5.5(j)
Error rate	0.023	0.0091	0.114	0.157

5.6 Discussions and Summary

In this chapter, we have proposed a method for embedding a watermark into a color image by using an image rescaling technique. The proposed methods are based on those proposed by Cheng and Tsai [23] and Yin and Tsai [5], which embed invisible watermarks and annotation information into images in the DCT domain. We improve it by rescaling a cover image to be a canonical image before the watermark embedding and extraction processes. By using the image rescaling technique, the synchronization problem of image blocks after scaling and line-removal attacks can be solved. The embedded watermark is shown by the experimental results to be robust and can survive scaling and line-removal attacks. The proposed method can achieve the goal of protecting image copyright of the owner.

Chapter 6
Tampering Detection in Color Images by Signature-Free Authentication Using DCT-Coefficient Relationship Comparison

In this chapter, the proposed method for color image authentication is presented. By adjusting of the magnitude relations of several pairs of the DCT coefficients of 8×8 image blocks in a color image, authentication signals, which are generated by a key and a DC coefficient value of image blocks, can be embedded in the image. The authentication signals can be extracted by checking the magnitude relations of the DCT coefficients of a given image in suspicion. The integrity of images can be verified by comparing the extracted authentication signals and the original ones.

The remainder of this chapter is organized as follows. In Section 6.1, an introduction is given first. In Section 6.2, the proposed authentication method is described. In Section 6.3, some experimental results are given to show the feasibility of the proposed approach. Finally, in Section 6.4 some discussions and a summary are made.

6.1 Introduction

6.1.1 Motivation

Because image transmission is a major activity in today's communication and digital images can be modified easily, it is necessary to design an effective algorithm for image authentication.

6.1.2 Problem Definition

When doing image authentication, one method is to use image features and save them as a signature file. Later, the signature file can be referenced to verify the integrity of the image in suspicious. But, an extra signature file for image authentication not only is a waste of the storage space but also makes the file management more complicated. On the contrary, in order to do image authentication without the use of signature files, authentication signals may be embedded into an image. We can then verify the extracted authentication signals to achieve the goal of image authentication. If the extracted authentication signals are altered, it means that the image is tampered, too. Our method has the merit of no signature files when conducting the authentication work.

In addition, some image processing operations like JPEG compression are normal behaviors to process an image, and they cannot be considered as tampering operations. Therefore, we also need semi-fragile watermarks to do image authentication.

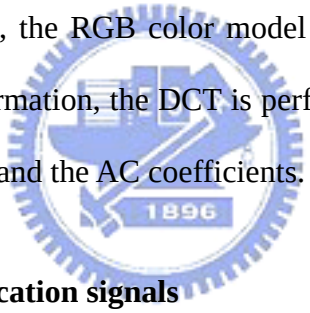
6.2 Proposed Authentication Method

In this section, the proposed method to embed authentication signals into a color

image and to authenticate the resulting image is introduced. The idea of adjusting the DCT-Coefficient relationship to embed authentication signals is employed in our method. The data hidden in the DCT coefficients in middle-high band have the semi-fragile property. That means that the embedded authentication signals will not be altered by high-quality JPEG compression. On the other hand, it is sensitive to tampering operations, like a drawing operation, in an 8×8 block.

6.2.1 Semi-Fragile Watermark Embedding Process

In the proposed method, we use an 8×8 image block to embed five bits of authentication signals. An input image is divided into non-overlapping 8×8 image blocks. For each image block, the RGB color model is transformed into the YC_bC_r color model. After the transformation, the DCT is performed on the Y channel. Then, we can get the DC coefficient and the AC coefficients.



A. Generating Authentication signals

In the proposed method, authentication signals are generated by a key and a DC coefficient value. For each 8×8 image block, we use a key and a random generator to randomly generate five bits. In addition, a DC coefficient value V is performed in the following way to get a result bit v.

$$v = \begin{cases} 1 & if \;\; 128 < V \le 255, \\ 0 & if \;\; 0 \le V \le 128, \end{cases} \tag{6.1}$$

where $0 \le V \le 255$. Then, we apply v to each of the five random bits with the exclusive-OR bit operation. The authentication signals of an image block can so be obtained.

B. Adjusting DCT-Coefficient relationship in the 8×8 Image Block

Later, we select five pairs of AC coefficients. For each pair of coefficients S_1 and S_2, the magnitude relation of the coefficients are adjusted according to Eq. 3.1 to embed a bit of the authentication signals.

After embedding the authentication signals, these DCT coefficients are transformed back into the spatial domain by the inverse DCT. Finally, each image block is transformed back into the RGB color model from the YCbCr color model. When all blocks are processed, a stego-image is obtained.

C. Detailed Algorithm

The inputs to the proposed authentication signal embedding process are a color image *C*, a random generator *G*, and a key *K*. The output is a stego-image *S*. The algorithm for the process is as follows. Figure 6.1 shows a flowchart for the algorithm.

Algorithm 1: *Authentication signal embedding process*.

Input: A given color image *C*, a random generator *G*, and a key *K*.

Output: A stego-image *S*.

Steps.

1. Divide *C* into non-overlapping 8×8 blocks.
2. For each 8×8 image block *D*, perform the following operations.
    2.1. Transform *D* from the RGB color model into the YC_bC_r color model.
    2.2. Perform the forward DCT on the Y channel.
    2.3. Check the DC coefficient value to get a bit *v* according to the method described in part A of Section 6.2.1.
    2.4. Use *G* and *K* to randomly generate five bits, called $b_1b_2b_3b_4b_5$.

2.5. Perform the exclusive-OR bit operation on each bit of $b_1b_2b_3b_4b_5$ with v to get authentication signals $a_1a_2a_3a_4a_5$ by using following equation, where $1 \leqq i \leqq 5$:

$$ai = bi \oplus v. \tag{6.2}$$

2.6. Select five pairs of DCT coefficients (S_1, S_2), (S_3, S_4), (S_5, S_6), (S_7, S_8), and (S_9, S_{10}) to embed the authentication signals.

2.7. Embed each authentication signals a_i by the following equations:

$$\begin{cases} if\ a_i = 1 \quad and\ S_{2i-1} < S_{2i}, \quad then \quad Swap(S_{2i-1}, S_{2i}), \\ if\ a_i = 0 \quad and\ S_{2i-1} \geq S_{2i}, \quad then \quad Swap(S_{2i-1}, S_{2i}), \end{cases} \tag{6.3}$$

where $1 \leqq i \leqq 5$ and $Swap(S_{2i-1}, S_{2i})$ means a process of swapping the values of S_{2i-1} and S_{2i} when the magnitude relation does not match the embedded bit.

2.8. Perform the inverse DCT on the Y channel.

2.9. Transform D from the YC_bC_r color model into the RGB color model.

3. Take the final result as the desired stego-image S.

6.2.2 Image Authentication Process

In the authentication signal embedding process, authentication signals are generated by a key and the DC coefficient value. Therefore, we can judge an image in suspicion as being tampered with or not by checking the difference between the generated authentication signals and the extracted ones.

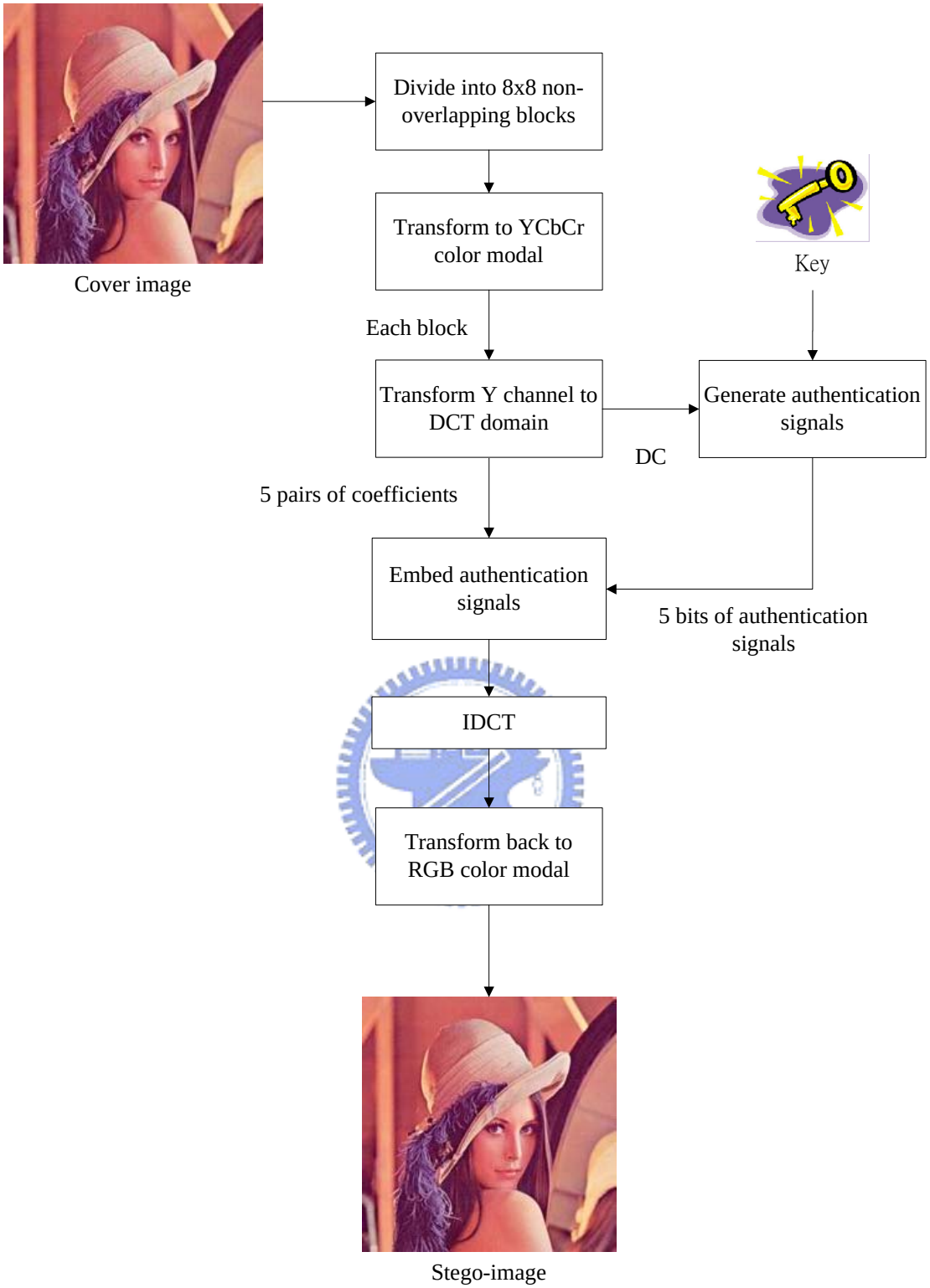


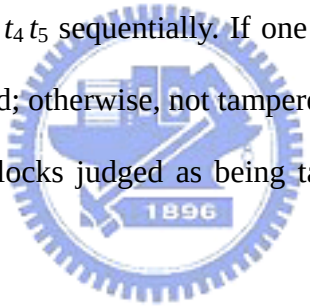
Figure 6.1 Flowchart of authentication signal embedding process.

The proposed image authentication process is essentially similar to the proposed authentication signal embedding process but in a reverse order. A suspicious image is

first divided into non-overlapping 8×8 blocks. For each image block, the RGB color model is transformed into the YC_bC_r color model. After the transformation, the FDCT is performed on the Y channel. Then, the DC coefficient and AC coefficients can be obtained. We select five pairs of AC coefficients to get the embedded authentication signals $e_1e_2\,e_3\,e_4\,e_5$ by the follow equation, where $1 \leqq i \leqq 5$:

$$e_i = \begin{cases} 1 & if\ S_{2i-1} \geq S_{2i}, \\ 0 & if\ S_{2i-1} < S_{2i}. \end{cases} \tag{6.4}$$

Then, we use the key and the random generator, which are employed in the authentication signal embedding process, to calculate the DC value to generate signals $t_1t_2\,t_3\,t_4\,t_5$ according to the method described in part A of Section 6.2.1. Then, we compare $e_1e_2\,e_3\,e_4\,e_5$ and $t_1t_2\,t_3\,t_4\,t_5$ sequentially. If one of them is different, the image block is judged as being altered; otherwise, not tampered with. In the output images of our experiments, the image blocks judged as being tampered with are marked with black color.



A. Detailed Algorithm

The proposed image authentication algorithm can be expressed as an algorithm as follows. The input is a stego-image S and the selected key K used in the authentication signal embedding. The output is an *authentication image A*. Figure 6.2 illustrates the process.

Algorithm 2: *Image authentication process*.

Input: A given stego-image S and the selected key K used in the authentication signal embedding.

Output: An authentication image A.

Steps.

1. Divide S into non-overlapping 8×8 blocks.
2. For each 8×8 image block D, perform the following operations.
   2.10. Transform D from the RGB color model to the YC_bC_r color model.
   2.11. Perform the forward DCT on the Y channel.
   2.12. Check the DC coefficient value to get a bit v according to the method described in part A of Section 6.2.1.
   2.13. Use the random generator G employed in Algorithm 1, with the input key K as a seed, to randomly generate five bits data, called $b_1b_2b_3b_4b_5$.
   2.14. Performed the exclusive-OR bit operation on each bit of $b_1b_2b_3b_4b_5$ with v to get signals $t_1t_2\,t_3\,t_4\,t_5$ by Eq. (6.2).
   2.15. Select five pairs of DCT coefficients (S_1, S_2), (S_3, S_4), (S_5, S_6), (S_7, S_8), and (S_9, S_{10}) to extract the authentication signals $e_1e_2\,e_3\,e_4\,e_5$ by (6.4).
   2.16. For all i, $i \leqq 5$, if $t_i \neq e_i$, regard the 8×8 image block as being tampered with and mark the same location in A with black color.
4. Take the final result as the desired authentication image A.

6.3 Experimental Results

Some experimental results of applying the proposed method are shown here. Figures 6.3(a), (b) and (c) show three color images all with size 512×512. And the stego-images resulting from embedding the authentication signals are shown in Figures 6.3(d), (e) and (f), respectively.

The corresponding PSNR values are shown in Table 6.1, which show that the quality of the recovered images is still good.

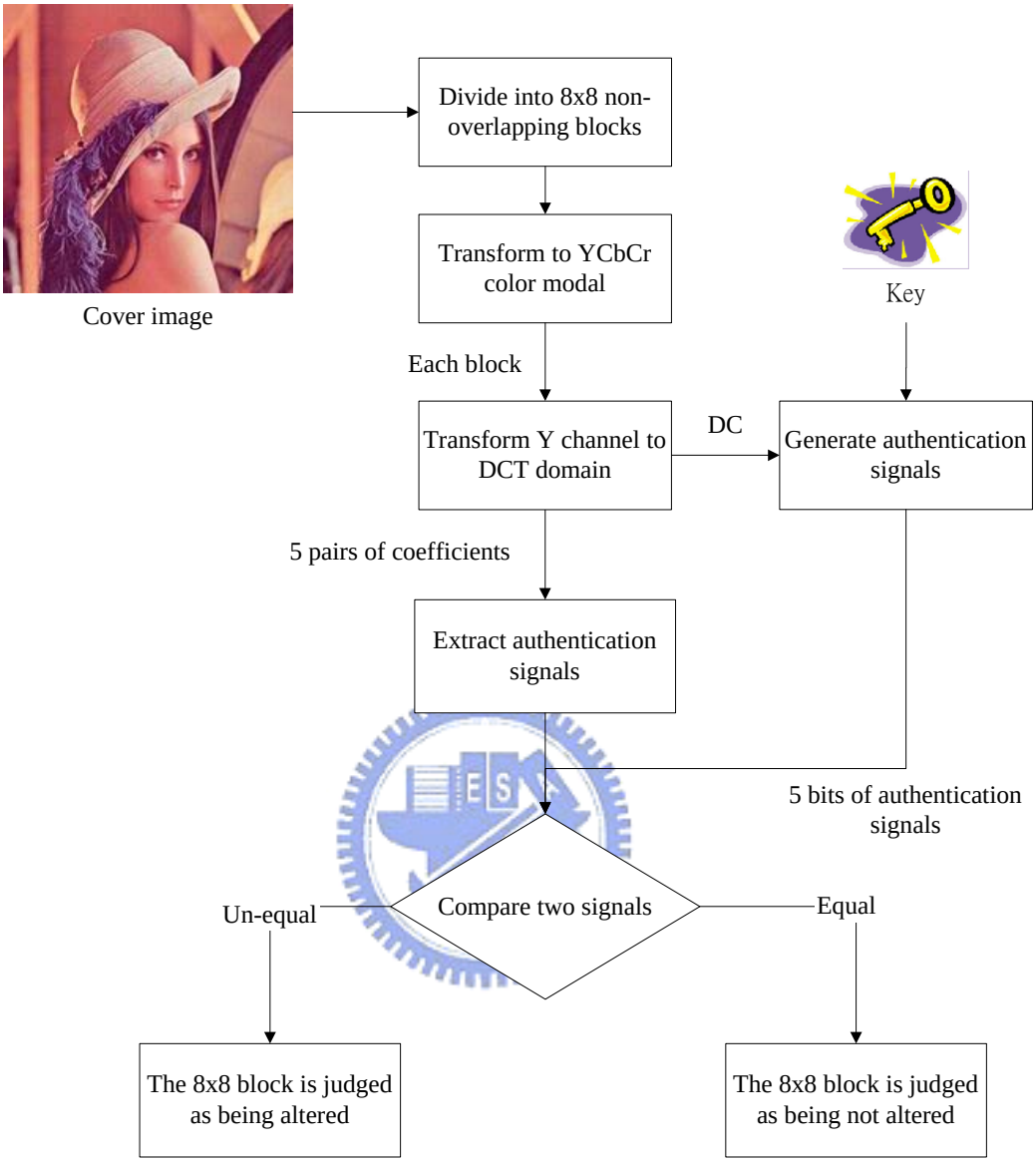


Figure 6.2 Flowchart of proposed image authentication process.

(a)

(d)

(b)

(e)

Figure 6.3 Input color images and output stego-images with authentication signals. (a) Color image “Lena”. (b) Color image “Plate”. (c) Color image “Jet”. (d), (e) and (f) Stego-images after embedding authentication signals, respectively.

(c) (f)

Figure 6.3 Input color images and output stego-images with authentication signals. (a) Color image “Lena”. (b) Color image “Plate”. (c) Color image “Jet”. (d), (e) and (f) Stego-images after embedding authentication signals, respectively (continued).

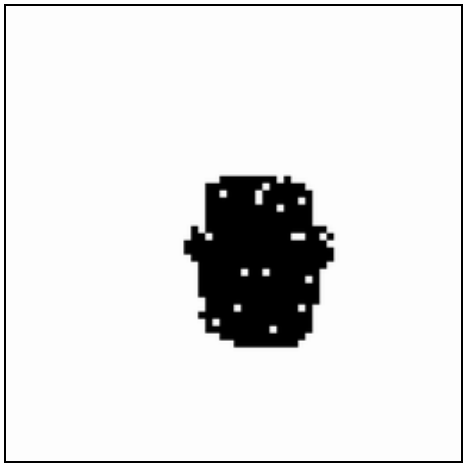
Table 6.1 The PSNR values of the stego-images after embedding the authentication signals.

	Lena	Plate	Jet
PSNR	31.1	30.5	30.5

Three tampered images are shown in Figures 6.4(a) through (c). And Figures 6.4(d), (e), and (f) show the respective authentication results. The black parts indicated the detected tampered areas. The authentication work can be seen to be successful.

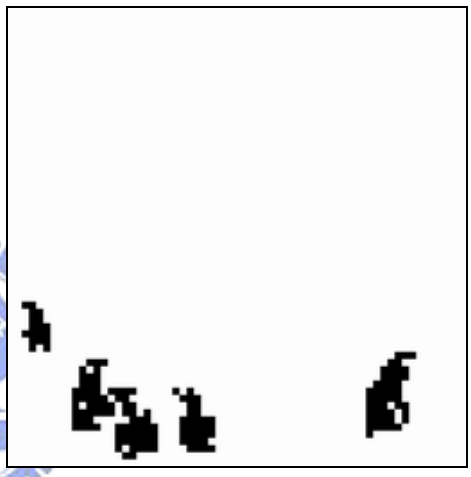
Finally, two compressed images are shown in Figures 6.5(a) and (b). And Figures 6.5(c) and (d) show the respective authentication results. There is no black part on the authentication results. The experimental results show that the embedded authentication signals have the semi-fragile property and can tolerate the JPEG compression operation.

(a)



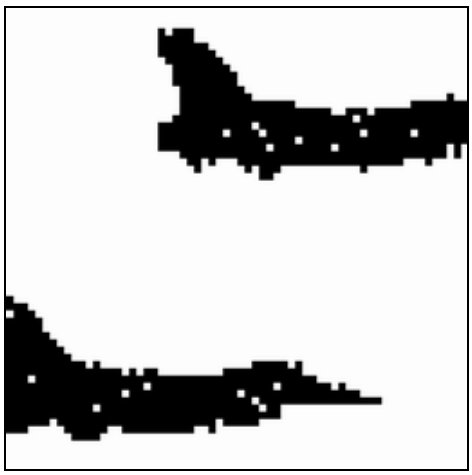
(d)

(b)



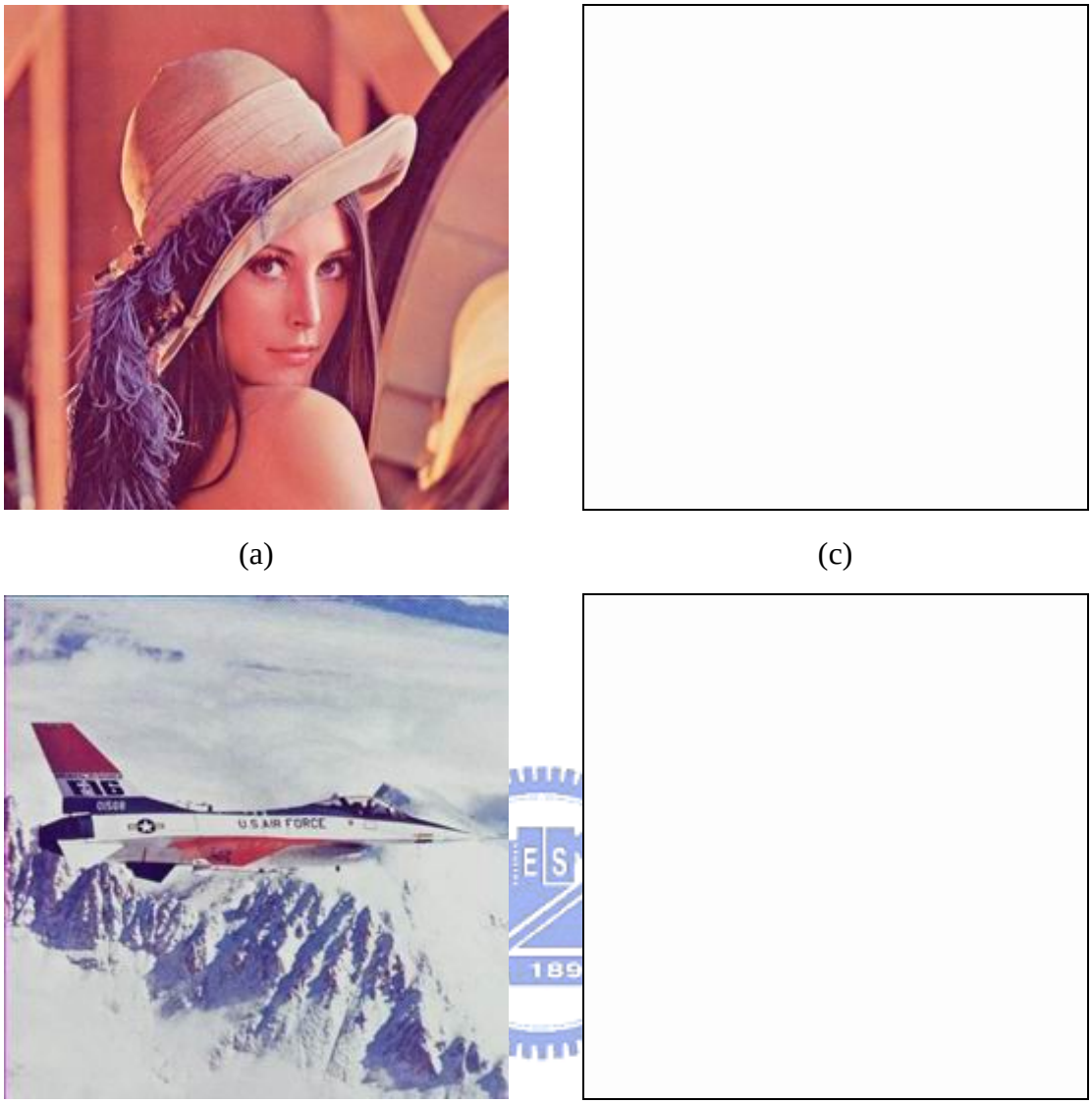
(e)

(c)



(f)

Figure 6.4 Some tampered images and authentication results. (a) Tampered image "Lena". (b) Tampered image "Plate". (d) Tampered image "Jet". (d) – (e) and (f) authentication results, respectively.



(a) (c)

(b) (d)

Figure 6.5 Some JPEG compressed images and authentication results. (a) Image "Lena" after JPEG compression with quality factor 80. (b) Image "Jet" after JPEG compression with quality factor 80. (c) and (d) authentication results, respectively.

6.4 Discussions and Summary

In this chapter, we have presented an authentication scheme to embed authentication signals in color images without the use of an extra signature file. We

use a key and the DC coefficient value to generate authentication signals, and embed them into an 8×8 block in the middle-high band of the DCT domain by adjusting the magnitude relations of pairs of the DCT coefficients. If somebody wants to tamper with the stego-image, the DCT-coefficient relationship and DC value will be changed. The generated authentication signals extracted from the tampered image will then not be the same as the extracted authentication signals. Therefore, the tampered areas can be detected and located.

In addition, the proposed method is provided with the semi-fragile property. It is common to save a digital image in the JPEG format. Therefore, JPEG compression should not be considered as a tampering operation, though it will degrade the quality of an image. The experimental results show that this concept is obeyed and no tampered areas are found after high-quality JPEG compression operations.

However, if the tampering operations are only applied on the C_b and C_r channels of a stego-image, the tampered areas cannot be detected by using the proposed method. It may be tried to solve this problem in the future.

Chapter 7
Conclusions and Suggestions for Future Works

7.1 Conclusions

In this study, we have proposed various methods based on information hiding techniques to solve two application problems, including copyright protection and tampering detection. For copyright protection, we have proposed four different methods to survive various attacks. First, a method for copyright protection by watermarking for color images against rotation and scaling attacks using coding and synchronization of peak locations in the DFT domain has been proposed, in which a serial number is embedded as a watermark into a cover image by adjusting the magnitude values of the coefficients in the middle band of the DFT domain to create peaks circularly and symmetrically and coding their locations in certain concentric circles. Furthermore, a peak for synchronization is also embedded in the middle band. Using coefficient-value peak detection in the DFT domain, the embedded watermark can be extracted from the stego-image. By this method, the embedded watermark becomes robust to survive rotation and scaling attacks.

Second, a method for copyright protection by watermarking for color images against rotation and cropping attacks using synchronization of peak locations in the DFT domain and DCT-coefficient relationship comparison has been proposed, in which two kinds of frequency domains are utilized. A binary image as a watermark

and verification codes are multiply duplicated and embedded into the 8×8 blocks of the cover image in the DCT domain by adjusting the magnitude relation of certain selected DCT coefficients. In addition, a synchronization peak is embedded into a cover image in the DFT domain. And with the detection of the synchronization later, the rotation of the stego-image can be checked. Finally, by checking the relationship of the DCT coefficients and the existence of the verification codes, the embedded watermark can be extracted from the DCT domain. Using this method, the embedded watermark becomes resistant to rotation and cropping attacks.

Third, a method for copyright protection by watermarking for color images against print and scan operations using coding and synchronization of peak locations in the DFT domain has been proposed, in which a watermark embedded in a rescanned image is provided with robustness against pixel-value distortion and geometric operation attacks. A watermark is embedded in certain concentric circles in the DFT domain by creating coefficient-value peaks in the middle band and using a combinatorial function to code the peak locations. In addition, an extra peak for synchronization is also embedded in the middle band. In the extraction process, the positions of the coefficient-value peaks are detected and mapped into a combinatorial function to get a watermark. Using this method, the watermark can survive print and scan operations.

Fourth, copyright protection by watermarking for color images against scaling and line-removal attacks using an image rescaling technique has been proposed. It is based on the works of Cheng and Tsai [23] and Yin and Tsai [5], which embed invisible watermarks and annotation information into images in the DCT domain. With the rescaling technique and insertion of certain verification codes, the embedded watermark can survive scaling and line-removal attacks.

Table 7.1 shows that the robustness of these four watermarking methods for

different kinds of attacks, where symbol "○" means that the watermarking methods have robustness against corresponding attacks, and vice versa.

Table 7.1 The robustness of the four watermarking methods for different kinds of attacks.

Robustness against attacks			Watermarking methods			
			First	Second	Third	Fourth
Signal enhancement	Gaussian Blur		X	○	○	○
	Median		X	○	X	○
	Sharpening		○	○	○	○
	JPEG	50	X	○	X	○
		60	X	○	X	○
		70	X	○	X	○
		80	X	○	X	○
		90	X	○	X	○
	GIF		X	○	X	○
Scaling		0.5	X	X	X	X
		0.7	X	X	X	○
		0.9	○	X	○	○
		1.5	○	X	○	○
		2	○	X	○	○
Cropping		25 %	X	○	X	X
		50 %	X	○	X	X
		75 %	X	○	X	X
Rotation		-10	○	○	○	X
		-5	○	○	○	X
		5	○	○	○	X
		10	○	○	○	X
Row /Column removal	Row	1	○	○	○	○
		2	X	○	○	○
	Column	1	○	○	○	○
		2	X	○	○	○
	Row/Column	1/1	○	○	○	○
		1/2	X	○	○	○
		2/1	X	○	○	○
		2/2	X	○	○	○

Finally, a method for image authentication in color images without the use of signatures has been proposed. Authentication signals, randomly generated by a key and the DC information of image blocks, are embedded into each 8×8 image block in the DCT domain by adjusting the magnitude relationship of the coefficients. There is no need to save an extra signature file when authenticating a suspicious image later. By checking the relationships of the selected DCT coefficients of the image block, the authentication signals can be extracted. With the help of the extracted authentication signals, tampered areas within a suspicious image can be detected and located, achieving the goal of verifying the integrity and fidelity of the image.

7.2 Suggestions for Future Works

Data hiding methods for copyright protection and tampering detection in digital images have been proposed in this study. However, some attractive and interesting topics that are related to this study are worth further woks. Several suggestions for future works are listed follows.

1. Extending the proposed methods to other kinds of digital media, such as video, and audio.
2. Increasing the capacity of data embedding.
3. Keeping the image quality good after embedding a large amount of data.
4. Authenticating the integrity of a stego image more precisely.
5. Resisting other attacks such as shearing.

References

[1] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," *Proceedings of IEEE International Conference on Image Processing*, Vol. 2, pp. 86-90, 1994.

[2] G. Voyatzis and I. Pitas, "Applications of total automorphisms in image watermarking," *Proceedings of IEEE International Conference on Image Processing (ICIP'96)*, Lausanne, Switzerland, Vol. 2, pp. 237-240, September 16-19, 1996.

[3] J. Fridrich, "Robust bit extraction form images," *Proceedings of IEEE ICMCS'99 Conference*, Florence, Italy, June 7-11, 1999.

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

[5] C. Y. Yin and W. H. Tsai, "Copyright and annotation protection in digital museums by using data hiding, watermarking, and image authentication techniques," *M. S. Thesis*, Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 2001.

[6] N. Chotikakamthorn and S. Pholsomboon, "Ring-shaped digital watermark for rotated and scaled images using random-phase sinusoidal function," *Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology*, Singapore, Vol. 1, pp. 321–325, August 19-22, 2001.

[7] 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.

[8] 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.

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

[10] S. D. lin and C. F. Chen, “A Robust DCT-Based Watermarking for copyright Protection,” *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 3, pp. 415-421, August 2000.

[11] J. J. K. O'Ruanaidh and T. Pun, “Rotation, scale and translation invariant digital image watermarking,” *Proceedings of IEEE International Conference on Image Processing*, Santa Barbara, CA USA, Vol. 1, pp. 536-539, Oct. 26-29, 1997.

[12] M. Wu and M. L. Miller, J. A. Bloom, and I. J. Cox, “A rotation, scale and translation resilient public watermark,” *Proceedings of 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, AZ USA, Vol. 4, pp. 2065, March 15-19, 1999.

[13] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y.M. Lui, “Rotation, scale, and translation resilient watermarking for images,” *IEEE Transactions on Image Processing*, Vol. 10, Issue 5, pp. 767-782, May 2001.

[14] D. J. Fleet and D. J. Heeger, “Embedding invisible information in color images,” *Proceedings of IEEE International Conference on Image Processing*, Santa Barbara, CA USA, Vol. 1, pp. 532-535, Oct. 26-29, 1997.

[15] V. Solachidis and L. Pitas, “Circularly symmetric watermark embedding in 2-D DFT domain,” *IEEE Transactions on Image Processing*, Vol. 10, Issue 11, pp. 1741-1753, Nov. 2001.

[16] P. C. Su and C. C. Kuo, “An Image Watermarking Scheme to Resist Generalized Geometrical Transformations,” *Proceedings of SPIE Conference on Multimedia*

Systems and Applications III, Boston, Massachusetts, Vol. 4209, pp. 354-365, November 5-8, 2000.

[17] F. Lefebvre, A. Gueluy, D. Delannay, and B. Macq, "A print and scan optimized watermarking scheme," *Proceedings of 2001 IEEE Fourth Workshop on Multimedia Signal Processing*, Cannes, France, pp. 511-516, Oct. 3-5, 2001.

[18] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proc. IEEE International Conference on Image Processing*, Vol. II, 1997, pp. 680-683.

[19] J. Fridrich, "Image watermarking for tamper detection," *Proceedings of IEEE International Conference on Image Processing*, Vol. 2, pp. 404-408, 1998.

[20] P. W. Wong, "A public key watermark for image verification and authentication," *Proceedings of IEEE International Conference on Image Processing*, Vol. 2, pp. 455-459, 1998.

[21] D. C. Wu and W. H. Tsai, "Embedding of any type of data in images based on a human visual model and multiple-based number conversion," *Pattern Recognition Letter*, Vol. 20, pp. 1511-1517, 1999.

[22] M. Wu and B. Liu, "Watermarking for image authentication," *Proceedings of IEEE International Conference on Image Processing*, Chicago, Illinois, Vol. 2, pp. 437-441, October 1998.

[23] Y. J. Cheng and W. H. Tsai, "Copyright and Integrity Protection for Images by Removable Visible Watermarking Techniques," *M. S. Thesis*, Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 2002.

[24] http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Properti-2.html, Properties of Fourier Transforms.

[25] A. Navarro and J. Tavares, "Joint Source-Channel PCM Image Coding for

Binary Symmetric Channels", *Proceeding of International Conference on Signal Processing Applications and Technology*, Orlando-USA, 1999.

[26] J. O'Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," *Proceeding of ICIP'96*, Lausanne, Switzerland, vol. 3, pp. 239–242, Sept. 1996.

[27] R.C. Gonzalez and R. E. Woods, "Digital Image Processing," second edition, pp. 155, 2002.

[28] C. Y. Lin and S. F. Chang, "Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process," *Proceeding of International Symposium on Multimedia Information Processing* (ISMIP), Taipei, Taiwan, Dec. 1999.