Expert Systems with Applications 38 (2011) 14357–14366

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An efficient wavelet-tree-based watermarking method ☆

Ray-Shine Run [a], Shi-Jinn Horng [b,*], Wei-Hung Lin [b], Tzong-Wann Kao [c], Pingzhi Fan [d], Muhammad Khurram Khan [e]

[a] Dept. of Electronic Engineering, National United University, Miao-Li 36003, Taiwan
[b] Dept. of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan
[c] Dept. of Electronic Engineering, Technology and Science, Institute of Northern Taiwan, Taipei, Taiwan
[d] Institute of Mobile Communications, Southwest Jiaotong University, Chengdu, Sichuan 610031, PR China
[e] Center of Excellence in Information Assurance, King Saud University, Saudi Arabia

ARTICLE INFO

Keywords:
Watermark
Blind watermarking method
Copyright protection
Local significant difference
Wavelet tree

ABSTRACT

This paper proposes a blind watermarking scheme based on wavelet tree quantization for copyright protection. In such a quantization scheme, there exists a large significant difference while embedding a watermark bit 1 and a watermark bit 0; it then does not require any original image or watermark during watermark extraction process. As a result, the watermarked images look lossless in comparison with the original ones, and the proposed method can effectively resist common image processing attacks; especially for JPEG compression and low-pass filtering. Moreover, by designing an adaptive threshold value in the extraction process, our method is more robust for resisting common attacks such as median filtering, average filtering, and Gaussian noise. Experimental results show that the watermarked image looks visually identical to the original, and the watermark can be effectively extracted.

1. Introduction

The Internet has popularized tremendously fast in our life in the last decade. Due to the digitalization of documents, images, music, videos, etc., people can access and propagate them easily via the network. The watermarking technique has been widely applied to digital contents for copyright protection, image authentication, proof of ownership, etc. This technique embeds information so that it is not easily perceptible; that is, the viewer cannot see any information embedded in the contents. The issue here is the detection of the existence of watermarks in the digital contents to prove ownership. There are several important issues in the watermarking system. First is the transparency. The embedded watermark should not degrade the quality of the image and should be perceptually invisible to maintain its protective secrecy. Second is the robustness. The watermark must be robust enough to resist common image processing attacks and not be easily removable; only the owner of the image should be able to extract the watermark. The third issue is the blindness. A watermarking technique is referred to as blind if the original image and watermark are not needed during extraction (Chen, Horng, & Lee, 2005; Dugad, Ratakonda, & Ahuja, 1997; Ganic & Eskicioglu, 2004; Inoue, Miyazaki, Yamamoto, & Katsura, 1998; Tsai, Yu, & Chen, 2000).

The spatial and spectral domains are two common methods for watermarking. In the spatial domain, a watermark is embedded in the selected areas on the texture of a host image (Mukherjee, Maitra, & Acton, 2004; Nikolaidis & Pitas, 1998). In the spectral domain, a host image is transformed to the frequency domain using methods such as Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT). A watermark is embedded in the mid-frequency to ensure the transparency and robustness of a watermarked image at the same time (Hernandez, Amado, & Perez-Gonzalez, 2000; Mahmood & Selin, 2006; Pickholtz, Schilling, & Milstein, 1982). See Potdar, Han, and Chang (2005), Sin-Joo and Sung-Hwan (2001), for details.

Cox, Kilian, Leighton, and Shamoon (1996, 1997) suggested inserting the watermark into the perceptually significant portion of the whole DCT-transformed image. The significant portion is a predetermined range of low frequency components excludes the DC component. This watermarking scheme has been shown to be robust against common attacks such as compression, filtering, and cropping. In Podilchuk and Zeng (1998), an image-adaptive watermarking scheme was proposed. Podilchuk and Zeng (1998) improved Cox's method and added the visual model of the just noticeable difference (JND) to select the maximum length and maximum power watermark sequence. The above methods do not fall

☆ This work was supported in part by the National Science Council under contract number NSC-98-2219-E-011-002, NSC-98-2221-E-011-133-MY3, NSC-98-2923-E-011-004-MY3, NSC-99-2916-I-011-002-A1, and it was also partially supported by the 111 Project under the Grant No. 111-2-1.

* Corresponding author. Tel.: +886 2 27376700; fax: +886 2 27301081.
E-mail addresses: run5116@ms16.hinet.net (R.-S. Run), horngsj@yahoo.com.tw (S.-J. Horng), weber3013@yahoo.com.tw (W.-H. Lin), tkao@tsint.edu.tw (T.-W. Kao), p.fan@ieee.org (P. Fan), mkhurram@ksu.edu.sa (M.K. Khan).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.03.024

into the blind watermarking scheme, since they require the original image for watermark retrieval.

Wang, Su, and Kuo (1998) proposed a watermarking method, in accordance with the multi-threshold wavelet coding (MTWC) (Wang & Kuo, 1997); the successive subband quantization (SSQ) is adopted in this method to search for the significant coefficients. The watermark is added by quantizing the significant coefficient in the significant band by using different weights. Hsieh, Tseng, and Huang (2001) proposed a watermarking method based on the qualified significant wavelet tree (QSWT). The QSWT is derived from the embedded zerotree wavelet algorithm (EZW) (Shapiro, 1993). The watermark is embedded in each of the two subbands of the wavelet tree. In Temi, Choomchuay, and Lasakul (2005) improved Hsieh et al.'s method, and the watermark is embedded in the LL subband. In Aboofazeli, Thomas, and Moussavi (2004), the watermark was embedded in wavelet coefficients, which are selected as they correspond to the points located in the neighborhood with maximum entropy. Lin et al. (2008) proposed a block-based significant difference quantization watermarking. Every seven wavelet coefficients in one subband are grouped into a block, the watermark bit is embedded into a block by quantizing the difference between two maximum wavelet coefficients. The embedding capacity is constrained by the block size.

Several wavelet tree-based watermarking methods (Lien & Lin, 2006; Lin, Wang, & Horng, 2009; Wang & Lin, 2004) are proposed recently. Wang and Lin (2004) grouped two wavelet trees into a super tree, and each watermark bit is embedded using two super trees. One of the two trees is quantized with respect to a quantization index, and both trees exhibit a large statistical difference between the quantized tree and the unquantized tree; the difference can later be used for watermark extraction. Lien and Lin (2006) improved Wang and Lin (2004) method by using four trees to represent two watermark bits in order to improve visual quality. One of the four trees is quantized according to the binary value of the two embedded watermark bits. But these methods (Lien & Lin, 2006; Wang & Lin, 2004) cannot effectively resist low-pass filtering such as the median filtering or the Gaussian filtering. In Lin et al. (2009), the watermark is embedded into insignificant coefficients of a wavelet tree. Although the method (Lin et al., 2009) can improve the quality of the watermarked image, the robustness of the watermarking is decreased.

In this paper, we propose a blind watermarking method based on wavelet tree quantization. A watermarking technique is denoted as *blind* if the original image is not needed for watermark extraction process (Wang & Lin, 2004). In previous researches, the watermark embedded in the significant coefficients was found to be robust. The common issue is the use of blind detection to find out whether the extracting order is the same as the embedding order (Aboofazeli et al., 2004; Cox et al., 1996, 1997; Hsieh et al., 2001; Podilchuk & Zeng, 1998; Temi et al., 2005; Wang & Huo, 1997; Wang et al., 1998). Hence, we propose a watermarking method which embeds a watermark bit in the maximum wavelet coefficient of a wavelet tree. The proposed method is different from those in Lin et al. (2008), Lien and Lin (2006), and Wang and Lin (2004) which use two trees or one block to embed a watermark bit. We embed the watermark by scaling the magnitude of the significant difference between the two largest wavelet coefficients in a wavelet tree to improve the robustness of the watermarking. The proposed method is different with Lin et al. (2008) that used block-based quantization. It is not constrained in a block size and can promote the capacity of embedding. The trees are so quantized that they exhibit a large enough energy difference between a watermark bit 1 and a watermark bit 0, which is then used for watermark extraction. During extraction, an adaptive threshold value is designed. A watermark bit 1 is extracted if the significant difference is greater than the adaptive threshold value; otherwise, a watermark bit 0 is extracted. Experimental results show that the proposed method is very efficient for resisting various kinds of attacks.

The rest of this paper is organized as follows: In Section 2, the wavelet trees and the proposed watermarking quantization method are introduced. The decoder and extraction algorithms are given in Section 3. In Section 4, the performance is analyzed by applying various attacks to the watermarked images, including nongeometric and geometric attacks. The conclusion is given in Section 5.

2. Watermarking by quantization of wavelet trees

2.1. Wavelet trees

A host image of size n by n is transformed into wavelet coefficients using the 4-level 5/3 discrete wavelet transform (DWT). With a 4-level decomposition, we have 13 frequency bands as shown in Fig. 1 for watermark embedding and the corresponding watermark extracting procedure is also shown in Fig. 2. The parent–child relationship can be connected between these subnodes to form a wavelet tree (Shapiro, 1993). If the root consists of more than one node, then an image will have many wavelet trees after the DWT, see Fig. 3. For example, we have three subbands LH4, HL4, and HH4 as roots, each subband has $n/2^4 \times n/2^4$ nodes. The total wavelet trees are $3 \times (n/2^4 \times n/2^4)$ after an image of size n by n is transformed by a 4-level wavelet transform. For example, when $n = 512$, there are 85 coefficients for a wavelet tree constructed from a node in LH4 to LH1 following the parent–child relationship.

A higher resolution level (such as level 3 in Fig. 1) has more significant coefficients than a lower resolution level (such as level 2 in Fig. 1). Using the LL4 subband as a root is not suitable for embedding a watermark, since it is a low frequency band that contains important information about an image and easily causes image distortions. Embedding a watermark in the HH4 subband is also not suitable, since the subband is easily eliminated, for example by a lossy compression. Hence, for the remaining two subbands, there are $S = 2 \times n/2^4 \times n/2^4$ wavelet trees which could be used for embedding. Then, the largest number of watermark bits which can be embedded is S. As for avoiding attacks such as low pass filters, in our proposed method we only need the largest two coefficients; these two coefficients are selected from one coefficient of LH4 and four coefficients of the same orientation in the same spatial location in LH3 as shown in Fig. 3.

2.2. The preprocess

In order to enhance the security for the embedded watermark, we shuffle the wavelet trees in a pseudorandom manner. A pseudorandom order of the numbers from 1 to S can be obtained by repeating two random numbers $S/2$ times; each time we generate

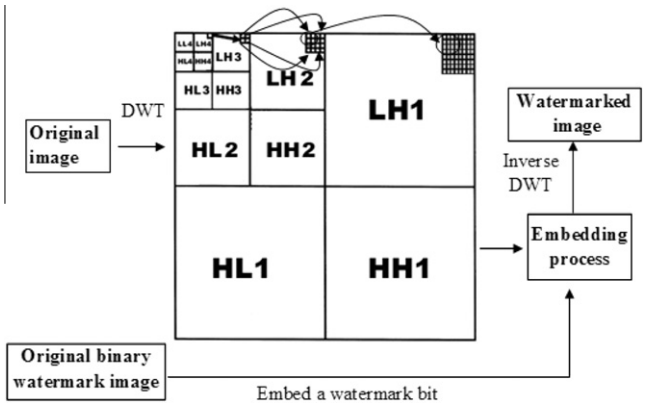

Fig. 1. The watermark embedding procedure.

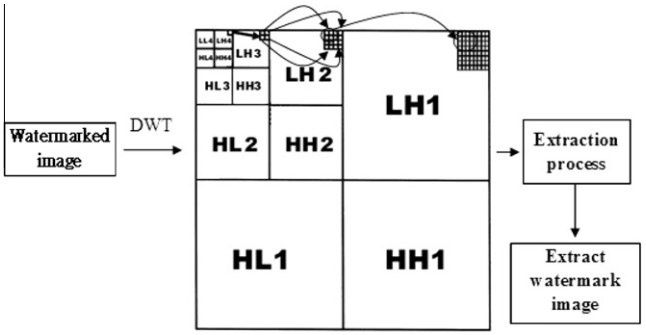

Fig. 2. Watermark extraction procedure.

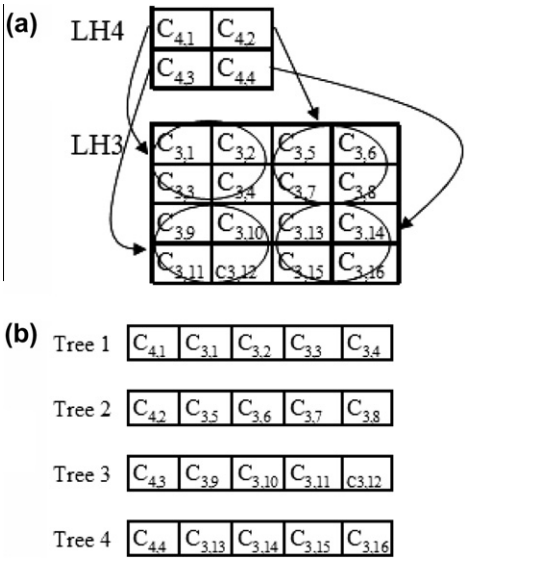

Fig. 3. (a) A wavelet coefficient from LH4, and four wavelet coefficients from LH3. (b) Based on (a), there are four wavelet tress each constructed from one coefficient of LH4 and four coefficients of LH3, respectively.

two random numbers using the same seed and using modulo $S + 1$. Note that we use two generated random numbers to denote the indexes of two wavelet trees; respectively, and exchange the two trees with the corresponding indexes. For example, suppose at the 300th time the two generated random numbers are 100 and 500, respectively, then the 100th wavelet tree and the 500th wavelet tree are exchanged. A binary watermark image W composed of size N_w ($\leqq S$) bits is embedded. We represent each watermark bit as 1 or 0, and use a pseudorandom function with another seed to shuffle N_w bits. According to the watermark bits embedded later, we select N_w non-overlapping wavelet trees in turn from S trees and compute the global average significant difference of the total number of the N_w wavelet trees using Eq. (1)

$$\varepsilon = \left\lfloor \frac{1}{N_w} \sum_{i=1}^{N_w} max_i - sec_i \right\rfloor, \tag{1}$$

where ε is the global average significant difference in all N_w wavelet trees; $\lfloor \cdot \rfloor$ is the floor function; max_i is the local maximum wavelet coefficient of the ith wavelet tree; sec_i is the local second maximum wavelet coefficient of the ith wavelet tree, $1 \leqq i \leqq N_w$. The difference between the two largest coefficients is positive. Quantization does not influence original rank of these five coefficients, since the value of the two largest coefficients are nearest. From Fig. 3, suppose, $N_w = 4$, $C_{4,1}, C_{4,2}, \ldots, C_{4,4} = 10, -21, 23, 31$ and $C_{3,1}, C_{3,2}, \ldots, C_{3,16} =$ 15, 4, 20, 2, 11, 51, 3, 5, 10, −7, 13, 18, 2, 1, 23, 41 (for simplicity, we use integers). Then, $\varepsilon = 15$ as computed by Eq. (1).

2.3. Watermark embedding

The idea is shown in Fig. 4. Let max_i and sec_i be the local maximum wavelet coefficient and the local second maximum wavelet coefficient in a wavelet tree; the difference between both of them is named as the local significant difference. We randomly select 512 wavelet trees from S wavelet trees. We select a threshold value β as an increment for quantization. The threshold value β provides a tradeoff between the strength of the watermark and the quality of the watermarked image. The larger β is, the more heavily quantized are the wavelet trees; using a larger β trades signal-to-noise ratio (SNR) quality of the image for more robustness of the watermark (Wang & Lin, 2004). In Fig. 5, when $\beta > 7$, the PSNR of Airport image is less than 40. For this reason, we select $\beta = 7$. Each time we embed a watermark bit, we quantize the maximum coefficient in a wavelet tree

$$pmax_i = \begin{cases} 0, & \text{if } max_i < 0 \\ max_i; & \text{otherwise.} \end{cases} \tag{2}$$

where $pmax_i$ is the adjusted max_i value before embedding. In Cox et al. (1997) we know a watermarking can effectively resist attacks, when a watermark is embedded in significant coefficients of an image which is transformed into frequency domain. Because some maximum coefficients in a wavelet tree may be negative, we then let the new maximum coefficient be positive. The reason is that a positive value has higher robustness than a negative value under attacks; moreover, it will result in two largest coefficients more significant if we modify the maximum coefficient from a negative value to a positive value. When the difference between the two largest coefficients is more significant, it will be more accurate during extracting watermark. To achieve the new maximum coefficient to be positive and to decrease the distortion of watermarked image due to quantization, the new maximum coefficient is set to a smallest positive value zero here. We then define

$$\Delta_i = pmax_i - sec_i, \tag{3}$$

where Δ_i denotes the significant difference between the maximum coefficient and the second maximum coefficient in the ith wavelet tree.

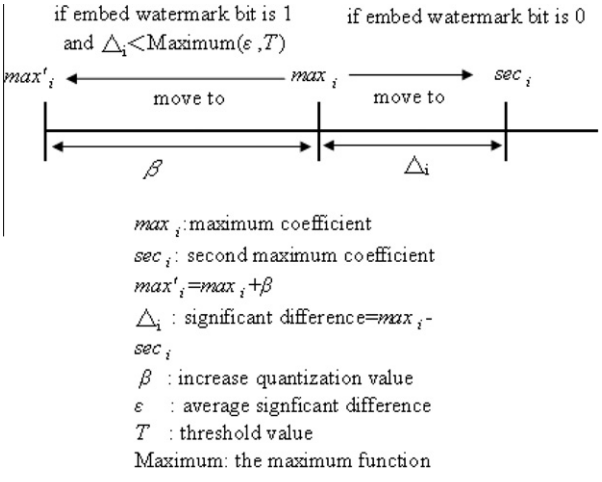

Fig. 4. The quantization method: If the embedded watermark bit is 1, the local maximum coefficient is not quantized under $\Delta_i \geqslant Maximum(\varepsilon, T)$ and is quantized under $\Delta_i < Maximum(\varepsilon, T)$. If the embedded watermark bit is 0, the local maximum coefficient is quantized by setting the local maximum coefficient to the local second maximum coefficient.

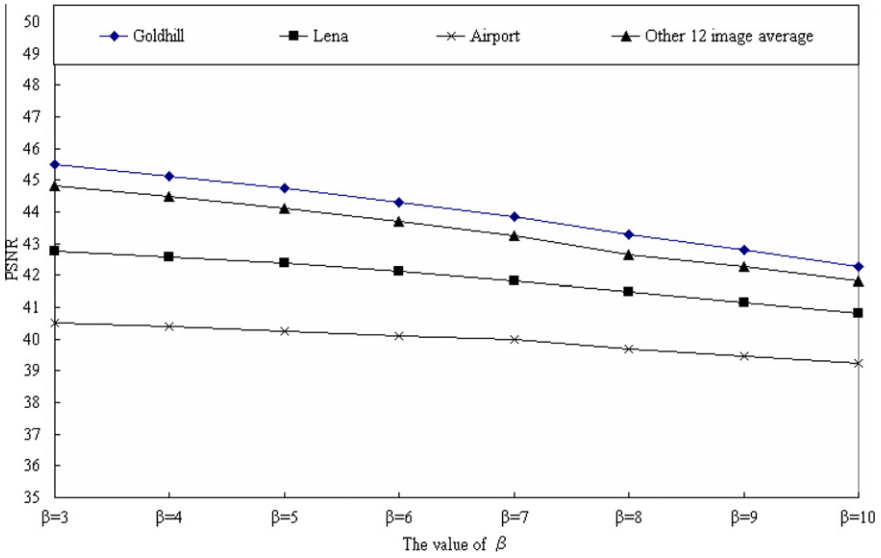

Fig. 5. PSNR vs. the value of β.

When we embed a watermark bit 1, $pmax_i$ is quantized by

$$max_i^{new} = \begin{cases} pmax_i, & \text{if } (\Delta_i \geqslant \text{Maximum}(\varepsilon, T)); \\ pmax_i + \beta, & \text{if } (\Delta_i < \text{Maximum}(\varepsilon, T) \text{ and } pmax_i \text{ is located at the highest resolution level}); \\ pmax_i + \beta \times \gamma, & \text{if } (\Delta_i < \text{Maximum}(\varepsilon, T) \text{ and } pmax_i \text{ is not located at the highest resolution level}), \end{cases} \tag{4}$$

where max_i^{new} denotes the new maximum coefficient in the ith wavelet tree after embedding the watermark bit 1 and Maximum(.) denotes a maximum function. The maximum significant coefficient is quantized and incremented by β if $\Delta_i <$ Maximum(ε, T); otherwise, it is kept the same as before. When $\Delta_i <$ Maximum(ε, T), the reason for not quantizing the maximum local significant coefficient is that we do not want to increase the distortion of the image. Some images may have smaller ε; this means that their significant difference is not obvious. We need an extra parameter T to improve the robustness. The larger T is, the higher probability the $pmax_i$ is quantized to a larger value; meanwhile, the more distortion of the image will be as well. For example, let $\varepsilon = 12$, and $\Delta_i = 13$. Suppose T is set to be less than ε, such as $T = 11$, $pmax_i$ will not be quantized as $\Delta_i > \varepsilon = 12$. On the other hand, if T is set to be larger than ε, such as $T = 14$, $pmax_i$ will be quantized and increased by β as $\Delta_i < T = 14$. As stated in Section 2.2, if $pmax_i$ is located at the lower resolution level then it has worse robustness than when it is located at the higher resolution level. According to the band sensitivity (Lewis & Knowles, 1992), the coefficients which are quantized at different resolution levels are given different weights. The quantized coefficient at the lower resolution level is given a heavier weight than that at the higher resolution level. Hence, if $pmax_i$ is not located at the highest resolution level, we will quantize it by adding an energy of γ times β, here γ is a scale parameter, and we set $\gamma = 1.5$.

On the contrary, we embed a watermark bit 0 according to Eq. (5)

$$max_i^{new} = \begin{cases} sec_i^{new} = 0, & \text{if } sec_i < 0 \\ sec_i; & \text{otherwise}, \end{cases} \tag{5}$$

where sec_i^{new} denotes the new second maximum coefficient in the ith wavelet tree after embedding the watermark bit 0. As we can see, the value of $pmax_i$ is quantized by decreasing it to the local second maximum sec_i and the new Δ_i will be equal to 0. Based on this strategy, there exists a large energy difference between embedding watermark bit 1 and watermark bit 0.

The detailed embedding algorithm is listed as follows:

Input: An original image and a watermark image W.
Output: A watermarked image.
Step 1. A binary watermark W comprised of N_w bits is randomly shuffled first using a seed.
Step 2. A 512×512 original image is decomposed using the 4-level DWT.
Step 3. The wavelet coefficients are grouped into the wavelet trees; the wavelet trees are then randomly shuffled using another seed.
Step 4. Compute the global average significant difference ε for all N_w wavelet trees using Eq. (1).
Step 5. Set $i = 1$.
Step 6. While ($i \leqq N_w$) do
  6.1. seek the values of max_i and sec_i from the wavelet tree i;
  6.2. compute the significant difference;
  6.3. embed the watermark bit by Eqs. (2)–(5).
  6.4. set $i = i + 1$.
Step 7. All wavelet trees are randomly reshuffled by using the same seed as used in Step 3.
Step 8. Transform the modified wavelet coefficients by inversing the DWT, and obtain a watermarked image.

3. Design of watermark decoder

3.1. The decoder design

In the proposed method, neither an original image nor an original watermark image is required for the extraction process. During the embedding process, embed a watermark bit 1 by adding an energy β (or $\beta \times \gamma$) to a maximum wavelet coefficient in the wavelet tree, and embed a watermark bit 0 by setting $max_i = sec_i$. Hence, if the wavelet tree was embedded a watermark bit 0, the local

significant difference between the largest two coefficients will be close to zero; otherwise, the wavelet tree was embedded a watermark bit 1, the local significant difference between the largest two coefficients will be greater than β.

To be more specific, we define the empirical CDF of the significant difference as

$$f(y) = \text{Prob}[(max'_i - sec'_i) \leqslant y], \tag{6}$$

where max'_i and sec'_i are the local maximum and second maximum wavelet coefficients of the ith wavelet tree in the watermarked image, respectively, y is a threshold value for extracting watermark.

Based on the 15 commonly images obtained from USC-SIPI (each using 512 wavelet trees, totally 15×512 wavelet trees), the empirical CDFs without attacks and with attacks are plotted in Fig. 6 for both embedded watermark bit 1 and bit 0, respectively.

In Fig. 6, for the curve of embedding a watermark bit 0, the CDF reaches 0.93 while the local significant difference is less than or equal to 1; on the other hand, for the curve of embedding a watermark bit 1, the CDF is increased from bin 7. The reason for the former the CDF not reaching 1 is that during embedding, we had modified the local maximum coefficients in the wavelet trees and this leads to a distorted watermarked image. We use y as the threshold to identify the embedded watermark bit. For a wavelet tree, when the local significant difference of it is satisfied with the condition $(max'_i - sec'_i) \leqslant y$, there is a large probability that it was embedded a watermark bit 0; otherwise, it was embedded a watermark bit 1. For example, when y is set to 5, the probability $f(y)$ for extracting the watermark bit 0 is 1 and that for extracting the watermark bit 1 is 0, respectively. Therefore, how to find a suitable y value becomes very important during the extraction process.

In some attacks, there is still a large difference for the CDFs between embedding a watermark bit 0 and a watermark bit 1. Fig. 6 shows the empirical CDFs for four types of attacks; JPEG compression with a quality factor (QF) of 50, median filtering of size 5×5 and 7×7, average filtering of size 5×5 and 7×7, and Gaussian filtering.

As shown in Fig. 6, the significant difference between embedding a watermark bit 0 and a watermark bit 1 will be much closer under the blur image processing attacks by median filtering and average filtering of sizes 5×5, and 7×7, respectively. In such a case, it would be hard to extract the embedded watermark bits. In order to extract watermark bits correctly, we define the value of y by Eq. (7)

$$Y = \left\lfloor \frac{1}{N_w \times \alpha} \sum_{j=1}^{N_w \times \alpha} \varphi_j \right\rfloor, \tag{7}$$

where

$$\phi = \{max'_1 - sec'_1, max'_2 - sec'_2 \ldots, max'_i - sec'_i\},$$
$$\text{for } i = 1, 2, \ldots, N_w.$$

$$\varphi(\phi) = \{\varphi_1, \varphi_2 \ldots \varphi_{N_w}\}, \quad \text{where } \varphi_1 < \varphi_2 < \cdots < \varphi_{N_w}.$$

Here, ϕ is the set of $\{max'_1 - sec'_1, max'_2 - sec'_2, \ldots, max'_i - sec'_i\}$; φ is a sort function; α is the scale parameter, $0 < \alpha \leqq 1$. α is crucial to y. α is used to determine how many percentages of the significant difference in ϕ can be used for the average. Hence, α marks the minimal y value for extracting the watermark. For example, in Fig. 6, when y is among 3 and 7, it can effectively extract the watermark from embedded images without attacks. On the contrary, when y is among 2 and 3, it would be better to extract the watermark in attacked images in Fig. 6. Therefore, when y is set to 3, it can effectively extract the watermarks both in normal images and attacked images simultaneously in Fig. 6. The larger the α is, the larger the y is. Suppose all embedded watermark bits are 1 in the watermark. This means that the difference between the maximum wavelet coefficients and the second maximum wavelet coefficients for any embedded wavelet tree is greater than β. The value of α should be set as small as possible to avoid extraction errors (see Eq. (8)); the reason for this is that it can exclude those big significant differences of the embedded wavelet trees in Eq. (7). On the other hand, if all embedded watermark bits are 0 in the watermark, the value of α must be set as large as possible. Therefore, α is sensitive to the content of the watermark. Fig. 7 shows the relationship between α and the content of the watermark embedded in the Lena image.

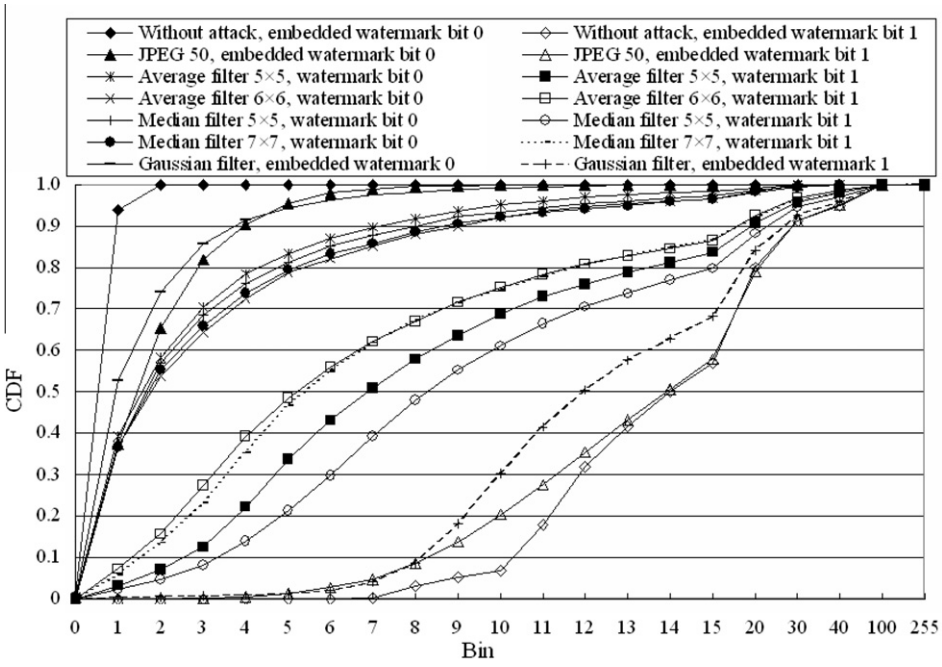

Fig. 6. CDFs of the significant difference of $(max_i - sec_i)$ without attacks and with attacks.

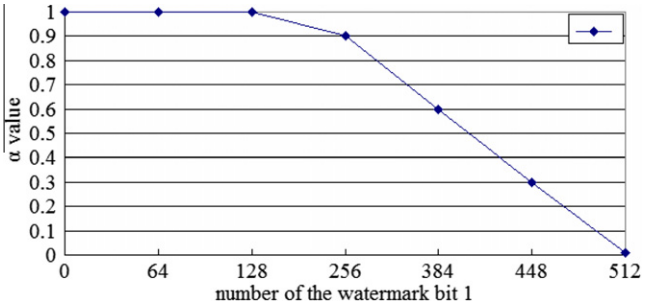

Fig. 7. The relationship between α and the content of a watermark embedded in the Lena image.

From α, we can determine y; for example, $N_w = 5$, $\phi = \{0.3, 2.2, 58.4, 130.5, 0.04\}$, hence $\varphi(\phi) = \{0.04, 0.3, 2.2, 58.4, 130.5\}$. Suppose $\alpha = 0.6$, then the value of y is 0.85 $((0.04 + 0.3 + 2.2)/(5 \times 0.6))$ as obtained via Eq. (7). Note that a wavelet tree is embedded with either a watermark bit 1 or 0, which is known in advance from the figures shown in Fig. 6. Without knowing the embedded information, as opposed to Fig. 6, we also plot the CDFs for the significant difference in Fig. 8 for the 15 images obtained from USC-SIPI, the images being under normal or attacks. In Fig. 8, while there were no attacks for the watermarked images, the number of bin 0 is 16 ($0.21\% \times 512 \times 15$), the number of bin 1 is 2208 ($28.75\% \times 512 \times 15$), the number of bin 2 is 143 ($1.86\% \times 512 \times 15$), the number of bin 3 is 2 ($0.03\% \times 512 \times 15$), the number of bin 7 is 9 ($0.11\% \times 512 \times 15$), the number of bin 8 is 156 ($2.03\% \times 512 \times 15$), the number of bin 9 is 115 ($1.5\% \times 512 \times 15$), the number of bin 10 is 86 ($1.12\% \times 512 \times 15$), the number of bin 11 is 587 ($7.65\% \times 512 \times 15$), the number of bin 12 is 743 ($9.67\% \times 512 \times 15$), the number of bin 13 is 517 ($6.73\% \times 512 \times 15$), the number of bin 14 is 446 ($5.81\% \times 512 \times 15$), and the number of bin 15 is 348 ($4.53\% \times 512 \times 15$). Since the number of bin 1 is 2208, the significant difference of bin 1 at 0.5 is roughly estimated to be 2208 in average. Similarly, the significant difference of bin 15 at 14.5 is about 348. Hence, by ignoring the number of other significant differences and setting $\alpha = 0.7$, we can roughly estimate y to be 5.52 $((16 \times 0 + 2208 \times 0.5 + 143 \times 1.5 + 2 \times 2.5 + 9 \times 6.5 + 156 \times 7.5 + 115 \times 8.5 + 86 \times 9.5 + 587 \times 10.5 + 743 \times 11.5 + 517 \times 12.5 + 446 \times 13.5 + 348 \times 12.5)/(512 \times 15 \times 0.70) = 6.67)$ as obtained via Eq. (7). Contrary to Fig. 6, we can precisely extract the watermark by setting y to 6. Hence, Eq. (7) is workable for defining the threshold value y under normal or attacks.

On the other hand, when the watermarked images are under a serious attack like median filtering with a mask of size 7×7 in Fig. 8, the number of bin 0 is 9 ($0.12\% \times 512 \times 15$), the number of bin 1 is 1170 ($15.23\% \times 512 \times 15$), the number of bin 2 is 854 ($11.12\% \times 512 \times 15$), the number of bin 3 is 754 ($9.82\% \times 512 \times 15$), the number of bin 4 is 845 ($11\% \times 512 \times 15$), the number of bin 5 is 726 ($9.45\% \times 512 \times 15$), the number of bin 6 is 543 ($7.08\% \times 512 \times 15$), the number of bin 7 is 416 ($5.41\% \times 512 \times 15$), and the number of bin 8 is 59 ($0.77\% \times 512 \times 15$). Following Eq. (7) by ignoring other significant differences and setting $\alpha = 0.7$, we get the y value to be 2.99 $((0 \times 9 + 0.5 \times 1170 + 1.5 \times 854 + 2.5 \times 754 + 3.5 \times 845 + 4.5 \times 726 + 5.5 \times 543 + 6.5 \times 416 + 7.5 \times 59)/(512 \times 15 \times 0.7))$. Contrary to Fig. 6, while y is set to 2, this is a reasonable threshold to extract the watermark while it is attacked by median filtering with a mask of size 7×7. Therefore, Eq. (7) is yet workable again for the watermarked image under serious attacks. Note that the adaptive threshold y is dynamically changed by α according to the watermarked image under different conditions.

3.2. Watermark extraction

Following Eqs. (6) and (7), it would be easy to extract the watermark. If the local significant difference is greater than or equal to y, where $0 < y \leqslant \beta$, the embedded watermark bit could be 1; otherwise, the embedded watermark bit could be 0.

The watermark bit can be extracted as

$$\textit{Watermark bit} = \begin{cases} 1 & \textit{if } (max'_i - sec'_i) \geqslant y, \\ 0 & \textit{otherwise}. \end{cases} \tag{8}$$

The detailed extraction algorithm is listed as follows.

Input: A watermarked image.

Output: A binary watermark image.

Step 1. A 512×512 watermarked image is decomposed using the 4-level DWT.

Step 2. Each wavelet trees is constructed as in embedding process; then the wavelet trees are randomly shuffled using the seed which was used in the embedding process.

Step 3. Set $i = 1$.

Step 4. While $(i \leqq N_w)$ do

1. seek the values of max'_i and sec'_i from the wavelet tree i;
2. extract the watermark bit using Eqs. (7) and (8).
3. set $i = i + 1$.

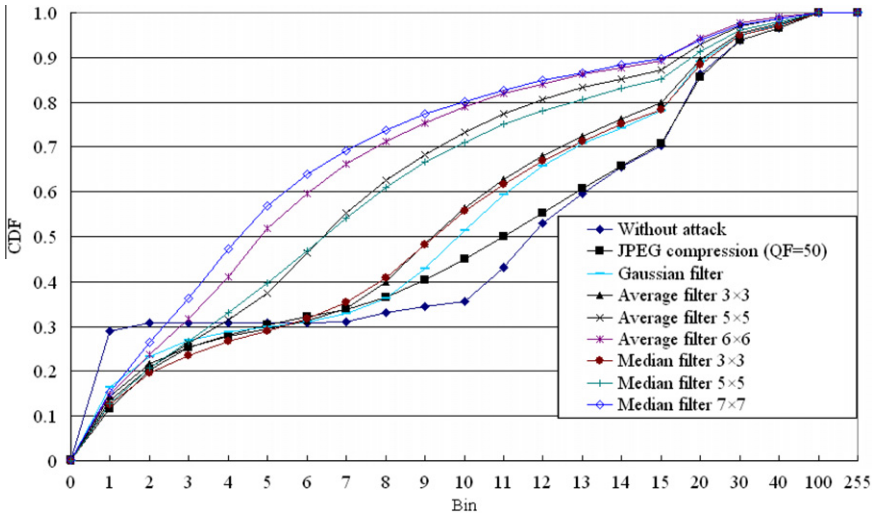

Fig. 8. The CDFs of the significant difference between $(max'_i - sec'_i)$ with attacks and without attack.

Step 5. The extracted watermark is randomly reshuffled with the seed which was used in the embedding process to obtain the binary watermark image.

In the proposed method, the human visual system (HVS) can be incorporated to improve the quality of watermarked images. Although the wavelet coefficients can be properly quantized according to HVS to reduce distortion of watermarked images, the watermark bits are easily detected errors in these wavelet trees of less quantization.

4. Experiment results

We use the peak signal-to-noise ratio (PSNR) to evaluate the quality between the attacked image and the original image. For the sake of completeness, the PSNR formula is formulated as follows:

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255 \times 255}{\frac{1}{H \times W} \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} [f(x,y) - g(x,y)]^2} dB, \tag{9}$$

where H and W are the height and width of the image, respectively. $f(x, y)$ and $g(x, y)$ are the values of the coordinate (x, y) in the original image and the attacked image, respectively.After extracting the watermark, the normalized correlation coefficient (NC) is computed using the original watermark and extracted watermark to judge the existence of the watermark. The value of the NC coefficient is defined as follows:

$$NC = \frac{1}{w_h \times w_w} \sum_{i=0}^{w_h-1} \sum_{j=0}^{w_w-1} w(i,j) \times w'(i,j), \tag{10}$$

where w_h and w_w are the height and width of the watermark. $w(i, j)$ and $w'(i, j)$ are the values of the coordinate (i, j) in the original watermark and the extracted watermark, respectively. Here $w(i, j)$ is set to 1 if it is a watermark bit 1; otherwise, it is set to -1. $w'(i, j)$ is set in the same way. So the value of $w(i, j) \times w'(i, j)$ is either -1 or 1.

We compare NC with the threshold value ρ. If NC $\geqq \rho$, the extracted watermark exists; otherwise, it does not. The probability of the false positive error P_{fp} can be computed by (Kundur, 1998),

$$P_{fp} = \sum_{A=((\rho+1)/2)\times N_W}^{N_W} \binom{N_W}{A} P_E^{N_W - A} (1 - P_E)^A, \tag{11}$$

where P_E is the probability when $w(i, j) \neq w'(i, j)$; it is reasonable to assume $P_E = 0.5$. We can choose an appropriate ρ to meet the requirement. For example, $\rho = 0.23$, $N_w = 1024$, and $P_E = 0.5$. The false positive error P_{fp} is 8.15×10^{-14}.

We use 15 images, namely Lena, Goldhill, and Peppers, and 12 other images (512×512 pixels, 8 bits/pixels) obtained from USC-SIPI for our experiments. For attacking, we use the Stirmark benchmark (Petitcolas, 1997) and PhotoImpact 11 software (Ulead system) tools to simulate common image attacks.

While there is no attack, for the sake of brevity only the Lena image and the binary watermark are shown in Fig. 9. Fig. 10 shows the watermarked image and the extracted result. We pre-determine the scale parameter T at 10, $\beta = 7$, $\gamma = 1.5$, and $\alpha = 0.7$. The PSNRs of the 15 images are shown in Fig. 11, and the watermark can be extracted from these 15 images with NC = 1.

In the following, we consider both geometric and nongeometric attacks. Nongeometric attacks include JPEG compression, low pass filtering, histogram equalization, and sharpening.

JPEG is one of the most used formats in the Internet and the digital camera. The JPEG quality factor is a number between 0 and 100 and associates a numerical value with a particular compression level. When the quality factor is decreased from 100, the image compression is improved, but the quality of the resulting image is significantly reduced. With the different quality factors of JPEG compression, the results are shown in Table 1. From Table 1, the proposed method can correctly extract the watermark while the quality factors are greater than 80 and it becomes worse if the quality factor is decreased. The less the quality factor is, the vague the extracted watermark is.

Fig. 9. (a) The original image of Lena of size 512×512. (b) The original binary watermark of size 32×16.

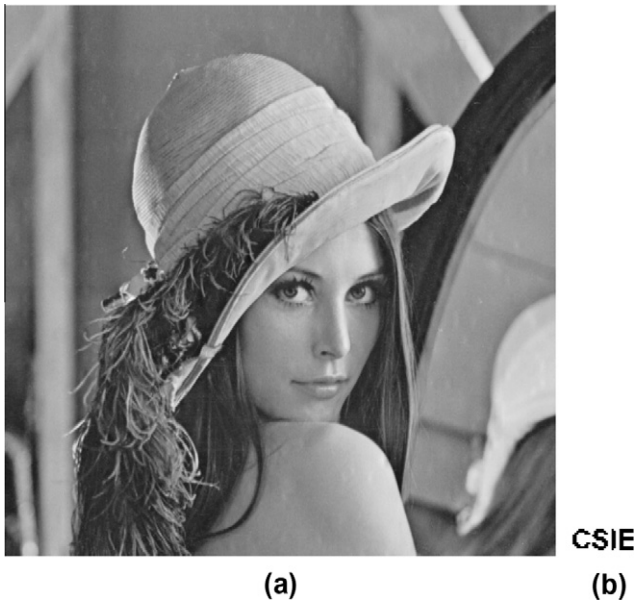

Fig. 10. (a) Watermarked Lena with PSNR = 41.84 dB. (b)The extracted watermark with NC = 1.

For other non-geometric attacks (Petitcolas, 1997), for example median filters, Gaussian filtering, average filtering, sharpening, and histogram equalization, after these attacks, the resulting image are blurred or sharpened on the edge (Gonzalez & Woods, 2002), the results are shown in Table 2. From Table 2, note that the proposed method can effectively resist attacks such as the median filtering

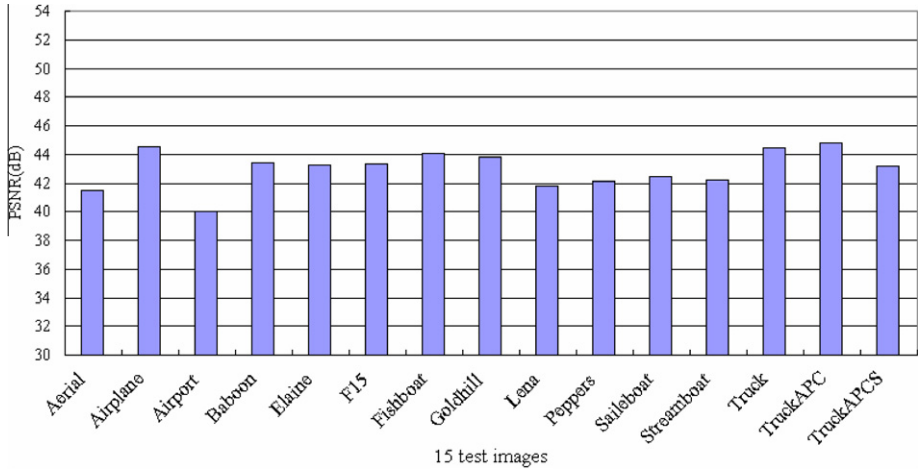

Fig. 11. The PSNRs of 15 watermarked images.

Table 1
Normalized correlation coefficients (NC) after attacks by JPEG compression with the quality factors (QF) 10, 15, 20, 25, 30, 35, 40, 50, 60, 70, 80, 90, 100 in 15 watermarked images.

	QF	10	15	20	25	30	35	40	50	60	70	80	90	100
(a) Lena	NC	0.32	0.53	0.68	0.77	0.85	0.93	0.93	0.96	0.99	0.99	1	1	1
	Extracted watermark	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE
(b) Goldhill	NC	0.36	0.52	0.69	0.78	0.85	0.89	0.92	0.96	0.99	0.99	1	1	1
	Extracted watermark	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE
(c) Peppers	NC	0.36	0.52	0.66	0.78	0.88	0.89	0.95	0.97	0.96	1	1	1	1
	Extracted watermark	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE
(d) Other 12 test images average	NC	0.35	0.52	0.67	0.80	0.86	0.90	0.92	0.96	0.98	0.99	1	1	1

Table 2
Normalized correlation coefficients (NC) after attacks by median filter (3×3, 5×5, 7×7), Gaussian filtering, average filtering attacks (3×3, 5×5, 6×6), sharpening, and histogram equalization in the 15 watermarked images.

(a) Lena	Nongeometric attacks	Median filter			Gaussian filter	Average filter			Sharpening	Histogram equalization
		(3×3)	(5×5)	(7×7)	PSNR = 25.42 dB	(3×3)	(5×5)	(6×6)	PSNR = 25.40 dB	PSNR = 19.44 dB
	NC	0.93	0.84	0.66	0.95	0.95	0.80	0.66	0.99	0.86
	Extracted Watermark	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE
(b) Goldhill	Nongeometric attacks	Median filter			Gaussian filter	Average filter			Sharpening	Histogram equalization
		(3×3)	(5×5)	(7×7)	PSNR = 25.11 dB	(3×3)	(5×5)	(6×6)	PSNR = 25.32 dB	PSNR = 17.78 dB
	NC	0.91	0.70	0.56	0.96	0.93	0.71	0.62	1	0.77
	Extracted Watermark	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE
(c) Peppers	Nongeometric attacks	Median filter			Gaussian filter	Average filter			Sharpening	Histogram equalization
		(3×3)	(5×5)	(7×7)	PSNR = 25.34 dB	(3×3)	(5×5)	(6×6)	PSNR = 25.33 dB	PSNR = 21.02 dB
	NC	0.91	0.78	0.62	0.98	0.96	0.74	0.60	0.99	0.86
	Extracted Watermark	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE
(d) Other 12 test images averge	Nongeometric attacks	Median filter			Gaussian filter	Average filter			Sharpening	Histogram equalization
		(3×3)	(5×5)	(7×7)		(3×3)	(5×5)	(6×6)		
	NC	0.85	0.69	0.52	0.95	0.93	0.71	0.54	0.97	0.77

with a mask of size up to 5×5 and the average filtering with a mask of size up to 5×5, where the extracted watermark can still be recognized clearly.

We also use other methods such as rotation, scaling, Gaussian noise and cropping to do geometric attacks. For the rotational attacks, it is first performed by rotating an image at a small angle, scaling the rotated image, and finally cropping the scaled image to the original image size (Petitcolas, 1997). The results are shown in Table 3. For the scaling attack, an image of size 512×512 is first scaled to 256×256 via PhotoImpact 11 software, then the scaled image is opened and resized back to 512×512. The results are shown in Table 3. For the Gaussian noise attack, the noise variance is varied from 1 to 3 with the step size of 1; for the cropping attack, an image of 1/4 size is cropped via PhotoImpact 11 software, the results for both attacks are shown in Table 3.

An intentional attack only in LH subband is shown in Fig. 12. The two largest coefficients are modified for all wavelet trees; both are set to zero in Fig. 12(a), $max_i = sec_i$. is set in Fig. 12(b). Although the watermark cannot be recognized, the watermarked image is seriously distorted.

4.1. Experimental analysis

We compare our method with three recently published papers, namely Lien and Lin (2006), Lin et al. (2009), and Wang and Lin (2004). A pseudo-random sequence that takes value 1 or -1 of size 512 bits is used as watermark and is embedded in Lena image. The results are shown in Table 4. In this case, $\rho = 0.23$ for a false positive probability P_{fp} is 1.03×10^{-7}. From Table 4, the PSNR of the proposed method is better than those of Wang and Lien's methods.

Table 3
Normalized correlation coefficients (NC) after attacks of scaling 256 × 256, cropping 1/4, Gaussian noise added by variations from 1 to 3, and rotation, followed by scaling and cropping to the original size in the 15 test images.

	Geometric attacks	Scaling	Cropping	Gaussian noise Variance			Rotation degree			
		256 × 256	1/4	1	2	3	0.25	0.3	−0.25	−0.3
(a) Lena	NC	0.86	0.68	0.89	0.52	0.39	0.65	0.58	0.70	0.65
	Extracted watermark	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE
(b) Goldhill	NC	0.82	0.68	0.90	0.55	0.42	0.55	0.47	0.60	0.54
	Extracted watermark	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE
(c) Peppers	NC	0.90	0.67	0.89	0.55	0.35	0.70	0.61	0.61	0.54
	Extracted watermark	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE	CSIE
(d) Other 12 test images average	NC	0.82	0.68	0.89	0.57	0.38	0.56	0.50	0.55	0.48

Fig. 12. An intentional attack for all wavelet trees on LH subband. (a) Both two largest coefficients are set to zero, where PSNR = 29.20 dB. (b) Both two largest coefficients are set to equal by $max_i = sec_i$, where PSNR = 33.93 dB.

Table 4
Comparison of the proposed method and the methods in Lin et al. (2009), Lien and Lin (2006), and Wang and Lin (2004).

Attacks/NC	Wang (PSNR = 38.2 dB)	Lien (PSNR = 41.54 dB)	Lin (PSNR = 45.59 dB)	Proposed method (PSNR = 42.98 dB)
Median filter (3 × 3)	0.51	0.79	0.92	0.93
Median filter (4 × 4)	0.23	0.51	0.75	0.80
JPEG (QF = 10)	NA	0.17	0.34	0.36
JPEG (QF = 20)	NA	0.61	0.59	0.68
JPEG (QF = 3Q)	0.15	0.79	0.81	0.91
JPEG (QF = 5G)	0.28	*0.39*	0.95	0.93
JPEG (QF = 7G)	0.57	0.97	0.99	0.93
JPEG (QF = 90)	1	1	1	1
Sharpening	0.46	0.88	0.99	0.93
Gaussian filter	0.64	0.84	0.96	0.96
Rotation (degree: 0.25°)	0.37	0.53	0.61	0.65
Rotation (degree: 0.75°)	0.26	0.16	0.33	0.32
Rotation (degree: 1°)	0.24	0.07	0.27	0.28
Rotation (degree: −0.25°)	0.32	0.47	0.65	0.72
Rotation (degree: −0.75°)	0.24	0.10	0.35	0.24
Rotation (degree: −l°)	0.16	0.16	0.28	0.26
Cropping 1/4	NA	0.92	0.60	0.67
Scaling 256 * 256	NA	0.79	0.86	0.85

Unit of rotation: degree (+: clockwise; −: counterclockwise).

In Lin's method, an insignificant coefficient in a wavelet tree is used, hence the PSNR is the highest in Table 4. In our method, it is not so good for the rotation attacks with degree greater than ±0.75; but it is far better than the listed methods; especially for low pass filtering attacks such as the median filtering, Gaussian filtering, sharpening, and JPEG compression.

5. Conclusion

In this paper, we proposed a wavelet-tree-based blind watermarking method by quantizing the maximum wavelet coefficient in a wavelet tree. The trees are so quantized that they exhibit a large enough energy difference between a watermark bit 0 and a

watermark bit 1; the energy difference is then useful for later watermark extraction. During extraction, an adaptive threshold value is designed. The magnitude of the significant difference in a wavelet tree is compared to the adaptive threshold value. Furthermore, regarding each wavelet tree embedded with a watermark bit, we not only can embed more bits in an image but can extract the watermark without any need of the original image and watermark. As a result, the watermarked images look lossless in comparison with the original ones, and the proposed method can effectively resist common image processing attacks; especially for JPEG compression and low-pass filtering. Moreover, by designing an adaptive threshold value in the extraction process, our method is more robust for resisting common attacks such as median filtering 5×5, average filtering 5×5, and Gaussian noise with a variance of less than 2. In addition to the copyright protection, the proposed method can also be applied to data hiding and image authentication.

References

Aboofazeli, M., Thomas, G., & Moussavi, Z. (2004). A wavelet transform based digital image watermarking scheme. In *Presented at the conference on electrical and computer engineering, Canadian*.

Chen, T. H., Horng, G., & Lee, W. B. (2005). A publicly verifiable copyright-proving scheme resistant to malicious attacks. *IEEE Transactions on Industrial Electronics, 52*, 327–334.

Cox, I. J., Kilian, J., Leighton, F. T., & Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing, 6*, 1673–1687.

Cox, I. J., Kilian, J., Leighton, T., & Shamoon, T. (1996). Secure spread spectrum watermarking for images, audio and video. In *Presented at the proceedings of the IEEE ICIP, Lausanne*.

Dugad, R., Ratakonda, K., & Ahuja, N. (1997). A new wavelet based scheme for watermarking images. In *IEEE ICIP* (pp. 4–7).

Ganic, E., & Eskicioglu, A. M. (2004). Robust DWT-SVD domain image watermarking: embedding data in all frequencies. In *Presented at the international multimedia conference on multimedia and security*.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing* (2nd ed.). Prentice Hall.

Hernandez, J. R., Amado, M., & Perez-Gonzalez, F. (2000). DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure. *IEEE Transactions on Image Processing, 9*, 55–68.

Hsieh, M. S., Tseng, D. C., & Huang, Y. H. (2001). Hiding digital watermarks using multiresolution wavelet transform. *IEEE Transactions on Industrial Electronics, 48*, 875–882.

Inoue, H., Miyazaki, A., Yamamoto, A., & Katsura, T. (1998). A digital watermark based on the wavelet transform and its robustness on image compression. In *Presented at the proceedings of the IEEE ICIP, Chicago*.

Kundur, D. & Hatzinakos, D. (1998). Digital watermarking using multiresolution wavelet decomposition. In *Presented at the proceedings of the IEEE ICASSP, Seattle*.

Lewis, A. S., & Knowles, G. (1992). Image compression using the 2-D wavelet transform. *IEEE Transactions on Image Processing, 1*, 244–250.

Lien, B. K. & Lin, W. H. (2006). A watermarking method based on maximum distance wavelet tree quantization. In *Presented at the proceedings of the 19th conference computer vision, graphics and image processing*.

Lin, W. H., Horng, S. J., Kao, T. W., Fan, P. Z., Lee, C. L., & Pan, Y. (2008). An efficient watermarking method based on significant difference of wavelet coefficient quantization. *IEEE Transactions on Multimedia, 10*, 746–757.

Lin, W. H., Wang, Y. R., & Horng, S. J. (2009). A wavelet-tree-based watermarking method using distance vector of binary cluster. *Expert Systems with Applications, 36*, 9869–9878.

Mahmood, A. K., & Selin, A. (2006). Spatially adaptive wavelet thresholding for image watermarking. In *Presented at the proceedings of the IEEE ICME, Toronto*.

Mukherjee, D. P., Maitra, S., & Acton, S. T. (2004). Spatial domain digital watermarking of multimedia objects for buyer authentication. *IEEE Transactions on Multimedia, 6*, 1–15.

Nikolaidis, N., & Pitas, I. (1998). Robust image watermarking in the spatial domain. *Signal Processing, 66*, 385–403.

Petitcolas, F. A. P. (1997). *Weakness of existing watermark scheme*. [online] Available from: <http://www.petitcolas.net/fabien/watermarking/stirmark/index.html>.

Pickholtz, R., Schilling, D., & Milstein, L. (1982). Theory of spread-spectrum communications – A tutorial. *IEEE Transactions on Communications, 30*, 855–884.

Podilchuk, C. I., & Zeng, W. (1998). Image-adaptive watermarking using visual models. *IEEE Journal of Selected Areas and Communications, 16*, 525–539.

Potdar, V. M., Han, S., & Chang, E. (2005). A survey of digital image watermarking techniques. In *Presented at the proceedings of the IEEE INDIN 2005*.

Shapiro, J. M. (1993). Embedded image coding using zerotrees of wavelet coefficients [see also IEEE Transactions on Acoustical, Speech, and Signal Processing]. *IEEE Transactions on Signal Processing, 41*, 3445–3462.

Sin-Joo, L., & Sung-Hwan, J. (2001). A survey of watermarking techniques applied to multimedia. In *IEEE ISIE* (pp. 272–277).

Temi, C., Choomchuay, S., & Lasakul, A. (2005). A robust image watermarking using multiresolution analysis of wavelet. In *Presented at the proceedings of the IEEE ISCIT*.

Tsai, M. J., Yu, K. Y., & Chen, Y. Z. (2000). Wavelet packet and adaptive spatial transformation of watermark for digital images authentication. In *Presented at the proceedings of the IEEE ICIP, Vancouver, BC*.

ed: Ulead system, Inc., p. PhotoImpact 11 software.

USC SIPI–The USC-SIPI Image Database. [Online]. Available: <http://sipi.usc.edu/services/database/Database.html>.

Wang, H. J. M., Su, P. C., & Kuo, C. C. J. (1998). Wavelet-based digital image watermarking. *Optics Express, 3*, 491–496.

Wang, H. J., & Kuo, C. C. J. (1997). High fidelity image compression with multithreshold wavelet coding (MTWC). In *Presented at the SPIE's annual meeting-application of digital image processing XX, San Diego*.

Wang, H. J., & Huo, C. C. J. (1997). A multi-threshold wavelet coder (MTWC) for high fidelity image compression. In *Presented at the Proceedings of the IEEE ICIP, Santa Barbara*.

Wang, S. H., & Lin, Y. P. (2004). Wavelet tree quantization for copyright protection watermarking. *IEEE Transactions on Image Processing, 13*, 154–165.