A wavelet-tree-based watermarking method using distance vector of binary cluster

Wei-Hung Lin\textsuperscript{a}, Yuh-Rau Wang\textsuperscript{b, *}, Shi-Jinn Horng\textsuperscript{a,c}

\textsuperscript{a} Dept. of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan
\textsuperscript{b} Dept. of Computer Science and Information Engineering, St. John’s University, 499, Section 4, Tam-King Road, Taipei, Taiwan
\textsuperscript{c} Dept. of Electronic Engineering, National United University, Miao-Li, Taiwan

\begin{abstract}
This paper proposes a wavelet-tree-based watermarking method using distance vector of binary cluster for copyright protection. In the proposed method, wavelet trees are classified into two clusters using the distance vector to denote binary watermark bits. The two smallest wavelet coefficients in a wavelet tree are used to reduce distortion of a watermarked image. The distance vector, which is obtained from the two smallest coefficients of a wavelet tree, is quantized to decrease image distortion. The trees are classified into two clusters so that they exhibit a sufficiently large statistical difference based on the distance vector, which difference is then used for subsequent watermark embedding. We compare the statistical difference and the distance vector of a wavelet tree to decide which watermark bit is embedded in the embedding process. The experimental results show that the watermarked image looks visually identical to the original and the watermark can be effectively extracted upon image processing attacks.
\end{abstract}

\section{Introduction}

In today's age of the Internet, the digital contents of images or texts, both audio and video, can easily be copied and circulated illegally. So the authors of digital multimedia may feel economically deprived or robbed of their reputations by fake copies. Watermarking is a process in which identifying information is embedded into a file, such as images, texts, audio, or video, enabling a file's author(s) to control the distribution of, and verify the ownership of, their digital contents. The purpose of watermarking is to protect the property that belongs to the said author(s). It is used widely in copyright protection and as proof of ownership. The watermarking method modifies the original data in a perceptually invisible manner to embed a watermark. By detecting the modifications to verify the existence of the watermark in order to prove an author’s ownership of some original data, or trace the illegal use of the data, the author’s identity and ownership rights are protected. Since a watermark is used to prove ownership, there are several important issues in the watermarking system including transparency, robustness, and blindness. In this paper, a watermark is applied to a digital image for copyright protection, hence robustness is essential. The watermark must be able to effectively resist common image attacks, such as low pass filtering or JPEG compression. Transparency is also required so that a watermark may remain invisible in order to maintain its secrecy. Transparency is also needed to prevent the visual distortion of an original image to maintain the commercial value of the image. A watermarking technique is referred to as blind if the original image and watermark are not needed during extraction (Chen, Horng, & Lee, 2005; Dugad, Ratakon, & Ahuja, 1998; Ganic et al., 2004; Inoue, Miyazaki, & Yamamoto, 1998; Tsai, Yu, & Chen, 2000). The blindness is necessary if it is difficult for us to obtain the original image and watermark.

The spatial and spectral domains are widely used in watermarking. Watermarking in the spatial domain embeds a watermark in selected areas on the texture of the host image (Mukherjee, Maitra, & Acton, 2004; Nikolaidis & Pitas, 1998). The advantage is that the algorithm is simple. The disadvantage is that the watermark cannot effectively resist image processing attacks. In recent years, the spectral domain has been widely used for in watermarking, since the spread spectrum communication is robust against many types of interference and jamming (Pickholz, Schilling, & Milstein, 1982). The host image is transformed to the frequency domain using methods such as the Discrete Cosine Transform (DCT) or the Discrete Wavelet Transform (DWT); next the watermark is embedded in the mid-frequency to ensure the simultaneous transparency and robustness of the watermarked image (Hernandez, Amado, & Perez-Gonzalez, 2000; Mahmood et al., 2006).

Cox, Kilian, and Leighton (1996, 1997) suggested inserting the watermark into the perceptually significant portion of the whole DCT-transformed image, wherein 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), Podilchuk and Zeng proposed an image
adaptive watermarking scheme to improve Cox’s method and added the visual model of the just noticeable difference (JND) to select the maximum length and maximum power watermark sequence. All the above methods do not fall into blind watermarking schemes, since they require the original image for watermark retrieval.

Wang, Su, and Kuo (1998) proposed a watermarking method, in accordance with 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.

Several watermarking methods (Lengelaar & Lagendijk, 2001; Lien & Lin, 2006; Li, Liang, & Niu, 2006; Wang & Lin, 2004) use two sets of coefficients, one to represent the watermark bit 0, and the other to represent the watermark bit 1. Each time according to the embedded watermark bit, only a set of coefficients is quantized. Lengelaar and Lagendijk (2001) proposed a blind watermarking approach called differential energy watermarking. A set of several 8 x 8 DCT blocks are composed and divided into two parts to embed a watermark bit. The high-frequency DCT coefficients in the JPEG/MPEG stream are selectively discarded to produce energy differences in the two parts of the same set. Wang and Lin (2004) proposed a wavelet based blind watermarking scheme. The wavelet coefficients of the host image are grouped into wavelet trees, and each watermark bit is embedded using two 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. Hu, Liu, and Deng (2004) incorporated Human Visual System (HVS), Cyclic Redundancy Check (CRC), and Error Correction Coding (ECC) into Wang and Lin’s (2004) method to reduce error of extraction.

This paper is organized as follows: In Section 2, the wavelet trees and the proposed watermarking quantization method are introduced. The decoder and the extraction algorithms are given in Section 3. In Section 4, the performance is analyzed by applying various attacks to the watermarked images, including non-geometric and geometric attacks. Finally, 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 L-level discrete wavelet transform (DWT). With an L-level decomposition, we have L x 3 + 1 frequency bands. As shown in Fig. 1, when L = 4, the lowest frequency subband is located in the top left (i.e., the LL4 subband); the highest frequency subband is at the bottom right (i.e., the HH1 subband). The relationship between these frequency bands from the blocks of variable size can be seen as a parent–child relationship (Shapiro, 1993). With the exception of the lowest frequency subband LL4, the parent–child relationship can be connected between these sub-

![Wavelet Coefficient and “Parent-Child” Relationship at 4-Level DWT](image-url)
nodes to form a wavelet tree. If the root consists of more than one node, then an image will have many wavelet trees after the DWT. For example, we use 3 subbands LH4, HL4, and HH4 as roots, each subband has $n/2^4 \times n/2^4$ nodes, and 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. When $n = 512$, every wavelet tree (such as from a node in LH4 to LH1 following the parent–child relationship) has 85 coefficients.

A higher level subband (e.g., the HL4 subband) is more significant than a lower level subband (e.g., the HL2 subband) (Shapiro, 1993). 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. The coefficients in high-frequency bands are not used since they often contain little energy and can easily be eliminated without causing visible image distortion, for example, by a lossy compression. Hence, in total there are $S = 2 \times (n/2^4 \times n/2^4)$ wavelet trees used where the roots are in the LH4 and HL4 subbands. The largest number of watermark bits which can be embedded is $S$. In the proposed method, we only use the smallest two coefficients; a wavelet tree is consisted of five wavelet coefficients which are selected from one coefficient of LH4 (HL4) and four coefficients in LH3 (HL3).

### 2.2. The preprocess

To reduce the probability of the embedded wavelet tree to be detected, we shuffle the embedded 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 two random numbers using the same seed and using modulo $S + 1$. Note that we use two generated random numbers to denote the indices of two wavelet trees; respectively, and exchange the two trees with the corresponding indices. For example, suppose that at the 300th time the two generated random numbers are 120 and 240, respectively, then the 120th wavelet tree and the 240th wavelet tree are exchanged. Assume that a binary watermark image $W$ composed of $N_w$ ($\leq S$) bits is embedded. We represent each watermark bit as either 1 or 0, and use a pseudorandom function with another seed to shuffle these $N_w$ bits. We select $N_w$ wavelet trees from $S$ wavelet trees sequentially and compute the global average distance vector of the total number of the $N_w$ wavelet trees using Eq. (1):

$$
\varepsilon = \frac{1}{N_w} \sum_{i=1}^{N_w} d_i,
$$

(1)

$$
d_i = sec_i - min_i,
$$

(2)

where $\varepsilon$ is the global average distance vector in all $N_w$ wavelet trees; $\lfloor \cdot \rfloor$ is the floor function; $min_i$ and $sec_i$ are the smallest and second minimum coefficients in $i$th wavelet tree, respectively. $d_i$ is the distance vector between $min_i$ and $sec_i$, $1 \leq i \leq N_w$. Note that we do not need to seek the maximum distance vector between two coefficients from a wavelet tree, such as maximum and minimum coefficients. When two coefficients are quantized by seeking maximum distance vector, it may influence original rank of these five coefficients. In the proposed method, the difference between the two smallest coefficients is positive. Quantization does not influence original rank of these five coefficients.

### 2.3. Watermark embedding

In order to blindly extract the watermark, we need to distinguish the watermark bit 0 and the watermark bit 1 among the extracted watermark bits. The original data is shown in Fig. 2a, with no relationship between them. According to the embedded watermark bit, the wavelet trees are classified into two clusters (shown in Fig. 2b). Assume that the two clusters are $o_{11}$ and $o_{12}$, where $o_{11}$ denotes the cluster of watermark bit 0, and $o_{12}$ denotes the cluster of watermark bit 1. The distance function is used in this clustering. For the sake of reducing distortion of watermarked image, we select the two smallest numbers from a wavelet tree, namely $min_i$ and $sec_i$. The $d_i$ in cluster $o_{11}$ contains no significant difference and in cluster $o_{12}$ contains a significant difference. Any $d_i$ in $o_{11}$ is equal to zero, and any $d_i$ in $o_{12}$ is greater than zero. Hence when embedded a watermark bit 0, we quantize the two smallest coefficients by

$$
Q_{min} = \frac{(sec_i + min_i)}{2},
$$

(3)

$$
Q_{sec} = Q_{min},
$$

(4)

where $Q_{min}$ and $Q_{sec}$ are the new smallest and second minimum coefficients in $i$th wavelet tree. In order to reduce distortion of a watermarked image, we set $Q_{min}$ and $Q_{sec}$ to be the mean value of $sec_i$ and $min_i$. Otherwise, when we embed a watermark bit 1, the $min_i$ is quantized by

$$
Q_{min} = \begin{cases} 
\min_i, & \text{if } d_i > \max(2 \times \alpha, \varepsilon), \\
\min_i - J_i, & \text{otherwise}
\end{cases}
$$

(5)

$$
J_i = \left\lfloor x + \frac{\Xi}{d_i} \right\rfloor, \quad \Xi = \varepsilon / \max(1, d_i)
$$

(6)

(7)

where $J_i$ is a threshold value, $\left\lfloor \cdot \right\rfloor$ is a ceiling function. $\alpha$ is a scale parameter. The $\alpha$ provides a tradeoff between the strength of the watermark and quality of the watermarked image. The larger $\alpha$ is, the more heavily quantized are the wavelet trees; using a larger $\alpha$

---

**Fig. 2.** (a) Before clustering the wavelet tree and (b) the wavelet trees are classified into two clusters.
trades signal-to-noise ratio (SNR) quality of the image for higher robustness of the watermark (Wang & Lin, 2004). In some images, if  is smaller, the value of  is given. The larger the  is, the larger the  will be, and the more wavelet trees are quantized. For example, let  = 12 and  = 13. Suppose  is set to be less than , such as  = 6,  will not be quantized as  = 12. On the other hand, if  is set to be larger than , such as  = 7,  will be quantized since  < 14. Smaller  = 1 will result in larger  and  based on Eqs. (6) and (7), respectively, and hence the difference between  and  will be more significant. In Eq. (7), the minimum value of denominator is set to one when  is less than one to avoid  to be too large.

When embedding a watermark bit 1, if  is not significant, the minimum coefficient is quantized; otherwise, it is kept the same as before, since we do not want to increase the distortion of the image. On the contrary, when embedding a watermark bit 0, the value of  and  are quantized, and the new  will be equal to 0. Based on this strategy, there exists a large energy difference between embedding watermark bit 1 and watermark bit 0.

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, the wavelet trees are classified into two clusters. The distance vectors in  do not have significant difference, since the difference between the two smallest coefficients will be close to zero. However, the distance vectors in  exist significant difference. The smallest wavelet coefficient  will be smaller than the original  and hence the distance value  will be larger than . Hence, we need to find a decision line to separate one cluster from the other. When we randomly select a wavelet tree, we just analyze whether its distance vector is greater than the decision value or not to decide which watermark bit has been embedded. So there are two hypotheses:

\[ H_0 : d_i \text{ does not have significant difference}, \quad H_1 : d_i \text{ has significant difference}, \]

\[ d_i = \text{sec}_i - \text{min}_i', \quad (8) \]

where  and  are the smallest coefficient and the second minimum coefficient of a wavelet tree in a watermarked image, respectively.  is a new distance vector in the extraction process.

Since there are only two clusters, the two hypotheses are equally likely. The decoder computes the  from  wavelet trees.

To find a decision value of a decision line, we sort all the  wavelet trees by

\[ \varphi(\phi) = \{\phi_1, \phi_2, \ldots, \phi_{Nw}\}. \quad (9) \]

where  is the set  is a sort function. The  correspond to wavelet tree  respectively.

If the numbers of watermark bits 0 and 1 are uniform distribution, the probability of  embedding a watermark bit 1 is higher than that of  embedding. The value of the decision line can be found in two wavelet trees where  and  have significant difference.

To find the decision value, recall in embedding process, in order to embed a watermark bit 1,  is increased, when  is smaller than the average distance vector  or 2; on the other hand, the  is not changed. But, after quantization,  which have been embedded a watermark bit 1, are not all greater than . Hence, an ideal decision value is greater than zero and smaller than . For example, in the embedding process, assume that  = 12 and  = 3. If  = 2, then based on Eq. (5), the new  will be 11. In addition, if  = 5, then based on Eq. (5), the new  will be 10. From Figs. 3–10 we use Lena image of size 512 × 512 (8 bits/pixel) and 512 watermark bits (the ratio of watermark bit 0 to watermark bit 1 is 1:1) to analyze the distribution of distance vector of wavelet trees and frequency count in two clusters. Fig. 3a shows that there exists a clearly discriminative between the two clusters and hence we can distinguish the two kinds of watermark bits easily. In Fig. 3b, we compute the frequency count of  distance vectors. Most of distance vectors of watermark bit 0 are close to zero and the distance vectors of watermark bit 1 are greater than seven. Note that the bin  denotes the distance vectors are within the range  (j = 1, j).

Note that after watermarked image is attacked, the difference between the two clusters decreases (see Figs. 4–10). If a watermarked image is attacked seriously, for example by median filtering 3 × 3 to 7 × 7, JPEG with QF = 70 to QF = 50, and Gaussian noise, it is difficult to distinguish the two clusters completely. From Figs. 3–10, we find two characteristics as follows. One is a standard deviation of these distance vectors will down, when two clusters are more and more approach. The other is that the motions (from a larger distance value to a smaller distance value) of those larger distance vectors (for example, distance vector greater than 30) are not significant, when two clusters are also more and more approach. Hence, in Cox et al. (1997), Cox et al. pointed out why the watermark embed in sig-

![Fig. 3.](image-url)
significant coefficient can obtain high robustness. If we use $N_w$ distance vectors $d_0^i$ to calculate the average distance vector, some smaller $d_0^i$ in $\omega_2$ will be misclassified. So, the adaptive decision line is necessary for the watermarked image under different attacks. The performance of extraction can be improved by using an adaptive decision line.

Hence, we compute the average distance vector $\bar{e}$ and standard deviation $\sigma$ by

$$
\bar{e} = \frac{1}{N_w} \sum_{i=1}^{N_w} d_0^i,
$$

$$
\sigma = \sqrt{\frac{1}{N_w-1} \sum_{i=1}^{N_w} (d_0^i - \bar{e})^2}.
$$

The performance of extraction can be improved by using an adaptive decision line.

Fig. 4. (a) The distribution of distance vector of wavelet trees with $\bar{e} = 8.78, \sigma = 11.49$ in two clusters upon attack of Gaussian filtering. (b) The frequency counts of $N_w$ distance vectors.

Fig. 5. (a) The distribution of distance vector of wavelet trees with $\bar{e} = 9.36, \sigma = 12.43$ in two clusters upon attack of JPEG compression within QF = 70. (b) The frequency counts of $N_w$ distance vectors.

Fig. 6. (a) The distribution of distance vector of wavelet trees with $\bar{e} = 9.68, \sigma = 12.33$ in two clusters upon attack of JPEG compression within QF = 50. (b) The frequency counts of $N_w$ distance vectors.
Then, we normalize the value of $d_0^i$ by

$$
\Delta_i = \left[ \frac{d_0^i - \mu'}{\sigma} \right].
$$

(12)

If $d_0^i$ is within one standard deviation, $\Delta_i$ is equal to zero. In order to let Eq. (12) be positive, we modify it as follows:

$$
\epsilon_0 = \frac{1}{N_w} \sum_{i=1}^{N_w} d_0^i.
$$

(10)

$$
\sigma = \sqrt{\frac{\sum_{i=1}^{N_w} (d_0^i - \epsilon')^2}{N_w}}.
$$

(11)
\[
\Delta_i = \left[ \frac{(d_i - e_i)^2}{\sigma} \right].
\]

(13)

Subsequently, we sum all \( \Delta_i \) by

\[
I = \sum_{i=1}^{N_w} \Delta_i.
\]

(14)

Finally, we calculate the decision value by

\[
T = \frac{1}{T} \sum_{i=1}^{N_w} \phi_i.
\]

(15)

When \( I \) is larger, \( T \) is closer to \( e \). If \( T \leq 1 \), \( T \) will be set to 1, the reason is like Eq. (7), \( T \in [1,e] \).

Note that in Eq. (13), we use floor function \( \lfloor \cdot \rfloor \) to discard the slight difference. If we do not use \( \lfloor \cdot \rfloor \), \( I' \) will be very close to \( N_w \).

Since \( d_i \) in the \( o_1 \) is close to zero, the hypothesis \( H_0 \) is more likely, if \( (d_i/T) < 1 \). Otherwise \( d_i \) in the \( o_1 \) is greater than \( T \), \( H_1 \) is more likely, if \( (d_i/T) \geq 1 \).

There are two special cases that we can not use Eq. (15) to compute the decision value. One is that all watermark bits are 1. Since the value of \( T \) is between the minimum \( d_i \) and maximum \( d_i \), the extraction errors will happen in some wavelet trees. The other case is that all watermark bits are 0. Since the value of \( T \) is equal to zero, the decision errors will happen when a few \( d_i \) are not equal to zero. Moreover, the extraction errors will be more serious under the condition that watermarked image is attacked.

3.2. Watermark extraction

We invoke Eqs. (16) to extract the watermark. If the significant difference (i.e. \( d_i \)) is greater than or equal to \( T \), the embedded watermark bit is 1; otherwise, the embedded watermark bit is 0.

\[
\text{watermark bit} = \begin{cases} 
1 & \text{if } (d_i/T) \geq 1, \\
0 & \text{otherwise.}
\end{cases}
\]

(16)

4. Experimental 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 stated as follows:

\[
\text{PSNR} = 10 \times \log_{10} \left( \frac{255 \times 255}{\frac{1}{W \times W} \sum_{x=0}^{W-1} \sum_{y=0}^{W-1} [f(x,y) - g(x,y)]^2} \right) \text{dB}.
\]

(17)

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 coordinates \( (x,y) \) 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:

\[
\text{NC} = \frac{1}{w_h \times w_w} \sum_{i=0}^{w_h-1} \sum_{j=0}^{w_w-1} w(i,j) \times w(i,j),
\]

(18)

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 coordinates \( (i,j) \) 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 0. \( w(i,j) \) is set in the same way. So the value of \( w(i,j) \times w(i,j) \) is either 1 or 0.

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

\[
P_{FP} = \sum_{A=(-\gamma)/2}^{(\gamma)/2} \left( \frac{N_w}{A} \right) p_{T\gamma}^{-A} (1 - p_{T\gamma})^A,
\]

(19)

where \( P_{T\gamma} \) is the probability when \( w(i,j) \neq w(i,j) \); it is reasonable to assume \( P_{T\gamma} = 0.5 \). We can choose an appropriate \( \rho \) to meet the requirement. For example, if \( \rho = 0.23 \), \( N_w = 1024 \), and \( P_{T\gamma} = 0.5 \), then the false positive error \( P_{FP} = 8.15 \times 10^{-14} \).

We use sixteen images, namely Elaine, Airplane, F16, and the other 13 images (512 × 512 pixels, 8 bits/pixel) obtained from USC SIPI Image Database (2008) for our experiments. For attacking, we use the Stirmark benchmark (Petitcolas, 1997) and Photomimic 11 software (Ulead Systems, 2005) tools to simulate common image attacks.

While there is no attack, for the sake of brevity only the three images are shown in Fig. 11. The binary watermark is shown in Fig. 12a. Fig. 12b shows the extracted result and Fig. 13 shows the watermarked image. We pre-determine the scale parameter \( \gamma = 3 \). The PSNRs of the sixteen images are shown in Fig. 14, and the watermark can be extracted from these sixteen images with NC = 1. In Fig. 14, all PSNRs of the watermarked image are greater than 40.

In the following, we consider both geometric and non-geometric attacks. Non-geometric attacks include JPEG compression, low pass filtering, histogram equalization, and sharpening.
JPEG is one of the most frequently used formats in the Internet and digital cameras. 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 smaller the quality factor is, the more unclear the extracted watermark will be.

For other non-geometric attacks (Petitcolas, 1997), for example median filtering, 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 by median filtering with a mask of size up to 5 x 5 and the average filtering with a mask of size up to 5 x 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. Rotational attacks are performed by first 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 x 512 is first scaled to 256 x 256 via PhotoImpact 11 software, then the scaled image is opened and resized back to 512 x 512. The results are shown in Table 3. For the Gaussian noise attack, the noise variance is varied from 1 to 3 with step size 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. Note that in the proposed method, it is possible that attackers modify the distance value of the two smallest coefficients to zero for all wavelet trees. It will result in the extracted watermark unrecognizable. In the meanwhile, the watermarked image will seriously be distorted as well and hence lose its commercial value. In our experiment, the PSNRs go down to less than 30.

### Fig. 11
(a) Elaine, (b) Airplane, and (c) F16, each of size 512 x 512.

### Fig. 12
(a) The original binary watermark of size 32 x 16, (b) All the extracted watermark with NC = 1.

### Fig. 13
(a) Watermarked Elaine with PSNR = 44.47 dB, (b) Watermarked Airplane with PSNR = 44.77 dB, (c) Watermarked F16 with PSNR = 41.89 dB.

### Fig. 14
The PSNRs of sixteen watermarked images.

### Table 1
<table>
<thead>
<tr>
<th>Image</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elaine</td>
<td>44.47</td>
</tr>
<tr>
<td>Airplane</td>
<td>44.77</td>
</tr>
<tr>
<td>F16</td>
<td>41.89</td>
</tr>
</tbody>
</table>
Table 1
Normalized correlation coefficients (NC) after attacked by JPEG compression with the quality factors (QF) 10, 15, 20, 25, 30, 35, 40, 50, 60, 70, 80, 90, 100 in (a) Elaine, (b) F15 and (c) Airplane.

<table>
<thead>
<tr>
<th>Test images</th>
<th>QF</th>
<th>NC</th>
<th>Extracted watermark</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Elaine</td>
<td>10</td>
<td>0.24</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.43</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.59</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.68</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.80</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.84</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.90</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.95</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>0.97</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>0.98</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>1</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>1</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1</td>
<td>CSIE</td>
</tr>
<tr>
<td>(b) F15</td>
<td>10</td>
<td>0.30</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.59</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.70</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.76</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.86</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.91</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.93</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.97</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>1.00</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>0.99</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>1</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>1</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1</td>
<td>CSIE</td>
</tr>
<tr>
<td>(c) Airplane</td>
<td>10</td>
<td>0.13</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.29</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.52</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.67</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.74</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.83</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.96</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.98</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>0.99</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>0</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>1</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>1</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1</td>
<td>CSIE</td>
</tr>
</tbody>
</table>

Table 2
Normalized correlation coefficients (NC) after attacked by median filtering $\left(\frac{3}{3}, \frac{5}{5}, \frac{7}{7}\right)$, Gaussian filtering, average filtering attacks $\left(\frac{3}{3}, \frac{5}{5}, \frac{6}{6}\right)$, sharpening, and histogram equalization in (a) Elaine, (b) F15 and (c) Airplane.

<table>
<thead>
<tr>
<th>Non-geometric attacks</th>
<th>Median filter $(\frac{3}{3})$</th>
<th>Gaussian filter $\frac{25.25,\text{dB}}{(\frac{3}{3})}$</th>
<th>Average filter $(\frac{3}{3}, \frac{5}{5}, \frac{6}{6})$</th>
<th>Sharpening PSNR = 25.15 dB</th>
<th>Histogram equalization PSNR = 18.57 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Elaine</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC</td>
<td>0.88</td>
<td>0.68</td>
<td>0.47</td>
<td>0.95</td>
<td>0.91</td>
</tr>
<tr>
<td>Extracted watermark</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
</tr>
<tr>
<td>(b) F15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC</td>
<td>0.85</td>
<td>0.74</td>
<td>0.66</td>
<td>0.92</td>
<td>0.90</td>
</tr>
<tr>
<td>Extracted watermark</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
</tr>
<tr>
<td>(c) Airplane</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC</td>
<td>0.88</td>
<td>0.63</td>
<td>0.61</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>Extracted watermark</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
</tr>
</tbody>
</table>

Table 3
Normalized correlation coefficients (NC) after attacks of scaling $256 \times 256$, cropping $1/4$, Gaussian noise added by variations from 1 to 3, and rotation, followed by scaling and cropping to the original size in (a) Elaine, (b) F15, and (c) Airplane.

<table>
<thead>
<tr>
<th>Test images</th>
<th>Geometric attacks</th>
<th>Scaling</th>
<th>Cropping</th>
<th>Gaussian noise</th>
<th>Rotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Geometric attacks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) Elaine</td>
<td>NC</td>
<td>0.85</td>
<td>0.63</td>
<td>0.85</td>
<td>0.44</td>
</tr>
<tr>
<td></td>
<td>Extracted watermark</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>(b) F15</td>
<td>0.84</td>
<td>0.63</td>
<td>0.88</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td>Extracted watermark</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
</tr>
<tr>
<td></td>
<td>(c) Airplane</td>
<td>0.92</td>
<td>0.61</td>
<td>0.81</td>
<td>0.31</td>
</tr>
<tr>
<td></td>
<td>Extracted watermark</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
<td>CSIE</td>
</tr>
</tbody>
</table>

Table 4

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Wang (PSNR = 38.2 dB)</th>
<th>Li (PSNR = 40.6 dB)</th>
<th>Lien (PSNR = 41.54 dB)</th>
<th>Proposed method (PSNR = 44.73 dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median filter $(\frac{3}{3})$</td>
<td>0.51</td>
<td>0.35</td>
<td>0.79</td>
<td>0.92</td>
</tr>
<tr>
<td>Median filter $(\frac{4}{4})$</td>
<td>0.23</td>
<td>0.26</td>
<td>0.51</td>
<td>0.75</td>
</tr>
<tr>
<td>JPEG (QF = 10)</td>
<td>NA</td>
<td>0.15</td>
<td>0.17</td>
<td>0.33</td>
</tr>
<tr>
<td>JPEG (QF = 20)</td>
<td>NA</td>
<td>0.34</td>
<td>0.61</td>
<td>0.59</td>
</tr>
<tr>
<td>JPEG (QF = 30)</td>
<td>0.15</td>
<td>0.52</td>
<td>0.79</td>
<td>0.81</td>
</tr>
<tr>
<td>JPEG (QF = 50)</td>
<td>0.28</td>
<td>0.52</td>
<td>0.89</td>
<td>0.95</td>
</tr>
<tr>
<td>JPEG (QF = 70)</td>
<td>0.57</td>
<td>0.63</td>
<td>0.97</td>
<td>1</td>
</tr>
<tr>
<td>JPEG (QF = 90)</td>
<td>1</td>
<td>0.78</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Sharpening</td>
<td>0.46</td>
<td>0.38</td>
<td>0.88</td>
<td>0.99</td>
</tr>
<tr>
<td>Gaussian filter</td>
<td>0.64</td>
<td>0.70</td>
<td>0.84</td>
<td>0.96</td>
</tr>
<tr>
<td>Rotation (degree: 0.25°)</td>
<td>0.37</td>
<td>0.46</td>
<td>0.53</td>
<td>0.61</td>
</tr>
<tr>
<td>Rotation (degree: 0.75°)</td>
<td>0.26</td>
<td>0.36</td>
<td>0.16</td>
<td>0.34</td>
</tr>
<tr>
<td>Rotation (degree: 1°)</td>
<td>0.24</td>
<td>0.33</td>
<td>0.07</td>
<td>0.27</td>
</tr>
<tr>
<td>Rotation (degree: -0.25°)</td>
<td>0.32</td>
<td>0.50</td>
<td>0.47</td>
<td>0.65</td>
</tr>
<tr>
<td>Rotation (degree: -0.75°)</td>
<td>0.24</td>
<td>0.29</td>
<td>0.10</td>
<td>0.22</td>
</tr>
<tr>
<td>Rotation (degree: -1°)</td>
<td>0.16</td>
<td>0.33</td>
<td>0.16</td>
<td>0.28</td>
</tr>
<tr>
<td>Cropping 1/4</td>
<td>0.61</td>
<td>0.92</td>
<td>0.60</td>
<td></td>
</tr>
<tr>
<td>Scaling 256 × 256</td>
<td>NA</td>
<td>0.35</td>
<td>0.79</td>
<td>0.86</td>
</tr>
</tbody>
</table>
4.1. Experiment analysis

We compare the proposed method with (Lien & Lin, 2006; Li et al., 2006; Wang & Lin, 2004) using the Lena image. The results are shown in Table 4. In this case, \( \rho = 0.23 \) for a false positive probability \( P_{fa} = 1.03 \times 10^{-7} \). From Table 4, the PSNR of the proposed method is better than those in Lien and Lin (2006), Li et al. (2006) and Wang and Lin (2004). In our method, it is not so good for the rotation attacks with degree greater than \( \pm 0.75 \); but it is far better than those in 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 use insignificant wavelet coefficients of wavelet trees to embed watermark. The method is different from previous researches which use significant coefficients to embed a watermark. The wavelet trees are classified into two clusters using a distance vector. In such a classification strategy, the wavelet trees exhibit a sufficiently large statistical difference in embedding a watermark bit 1 and a watermark bit 0. Owing to the statistical difference, a watermark can be effectively extracted without requiring any original image or watermark. Since the two smallest wavelet coefficients are used in classifying the above clusters, the watermarked images look lossless. As a result, the proposed method can effectively resist common image processing attacks, especially JPEG compression and low pass filtering. Moreover, owing to the significant statistical difference, our method is more robust in cases of resistance against attacks by Gaussian noise with a variance of less than 2, as well as against filtering attacks which employ larger masks such as median filtering \( 5 \times 5 \) or average filtering \( 5 \times 5 \).

Acknowledgement

This work was partially supported by National Science Council under the Contract Numbers NSC-97-2221-E-129-007 and NSC-97-2221-E-239-022.

References


