Copyright Protection by Watermarking for Color Images against Print-and-Scan Operations Using Coding and Synchronization of Peak Locations in Discrete Fourier Transform Domain*

Yen-Chung Chiu (邱彥中) and Wen-Hsiang Tsai (蔡文祥)

Department of Computer & Information Science
National Chiao Tung University
Hsinchu, Taiwan 300
E-mails: gis91528, whtsai@cis.ncu.edu.tw

Abstract

A watermarking method for copyright protection of color images against print-and-scan operations is proposed. A watermark is embedded in an input image as coefficient-value peaks circularly and symmetrically distributed in a middle band of the discrete Fourier transform (DFT) domain of the input image. By detecting the robust peaks in the DFT domain of a reproduced image resulting from scanning a printed version of a watermarked image, the embedded watermark can be extracted for copyright proof of the reproduced image. Experimental results are shown to prove the feasibility of the proposed method.

Keywords: digital watermarking, color image, copyright protection, print-and-scan operations, discrete Fourier transform, reproduced image.

1. Introduction

Because of the rapid development of electronic products, printers and scanners are commonly used for publications and reproductions of documents. Digital images can be printed to spread around. And when a printed image is scanned again, the resulting image, called *reproduced image* in this paper, becomes a digital version similar to the original one, though with some distortion sometimes. Such reproduced images might be misused against the copyright of the original digital image. It is desired to have a certain way to counteract such illegal print-and-scan operations, called *print-and-scan attacks* sometimes, on protected digital images.

Digital watermarking is a technique for embedding a watermark into a digital image to protect an owner's copyright of the image. The resulting watermarked digital image is called a *stego-image*. One way to solve the above-mentioned print-and-scan problem is to make the embedded watermark *robust* against print-and-scan operations, so that after applying these operations on a stego-image to yield a reproduced image, the watermark is not fully destroyed and can still be extracted from the reproduced image to verify the copyright of the image.

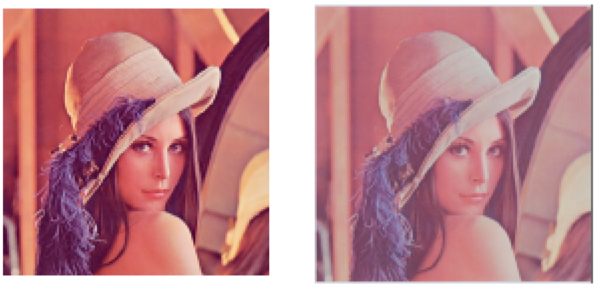
Some researches about watermarking techniques for copyright protection against print and scan attacks have been proposed in recent years. Fleet and Heeger [1] described a human color vision model to ensure that the embedded signal is invisible and proposed a method for embedding sinusoidal signals, which act as a grid and provide a coordinate frame on the image. In Solachidis and Pitas [2], a private key, which allows a very large number of possible watermarks, was proposed to determine a watermark, which was then embedded in a ring in the DFT domain. And the measure of correlation was used for watermark detection. Lefebvre et al. [3] proposed a method, which combines an additive watermarking algorithm in the spatial domain and a synchronization template in the Fourier domain. In Chotikakamthorne and Pholsomboon [4], 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 a reproduced image, there are two categories of distortions, namely, geometric transformations and pixel-value changes. The former category includes rotation, scaling, padding, etc., and the latter includes changes of pixel values in luminance, contrast, gamma correction, chrominance, blurring, etc. [5]. Geometric transformations do not cause significant effects on the visual quality but the pixel-value changes do, as seen in Fig. 1 for example.

A reproduced image in general has both pixel-value changes and geometric transformations. Therefore, a watermark embedded in a reproduced image must have a certain degree of robustness against attacks of pixel-value changes and geometric operations. In order to embed watermarks in color images to survive

* This work was supported by the NSC93-2752-E009-006-PAE

geometric operation attacks, invariant features of images with respect to geometric transformations should be adopted. And the embedded watermark must be imperceptible, of course. In this paper, we propose a robust method for embedding a watermark in an input image as coefficient-value peaks circularly and symmetrically distributed in a middle band of the discrete Fourier transform (DFT) domain of the input image. The peaks are found robust in this study in the DFT of a reproduced image, and can be extracted for copyright proof of the image. Experimental results are shown to prove the feasibility of the proposed method.



(a) (b)

Fig 1 A color image and a reproduced image with degraded quality. (a) Color image "Lena". (b) Reproduced image of (a) with quality of 100dpi.

The remainder of this paper is organized as follows. In Section 2, the ideas of the proposed method are described. In Section 3, the proposed watermark embedding process is presented. In Section 4, the proposed watermark extraction process is described. In Section 5, some experimental results are shown. Finally, some conclusions are made in Section 6.

2. Ideas of Proposed Method

2.1 Properties of DFT and Color Images

The DFT $F(u, v)$ of an input image $f(x, y)$ of size $M \times N$ can be described by:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2p(ux/M+vy/N)}. \quad (1)$$

This transform has several properties useful for this study. First, the transform has a symmetry property [6] shown by

$$F(u, v) = F^{*}(-u, -v), \quad (2)$$

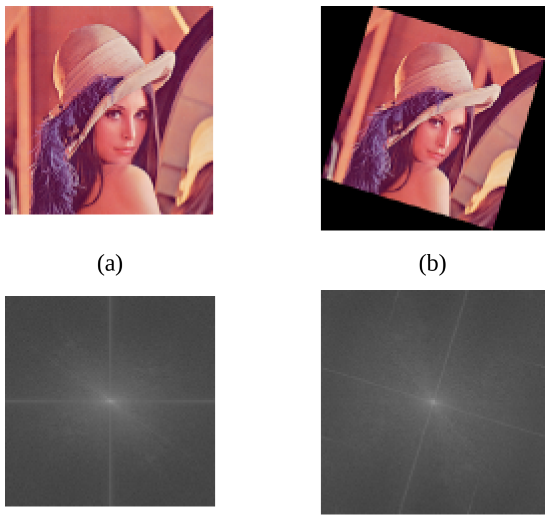
where the symbol F^{*} means the complex conjugate of F. Also, the complex transform $F(u, v)$ can be divided into two parts, the *magnitude function* (or called *spectrum*) $|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$ and the *phase function* $f(u, v) = \tan^{-1}[I(u, v)/R(u, v)]$, where $R(u, v)$ and $I(u, v)$ are the real and imaginary parts of $F(u, v)$, respectively. For real inputs like images, Eq. (2) leads to:

$$|F(u, v)| = |F(-u, -v)|, \quad (3)$$

which means that a coefficient value and its symmetric version in the DFT domain are equal in magnitude. Both the magnitude and the phase functions are required for reconstruction of an input image from its DFT. The magnitude function is less important than the phase function. The magnitude-only image is unrecognizable, while the phase-only image is barely recognizable [7]. Therefore, we may compute and adjust the magnitudes of the DFT coefficients to embed information without causing significant loss of the image quality, as is done in this study.

Furthermore, it is known [5] that the rescaling operation has almost no effect on the DFT coefficients, while image rotation in the spatial domain will cause the coefficient values to have the same rotation in the frequency domain. Figs. 2(a) and (b) show an image and a rotated version of it. And the corresponding *spectrum images*, in which each pixel value is taken to be the magnitude of a DFT coefficient, are shown in Figs. 2(c) and (d), respectively. Notice the same rotation of the spectrum image in 2(d) as that of the image in 2(b).

Finally, it is mentioned that although we can embed watermark information into all of the three color channels of an image, experiments shows that this work can only be conducted in the red and blue channels in the DFT domain because hiding information in the green channel is too sensitive to the human vision [8] and will create perceivable effects.



(a) (b) (c) (d)

Fig. 2 Input images and Fourier spectrums of G channel. (a) Image "Lena". (b) Image "Lena" after rotation. (c) Fourier spectrum of "Lena" (d) Fourier spectrum with the same rotation of (b).

2.2 Proposed Watermarking Technique Using Coefficient-value Peaks in DFT

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 $R_1 < R_2$, as shown in Fig. 3. Next, we divide B into n equally-spaced concentric circular stripes with outer radii $r_1, r_2, \cdots, r_n$, and each stripe into m angle ranges with starting angles $q_1, q_2, \cdots, q_m$, as seen in Fig. 4. Then, for watermark embedding we select $n\times m$ locations P = $\{p_1, p_2, \cdots, p_{n\times m}\}$, called *embeddable positions*, in the frequency domain with their coordinates described by

$$p_k = (u_k, v_k) = (r_i \cos q_j, r_i \sin q_j), \tag{4}$$

where $1 \le i \le n$, $1 \le j \le m$, and $1 \le k \le \ell$ with $\ell = n\times m$. 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:

$$M'(u_k, v_k) = M(u_k, v_k) + c \tag{5}$$

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

It is noted that, when changing the coefficient value to be a peak at each $p_k = (u_k, v_k)$ for the amount of c, we must preserve the *positive symmetry* property of the DFT [9] by changing the corresponding coefficient value at $p_k' = (-u_k, -v_k)$ for the same amount c. Otherwise, the peak created at p_k will be counteracted by the unchanged 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

$$M'(-u_k, -v_k) = M(-u_k, -v_k) + c \tag{6}$$

each time when we perform an operation of Eq. (5).

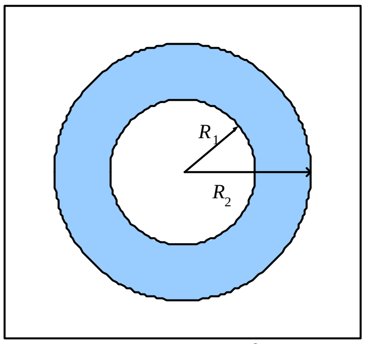


Fig. 3 A ring region in middle frequency band.

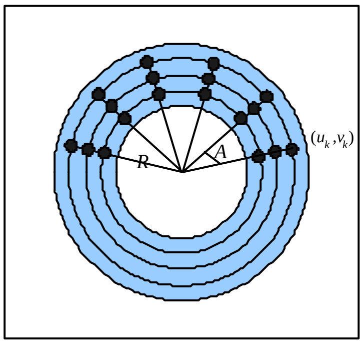


Fig. 4 The ring region in Fig, 3 is divided into concentric circular stripes and each stripe into angular sectors.

2.3 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_s, called *synchronization peak*, is created in the DFT domain to serve as a signal for *synchronizing* the peak locations P = $\{p_1, p_2, \ldots, p_{n\times m}\}$ mentioned previously in a way described later. P_s is embedded into the previously-mentioned middle frequency band B as well at a location p_s described by

$$p_s = (u_s, v_s) = (r_s \cos q_s, r_s \sin q_s) \tag{7}$$

where r_s is selected to be larger than R_2 (the outer radius of the band B) and q_s is a pre-selected angle value. We adjust the DCT value of P_s and that of its symmetric version to be peak values also by Eqs. (5) and (6).

We now describe how we use the synchronization peak P_s in the proposed watermark extraction process to calculate the rotation angle of a suspicious stego-image which suffered possibly from a rotation attack. Because of the DFT properties mentioned previously and illustrated by Fig. 2, if a stego-image is rotated, the location of P_s will also be changed with the same rotation angle. We may calculate first the new angle q_s' of P_s and take the difference Δq between q_s' and q_s to decide whether the stego-image has been rotated: if $\Delta q \neq 0$, then rotated; else, not. If rotated, then we find the angles q_k' of the other local peaks, and compute their original angles q_k'' by

$$q_k'' = q_k' - \Delta q. \tag{8}$$

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.

2.4. 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 ith iteration according to the following rule:

$$T_i = \begin{cases} T_{i-1} + d & \text{if } e_i > h, \\ T_{i-1} - d & \text{if } e_i < h, \end{cases} \quad (9)$$

where T_i is the value for T in the ith 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 d 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 d 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.

3. Watermark Embedding Process

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. A detailed algorithm of this process is described as follows.

Algorithm 1: *Watermark embedding process.*
Input: a 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 as a watermark W' 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 $\Theta = \{q_1, q_2, ..., q_m\}$ 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 ($r_i \cos q_j$, $r_i \sin q_j$) 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 Eqs. (5) and (6).
   - 3.6 Add a synchronization peak P_s according to the scheme described in Section 2.3.
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. Watermark Extraction Process

In the proposed watermark extraction process, no other information but a stego-image in suspicion is needed as the input. 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 2.1, 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 P = $\{p_1, p_2, \cdots, p_h\}$ by Eq. (8) to get their new locations 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 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:

for all k = 1, 2, …, ℓ and i = 1, 2, …, h,

$$\text{if } p_i' \text{ falls in } d_k, \text{ then regard } d_k \text{ to be in A.} \quad (10)$$

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* 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.

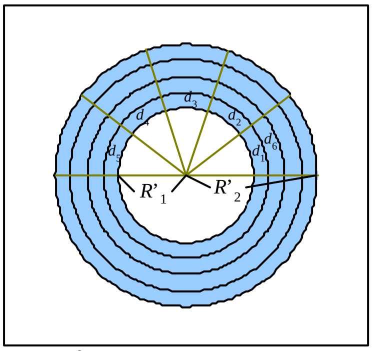


Fig. 5 The middle frequency band is divided into concentric sectors.

The detailed watermark extraction process can be expressed as an algorithm as follows

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 2.4.
   - 2.2 Select a peak with the longest radius to be the synchronization peak, and calculate its angle change Δq with respective to the original angle of the synchronization peak.
   - 2.3 Reconstruct the angles of the remaining h peaks by Eq. (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. (10), 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.

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. Fig. 6 shows an input image with size 512×512. And Fig. 7(a) shows the stego-image of Fig. 5 after embedding the watermark. In addition, Figs. 7(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. Fig. 7(d) shows that Fig. 7(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 Figs. 7(e) and (f), respectively. The embedded peaks can be successfully detected in our experiments.

Figs. 8(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 Figs. 8(c) and (d), respectively. The corresponding PSNR values are shown in Table 1, which show that the quality of each of the stego-images is still good. And the embedded watermark is imperceptible by human vision.

In addition, two reproduced images of Figs. 8(a) and (b) are shown in Figs. 9(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.

6. Conclusions

In this paper, 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 operation 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 operation 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 is not large for hiding data. It is not enough to embed a common logo image. In future works, it may be tried to solve this problem.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

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

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

[8] 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.

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

Fig. 6 An input image "Lena".

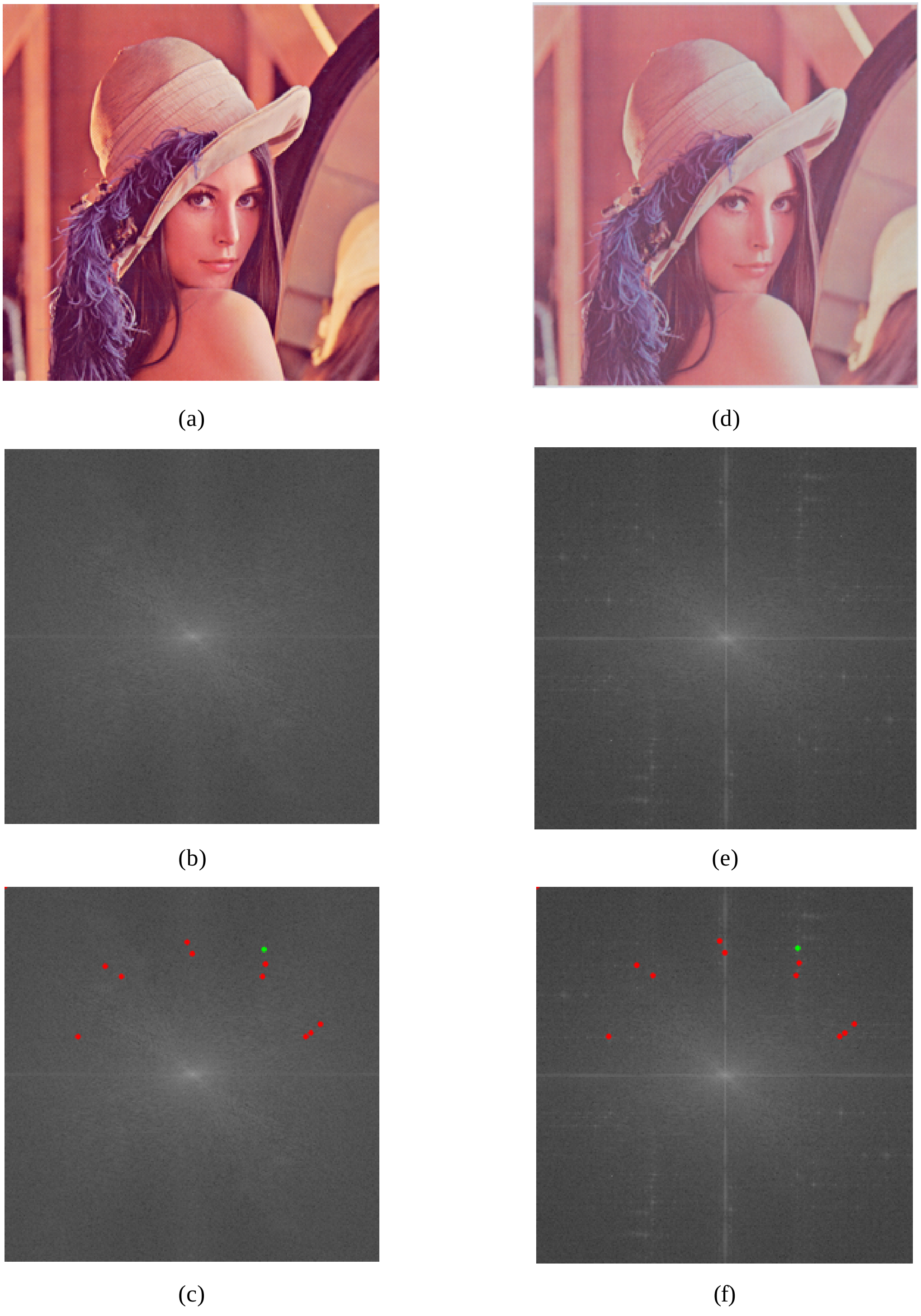


Fig. 7 An output stego-images with the watermark, the reproduced image and Fourier spectrum images. (a) Stego-Image “Lena”. (b) Fourier spectrum image of (a). (c) Peak locations of (c). (d) Reproduced image with the resolution of 100dpi. (e) Fourier spectrum image of (d). (f) Peak locations of (e).

(a)

(c)

(b)

(d)

Fig. 8 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 1 The PSNR values of recovered images after embedding watermarks.

	Lena	Pepper	Jet
PSNR	33.0	33.0	32.4

(a)

(b)

Fig. 9 Some reproduced images with different quality. (a) Reproduced image with the resolution of 100dpi. (b) Reproduced image with the resolution of 150dpi.