Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2010, Article ID 348321, 6 pages
doi:10.1155/2010/348321

Research Article

Adaptive Watermarking Scheme Using Biased Shift of Quantization Index

Young-Ho Seo,[1] Hyun-Jun Choi,[2] Ji-Sang Yoo,[3] and Dong-Wook Kim[4]

[1] *College of Liberal Arts, Kwangwoon University, 447-1, Welgye-1Dong, Nowon-Gu, Seoul 139-701, South Korea*
[2] *Deptartment of Information and Communications Engineering, Anyang 5-Dong, Manan-Gu, Anyang, Gyeonggi-do, 430-714, South Korea*
[3] *Department of Electronic Engeering, Kwangwoon University, 447-1, Welgye-1Dong, Nowon-Gu, Seoul 139-701, South Korea*
[4] *Department of Electronic Materials Engeering, Kwangwoon University, 447-1, Welgye-1Dong, Nowon-Gu, Seoul 139-701, South Korea*

Correspondence should be addressed to Young-Ho Seo, yhseo@kw.ac.kr

Received 19 October 2009; Revised 8 February 2010; Accepted 13 February 2010

Academic Editor: Liang-Gee Chen

We propose a watermark embedding and extracting method for blind watermarking. It uses the characteristics of a scalar quantizer to comply with the recommendation in JPEG, MPEG series, or JPEG2000. Our method performs embedding of a watermark bit by shifting the corresponding frequency transform coefficient (the watermark position) to a quantization index according to the value of the watermark bit, which prevents from losing the watermark information during the data compression process. The watermark can be embedded simultaneously to the quantization process without an additional process for watermarking, which means it can be performed at the same speed to the compression process. In the embedding process, a Linear Feedback Shift Register (LFSR) is used to hide the watermark informations and the watermark positions. The experimental results showed that the proposed method satisfies enough robustness and imperceptibility that are the major requirements for watermarking.

1. Introduction

As digital data, especially the image/video data that is the most information-rich, has been used as a commercial media very popularly, the ownership of the digital contents has been major issue to be solved. Digital watermarking has been known as the best solution as the ownership protection technique.

A watermarking scheme mainly consists of two processes: (1) deciding target data to embed the watermark in (watermark positions) and (2) embedding watermark data. Both affect the watermarking performance, robustness and imperceptibility. In this paper, we only concern with the embedding method itself, not proposing the watermark positioning method.

The target data to embed watermark in has been transferred from the ones in spatial domain to frequency domain as the watermarking techniques have been developed. A watermark embedded in a frequency domain data has been known to show better robustness against attacks than the spatial domain data. Thus, recent research has focused on the frequency domain watermarking.Cox et al. [1] proposed watermarking schemes embedding the watermark into the frequency domain based on the Discrete Cosine Transform (DCT).Hsieh et al. [2] used the multiresolution property of the Discrete Wavelet Transform (DWT). The adoption of JPEG2000 [3] as a standard for still image compression technique has speeded up these researches [2, 4].Yasein and Agathoklis proposed a JPEG2000-based watermarking technique that embeds watermark into the lowest subband using error correction code [5]. As 3D (3-dimensional) image and graphics are widely researched recently, watermarking schemes for 3D mesh object have been also proposed [6].

This paper is to propose a watermark embedding method that uses characteristics of a scalar quantization which is commonly used in a data compression technique for 2D images such as JPEG [3], MPEG series [7], and JPEG2000 [3]. That is, the proposed method can be applied to any watermark positioning method complying with such a compression technique. Thus, we do not include a specific

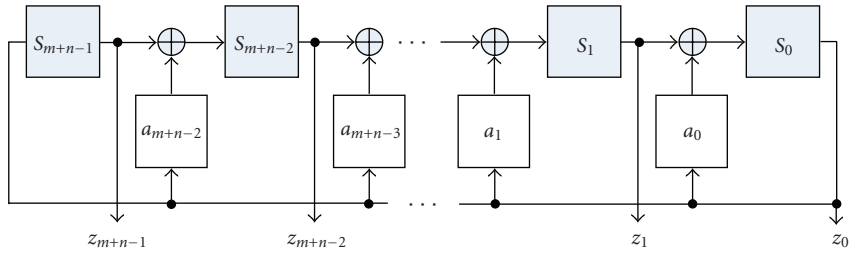

Figure 1: A linear feedback shift register (LFSR).

watermark positioning scheme here. The purpose of our method is to embed the watermark during the quantization process without additional process for watermarking itself. Quantization is generally used for a lossy compression in JPEG, MPEGs, or JPEG2000 standard. Thus, the speed to perform the watermarking scheme with our method is exactly the same as to perform the data compression process. Another intention of our method is to lose as small as possible watermarked data by quantization robustness. One more purpose of it is to minimize the quality degradation due to watermarking (imperceptibility) by limiting the variance of a coefficient within a half of a quantization step size. Since the watermarking is executed without a special calculation process during quantization, it is possible to minimize the delay time to embed/extract watermark in a realtime.

2. Proposed Watermarking Scheme

Since an image/video is mostly used in a compressed form, one must consider the information loss during the compression process. The main reason is to quantize coefficients, which corresponds to an essential tool for lossy compression. Therefore our research deeply considers the relationship between quantization and watermarking to propose an adaptive watermark embedding scheme.

The quantization in the JPEG, MPEGs, and JPEG2000, the international standards for the still image or motion picture, is a lossy compression process which is executed by dividing a coefficient by a predefined value that is called quantization step size. Equations (1) or (2) shows the quantization process for JPEG/MPEGs and JPEG2000, respectively.

$$q_b(u,v) = \text{round}\left(\frac{a_b(u,v)}{\Delta_b(u,v)}\right), \tag{1}$$

$$q_b(u,v) = \text{sign}(a_b(u,v)) \cdot \left\lfloor \frac{|a_b(u,v)|}{\Delta_b} + 0.5 \right\rfloor, \tag{2}$$

where, $a_b(u,v)$ and $q_b(u,v)$ are the pixel values at (u,v) before and after quantization, respectively, with the quantization step size of $\Delta_b(u,v)$. Here, the subscript b represents a subband or a region resulting from a transformation from a space domain into a frequency domain. Also, round(x) means the function is to round up to integer, which is the same function to $\lfloor x + 0.5 \rfloor$ in (2), in which $\lfloor x \rfloor$ means the largest integer not greater than x. sign(x) is the sign of x.

2.1. Watermark Embedding. The proposed scheme uses a biasedshift of a quantized coefficient (or index) according to the watermark value and the coefficient to embed the watermark in. Before embedding the watermark, we scatter the watermark itself for safety. In this paper, we use an $(m+n)$-bit LFSR whose representative feedback characteristics are expressed by (3) when the watermark data is a 2D data with the size of $2^m \times 2^n$, which is shown in Figure 1

$$P(x) = x^{m+n-1} + a_{m+n-2}x^{m+n-2} + \cdots + a_1 x + a_0. \tag{3}$$

In this equation, the coefficients ($a_i \in \{0,1\}$) have a set of values to make the polynomial primitive. In Figure 1, each block of a_i is a switch to connect the feedback to the corresponding XOR gate ($\oplus$). In the figure, S_i is the storage element that operates by a clock and its output z_i is one of the outputs of this LFSR. The initial value of the LFSR (key_{LFSR} in Algorithm 1) can be arbitrary except all "0" and is used as a secret key in this paper to hide the watermark to an unauthorized person.

With this LFSR, we composed a data rearrange scheme as shown in Algorithm 1. It first converts the accepted 2D watermark data into a serial data. Then, it is repositioned by the parallel outputs ($Z_i's$) of LFSR such that they are partitioned into two parts to decide the x and y coordinates.

The proposed watermark embedding scheme directly adjusts the transformed coefficients by considering the quantization scheme and each watermark bit. The data adjustment is performed by a biased shift method shown in (4) or (5), where $\lceil x \rceil$ means the smallest integer not less than x,

$$a'_b(u,v) = \text{sign}(a_b(u,v)) \cdot \left\lfloor \frac{|a_b(u,v)|}{\Delta_b(u,v)} \right\rfloor \times \Delta_b(u,v), \tag{4}$$

$$a'_b(u,v) = \text{sign}(a_b(u,v)) \cdot \left\lceil \frac{|a_b(u,v)|}{\Delta_b(u,v)} \right\rceil \times \Delta_b(u,v). \tag{5}$$

If watermarking is not considered, the quantization makes the transformed coefficient $a_b(u,v)$ the value of the nearest multiple of the quantization step $\Delta_b(u,v)$ as (1) or (2). Our watermarking scheme shifts this quantized coefficient (QC) or index. The amount and the direction of the shift are determined by the Least Significant Bit (LSB) of the resulting QC and the corresponding watermark bit in the manner that the LSB of the QC would be the same as the watermark bit by using (4) or (5). This process includes four cases, and the processing method for each of them is described in Algorithm 2.

With this watermarking process, the maximum amount of shift by watermarking is one quantization step. If the occurrence probabilities of the four cases are the same, the average amount of shift is zero. In Algorithm 2, $\mathrm{LFSR}_p(i)$ means the value of output z_p by applying i clocks with the LFSR initial value $\mathrm{key}_{\mathrm{embed}}$. Also $\mathrm{LSB}(x)$ means the LSB of x.

In this paper, we consider one more scheme to hide the watermarking positions with an LFSR, which might be the same to or different from the one to scatter the watermark data. The method is to select some QCs from the mark space, the target domain of the QCs to be watermarked, according to the value from the LFSR. There can be two schemes as follows. Note that it is out of this paper's scope how to select the mark-space.

(1) Use a serial output of an LFSR. The output bit stream is matched to the QCs in the mark space and only the QCs whose LFSR bits are "1" are selected as the watermarking positions. This method only embeds in the half of the QCs in the mark space if "0" and "1" have the same occurrence probability in the LFSR output.

(2) Use k parallel outputs. The value with k LFSR outputs is used as the value to skip the QCs in the mark space. That is, if the value is q, the next watermark position would be the $(q+1)_{\mathrm{th}}$ QC from the previously selected QC. With this method, only $2/(2^k+1)$ QCs from the mark space would be selected as the watermark positions if each k-bit combination by the k outputs has the same occurrence probabilities.

Algorithm 2 includes this random selection scheme. Here, only the first method is included such that $\mathrm{LFSR}_p(j)$.

2.2. Watermark Extracting. The embedded watermark is extracted only when the watermark information is required. It should be extracted from the watermarked data without support of the original data since the scheme is blind watermarking. The extraction process of our scheme needs to decide if a watermark bit resides in a specific data and the value of the watermark bit if resides.

Existence of watermark bit is determined using the same LFSR(s) and their initial values as the one(s) used in embedding process. The procedure is exactly the same as in embedding, and it is not mentioned further here. The embedded watermark is extracted only when the watermark information is required. It should be extracted from the watermarked data. The extraction process of our scheme needs to decide if a watermark bit resides in a specific data and the value of the watermark bit if resides.

Existence of watermark bit is determined using the same LFSR(s) and their initial values as the one(s) used in embedding process. The procedure is exactly the same as the one of embedding, and it is not mentioned further, here.

The process to read out the watermark bit is depicted in Algorithm 3. If the transmitted data containing a watermark bit has the value of even (odd) step of quantization, the corresponding watermark bit is "0" ("1"). In a real process, the received content data $r_b(u,v)$ is dequantized as

```
Procedure{Watermark_Mix}
input: 2^m × 2^n original 2D watermark (w(i, j))
       LFSR = (z_{m+n−1}, z_{m+n−2}, ..., z_1, z_0)
output: 2^m × 2^n rearranged 2D watermark (w'(i, j))
begin {
  k = 1;
  for i = 0 to 2^m − 1 {
    for j = 0 to 2^n − 1
      w(k++) = (w(i, j)); }
  initialize LFSR with key (key_LFSR);
  w'(0, 0) = w(0);
  for I = 1 to 2^{m+n} − 1 {
    apply one clock to LFSR;
    x_i = z_{m+n−1}2^{m−1} + z_{m+n−2}2^{m−2} + ··· + z_{n+1}2^1 + z_n2^0;
    y_i = z_{n−1}2^{n−1} + z_{n−2}2^{n−2} + ··· + z_1 2^1 + z_0 2^0;
    w'(x_i, y_i) = w(i); }}
end{Watermark_Mix}
```

Algorithm 1: Rearrangement algorithm for watermark data.

```
Procedure{Watermark_Embedding}
begin {
  initialize LFSR with a key (key_embed);
  for i = 0 to 2^{m+n} − 1 {
    if (LFSR_p(i) = "1") then {
      if (LFB(⌊|a_b(u, v)|/Δ_b(u, v)⌋) = "0") then {
        if (w_i = 0) then embed w_i by (4);
        else embed w_i by (5);}
      if (LFB(⌊|a_b(u, v)|/Δ_b(u, v)⌋) = "1") then {
        if (w_i = 0) then embed w_i by (5);
        else embed w_i by (4);}}
    apply one clock to LFSR; }
end{Watermark_Embedding}
```

Algorithm 2: Watermark embedding procedure.

```
Procedure{Watermark_Extraction}
begin {
  initialize LFSR with a key (key_embed);
  for i = 0 to 2^{m+n} − 1 {
    if (LFSR_p(i) = "1") then {
      if(⌊(a''_b(u, v) + Δ_b(u, v)/2)/Δ_b(u, v)⌋ = even) then
          w'_i = "0" by (6);
      else w'_i = "1" by (6); }}
  initialize LFSR with key (key_LFSR);
  for I = 1 to 2^{m+n} − 1 {
    x_i = z_{m+n−1}2^{m−1} + z_{m+n−2}2^{m−2} + ··· + z_{n+1}2^1 + z_n2^0;
    y_i = z_{n−1}2^{n−1} + z_{n−2}2^{n−2} + ··· + z_1 2^1 + z_0 2^0;
    w''(x_i, y_i) = w'(i); }}
    apply one clock to LFSR; }
end{Watermark_Extraction}
```

Algorithm 3: Watermark extraction procedure.

$a_b''(u,v) = r_b(u,v) \times \Delta_b'(u,v)$, where $\Delta_b'(u,v)$ is the quantization step for real compression, and the result is analyzed by (6) where m and n are arbitrary integer, which decides the extracted watermark bit w_i'. Note that Algorithm 3 includes the process to decide existence of a watermark bit when a serial LFSR bit is used. Also, it includes the step to reform the 2D watermark data

$$w_i' = \begin{cases} 0, & (2m-1)\Delta_b(u,v) \le a_b''(u,v) < (2m+1)\Delta_b(u,v), \\ 1, & (2n+1)\Delta_b(u,v) \le a_b''(u,v) < (2n+3)\Delta_b(u,v). \end{cases} \tag{6}$$

The extraction process needs only a few decision tasks so that its speed degradation on the decoding time is negligible, even though the watermark is not necessarily extracted in each decoding process.

3. Applying the Proposed Watermarking Scheme

Since the proposed scheme uses the properties of the coefficients in the frequency domain resulting from DCT or DWT and accomplishes the watermarking during scalar quantization, it can be applied into the JPEG/MPEGs or JPEG2000 standards without any modification. It also has inherently an adaptability property for image and video application because it is applicable to a predefined quantizer with any quantizaton step size.

All of JPEG and MPEGs use DCT and scalar quantization for intra- or interframes. Thus our scheme is applicable to any of them because ours concerns only with DCT and quantization. To obtain higher robustness in watermarking on a DCT domain, a DC coefficient which is less sensitive to a frequency change is a good candidate for watermarking [8]. Also watermarking in DC coefficients would degrade less image quality than AC coefficients, which was approved by a quantitative analysis [8]. Thus, we apply the proposed watermarking scheme into the DC coefficients after DCT for experimental purpose, without suggesting a new scheme to find the watermarking positions.

Each of the resulting sub-bands from DWT in JPEG2000 has the spatial information of the original image as well as the corresponding frequency band information. Thus, differently from DCT, one or a few subbands are usually selected to position the watermark. With considering the human visual system and difference in the quantization step in each subband, we predecided the four lowest subbands as the mark space candidates. Then, we performed pre-experiments with about 500 test images to see which sub-band is the robust for attacks. In the experiments, 4-level DWT has been performed and JPEG compressions in various strengths were applied as the attack processes. The experimental results are summarized in Table 1. In the table, XY_k means a particular sub-band after k-level DWT, where $(X \in \{L,H\})$ and $(Y \in \{L,H\})$ are the horizontal and vertical tranformation, respectively, with $L(H)$ meaning the result from low- (high-) pass filtering. It shows that the lowest DC subband (LL_4) is most robust, but usually it does not quantized. Thus, we chose the next robust subband, LH_4, as the watermarking sub-band.

Table 1: Subband robustness in JPEG.

JPEG Quality	Error ratio (%)			
	LL_4	LH_4	HL_4	HH_4
12	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.5
8	0.0	0.3	0.3	20.9
6	0.0	3.6	3.3	34.3
4	2.9	23.8	24.7	55.5
2	1.6	43.0	48.8	69.2
0	36.7	58.2	65.4	76.1

(a) (b)

Figure 2: JPEG2000 watermarking example for Lena image: (a) before, (b) after watermarking.

4. Experimental Result

The proposed embedding/extracting schemes and the arbitrarily assumed positioning schemes were implemented in C++ language on Pentium IV 2.0 GHz CPU. For the test, we used about 500 gray-scale images of pixels in size. Among them, we chose Lena image for demonstration because it contains both low and high frequency components. As the watermark, we used a binary image of 512×512 pixels in size which has a special logo for visual confirmation after extracting it. For visibility information we estimated Peak Signal to Noise Ratio (PSNR). For robustness against attacks to an image the error ratios of the extracted watermarks to the original ones were estimated for various malicious and nonmalicious attacks such as JPEG and JPEG2000 compression in different compression ratios, Gaussian noise addition, sharpening, blurring, and various amounts of cropping.

Figure 2 shows the resulting Lena image example before (31.54 dB in PSNR) and after (31.50 dB) watermarking for JPEG2000, in which the quality degradation (0.04 dB) by watermark is quite negligible. In Table 2, the error ratios of the extracted watermarks after each attack to the images compressed by JPEG or JPEG2000 (J2K) are summarized, where all the ratios are the average values for the 500 test images. It shows that the proposed scheme has an excellent

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

Figure 3: Various examples of extracted watermarks: (a) original watermark, the extracted watermark with error ratio of (b) 1.5%, (c) 5.6%, (d) 10.7%, and (e) 14.2%.

Table 2: Error ratios resulting from various attacks for JPEG and JPEG2000 watermarking.

Attack			Error ratio(%)	
	JPEG	J2K	JPEG	JPEG2K
JPEG Quality (%)	40	60	0	0
	20	40	0	0
	0	20	0	0.8
JPEG2000 (bpp)		1	0.3	0
		0.5	0.3	0
		0.25	0.7	0
Gaussian Noise (%)		3	3.5	1.7
		5	5.5	4.3
Sharpening			2.2	0.7
Blurring			2.7	1.3
Cropping (%)		10	1.5	1.5
		15	3.2	3.8
		20	5.7	5.0

Table 3: Comparison to Yasein's scheme.

Attack		Error ratio (%)	
		Yasein's	Ours
JPEG quality (%)	60	0	0
	40	1	0
	20	4	0.8
JPEG2000 (bpp)	0.5	0.5	0
	0.4	0.7	0
	0.2	14	3.5

property on robustness for all the considered attacks such that even for the 5% Gaussian noise addition, the error ratio was only 5.5% for JPEG watermarking and 4.7% for JPEG2000 watermarking. In Figure 3, various examples of the extracted watermarks according to their error ratios are shown. As can be seen in the figures, an watermark with error ratio bigger than 15% is hard to recognize.

To show the excellence of our scheme, we compared the robustness with the algorithm previously proposed by Seo et al. [4] which is the most similar to ours. The results are shown in Table 3, which includes the JPEG attacks and JPEG2000 attacks. This table shows that for all the attacks ours show 4∼5 times better in performance than Yasein's.

5. Conclusion

In this paper, we proposed an adaptive blind watermark embedding/extracting scheme which is applicable for both JPEG/MPEGs and JPEG2000 standard compression technologies. It uses the properties of the scalar quantization to comply the compression technologies such that a watermark bit is embedded by a value-shift method for the frequency transformed coefficients. Also it includes a scheme to select coefficients in the mark space randomly with a LFSR.

Experimental results with about 500 test images showed that the proposed scheme has enough robustness less than the error ratio of 5% and very low visibility of the embedded watermark. Even in comparison with the most similar previous method, ours showed much better performance.

Consequently, we expect the proposed scheme to be used as a good watermarking scheme with an appropriate watermark positioning scheme in the application areas that need high invisibility and robustness against various attacks of the embedded watermark.

Acknowledgment

This work was supported by the IT R&D program of KEIT. [KI002058, Signal Processing Elements and their SoC

Developments to Realize the Integrated Service System for Interactive Digital Holograms.]

References

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

[2] M.-S. Hsieh, D.-C. Tseng, and Y.-H. Huang, "Hiding digital watermarks using multiresolution wavelet transform," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 5, pp. 875–882, 2001.

[3] http://www.jpeg.org/.

[4] Y.-H. Seo, S.-Y. Choi, S.-H. Park, and D.-W. Kim, "A digital watermarking algorithm using correlation of the tree structure of DWT coefficients," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E87-A, no. 6, pp. 1347–1354, 2004.

[5] M. S. Yasein and P. Agathoklis, "A wavelet-based blind and readable image watermarking algorithm," in *Proceedings of the 36th Asilomar Conference on Signals Systems and Computers*, vol. 2, pp. 1215–1219, Pacific Groove, Calif, USA, November 2002.

[6] S. Zafeiriou, A. Tefas, and I. Pitas, "Blind robust watermarking schemes for copyright protection of 3D mesh objects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 5, pp. 596–607, 2005.

[7] http://www.itscj.ipsj.or.jp/sc29/.

[8] J. Huang, Y. Q. Shi, and Y. Shi, "Embedding image watermarks in DC components," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 974–979, 2000.