

## Research Article

# Wavelet-Based Color Pathological Image Watermark through Dynamically Adjusting the Embedding Intensity

Guoyan Liu,<sup>1</sup> Hongjun Liu,<sup>2</sup> and Abdurahman Kadir<sup>3</sup>

<sup>1</sup>Department of Dermatology, Affiliated Hospital of Weifang Medical University, Weifang 261031, China

<sup>2</sup>Department of Information Engineering, Weifang Vocational College, Weifang 261041, China

<sup>3</sup>School of Computer Science and Engineering, Xinjiang University of Finance and Economics, Urumqi 830012, China

Correspondence should be addressed to Hongjun Liu, smithliu@126.com

Received 24 September 2012; Revised 20 October 2012; Accepted 22 October 2012

Academic Editor: Kumar Durai

Copyright © 2012 Guoyan Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a new dynamic and robust blind watermarking scheme for color pathological image based on discrete wavelet transform (DWT). The binary watermark image is preprocessed before embedding; firstly it is scrambled by Arnold cat map and then encrypted by pseudorandom sequence generated by robust chaotic map. The host image is divided into  $n \times n$  blocks, and the encrypted watermark is embedded into the higher frequency domain of blue component. The mean and variance of the subbands are calculated, to dynamically modify the wavelet coefficient of a block according to the embedded 0 or 1, so as to generate the detection threshold. We research the relationship between embedding intensity and threshold and give the effective range of the threshold to extract the watermark. Experimental results show that the scheme can resist against common distortions, especially getting advantage over JPEG compression, additive noise, brightening, rotation, and cropping.

## 1. Introduction

For content owners and distributors, there emerged a necessary concern in regard to the content authentication of pathological images as well as copyright protection. A latent solution to this issue is bestowed by digital watermarking. In general, image watermarking can be divided into two categories, according to the processing domain of cover images where the watermark is embedded: (1) the spatial domain method, which directly modifies the intensity value of the image and these algorithms are simple and speedy but not robust; (2) the frequency domain method, which is to modify the frequency coefficients.

In recent years, chaos is employed to encrypt the image, for it has sensitive dependence on initial conditions and can be employed to generate pseudorandom sequences, so the algorithm has large key space. Shyamsunder and Kaliyaperumal [1] proposed an image encryption scheme, which incorporates the concept of modular arithmetic and chaos theory. A necessary random matrix is generated for image encryption, and the look-up table is used to find

the element by modular inverse of the random matrix for decryption. Rawat and Raman [2] proposed a chaos-based watermarking scheme for image authentication and tamper detection. Their scheme can detect any modification made to the image and can also indicate the specific locations that have been modified. To improve the security of the proposed scheme, two chaotic maps are employed. Li et al. [3] used logistic system and Chebyshev maps to construct a hybrid chaotic mapping system, the aim is to set risk transfer, process and improve risk management efficiency in projects management. That is a good case for applying chaos in enterprise management.

The DWT approach remains one of the most effective ways for image watermarking. Lin et al. [4] proposed a blind watermarking algorithm based on maximum wavelet coefficient quantization; the blocks are randomly selected from different subbands, by adding different energies to the maximum wavelet coefficient to embed the watermark. The wavelet functions will analyze image features such as edges and borders through good space-frequency localization. They are used in several fields: image compression, signal

denoising, image smoothing, and texture analysis. The main advantages of embedding watermarks in the DWT can be found in [5, 6].

The most important issue in DWT-based image watermarking is how to choose the effective coefficients to be embedded and extracted. Liu [7] estimates the noise detection threshold of each wavelet coefficient in luminance and chrominance components of color image. The thresholds are derived into a locally dynamic fashion based on the wavelet decomposition, through which the perceptually significant coefficients are selected for embedding watermark. Al-Otum and Samara [8] proposed a watermark scheme based on the wavelet trees, which exploits the significant features and relations between the color pixel components in the wavelet domain. The watermark is embedded by spreading it that the interpixel robust relations carry the watermark bit sign with sufficient energy. Zhang et al. [9] proposed an adaptive block-based for embedding binary watermark into grayscale image. In these articles, the watermark bits are only shuffled by pseudorandom sequence, they are not scrambled to uniformly, and randomly distributed in the host image, which will lead to be attacked easily. The threshold is a fixed experimental value, and the authors did not explain how to get the threshold.

Hornig et al. [10] proposed a blind watermarking algorithm based on the significant difference of wavelet coefficient quantization. The maximum wavelet coefficients are quantized that their significant difference between watermark bit 0 and 1 exhibits a large energy difference, which can be used for watermark extraction. An adaptive threshold is designed to extract the watermark.

In this paper, we propose a dynamic blind watermarking scheme for color pathological image based on DWT; the watermark is scrambled firstly by Arnold cat map and XOR with pseudorandom sequence generated by Chebyshev map. The host image is divided into  $n \times n$  blocks, and each bit of the encrypted watermark is embedded into the detail wavelet sub-band, that is, the higher-frequency domain of blue component. We modify the wavelet coefficient dynamically according to the mean and variance of the subbands to embed "0" or "1." In addition, we research the relationship between the embedding intensity and the threshold and deduce the range of the threshold, by which to extract the watermark correctly.

## 2. Watermarking Algorithm

*2.1. Preprocessing the Watermark Image.* The watermark used for embedding is a binary logo image, which is very small in size compared to the host image. The watermark needs to be very small so that it is spatially localized and becomes robust against the intentional and unintentional attacks.

For a binary logo image, in order to uniformly and randomly spread the bits over the host image, it is needed to be preprocessed, or the watermarked image cannot resist against even the simplest attack: cropping. The watermark can be preprocessed in numerous ways, such as randomly select the embedding position, generate pseudorandom sequences

to shuffle these bits [11], or the original watermark itself is dynamically generated from a pseudorandom Gaussian sequence [12, 13].

Here we take two measures to preprocess the watermark bits. Firstly, in order to make the watermarked image be more robust to resist against cropping attack, the two-dimensional Arnold cat map [14] is employed to scramble these binary bits. Suppose the watermark is an  $n \times n$  binary image  $W$ ; after scrambling, we get the binary image  $W_A$ ; the aim is to uniformly spread the watermark bits in the host image. By this means, the watermarked image can be more robust against cropping than the method in [7]. The iteration times  $t$  is served as one of the keys. The Arnold cat map can be described as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod n. \quad (1)$$

Secondly, watermarks generated from low-pass chaotic signals have superior performance over other signal types [13]. In order to eliminate the statistical significance and make the numbers of 0 and 1 approximately equal, a pseudorandom sequence  $S_n \in \{0, 1\}$  will be generated by the Chebyshev map.

The expression of Chebyshev map is as follows:

$$z_{i+1} = \cos(w(\arccos z_i)), \quad -1 \leq z_i \leq 1, \quad (2)$$

where  $w$  is the degree of Chebyshev map. Its corresponding invariant density is as follows:

$$\rho(z) = \frac{1}{(\pi\sqrt{1-z^2})}. \quad (3)$$

Chebyshev map has important properties of excellent cryptosystem [14, 15]. While  $w \geq 2$ , the Lyapunov exponent of the Chebyshev map is positive, which predicates that Chebyshev map is chaotic, as shown in Figure 1.

Different sequences can be generated with different initial values; we get the sequence  $S_n \in \{0, 1\}$  by

$$S_n = \begin{cases} 0, & z_n \in [-1, 0), \\ 1, & z_n \in [0, 1]. \end{cases} \quad (4)$$

Finally, we can get the binary image  $W_{AX}$  by the XOR operation:

$$W_{AX} = W_A \oplus S_n, \quad (5)$$

which is a binary sequence to be embedded into the host image.

The preprocessing process, including the original image  $W$ , the scrambled image  $W_A$ , and the image  $W_{AX}$ , is shown in Figure 2.

*2.2. Analysis of Wavelet Coefficients.* The host image  $I$  of size  $w \times h$  is transformed into wavelet coefficients using the  $L$ -level discrete wavelet transform (DWT). With  $L$ -level decomposition, we can get  $3 \times l + 1$  subbands, as shown in Figure 3. The lowest band (approximation band)  $LL_l$  is

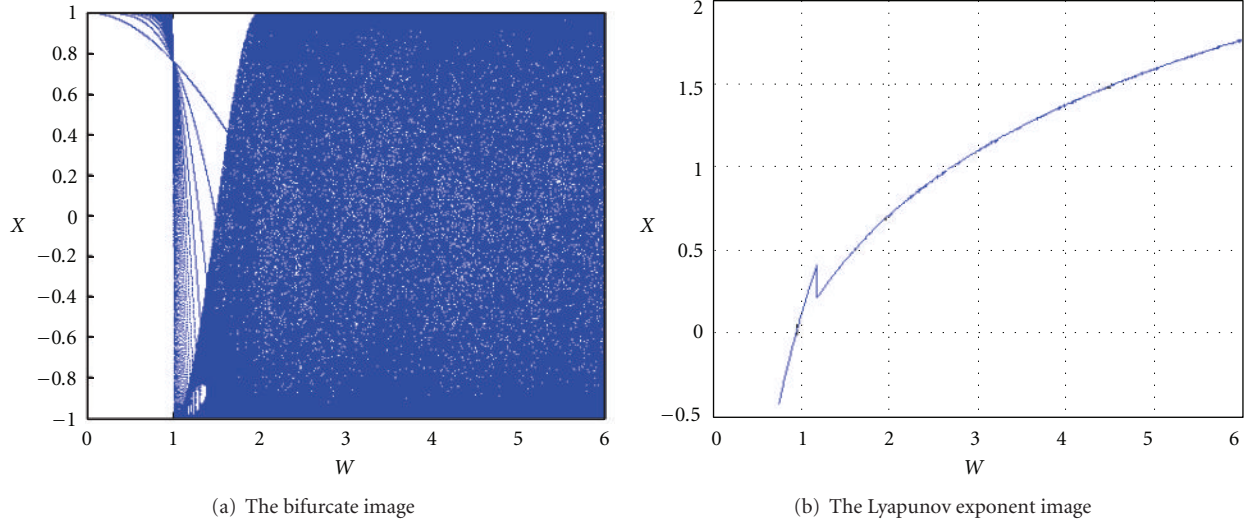


FIGURE 1: The bifurcate and Lyapunov exponent figures of Chebyshev map.

the basic band of the decomposed wavelet, which includes most of the energy from the original image; it has a crucial effect on quality; therefore,  $LL_l$  frequency band is unsuitable to be modified.

Only embedding the watermark into  $HH_l$ ,  $HH_{l-1}, \dots$ ,  $HH_1$  subbands is also unsuitable, for they have the highest wavelet coefficients; the subbands can easily be eliminated and modified by lossy compression or other processing [14].

According to these characteristics, we design to adaptively adjusting, the high-frequency coefficient of the subbands  $HL_l$ ,  $LH_l$ , and  $HH_l$ , to embed the watermark bits.

Generally, watermark embedding is realized by modifying some special values of pixels or transforming domain coefficients [16]. When the watermarked image receives some attacks, though the quality is still high, the pixel value or coefficient may be seriously eliminated, which will lead to the watermark detecting failed [17, 18]. Here we take advantage of the statistic characteristic, such as mean and variance of the  $LH_l$ ,  $HL_l$ , and  $HH_l$  subbands, to modify the whole coefficients of them to dynamically embed the watermark and implement the compromise between quality and robustness.

**2.3. Watermark Embedding.** After preprocessing the binary watermark image  $W$  to get  $W_{AX}$ , we reshape  $W_{AX}$  to binary sequence  $wm$ ; its length is still  $n \times n$ . According to Human Visual System (HVS) [19], small changes to the blue component of color image are the most difficult to detect by human eyes; then we select the blue component to embed.

Here we improve the method in [9] to embed pre-processed watermark into the blue component of color image. Suppose the host color image is  $I$ . Firstly, we divide the blue component of  $I$  into a set of nonoverlapping  $n \times n$  subblocks  $b_j \in B$ ,  $j = 1, 2, \dots, n^2$ . Then a  $l$ -level DWT decomposition of each sub-block of image  $I$  is performed

using Haar wavelet, and then compute the mean  $E_j$  of each wavelet block  $S_D$  by

$$E_j = \frac{1}{M} \left( \sum_{(x,y) \in S_D} c_j^{(l,s)}(x,y) \right), \quad (6)$$

where  $c_j^{(l,s)}(x,y)$  is the wavelet coefficient of the  $j$ th sub-block,  $l$  is the level of wavelet decomposition,  $s$  refers to the three subbands of  $LH_l$ ,  $HL_l$  and  $HH_l$ .  $(x,y)$  is the coordinate for wavelet coefficient in the wavelet blocks, and  $M$  is the total number of wavelet coefficient in  $S_D$ .

For example, here we set  $l = 2$ , for the  $j$ th sub-block  $b(j) \in B$ ; we use the  $dwt2()$  function in Matlab to perform a single-level two-dimensional wavelet decomposition. Finally,  $c_j^{(2,s)}(x,y)$  can be gotten by (9)

$$\left[ LL_{j1}, LH_{j1}, HL_{j1}, HH_{j1} \right] = dwt2(b(j), 'haar'), \quad (7)$$

$$\left[ LL_{j2}, LH_{j2}, HL_{j2}, HH_{j2} \right] = dwt2(LL_{j1}, 'haar'), \quad (8)$$

$$c_j^{(2,s)}(x,y) = \left[ LH_{j2}, HL_{j2}, HH_{j2} \right]. \quad (9)$$

The purpose of  $E_j$  is to keep the average of  $c_j^{(l,s)}(x,y) - \lambda E_j$  in (10) and (11) close to zero.

If  $wm(j) = 1$ , we modify the whole wavelet coefficients in  $S_D$  by

$$c_j'^{(l,s)}(x,y) = c_j^{(l,s)}(x,y) - \lambda E_j. \quad (10)$$

If  $wm(j) = 0$ , we modify the whole wavelet coefficients in  $S_D$  by

$$c_j'^{(l,s)}(x,y) = c_j^{(l,s)}(x,y) - \lambda E_j + P. \quad (11)$$

Here  $c_j'^{(l,s)}(x,y)$  is the modified wavelet coefficient; the distribution of wavelet coefficient is modulated by  $wm(j)$ .

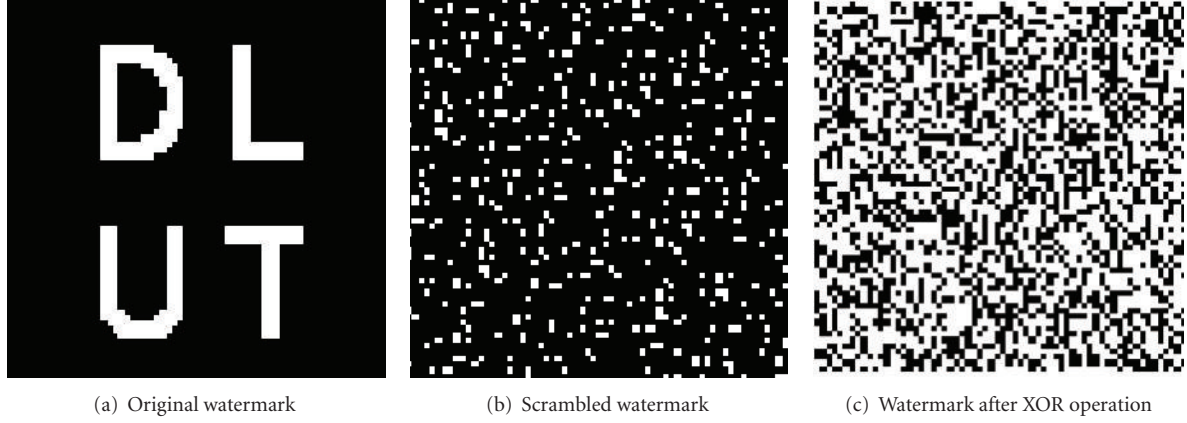


FIGURE 2: The preprocessing process of the original watermark.

LL3	LH3	LH2	LH1
HL3	HH3		
HL2		HH2	
HL1		HH1	

FIGURE 3: Wavelet subbands of 3-level.

$s$  refers to the three detail subbands of  $LH_i$ ,  $HL_i$ , and  $HH_i$ . That is to say, embedding “0” means that the coefficient mean of wavelet block  $S_D$  is  $P$ , and embedding “1” means 0.  $\lambda$  refers to the magnification of  $E_j$ ; here we set  $\lambda = 1.5$  to ensure the range of threshold  $Th$  larger enough to extract the watermark.

Because the variance is bigger in the textures and edges than that in the smooth region, we use the variance of the coefficient as the modulation coefficient  $P$  to control the dynamically embedding intensity [19]. Raising embedding intensity within the definition can enhance the robustness [15]. So  $P$  can be defined by

$$P = A + T(j). \quad (12)$$

$T(j)$  is defined as

$$T(j) = \left( \sqrt{\left( \sum_{(x,y) \in S_D} (c_j^{(l,a)}(x,y) - c_j^{(l,b)}(x,y)) - \lambda E_j \right)^2} \right)^\beta. \quad (13)$$

We can find that  $T(j)$  is the local variance of the  $j$ th wavelet sub-block except the approximate sub-band  $LL_i$ , and  $\beta$  is a constant. In (12),  $A$  is the intensity coefficient, which ensures to embed some watermark in the smooth region while  $T(j)$  is approximately 0. So (11) can be fully expressed by (14).

$$c_j'^{(l,s)}(x,y) = c_j^{(l,s)}(x,y) - \lambda E_j + A + \left( \sqrt{\left( \sum_{(x,y) \in S_D} (c_j^{(l,a)}(x,y) - c_j^{(l,b)}(x,y)) - \lambda E_j \right)^2} \right)^\beta. \quad (14)$$

Using (10) and (14) we embed watermark by modify the wavelet coefficients. Finally, the IDWT is applied to each block, to get the watermarked image  $I_w$ .

**2.4. Watermark Extraction.** The watermark can be extracted correctly from the watermarked image, without the original host image or the watermark image, so the scheme belongs to blind watermark algorithm. The transformed coefficients are compared with the threshold  $Th$  of the coefficients, those coefficients above the threshold are retained, and all the others are discarded [20].

The extracting process is as follows, which is mirroring the embedding process.

*Step 1.* We divide the blue component of the watermarked color image  $I$  into a set of nonoverlapping  $n \times n$  subblocks.

*Step 2.* To extract the embedded bit from a sub-block, the  $L$ -level DWT decomposition of each block is firstly obtained. Then we compute  $\text{Sum}(j)$ , the coefficients sum of the wavelet block  $S_D$  using

$$\text{Sum}(j) = \sum_{(x,y) \in S_D} c_j'^{(l,s)}(x,y). \quad (15)$$

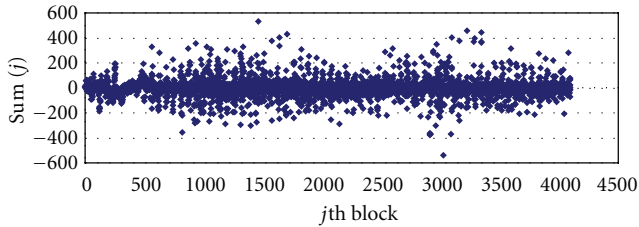


FIGURE 4: Distribution:  $\text{Sum}(j)$  of the wavelet blocks  $S_D$  before embedding.

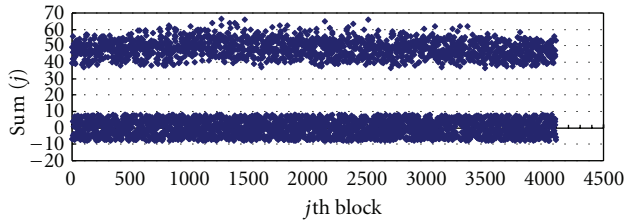


FIGURE 5: Distribution:  $\text{Sum}(j)$  of  $S_D$  after embedding ( $\beta = 0.218$ ).

For all the wavelet blocks of  $j = 1, 2, \dots, n^2$ ,

$$\begin{aligned} \text{Sum}(j) &\in [r1, r2], & \text{if } S_D \text{ carries bit "1"}, \\ \text{Sum}(j) &\in [r3, r4], & \text{if } S_D \text{ carries bit "0"}, \end{aligned} \quad (16)$$

where  $r1 < r2 < r3 < r4$ , then we can get the range of  $\text{Th} \in (r2, r3)$ .

*Step 3.*  $\text{Sum}(j)$  is compared with the threshold  $\text{Th}$ , to decide whether the coefficient carries a bit "1" or "0" using

$$wm(j) = \begin{cases} 1 & \text{Sum}(j) < \text{Th}, \\ 0 & \text{Sum}(j) \geq \text{Th}, \end{cases} \quad (17)$$

where  $wm(j)$  is the extracted bit.

*Step 4.* Finally, the extracted watermark bits are recovered into a binary image according to the inverse process.

During the embedding and extracting processes, the iteration times  $t$  of Arnold transform, the initial value of Chebyshev map, the size  $n \times n$  of each block, the values of  $A$ ,  $\lambda$ , and  $\beta$ , the threshold  $\text{Th}$ , and the type of wavelet are all served as keys, which can guarantee the security of the watermarked image; it is impossible to extract the watermark without these keys.

### 3. Experimental Result Analysis

*3.1. Get the Effective Range of the Threshold in Experiment.* The threshold is a critical factor, which can ensure to extract the watermark correctly from watermarked image under some attacks, and some papers only provide a fixed value [3, 7, 8]. The proposed scheme provides an exact range of the threshold by testing dozens of images, and it keeps constant even if the watermarked image has been attacked in some degree.

Figure 4 is Image1 distribution of  $\text{Sum}(j)$  before embedding; we can find that most of the coefficients in the detail

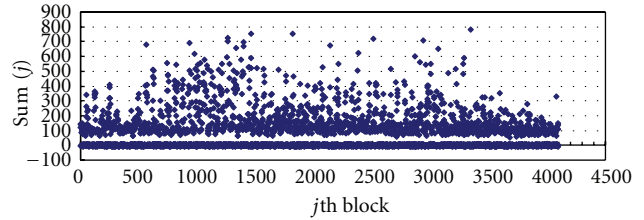


FIGURE 6: Distribution:  $\text{Sum}(j)$  of  $S_D$  after embedding ( $\beta = 0.818$ ).

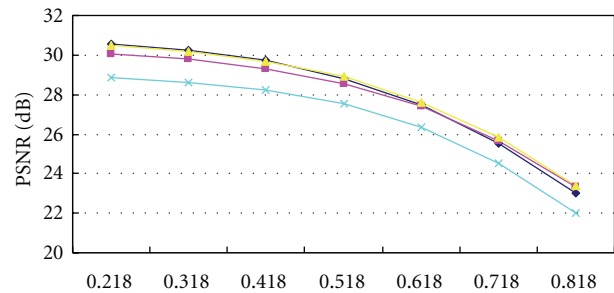


FIGURE 7: The relationship between  $\beta$  and PSNR (dB).

subbands after DWT are close to zero, and the mean of each detail sub-band is approximately zero.

Figures 5 and 6 are the Image1 distribution of  $\text{Sum}(j)$  after embedding with  $\beta = 0.218$  and  $\beta = 0.818$ ; from them we can find that when  $\beta = 0.218$ , the distribution of  $\text{Sum}(j)$  is relatively concentrated, but when  $\beta = 0.818$ , the values of  $\text{Sum}(j)$  distribute in larger range. The distribution of other images, such as Image3 and Airplane, is similar to Image1 when we set  $\beta = 0.218$ .

Experiment results demonstrate that when we set  $\beta \in [0.218, 0.818]$ , the threshold  $\text{Th} \in (10, 35)$ , which is consistent with the result of (16), the range of  $\text{Th}$  watermark can ensure the watermark to be extracted clearly from the watermarked images.

*3.2. The Relationship between  $\beta$  and PSNR.* The quality of watermarked image is based on the PSNR [12]. From Figure 7 we can find that the PSNRs decrease smoothly with the increase of  $\beta \in [0.218, 0.818]$ , which can increase the embedding intensity  $P$ . What's more, Figure 7 shows that the PSNR values of Image1, Image2, and Image3 are similar, and they are greater than that of Image4, for they have richer edges and borders than Image4, as shown in Figure 8.

### 4. Experimental Results and Key Space

*4.1. Experimental Result.* Here we made a  $64 \times 64$  binary image as the watermark. Some standard color images with the size of  $512 \times 512 \times 24$  bits, including Image1, Airplane, and Image3, are being tested. The size of the sub-block is set to  $8 \times 8$ ,  $\lambda = 1.5$ ,  $l = 2$ , when  $A \in [4, 22]$ ,  $\beta \in [0.218, 0.818]$ , and  $\text{Th} \in (10, 35)$ ; the results of watermarked images are satisfactory, as shown in Figure 9.

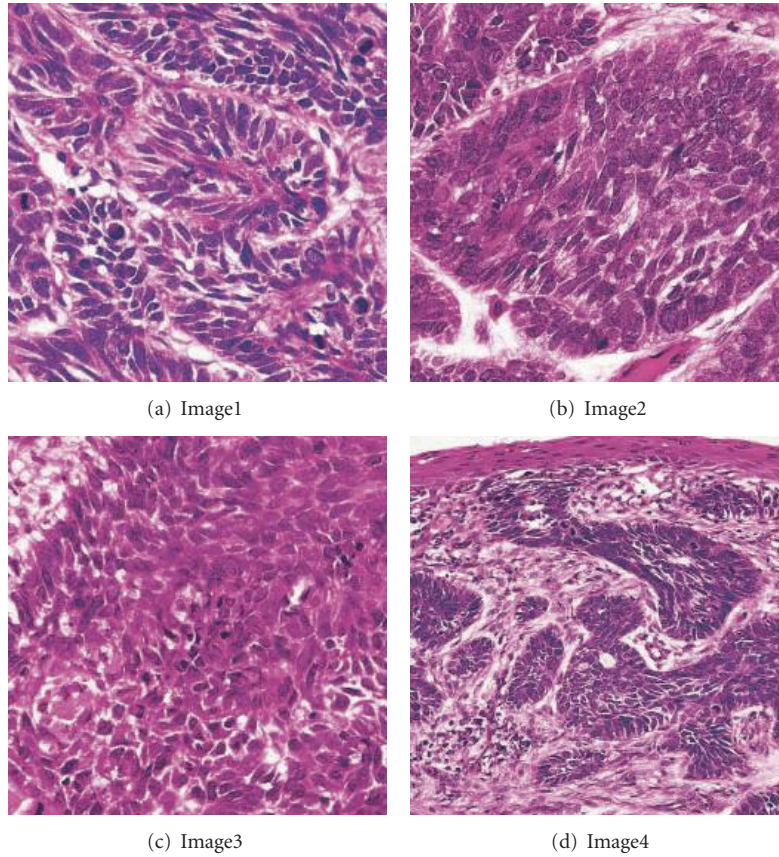


FIGURE 8: Sample images.

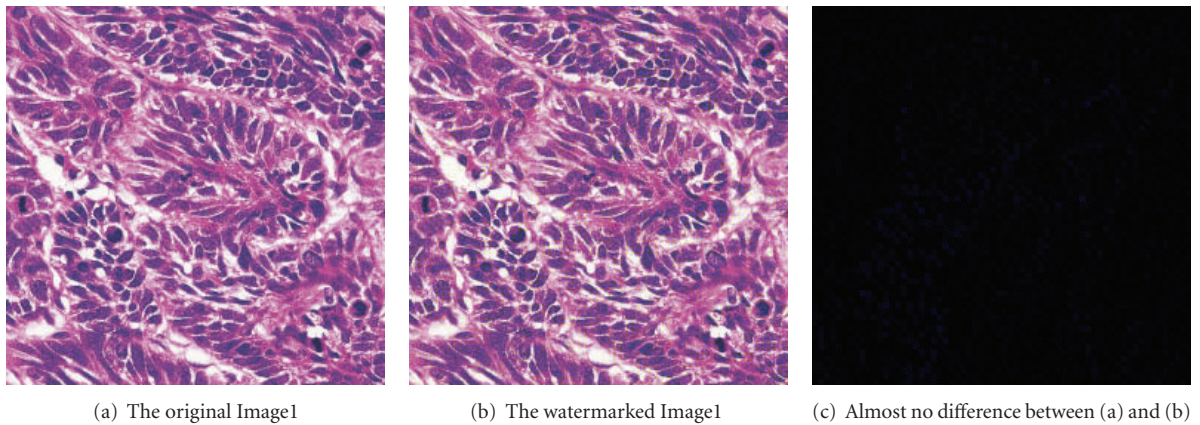


FIGURE 9: The original and the watermarked Image1, and the difference between them.

**4.2. Key Space Analysis.** The high sensitivity to initial conditions is inherent to any chaotic system. To provide an secure algorithm, the key space should be large enough to make any brute force attack ineffective. Our scheme has some of the following secret keys: (1) for the Arnold cat map, the initial values of  $x_0$  and  $y_0$ , the iteration times  $t$ ; (2) Chebyshev maps, the initial value  $z_0$ , and the parameter  $w$ .

For the Arnold cat map,  $x_0, y_0 \in [1, n]$  and  $t \in [1, 256]$ , here  $n = 64$ . For the Chebyshev maps, when the tiny

change in the initial value  $\Delta z_0 = 10^{-16}$ , the scrambled watermark image is completely different. A large number of experimental results indicate that the key spaces for initial values are  $S_{z_0} = 10^{16}$ . Similarly, the variation of the parameter  $w$  in the chaotic region is between 2 and 6 with a step of  $10^{-16}$ , so  $S_w \approx 4 \times 10^{16}$ .

Finally, the total key space is  $S = S_{x_0} \times S_{y_0} \times S_t \times S_{z_0} \times S_w \approx 4.19 \times 10^{38}$ , which is larger than  $2^{100}$ , so even the scrambled watermark is extracted; it's difficult to recover it.

TABLE 1: Noise attacking results for Image1.


























Density	$d = 0.001$	$d = 0.005$	$d = 0.01$	$d = 0.02$	$d = 0.05$
Salt and pepper					
NC	0.9585	0.8520	0.7462	0.6488	0.5309
Mean and variance	$m = 0, v = 0.0005$	$m = 0, v = 0.001$	$m = 0, v = 0.002$	$m = 0, v = 0.005$	$m = 0, v = 0.01$
Gaussian					
NC	0.8503	0.7124	0.6258	0.5615	0.5137

TABLE 2: Compression results with different ratios.

$A = 20, \beta = 0.128$	0.80	0.65	0.50	0.25	0.10
Image1					
NC	0.8071	0.7325	0.7143	0.5497	0.4449
Image2					
NC	0.7765	0.6970	0.6603	0.5368	0.4186
Image3					
NC	0.6363	0.5789	0.5621	0.5100	0.5037

## 5. Attack Test Results

The attacks to the whole watermarked image can modify the red, green, and blue components simultaneously. The modifications in the red or green component are easier to be observed, that is only because of the Human Visual System. There is no evidence to show that the modification in the blue component can be more robust than the other components.

The peak signal-to-noise ratio (PSNR) is used to evaluate the quality between an attacked image and the original image. For the sake of completeness, we list the PSNR formula as follows:

$$\text{PSNR} = 10 \times \log_{10} \frac{255 \times 255}{1/(I_h \times I_w) \sum_{x=0}^{I_h-1} \sum_{y=0}^{I_w-1} [f(x, y) - g(x, y)]^2} \text{dB}, \quad (18)$$

where  $I_h$  and  $I_w$  are the height and width of the image, respectively.  $f(x, y)$  and  $g(x, y)$  are the values located at coordinates  $(x, y)$  of the original image and the attacked image, respectively.

After extracting the watermark, the normalized correlation coefficient (NC) is computed using the original watermark and the extracted watermark to judge the existence of the watermark and to measure the correctness of an extracted watermark. It can be defined as

$$\text{NC} = \frac{1}{W_h \times W_w} \sum_{i=0}^{W_h-1} \sum_{j=0}^{W_w-1} W(i, j) \times W'(i, j), \quad (19)$$

where  $W_h$  and  $W_w$  are the height and width of the watermark, respectively.  $W(i, j)$  and  $W'(i, j)$  are the watermark bits located at coordinates  $(i, j)$  of the original watermark and the extracted watermark.

*5.1. Attack of Noise.* Noise attack is a common attack during the transmission of the watermarked image on the network

TABLE 3: Extracting results for Image1 and Image2 after being cropped.

Cropping	5%	10%	25%	50%	75%
Image1 Watermark extracted					
NC	0.9128	0.8245	0.6617	0.4701	0.3470
Image2 Watermark extracted					
NC	0.8911	0.8245	0.6320	0.4622	0.3332

TABLE 4: Randomly select the cropping position and the extracting result.

$A = 20, \beta = 0.128$	Randomly cropped	Quadrately cropped	Ambiently cropped
Cropped watermarked image			
Extracted watermark before reverse transformation			
Extracted watermark after reverse transformation			

TABLE 5: Extracting results from the image with enhanced and decreased brightness.

Enhanced brightness	10%	30%	50%
Extracted watermark			
Decreased brightness	-10%	-30%	-50%
Extracted watermark			

set to 0.05, the NC is 0.5309, and the extracted watermark can still be distinguishable. Similarly, for the watermarked image attacked by Gaussian white noise, when the mean value is set to zero, more noise points appear in the extracted watermark with the increase of variance; when variance is set to 0.01, the extracted watermark can still be distinguishable.

**5.2. Attack of JPEG Compression.** JPEG lossy compression is the most common image compression technique. The watermarked image Image1 is compressed using JPEG with different value of  $\beta$ , horizontal resampling by 2:1, and compression ratios, 0.80, 0.65, 0.50, 0.25, and 0.10 respectively. The results obtained are summarized in Table 2; the distinguishable watermark can still be extracted even the ratio is set to 0.10, so the algorithm is very robust against compression.

[21]. In the experiment, we add salt and pepper noise and Gaussian white noise to Image1 image, as shown in Table 1.

For the watermarked image with salt and pepper noise, with the increase of noise density, more and more noise points appear in the extracted watermark, when the density is

**5.3. Attack of Cropping.** The algorithm is also very robust against the cropping operation. In order to resist true cropping, that is, only a portion of the image is kept and the remaining part is thrown away, the width and height of the host image, along with the other keys, are sent to the



TABLE 6: Compare the proposed method with the algorithms in [4, 11, 18].

Attacks	NC			
	Reference [4] (PSNR = 42.02 dB)	Reference [11] (PSNR = 44.73 dB)	Reference [18] (PSNR = 42.98 dB)	Our algorithm (PSNR = 31.14 dB)
Median filter ( $3 \times 3$ )	0.90	0.88	0.93	1
JPEG (ratio = 0.80)	0.99	NA	1	0.9997
JPEG (ratio = 0.50)	0.96	0.95	0.96	0.9976
JPEG (ratio = 0.10)	0.34	0.33	0.32	0.9987
Rotation (degree: $0.25^\circ$ )	0.59	0.61	0.65	0.9897
Rotation (degree: $-0.25^\circ$ )	0.60	0.65	0.72	0.9900
Cropping 1/4	0.66	0.60	0.67	0.8561
Scaling $256 \times 256$	0.88	0.71	0.85	0.9999

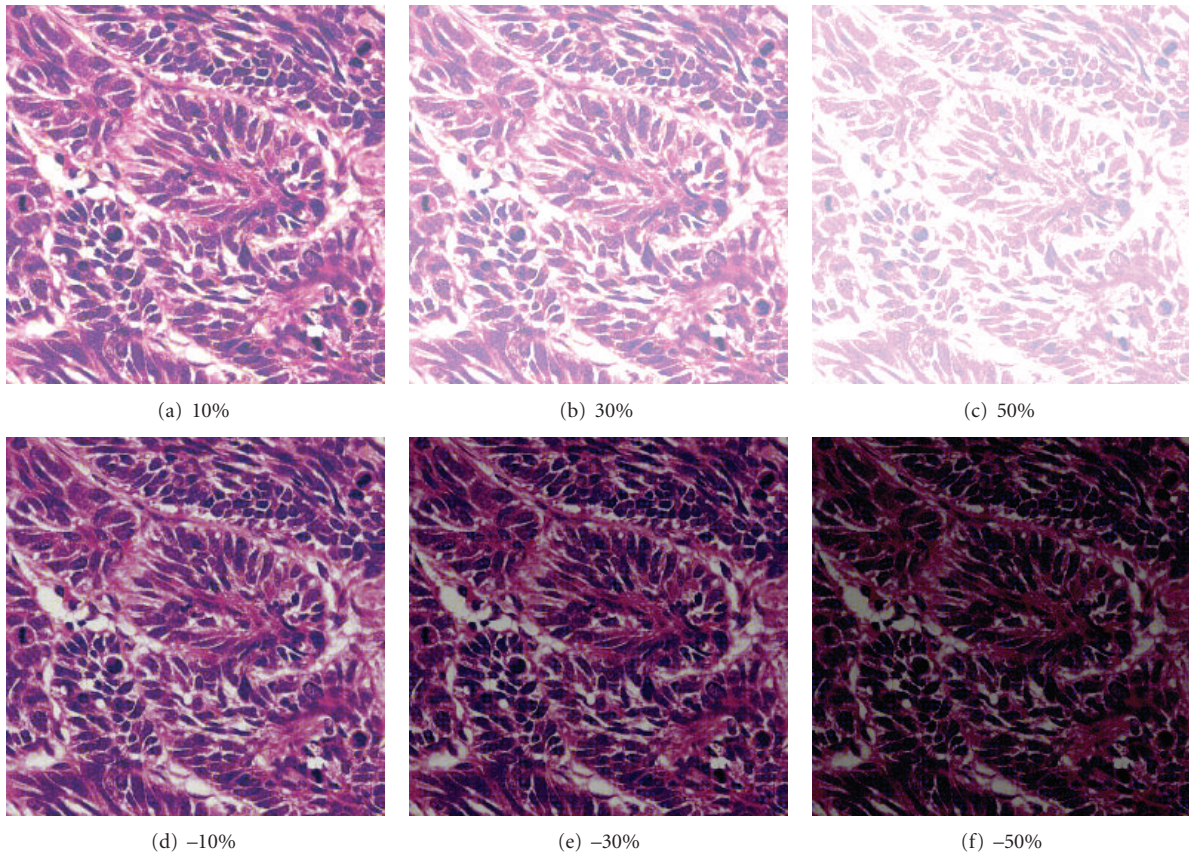


FIGURE 10: Image1: enhance and decrease the brightness.

detector. If the detector receives a portion of the image, he can recover it to the original size by zero padding, to extract the watermark.

After randomly selecting the position and crop 5%, 10%, 25%, 50%, and 75% of the watermarked image, respectively, then extract the watermark, as shown in Table 3. When 50% of the whole image is cropped, we can still extract the distinguishable watermark. That is to say, the algorithm has high robustness against cropping.

After cropping the randomly selected position, and filling the cropped region with zero, the extracted watermark before and after the reverse transformation is shown in Table 4;

from it we can find that the randomly cropping has no effect on the extracted watermark.

Without the preprocessing to scramble the watermark, the four letters "DLUT" on the original watermark may be unexpectedly cropped; that is, the extracted watermark may be fragmentary, so scrambling the watermark by Arnold cat map in Section 2.1 can help to make the bits randomly distributed, which contributes to the robustness.

*5.4. Attack of Brightness.* We increase and decrease the brightness by 10%, 30%, and 50%, respectively, as shown in

Figure 10. The extracted watermarks demonstrate that increasing the brightness has less effect on the watermark than decreasing the brightness, as shown in Table 5. So the algorithm has high robustness against the attack of brightness.

5.5. *Comparison.* Finally, we use Image1 with  $\beta = 0.218$  to compare our algorithm with the algorithms proposed in [4, 11, 18]. The results show that our algorithm gets advantages over the others in the attacks of median filter, compression, rotation, and cropping, as shown in Table 6.

## 6. Conclusion

We propose a dynamic block-based blind watermark algorithm based on a 2-level DWT, using Haar filter to embed a preprocessing binary image into the blue component of the color pathological image. By analyzing the coefficients' characteristic after wavelet decomposition, we select the detail subbands to embed watermark. During the embedding procedure, the statistic characteristics of the mean and variance are applied to dynamically adjust the embedding intensity, so as to generate the effective range of threshold to extract the watermark. Experiment results demonstrate that the quality of the watermarked image, which is based on the PSNR, decreases smoothly with the increase of the embedding intensity, and the algorithm is robust against common distortions, especially, getting advantages over noise, compression, cropping, rotation, and brightening.

## Acknowledgments

The research is supported by the Science and Technology Development Plan of Weifang (no. 20121346) and the Minority Nationality Technology Talent Cultivation Plan of Xinjiang (no. 201123116).

## References

- [1] S. Shyamsunder and G. Kaliyaperumal, "Image encryption and decryption using chaotic maps and modular arithmetic," *American Journal of Signal Processing*, vol. 1, no. 1, pp. 24–33, 2011.
- [2] S. Rawat and B. Raman, "A chaotic system based fragile watermarking scheme for image tamper detection," *International Journal of Electronics and Communications*, vol. 65, no. 10, pp. 840–847, 2011.
- [3] C. B. Li, P. Li, and G. S. Lu, "Enterprise projects set risk element transmission chaotic genetic model," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 17, pp. 3162–3167, 2012.
- [4] W. H. Lin, Y. R. Wang, S. J. Horng, T. W. Kao, and Y. Pan, "A blind watermarking method using maximum wavelet coefficient quantization," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11509–11516, 2009.
- [5] S. Agreste and G. Andaloro, "A new approach to pre-processing digital image for wavelet-based watermark," *Journal of Computational and Applied Mathematics*, vol. 221, no. 2, pp. 274–283, 2008.
- [6] S. Agreste, G. Andaloro, D. Prestipino, and L. Puccio, "An image adaptive, wavelet-based watermarking of digital images," *Journal of Computational and Applied Mathematics*, vol. 210, no. 1–2, pp. 13–21, 2007.
- [7] K. C. Liu, "Wavelet-based watermarking for color images through visual masking," *International Journal of Electronics and Communications*, vol. 64, no. 2, pp. 112–124, 2010.
- [8] H. M. Al-Otum and N. A. Samara, "A robust blind color image watermarking based on wavelet-tree bit host difference selection," *Signal Processing*, vol. 90, no. 8, pp. 2498–2512, 2010.
- [9] G. Zhang, S. Wang, and Q. Wen, "An adaptive block-based blind watermarking algorithm," in *Proceedings of the 7th International Conference on Signal Processing Proceedings (ICSP '04)*, vol. 3, pp. 2294–2297, September 2004.
- [10] S. J. Horng, T. W. Kao, P. Fan, C. L. Lee, and Y. Pan, "An efficient watermarking method based on significant difference of wavelet coefficient quantization," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 746–757, 2008.
- [11] W. H. Lin, Y. R. Wang, and S. J. Horng, "A wavelet-tree-based watermarking method using distance vector of binary cluster," *Expert Systems with Applications*, vol. 36, no. 6, pp. 9869–9878, 2009.
- [12] W. Lu, W. Sun, and H. Lu, "Robust watermarking based on DWT and nonnegative matrix factorization," *Computers and Electrical Engineering*, vol. 35, no. 1, pp. 183–188, 2009.
- [13] H. A. Al-Otum and A. O. Al-Taba'a, "Adaptive color image watermarking based on a modified improved pixel-wise masking technique," *Computers and Electrical Engineering*, vol. 35, no. 5, pp. 673–695, 2009.
- [14] X. Wu and Z. H. Guan, "A novel digital watermark algorithm based on chaotic maps," *Physics Letters A*, vol. 365, no. 5–6, pp. 403–406, 2007.
- [15] A. Mooney, J. G. Keating, and I. Pitas, "A comparative study of chaotic and white noise signals in digital watermarking," *Chaos, Solitons and Fractals*, vol. 35, no. 5, pp. 913–921, 2008.
- [16] J. M. Amigó, L. Kocarev, and J. Szczepanski, "Theory and practice of chaotic cryptography," *Physics Letters A*, vol. 366, no. 3, pp. 211–216, 2007.
- [17] D. Tsolis, S. Nikolopoulos, L. Drossos, S. Sioutas, and T. Papatheodorou, "Applying robust multibit watermarks to digital images," *Journal of Computational and Applied Mathematics*, vol. 227, no. 1, pp. 213–220, 2009.
- [18] R. S. Run, S. J. Horng, W. H. Lin, T. W. Kao, P. Fan, and M. K. Khan, "An efficient wavelet-tree-based watermarking method," *Expert Systems with Applications*, vol. 38, no. 9, pp. 14357–14366, 2011.
- [19] H. Chen and P. K. Varshney, "A human perception inspired quality metric for image fusion based on regional information," *Information Fusion*, vol. 8, no. 2, pp. 193–207, 2007.
- [20] G. Bhatnagar and B. Raman, "A new robust reference watermarking scheme based on DWT-SVD," *Computer Standards and Interfaces*, vol. 31, no. 5, pp. 1002–1013, 2009.
- [21] A. Mooney, J. G. Keating, and D. M. Heffernan, "Performance analysis of chaotic and white watermarks in the presence of common watermark attacks," *Chaos, Solitons and Fractals*, vol. 42, no. 1, pp. 560–570, 2009.



**Hindawi**  
Submit your manuscripts at  
<http://www.hindawi.com>

