Blockwise Image Watermarking System with Selective Data Embedding in Wavelet Transform Domain

Pao-Chi Chang[1], Ta-Te Lu, and Li-Lin Lee
Department of Electrical Engineering, National Central University, Chung-Li, Taiwan 320

ABSTRACT

In this paper, we propose an image watermarking system that is highly robust against various attacks without perceivable image degradation. The cover image is first discrete wavelet transformed (DWT), and then the low and middle subbands are divided into wavelet blocks. A selective watermark embedding method is used in which a DWT block is chosen for watermark embedding only when its coefficients clearly indicate the block polarity. Instead of the original image, a key is used in the watermark extraction to indicate the locations where watermark bits are embedded. The key is generated by a Tri-state Exclusive OR (TXOR) operation on the randomized watermark and the randomized DWT coefficients of the original image. Finally, a deadzone evacuation procedure is performed to ensure an adequate noise margin. If a DWT coefficient is very close to the polarity threshold, e.g., the median, then it will be forced to shift to the positive or the negative end of the deadzone depending on its polarity. Simulation results show that the key method proposed herein achieves excellent performance for Checkmark non-geometric attacks, such as filtering, compression, and copy attacks. The proposed scheme is also robust for image cropping at different positions.

Keywords: Watermarking, selective data hiding, DWT, XOR.

1. INTRODUCTION

The use of images, audio, and video on Internet or wireless networks becomes increasingly popular. Since these digital data are easy to modify and copy, legal copyright and media protection are essential. Digital watermarking is a conventional means of claiming ownership of a data source. Many research results of watermarking appeared in recent years. However, the performance of the current watermarking techniques is still not satisfactory. For instance, the requirement of the original image for watermark extraction is inconvenient, and they are either perceivable for the watermarked image quality or vulnerable against various attacks.

Cox [1] suggested that the watermark method should be robust, invisible and unambiguous. Most image watermarking procedures operate in the spatial domain or the frequency domain [2]-[5]. Spatial domain watermarking techniques are relatively simple. However, data compression and noise attacks can easily remove and distort these embedded watermarks. In contrast, watermarking schemes operated in the frequency domain can generally embed more bits and are more robust against attacks [4]-[8].

Watermarking techniques in the wavelet domain have been proposed [8]-[10]. Most of them inserted watermarks into high or middle frequency subbands because these subbands include the edges or less important information of an image. However, most of the wavelet coefficients in high or middle frequency subbands are small in magnitude, the embedded watermarks can easily be removed by lossy compression, filtering, and cropping attacks. On the contrary, the lowest subband usually contains the largest portion of energy. Therefore, the lowest subband is more robust than others against attacks. However, the human visual system is sensitive to the change of the wavelet coefficients in the lowest subband, the embedded watermarks are relatively easy to be visually detected. In this paper, we use a selective embedding algorithm that jointly considers all wavelet coefficients in a block to identify suitable blocks for watermark embedding to filter out those blocks that are either perceivable for image quality or are vulnerable against attacks.

[1]Correspondence: e-mail: pcchang@ee.ncu.edu.tw; phone: 886 3 4227151 ext. 4466; fax: 886 3 4255830; http://vaplab.ee.ncu.edu.tw/

Security and Watermarking of Multimedia Contents IV, Edward J. Delp III, Ping Wah Wong, Editors, Proceedings of SPIE Vol. 4675 (2002) © 2002 SPIE · 0277-786X/02/$15.00

In the watermarking extraction scheme, it is desirable to have the blind detection capability, i.e., extracting the watermark without referring to the original unwatermarked image. Instead of the original image, a secret key that is generated based on the watermark logo and the original image is used in the embedding and extraction processes [11]-[13]. Kutter [11] proposed a method that the embedded watermark data cannot be extracted without having the secret key. Miaou [12] embedded a watermark in the middle frequency subband of wavelet domain and proposed a key watermarking method.

In our proposed system, a key method is used with the selective watermark embedding. In the extraction stage, based on the polarity of the wavelet coefficients and the key, the original watermark can accurately be extracted. The key is generated by a Tri-state Exclusive OR (TXOR) operation on the randomized watermark and the randomized wavelet blocks. If most of the wavelet coefficients in one wavelet block are close to the median value, the wavelet block is classified as the third state or the undefined state, which will not be used to generate the key. The process can substantially increase the robustness.

The rest of this paper is organized as follows. Section 2 describes the embedding watermarking structure while Section 3 focuses on the extraction of the watermark. Section 4 presents the simulation results and conclusions are finally made in Section 5.

2. WATERMARK EMBEDDING STRUCTURE

The block diagram of the proposed watermark embedding scheme is illustrated in Fig. 1. An $M \times M$ image is first processed by L level discrete wavelet transform (DWT) with Daubechies-4 orthogonal filters. These wavelet coefficients are then divided into $m \times m$ wavelet blocks with the block size $r \times r$. Most of the DWT coefficients in high frequency bands have small values that are not suitable for watermark embedding. Therefore, only coefficients with relatively high energy are selected. We use block composition to filter out wavelet blocks which are not suitable for watermark embedding. We notice that some wavelet coefficients are close to the median values and therefore not suitable for watermarking, because after compression or other DSP attacks, the polarity of the coefficient may be changed and errors in watermark extraction will occur. The wavelet coefficients in each band are composed into blocks C_b^i, where b is the block sequence number, b = 0, 1, 2, …., $m \times m - 1$, i is the coefficient index, i = 0, 1, 2, …. , $r \times r$ -1. In this work, a block is composed of four coefficients, which form a 2×2 square block.

The watermark logo is a $M_w \times M_w$ bi-level image which can be represented as one dimension sequence W_l that consists of binary values "1" and "0", where l = 0, 1, 2, …., $M_w \times M_w - 1$, representing the binary sequence index. In order to enhance the security of the key sequence, block sequence C_b^i and watermark sequence W_l perform random permutations to get $C_b^{i'}$ and W_l', respectively, before mutual operations. The detailed embedding procedure is described as follows.

2.1 Block Mapping

The polarity of each coefficient in a block is first determined. Let M_i be the median value of the i-th coefficients of all wavelet blocks. If an element in block b exceeds M_i, then the polarity of this coefficient is defined as positive and the block counter N_b increases by one. In the block mapping, only when all or almost all coefficients in a block are at the same side of the median, i.e., the same polarity, this block is qualified to embed watermark. This technique further improves the robustness because errors occur in the watermark extraction only when multiple DWT coefficients are affected by attacks. In this work, two three-level block mapping methods are used.

(a) In the lowest wavelet band (LL): If $N_b = 3$ or 4, i.e., most coefficients are positive polarity, the block polarity B_b' is set as "1"; if $N_b = 0$ or 1, i.e., most coefficients are negative polarity, B_b' is set as "0"; moreover, if $N_b = 2$, the block polarity B_b' is set as "U" (Undefined), that means block coefficients are equally divided into positive and negative, which are vulnerable against attacks.

(b) In other bands (LH, HL, HH): More restrictions are imposed on the mapping to avoid ambiguity in extraction. If $N_b = 4$, the block polarity $B_b^{'}$ is set as "1"; if $N_b = 0$, $B_b^{'}$ is set as "0"; and if $N_b = 1, 2, 3$, that means the high frequency block does not imply a strong polarity and the block polarity may become ambiguous after attacks. Therefore, it is not suitable for watermark embedding and the block polarity $B_b^{'}$ is set as "U" (Undefined).

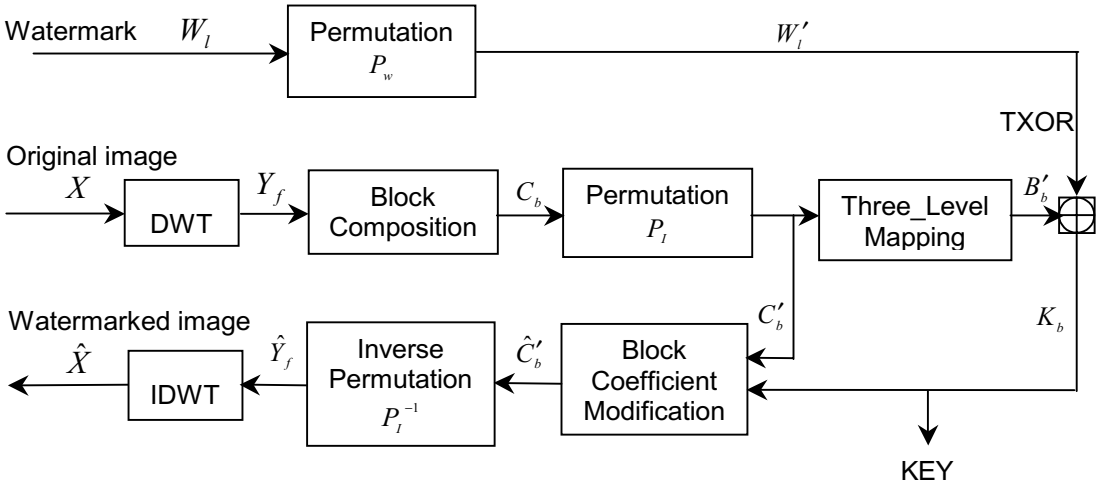

Fig.1: The block diagram of the proposed watermark embedding scheme

2.2 Selective Watermark Embedding and Key Generation

Before the cover image and watermark logo are mutually operated, both the wavelet blocks and the watermark are processed by random permutations to disperse the spatial relationship with the neighborhood. A Tri-state Exclusive OR (TXOR) operation is then undertaken for permutated watermarks $W_I^{'}$ and block polarity $B_b^{'}$, from the lowest band to high frequency bands, to produce a key K_b

$$K_b = \begin{cases} W_I^{'} \oplus B_b^{'} & , \quad B_b^{'} = 0, 1 \\ 0 & , \quad B_b^{'} = U \end{cases} \qquad (1)$$

which is also shown in Fig. 2. When a block has a clear polarity, i.e., 0 or 1, XOR is performed to generate a key bit that is either 0 or 1. Otherwise, if the block polarity is undefined, the key bit associated with the block is set to 0. Therefore, the key length will be longer than the watermark sequence length in general. Fig. 2 shows an example for the key generation algorithm. The key sequence $K_b = 100111$ can be obtained by $W_I^{'} = 11010$ and $B_b^{'} = 01U010$.

2.3 Deadzone Evacuation

In watermark embedded blocks, if a coefficient is located very close to the median, it is vulnerable against attacks because it is very possible that its polarity can be changed by attack. In this case, we force the coefficient to shift toward the positive or negative end by the just-noticeable-difference (JND) amount, so that it is not noticeable but more robust. The deadzone or the neutral zone of coefficient *i* is defined as the interval $[M_i - T_i,\ M_i + T_i]$, where M_i and T_i represent the median and the JND threshold of coefficient *i*, respectively. Depending on the coefficient location, some coefficient may need to be adjusted as follows

$$\hat{C}_b^{i'} = \begin{cases} M_i + T_i & , \quad M_i \le C_b^{i'} < M_i + T_i \\ M_i - T_i & , \quad M_i - T_i < C_b^{i} < M_i \\ C_b^{i'} & , \quad otherwise \end{cases} \qquad (2)$$

where $\hat{C}_b^{i'}$ is the modified coefficient. After this operation, no coefficients in watermarked blocks are allowed to appear in the deadzone.

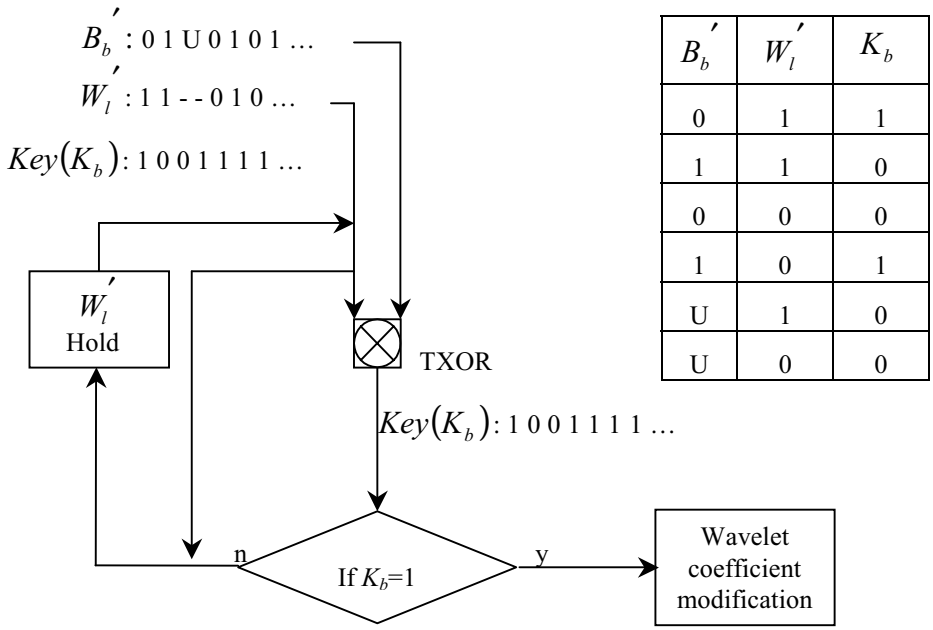

Fig.2: An example of the selective watermark embedding algorithm

3. WATERMARK DETECTION

This method uses the key to locate the image positions and to extract the embedded watermark. Fig. 3 illustrates the block diagram of the watermark detection scheme. The major steps in watermark detection are described as follows:

A. Discrete wavelet transform: the watermarked image is DWT transformed;
B. Block mapping and pseudo random permutation: with the same procedure of the watermark embedding to obtain $B_b^{*'}$;
C. TXOR Processing: If $K_b = 1$ then it performs XOR with $B_b^{*'}$ to obtain $W_l^{*'}$. Otherwise, $B_b^{*'}$ is erased;
D. Inverse permutation: perform reverse permutation to extract the watermark;

Once the watermark is extracted, the normalized correlation (NC) is used to calculate the similarity between the extracted and the original watermarks. For the extracted watermark W_l^* and the referenced watermark W_l with the size $M_w \times M_w$, the NC is defined as

$$NC = \frac{\sum_{i=0}^{M_W \times M_W - 1} W_l W_l^*}{\sum_{l=0}^{M_W \times M_W - 1} W_l^2} \qquad (3)$$

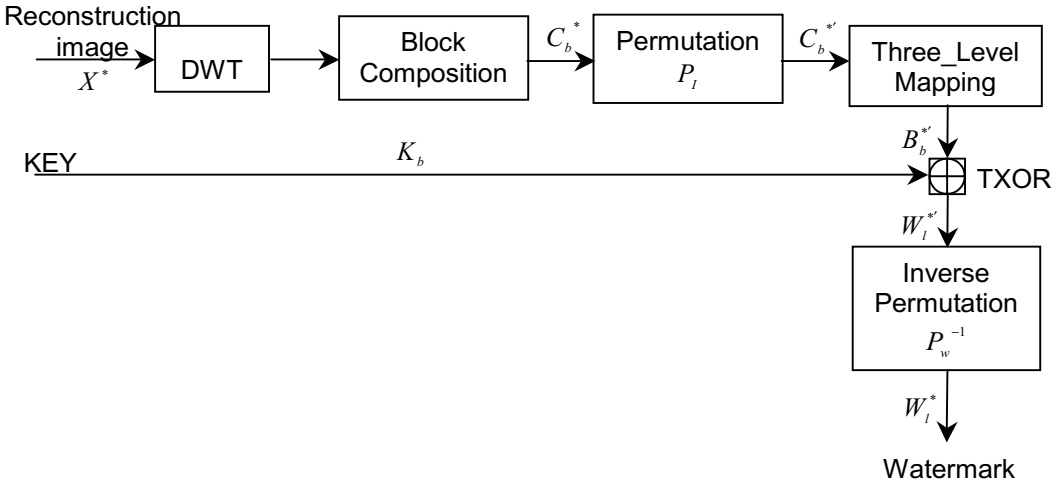

Fig.3: The block diagram of the proposed watermark detection scheme

4. EXPERIMENTAL RESULTS

In the simulations, method1 uses the proposed key method with one-level DWT, while method2 is with two-level DWT and method3 is with three-level DWT. The performances without attacks are first measured. Table I compares the PSNR of the watermarked images with method1, method2, and method3 for Lena, Barbara and Pepper (size 512×512, Daubechies-4 orthogonal filters), with the deadzone threshold $T = 10$. Two different digital watermarks (size of 32×32), watermark1 in English and watermark2 in Chinese, are used in simulations. Simulation results show that both watermarks can be embedded completely into the lowest band (LL) for method1 and method2. However, method3 needs to use LL, LH, and HL bands to embed the watermark. Therefore the key is longer than that in method1 and method2 due to more restricted three-level mapping in high bands. In all methods, both watermark1 and watermark2 are invisible after embedding. In fact, the three key methods achieve PSNR over 55 dB which yields excellent image quality.

The performances of the proposed system with Voloshynovskiy's Checkmark attacks [14] such as image denosing, lossy compression, quantization, remodulation, cropping, and scale are then measured. Tables II, III, and IV compare the NC values and numbers of erroneous pixels with method1, method2, and method3 for JPEG2000, JPEG, and SPIHT compression. Even at a low quality factor (Q=10) for JPEG compression, method1 can extract watermark1 and watermark2 (NC=0.91), and method2 can extract watermark1 (NC=0.966) and watermark2 (NC=0.958) successfully for Lena. At 0.1 bpp for JPEG2000 compression, method1 can extract watermark1 (NC=0.945) and watermark2 (NC=0.947), and method2 can extract watermark1 and watermark2 (NC=0.986) for Lena. These results confirm that key method proposed herein is extremely robust with method1 and method2 against JPEG, JPEG2000, and SPIHT compression. Method3 is less robust against compression attacks because in this case the key cannot be embedded in LL band only, and high bands are more difficult to hide data. Table V shows that the NC values of method2 for Barbara after rescaling to various size from 0.5~2.0 and maximum a posteriori probability (MAP) attacks that include Wiener filter, soft shrinking and hard shrinking [14]. Fig. 4 illustrates cropping at different positions for Lena, Barbara, and Pepper. Fig. 5 illustrates the cropping results of extracted watermark for Lena, and Barbara. It clearly shows that the key technique is robust against non-geometrical, scaling, and cropping attacks.

5. CONCLUSIONS

This work has presented a key watermarking method for embedding watermarks at the lowest band of wavelet transform, and the method is extremely robust under attacks such as compression and cropping. The simulation results demonstrate that when using method2, no error exists in the extracted watermark1 up to 20:1 JPEG2000 compression, i.e., quality=40. Furthermore, NC exceeds 0.9 when using method1 and method2 for JPEG, JPEG2000, and SPIHT compression. Additionally, without using the original images makes it easy to determine the ownership of the watermarked images.

REFERENCES

[1] I. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia, " *IEEE Trans. Image Processing, vol. 6, pp. 1637-1687, Dec. 1997*.

[2] O. Bruyndonckx, J. J. Quisquater, and B. Macq, "Spatial Method for Copyright Labeling of Digital Images," *in Proc. IEEE Nonlinear Signal and Image Processing, pp. 456-459, June 1995.*

[3] R. B. Wolfgang and E. J. Delp, "A Watermark for Digital Images," in *IEEE Int. Conf. Image Processing, vol. 3, pp. 219-222,1996.*

[4] Y. Kim, I. Choi, I. Lee, T. Yun, and K. T. Park, "Wavelet Transform Image Compression Using Human Visual Characteristics and a Tree Structure with a Height Attribute," *Optical engineering, vol. 35, pp. 204-212, 1996.*

[5] C. I. Podilchuk, W. Zeng, "Image-Adaptive Watermarking Using Visual Models," *IEEE J. Select. Areas Commum., vol. 16, No. 4, pp.525-539, May 1998.*

[6] H. Inoue, A. Miyazaki, A.Yamamoto, and T. Katsura, "A Digital Watermark Based on the Wavelet Transform and its Robustness on Image Compression," *Image Processing, 1998. ICIP 98. Proceedings*.

[7] C. T. Hsu and J. L. Wu, "Hidden Digital Watermarks in Images," *IEEE Trans. Image Processing, vol. 8, pp. 58-68, January 1999*.

[8] C. T. Hsu and J. L. Wu, "Multiresolution Watermarking for Digital Images," *IEEE Trans. on Circuits and Systems, vol. 45, pp. 1097-1101, August 1998*.

[9] W. Zhu, Z. Xiong and Y. Q. Zhang, "Multiresolution Watermarking for Images and Video," *IEEE Trans. on Circuits and Systems for video technology, vol. 9, pp. 545-550, June 1999*.

[10] M. J. Tsai, K. Y. Yu and Y. Z. Chen, "Joint Wavelet and Spatial Transformation for Digital Watermarking," *IEEE Tran. on Consumer Electronics, vol. 46, pp. 241-245, Feb. 2000.*

[11] F. Hartung and M. Kutter, "Multimedia Watermarking Techniques," *in Proc. IEEE vol. 87, No. 7, pp. 1079-1107, July 1999*

[12] S. G. Miaou and Z. M. Chen, "A Robust Image Watermarking Technique Based on Wavelet Transform and Human Visual System," *Computer Vision Graphics and Image Processing*, *pp. 82-89, July 2000*.

[13] G. W. Braudaway, K. A. Magerlein, and F. C. Mintzer "Color correct digital watermarking of images," *U.S Patent. 5 530 759, June 1996.*

[14] S. Voloshynovskiy, S. Pereira, V. Iquise, and T. Pun, "Attack modeling: towards a second generation watermarking benchmark," *Signal Processing, pp.1177-1214, 2001*

Table I: The PSNR of three watermarking methods for three images: Lena, Barbara, and Pepper

	watermark1			watermark2		
	method 1	method 2	method 3	method 1	method 2	method 3
Lena	55.11	61.79	70.11	55.43	62.17	70.12
Barbara	62.59	63.89	68.5	62.59	63.65	68.48
Pepper	64.12	66.29	71.31	64.61	65.72	72.11

Table II: The NC and the number of error pixels under JPEG2000 compression attacks

Lena	watermark 1						watermark 2					
	method 1		method 2		method 3		method 1		method 2		method 3	
	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points
0.1bpp	0.945313	28	0.986328	7	0.691406	158	0.947266	27	0.986328	7	0.697266	155
0.2bpp	0.96875	16	0.992188	4	0.744141	131	0.970703	15	0.992188	4	0.742188	132
0.4bpp	0.982422	9	1	0	0.78125	112	0.986328	7	1	0	0.796875	104
0.6bpp	0.990234	5	1	0	0.808594	98	0.988281	6	1	0	0.816406	94
0.8bpp	0.998047	1	1	0	0.865234	69	0.996094	2	1	0	0.871094	66
1bpp	0.998047	1	1	0	0.884766	59	0.998047	1	1	0	0.890625	56

Barbara	watermark 1						watermark 2					
	method 1		method 2		method 3		method 1		method 2		method 3	
	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points
0.1bpp	0.976563	12	0.974609	13	0.765625	120	0.964844	18	0.982422	9	0.792969	106
0.2bpp	0.986328	7	0.978516	11	0.822266	91	0.984375	8	0.978516	11	0.841797	81
0.4bpp	0.990234	5	0.992188	4	0.886719	58	0.990234	5	0.990234	5	0.886719	58
0.6bpp	0.994141	3	0.996094	2	0.902344	50	1	0	0.998047	1	0.916016	43
0.8bpp	0.998047	1	1	0	0.917969	42	1	0	1	0	0.927734	37
1bpp	0.998047	1	1	0	0.931641	35	1	0	1	0	0.945313	28

Pepper	watermark 1						watermark 2					
	method 1		method 2		method 3		method 1		method 2		method 3	
	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points
0.1bpp	0.974609	13	0.980469	10	0.787109	109	0.974609	13	0.988281	6	0.804688	100
0.2bpp	0.988281	6	0.996094	2	0.845703	79	0.986328	7	0.998047	1	0.845703	79
0.4bpp	0.996094	2	0.998047	1	0.839844	82	0.996094	2	0.998047	1	0.876953	63
0.6bpp	0.998047	1	1	0	0.869141	67	0.998047	1	1	0	0.892578	55
0.8bpp	1	0	1	0	0.892578	55	1	0	1	0	0.898438	52
1bpp	1	0	1	0	0.898438	52	1	0	1	0	0.917969	42

Table III: The NC and the number of error pixels under JPEG compression attacks

Lena	watermark 1						watermark 2					
	method 1		method 2		method 3		method 1		method 2		method 3	
JPEG quality	NC	error points	NC	error points	NC	error points	NC	error points	NC	error points	NC	error points
10	0.910156	46	0.966797	17	0.685547	161	0.910156	46	0.958984	21	0.6875	160
20	0.986328	7	0.994141	3	0.767578	119	0.982422	9	0.996094	2	0.765625	120
30	1	0	0.998047	1	0.814453	95	1	0	1	0	0.820313	92
40	1	0	1	0	0.861328	71	1	0	1	0	0.853516	75
50	1	0	1	0	0.876953	63	1	0	1	0	0.880859	61
60	1	0	1	0	0.919922	41	1	0	1	0	0.914063	44
70	1	0	1	0	0.957031	22	1	0	1	0	0.962891	19
80	1	0	1	0	0.964844	18	1	0	1	0	0.966797	17
90	1	0	1	0	0.986328	7	1	0	1	0	0.990234	5
100	1	0	1	0	0.988281	6	1	0	1	0	0.990234	5

Babara	watermark 1						watermark 2					
	method 1		method 2		method 3		method 1		method 2		method 3	
JPEG quality	NC	error points	NC	error points	NC	error points	NC	error points	NC	error points	NC	error points
10	0.988281	6	0.992188	4	0.826172	89	0.990234	5	0.992188	4	0.826172	89
20	0.994141	3	0.996094	2	0.871094	66	0.996094	2	1	0	0.882813	60
30	0.998047	1	1	0	0.896484	53	0.998047	1	1	0	0.900391	51
40	0.998047	1	1	0	0.912109	45	0.998047	1	1	0	0.914063	44
50	1	0	1	0	0.931641	35	1	0	1	0	0.935547	33
60	1	0	1	0	0.947266	27	1	0	1	0	0.957031	22
70	1	0	1	0	0.96875	16	1	0	1	0	0.976563	12
80	1	0	1	0	0.982422	9	1	0	1	0	0.982422	9
90	1	0	1	0	0.990234	5	1	0	1	0	0.988281	6
100	1	0	1	0	0.996094	2	1	0	1	0	0.994141	3

Pepper	watermark 1						watermark 2					
	method 1		method 2		method 3		method 1		method 2		method 3	
JPEG quality	NC	error points	NC	error points	NC	error points	NC	error points	NC	error points	NC	error points
10	0.986328	7	0.992188	4	0.783203	111	0.988281	6	0.990234	5	0.806641	99
20	1	0	0.996094	2	0.835938	84	0.996094	2	0.998047	1	0.855469	74
30	0.998047	1	0.998047	1	0.867188	68	0.998047	1	1	0	0.892578	55
40	1	0	1	0	0.898438	52	1	0	1	0	0.90625	48
50	1	0	1	0	0.916016	43	1	0	1	0	0.921875	40
60	1	0	1	0	0.943359	29	1	0	1	0	0.953125	24
70	1	0	1	0	0.964844	18	1	0	1	0	0.957031	22
80	1	0	1	0	0.976563	12	1	0	1	0	0.976563	12
90	1	0	1	0	0.978516	11	1	0	1	0	0.982422	9
100	1	0	1	0	0.994141	3	1	0	1	0	0.992188	4

Table VI: The NC and the number of error pixels under SPIHT compression attacks

Lena	watermark_1						watermark_2					
	method 1		method 2		method 3		method 1		method 2		method 3	
	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points
0.1bpp	0.939453	31	0.988281	6	0.707031	150	0.9375	32	0.990234	5	0.697266	155
0.2bpp	0.949219	26	0.988281	6	0.710938	148	0.962891	19	0.990234	5	0.724609	141
0.4bpp	0.984375	8	1	0	0.767578	119	0.978516	11	0.998047	1	0.798828	103
0.6bpp	0.988281	6	0.998047	1	0.808594	98	0.986328	7	1	0	0.824219	90
0.8bpp	0.992188	4	0.998047	1	0.830078	87	0.990234	5	1	0	0.849609	77
1bpp	0.994141	3	1	0	0.875	64	0.996094	2	1	0	0.871094	66

Barbara	watermark_1						watermark_2					
	method 1		method 2		method 3		method 1		method 2		method 3	
	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points
0.1bpp	0.974609	13	0.974609	13	0.755859	125	0.974609	13	0.982422	9	0.810547	97
0.2bpp	0.976563	12	0.990234	5	0.806641	99	0.988281	6	0.994141	3	0.837891	83
0.4bpp	0.992188	4	0.988281	6	0.875	64	0.996094	2	1	0	0.867188	68
0.6bpp	0.990234	5	0.994141	3	0.900391	51	0.998047	1	1	0	0.896484	53
0.8bpp	0.998047	1	1	0	0.927734	37	1	0	1	0	0.898438	52
1bpp	1	0	1	0	0.929688	36	1	0	1	0	0.910156	46

Pepper	watermark_1						watermark_2					
	method 1		method 2		method 3		method 1		method 2		method 3	
	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points	NC	Error points
0.1bpp	0.974609	13	0.982422	9	0.779297	113	0.966797	17	0.980469	10	0.802734	101
0.2bpp	0.990234	5	0.992188	4	0.814453	95	0.966797	17	0.986328	7	0.84375	80
0.4bpp	0.996094	2	0.998047	1	0.847656	78	0.990234	5	0.988281	6	0.898438	52
0.6bpp	0.998047	1	1	0	0.869141	67	0.998047	1	0.996094	2	0.912109	45
0.8bpp	1	0	1	0	0.880859	61	1	0	1	0	0.941406	30
1bpp	1	0	1	0	0.896484	53	1	0	1	0	0.947266	27

Table V: Performance of the proposed system under MAP and Scaling attacks

MAP Type	NC	Scaling Type	NC	PSNR
Wiener1_J100	0.998047	0.5	0.996	28.94
Wiener2_J100	0.996094	0.75	1	35.35
Soft shrinking1_J100	0.994141	0.9	0.998	39.62
Soft shrinking2_J100	0.994141	1.1	1	42.2
Hard shrinking1_J100	0.996094	1.5	1	40.96
Hard shrinking2_J100	0.994141	2.0	1	38.26

Fig.4: Cropping attack, (a)(b)(c) different cropping positions for watermark1, (d)(e)(f) different cropping positions for watermark2

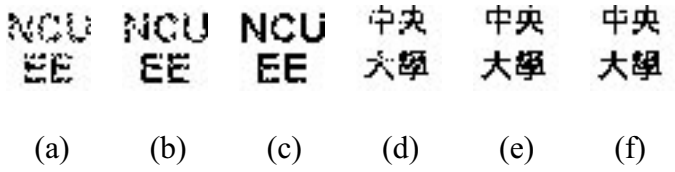

Fig.5: Extracted watermarks. (a) from Fig.4 (a), NC= 0.812500; (b) from Fig.4 (b), NC= 0.890625; (c) from Fig.4 (c), NC= 0.972656; (d) from Fig.4 (d), NC= 0.941406; (e) from Fig.4 (e), NC= 0.980469; (f) from Fig.4 (f), NC= 0.988281.