Regular Article

Efficient Image Watermarking Using Filtered DWT-Blocks for Quantization of Significant Differences

Thien Huynh-The[1], **Thuong Le-Tien**[2], **Tuan Nguyen-Thanh**[2]

[1] Department of Computer Engineering, Kyung Hee University, Korea
[2] Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology, Vietnam

Correspondence: Thien Huynh-The, thienht@oslab.khu.ac.kr

Manuscript communication: received 8 August 2013, accepted 29 September 2014

Abstract– **In the paper, a robust blind watermarking method is introduced for gray-scale images based on wavelet tree quantization with an adaptive threshold in the extraction. Every block of 2×2 coefficients of High-Low subbands of the Wavelet tranform are grouped in a block through the parent-child relationship of the wavelet tree. Every scrambled binary watermark bit is embedded into each block based on the difference value of two largest coefficients. The watermark is recovered by comparing the difference values in each block to an adaptive threshold. The accuracy of an extracted watermark depends on the threshold which is determined by minimizing the sum of weighted within-class variance. The performance of the proposed watermarking method is represented through experimental results under various types of attack such as, Histogram Equalization, Cropping, Low-pass Filtering, Gaussian noise, Salt & Pepper noise and JPEG compression. In additions, the proposed method is also compared to recent methods in the extraction performance.**

Keywords– **blind watermarking, 3-level DWT, wavelet tree, significant difference, adaptive threshold, within-class variance.**

1 Introduction

Due to the rapid advance of the Internet and the easiness of a digitalization, people can arbitrarily access and distribute any digital products. Hence, the digital watermarking technique has been applied to multimedia products for the copyright protection or the image authentication... This technique embeds an information into digital contents so that the viewer cannot see any information. However, there are many problems in the watermarking system. First, the watermark has to ensure no degrading a quality of the cover image and being perceptually invisible for human eyes. Second, the watermark must be robust to resist different attacks. Finally, the watermarking system is blind watermarking model, that is, the watermark can be recovered without any information of an original image.

The wavelet tree-based watermarking method is proposed based on the Qualified Significant Wavelet Tree (QSWT) [1]. The authors in this paper embedded a watermark into each of two wavelet sub-bands. Lien et al. [2] improved QSWT-based watermarking method [1] by using four wavelet trees to represent two watermark bits for enhancing the visual quality. But this method cannot effectively resist low-pass filter attacks such as median filters or Gaussian filters. The watermarking method based on the significant difference quantization technique was recommended by Lin et al. [3]. As the main stage of process, every seven-coefficients of 3-level Discrete Wavelet Transform (DWT) sub-bands grouped into a block, the watermark bit is embedded into a block by quantizing the difference of two largest coefficients. In the study [4], the embedding process using the insignificant coefficients of wavelet trees to code a information was introduced. Although the method in the article [4] improved the quality of the output image, the degradation of watermark robustness is the shortcoming of this technique. Moreover, Run et al. in the research [5] utilized scaling the magnitude of the significant difference of two largest wavelet coefficients in a wavelet tree for improving the robustness. However, extracted information cannot ensure the quality under JPEG compression. Also with the structure-based quantization, watermark bits in the study [6] were embedded into significant coefficients of super-trees based on an energy distribution for improving the robustness. In order to extract, the hidden bits are recovered by the de-quantized super-tree structures, however, the robustness of this method is not effective in low-frequency filters.

Another approach introduced in [7] used the wavelet packet decomposition for both an original image and a watermark. Due to embedding into low-frequency subbands, it is easy to decide the existence of watermark by comparing the correlation between the recovery and the original watermark. In additions, Fan et al. [8] proposed the method in which the watermark were embedded into the cover image by modifying the center coefficient in each block based on using statistical characteristics of coefficients for two-phase quantization in the watermarking system. The values of coefficients which would be modified mostly depended on the watermark bits and the mean of four cross neighbor

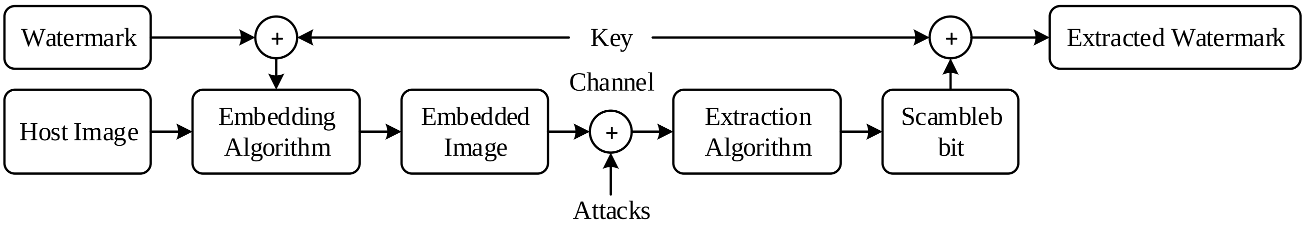


Figure 1. The flowchart of the general model for digital image watermarking.

coefficients. The basic method in [9] embedded each watermark bit by comparing the difference of two largest significant coefficients in each wavelet tree to the average value. The maximum coefficients in blocks are then modified based on the value of watermark bits. Embedding each pixel of a watermark image into DWT sub-band blocks of the cover image was previously proposed by Huang et al. [10]. The limitation of this method is non-blind watermarking system, that is, the original image is fully necessary for the extraction process. Furthermore, Uhl et al. [11] utilized the re-watermarking approach as the quantization-based and robust technique for embedding the multiple fingerprinting. They also investigated two extensions to a wavelet coefficient-tree based embedding technique which turns out to improve the detection performance. In this paper, a blind watermarking method for the gray-scale is developed from the method in [3]. A binary watermark is embedded into high-low (HL) subbands of the host image by comparing the difference of two largest coefficients to the quantization value. The wavelet-tree algorithm is utilized for decomposing the host image. The host image is then recovered by using the Inverse Discrete Wavelet Transform (IDWT). For extracting, the watermark bits are obtained by comparing to the significant differences to the threshold which is determined based on the analysis of the probability distribution function (pdf) of the significant difference of wavelet coefficients. In order to achieve the minimum of error, the adaptive threshold is defined as the value at which the sum of weighted within-class variances is smallest. The proposed method is assessed through two measurements: Peak Signal to Noise (PSNR) of the embedded image and Normalized Correlation (NC) of the extracted watermark. This paper is organized as follow. Section 2, the proposed method is described in detail. The experimental results and conclusion of this work are then represented in Section 3 and 4, respectively.

2 The Proposed Method

2.1 The Watermarking Model and Pre-Process

The Figure 1 represents the digital watermarking model for the gray-scale images. The proposed method is also used for color images by applying it to the luminance channel of the YCbCr color space. The first stage of the model is that the watermark is embedded into the host images with the security key to improve the safety. The embedded images could be then transferred, stored or modified for personal purposes. In this stage, the images could be damaged by attackers in exertion of modifying or even removing the watermark information. Finally, in order to examine the copyright of these images, the hidden information is recovered without the original one.

In this paper, a watermark is embedded into the cover images on the wavelet domain to improve the robustness. Based on the wavelet tree technique, there are 13 frequency sub-bands of three super wavelet trees. The embedding process is implemented on these sub-bands. Using LL4 sub-band as a root component is not suitable for embedding a watermark since it is a low-frequency band corresponding to the important information of an image and easily causes the image distortion. The HH4-HH3-HH2-HH1 super tree is easily eliminated when using the JPEG lossy compression on a watermark image. Only two super wavelet trees HL4-HL3-HL2-HL1 and LH4-LH3-LH2-LH1 corresponding to $2 \times n/2^4 \times n/2^4$ wavelet trees host images $(n \times n)$ could be used for embedding. In this research, the authors used super wavelet tree HL4-HL3-HL2-HL1 to hide watermark. To avoid the complex computation and the effects of various types of attack, such as low pass filter, the proposed method only utilized two largest coefficients selected from each block that includes one coefficient of HL4 sub-band and four coefficients of HL3 sub-band, shown in the Figure 2. Therefore, the maximum number of wavelet trees is $\left(n/2^4\right)^2$ corresponding to the maximum number of watermark bits can be embedded. However, due to the size of host images 512×512 and 16×32 for the watermark image used in this research, there are only 512 blocks or 512 wavelet trees used for embedding and extracting 512 watermark bits. It is noted that these blocks are also scrambled with a security key to improve the security of a watermark. In the main algorithm for embedding, the difference of two largest coefficients in each block is modified based on the watermark bit and the fixed quantity, called the quantization value.

2.2 The Embedding Process

In this paper, the watermark is used for embedding into the host gray-scale image is the 16×32 binary image with 512 bits of 0 or 1 value. The watermark bits are embedded into wavelet-tree blocks which include five coefficients (4 HL3 coefficients and 1 HL4 coefficient). The embedding algorithm is implemented based on the

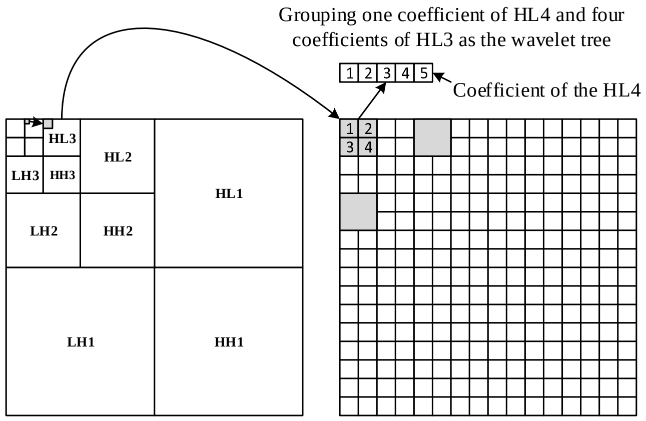

Grouping one coefficient of HL4 and four coefficients of HL3 as the wavelet tree
1 2 3 4 5
Coefficient of the HL4
HL3
HL2
LH3
HH3
HL1
LH2
HH2
LH1
HH1
1 2
3 4

Figure 2. The manner of grouping coefficients as a block based on the wavelet tree technique.

modification of wavelet coefficients corresponding to the watermark bits. This can be represented as follow:

• With watermark bit 1:

$$\max_i = \begin{cases} \max_i + \Delta & \text{if } (d_i \leq T) \\ \max_i & \text{otherwise} \end{cases} \qquad (1)$$

• With watermark bit 0:

$$\max_i = \sec_i \qquad (2)$$

where $\max_i$ and $\sec_i$ are the two largest coefficients in examining block i^{th}. The d_i which is the significant difference of $\max_i$ and $\sec_i$ is calculated as $d_i = \max_i - \sec_i$. In Equation (1), $\Delta = T - d_i$ is the value to ensure that the difference would be larger than the quantization value T. In the quantization technique, the value of T needs to be turn on at first. After applying the quantization algorithm representing by Equations (1) and (2), the different values in embedded blocks have been changed to be zero for the 0-bit watermark and greater than T for 1-bit watermark, respectively. Therefore, based on this specification in the inverse process, the watermark information can be extracted.

As a result, the value of watermark bit is determined based on the difference in embedded blocks by comparing this value to a threshold in the extraction stage. The modification of these coefficients would degrade the image quality. As effort to minimize the effect of modification, the blocks will be arranged based on the differences before embedding. It can be seen that the sum of differences after embedding is expressed in following equation:

$$D = \Delta n_1 + d n_0, \qquad (3)$$

where n_1 and n_0 are the number of 1-bits and 0-bits, respectively. Clearly, this value can be only minimized through choosing compatible block. The wavelet-tree blocks are firstly sorted based on different values in ascend. The blocks with the minimum difference are then utilized to embed watermark 0-bits and the remains of blocks would be used for 1-bits. In order to be clear, this process is shown as in the Figure 3.

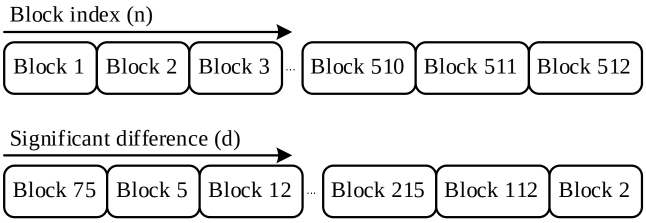


Figure 3. Coefficient-blocks are chosen for embedding based the difference values to minimize the degradation in an output image.

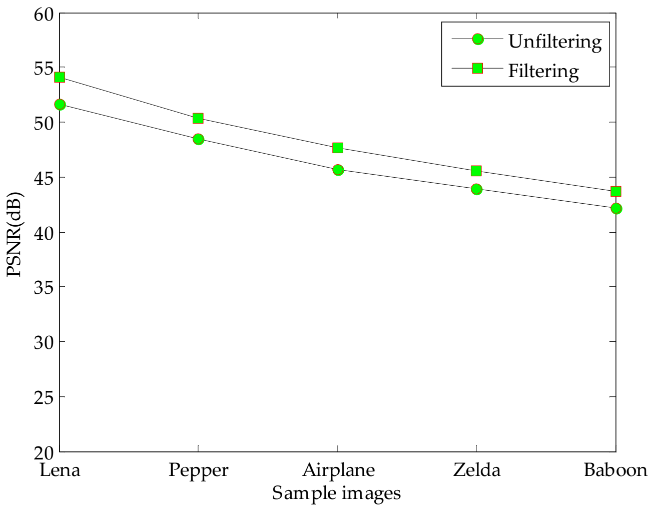


Figure 4. Comparing two cases of manner of choosing blocks for embedding.

Comparing with the case of unfiltering of blocks [12], the output image is better in the quality at the arbitrary value of quantization value. In order to demonstrate this statement, the small estimation for two above cases is performed and displayed as in the Figure 4.

From Equation (1), the quality of an embedded image also depends on the quantization value T, so some experimental results as in Figure 5 from simulation are shown to assess the effect of this parameter on the embedded image quality. The assessment is run for many gray-scale images with various values of T. In practice, the quality of output images were patently decreased when increasing the value of T. However as the tradeoff in the almost watermarking methods, the robustness of a watermark will be raised under various types of attack.

The detail of embedding process can be shown in Figure 6 and listed as follows:

Input: An original image, a watermark image and a security key containing positions of watermark bits.

Output: An embedded image.

• Step 1: The binary watermark is segmented into two partitions (one for 0-bit and another for 1-bit) and save the original position of them as the key for extraction process.

• Step 2: The original image is decomposed at 4-level of DWT.

• Step 3: The wavelet coefficients of HL3 and HL4 are grouped into the blocks and then sorted fro m small to large of significant difference.

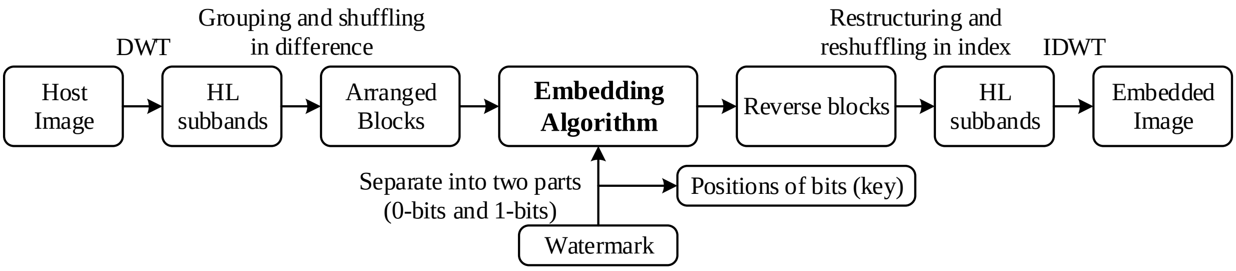


Figure 6. The flowchart of the embedding process for gray-scale image using a binary image as a watermark.

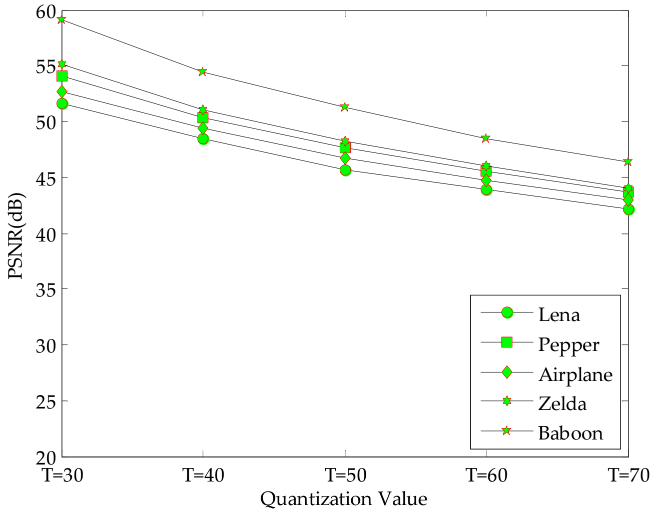


Figure 5. The PSNR of embedded images for various quantization values.

• Step 4: Embedding the watermark into the host image is performed by using the equation (1) for 1-bit and (2) for 0-bit. It is important to note that these watermark bits are embedded into compatible blocks which were determined before as in the Figure 3.

• Step 5: All blocks are reshuffled as the origin.

• Step 6: In order to echieve the embedded image, the modified wavelet coefficients is transformed based on the IDWT technique.

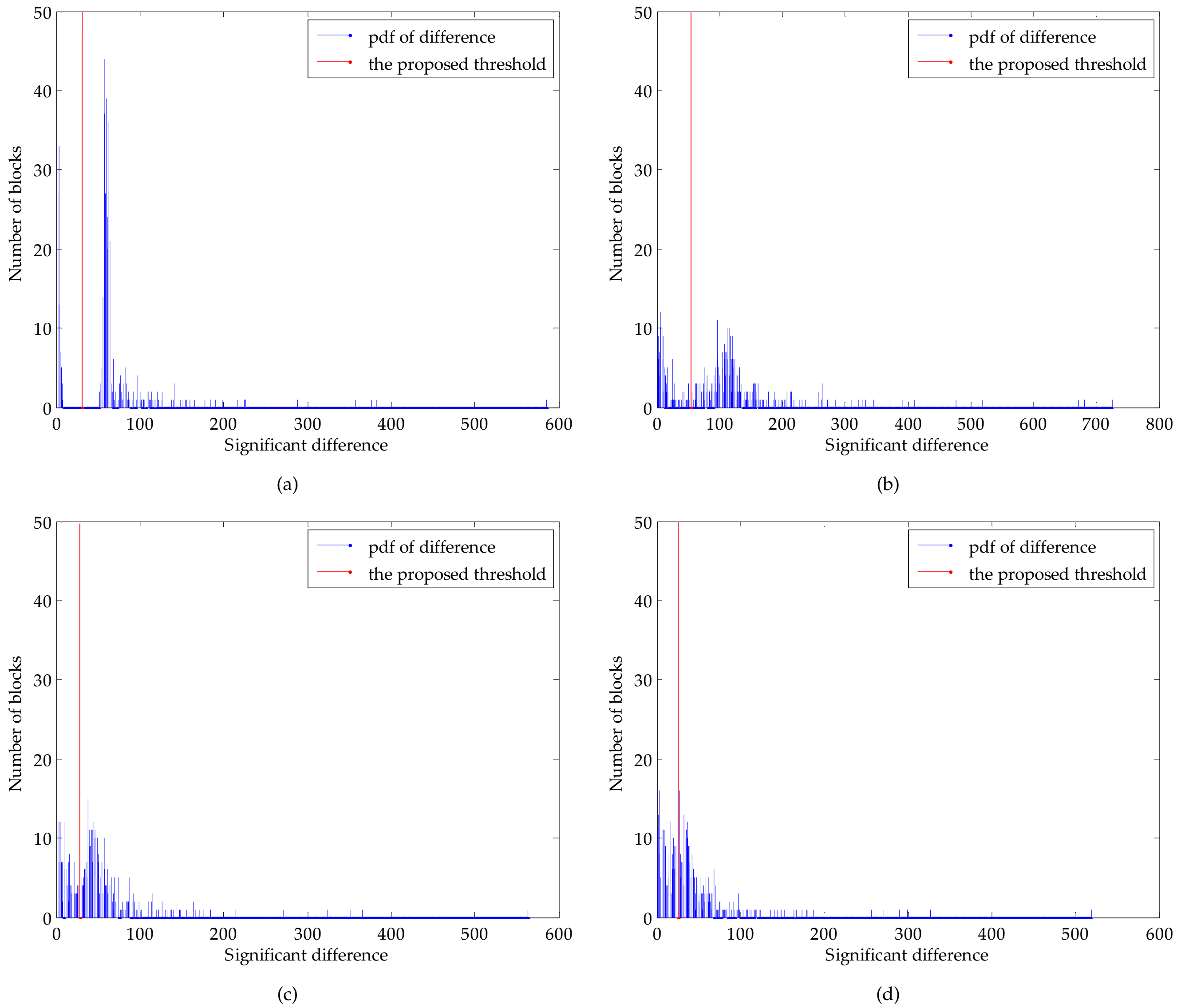
2.3 The Extraction Process Using Weighted-Variance Threshold

In the extraction, watermark bits embedded into each block can be recovered based on comparing the significant difference to the threshold - denoted y. It can be seen that the difference in each block of a host image after embedding will be either 0 or larger than T as following the Equations (1) and (2). Therefore, the watermark can be easily extracted through the below equation as:

$$\text{bit}_i = \begin{cases} 1 & \text{if } (d_i \geq y) \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where d_i is significant difference of i^{th} block and y is the threshold to extraction. It is noted that the watermark can be only extracted exactly if the threshold in the equation (4) is $(1 \leq y \leq T)$ for the non-attack case. However, there are many errors in extracting under various attack types, that is, an extracted watermark bit can be 1-bit instead of 0-bit as the correct result or vice versa. An inaccuracy in extraction is explained that significant differences of embedded blocks have been modified unexpectedly by attacks, therefore, the determined value of the threshold y is important task because it affects directly to the extracted watermark quality. In this paper, the authors focus on discussing and proposing the algorithm to determine an effective threshold automatically. This work is implemented by considering the probability distribution function (pdf) of significant differences. Figure 7s (a)-(d) shown the pdf of different values of a watermarked image under non-attack, histogram equalization and average filter (3×3), and (5×5), respectively. For the non-attack case, it is easy to observe two distinguished regions (one for 1-bits and another for 0-bits), therefore, the threshold is easily determined as the value in the range of two regions. However, this task is more difficult for attack cases such as average filtering. The purpose in this stage is determination of a threshold y to minimize the error of recovered bit in extraction.

In the sequence of proposed method, the authors recommend the algorithm to define the threshold based on the Otsu method which is usually utilized in the image segmentation application. The basic idea of the Otsu method is definition of the value which will separate the image histogram into two segments, called the object and the background with the minimization of the intra-class variance. In order to apply this method as the solution for this challenge, the authors consider 1-bits and 0-bits regions of the distribution of difference in the Figure 7 (a) as two segments of background and object need to be separated. The different value as the threshold is automatically made by computing the sum of weighted within-class variances of two classes at each significant different value. And the threshold is chosen as the value, in which the sum is the minimum value. Besides that, the other advantage of this method is that the threshold is given adaptively corresponding to different images and attacking types. It is not difficult to note that the decision of value is produced based on distribution of block difference which is easy to be affected by attacking types. However, there is a limitation in some strong attack cases whenever the different values become very large and affect directly on computing the threshold. In order to solve this



(a)

(b)

(c)

(d)

Figure 7. The probability distribution function of significant differences of embedded image under various cases: (a). Non-attack (b). Histogram equation (c). Average filter (3×3) (d). Average filter (5×5).

shortcoming, only the differences $(d_i > T)$ of the blocks are modified to satisfy for computation of variances. It is noted that this alteration hardly ever effect the quality of a recovered watermark. The modifying is computed as the following equation:

$$d_i = \begin{cases} d_i & \text{if } (d_i \leq T) \\ T & \text{otherwise} \end{cases} \tag{5}$$

The weighted-variance threshold can be defined through the weighted sum of variance of two classes σ_W^2 as

$$\sigma_W^2(d) = W_0(d)\,\sigma_0^2(d) + W_1(d)\,\sigma_1^2(d), \tag{6}$$

where class probabilities W_0 and W_1 of difference d are estimated:

$$W_0(d) = \sum_{i=1}^{d} p(i) \tag{7a}$$

$$W_1(d) = \sum_{i=d+1}^{\max(d_i)} p(i) \tag{7b}$$

where $p(i)$ is the pdf of difference values. And the class means are given by:

$$\mu_0(d) = \sum_{i=1}^{d} \frac{i \times p(i)}{W_0(d)};\, \mu_1(d) = \sum_{i=d+1}^{\max(d_i)} \frac{i \times p(i)}{W_1(d)} \tag{8}$$

Finally, the individual class variances are:

$$\begin{aligned} \sigma_0^2(d) &= \sum_{i=1}^{d} \left((i - \mu_0(d))^2 \frac{p(i)}{W_0(d)} \right) \\ \sigma_1^2(d) &= \sum_{i=d+1}^{\max(d_i)} \left((i - \mu_1(d))^2 \frac{p(i)}{W_1(d)} \right) \end{aligned} \tag{9}$$

The significant different value with minimum of weighted sum of variance is defined as the threshold for extraction:

$$y = d | \sigma_W^2(d) = \min_i \left(\sigma_W^2(i) \right) \tag{10}$$

After determining threshold, the equation (4) is used to

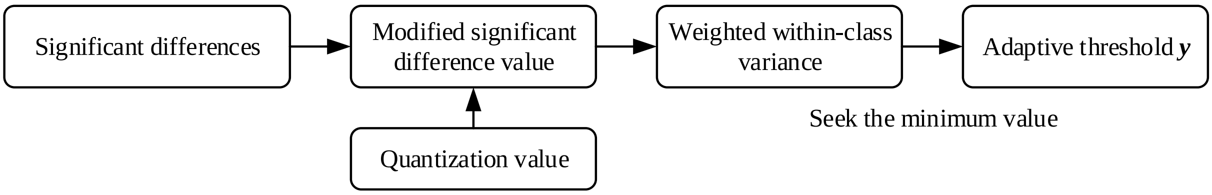


Figure 8. The flowchart for the process of determining adaptive threshold using Otsu method.

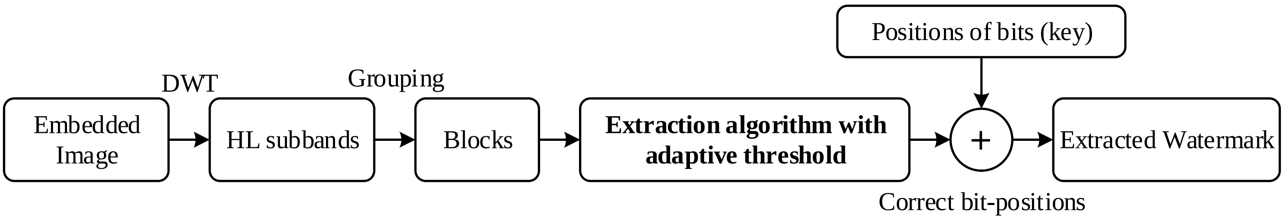


Figure 9. The flowchart of extraction process. The bit position achieved in the embedding stage is used as the key for recover the watermark.

extract the watermark. All step for determining threshold are showed in the Figure 8. The detail extraction algorithm is shown in the Figure 9 or listed as follows:

Input: An embedded image and a security key.

Output: An extracted binary watermark.

- Step 1: The embedded image is decomposed at 4-level of DWT.
- Step 2: The wavelet coefficients of HL3 and HL4 are grouped into the blocks.
- Step 3: Compute and modify the significant differences using the equation (5) to define the threshold y based on the proposed algorithm (the equation (6) - (10)).
- Step 4: Extract watermark bits by using the equation (4).
- Step 5: The extracted watermark is reshuffled with the key which includes the bit-positions to recovery the watermark.

3 Experimental Result

The peak signal-to-noise ratio PSNR is used to evaluate the quality between the embedded image and the original image. This formula is defined as follows:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{11}$$

where MSE is computed as below equation:

$$MSE = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I\left(i,j\right) - I'\left(i,j\right)\right)^2 \tag{12}$$

where M and N are the height and width of the image, respectively. $I\left(i,j\right)$ and $I'\left(i,j\right)$ are the grey value located at coordinate $\left(i,j\right)$ of the original image and embedded image. After recovering, an estimation for quality of extracted watermark is compared to the original watermark is expressed according the normalized correlation NC value:

$$NC = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n} w(i,j) \times w'(i,j) \tag{13}$$

where $m \times n$ is the watermark of size. $w(i,j)$ and $w'(i,j)$ are the values located at coordinate (i,j) of original watermark and extracted watermark. Value of $w(i,j)$ is set 1 if it is a watermark bit 1, otherwise, it is set -1 and similar to $w'(i,j)$. So the value of $w(i,j) \times w'(i,j)$ is either 1 or -1. If number of bits is extracted correctly, the NC value is positive; otherwise, it is negative. In this paper, the proposed method for watermarking digital image will be estimated under various aspects to consider the robustness of a watermark on the Computing and Simulation Software MATLAB.

3.1 Quality of embedded images

In the below simulation, the authors used 15 gray-scale images, namely Lena, Goldhill, Baboon, Pepper, Zelda and 10 other images (512 × 512 pixels, 8 bits/pixel) obtained from the USC-SIPI set; and the watermark is the binary image (16 × 32 pixels, 1 bits/pixel). Some example images and the watermark are shown in the Figure 10. In additions, the authors apply the Haar wavelets to decompose and reconstruct the cover image. The simulation results of embedding and extraction process in non-attack case are shown in the Figure 11. Based on the experience of quantization value as in the Figure 5, the parameter was set at $T = 60$.

3.2 Robustness of extracted watermark

Under various attacks, the authors also assess the proposed method for 15 gray-scale images, however, only *Lena* image have been shown in the Figure 12 and Figure 13. The experimental results of the other images can be observed in the Table I. There are some geometric attacks are considered in this simulation, such as, cropping, scaling, rotation and Gaussian noise. For the cropping attack, $^1/_4$ image removed at the center of the embedded image has been replaced by the gray value 0 pixels. In the scaling case, the host image is scaled to 256 × 256 and then scaled one more time to the original size. For two last cases of geometric attack, the cover image is rotated by small degree with

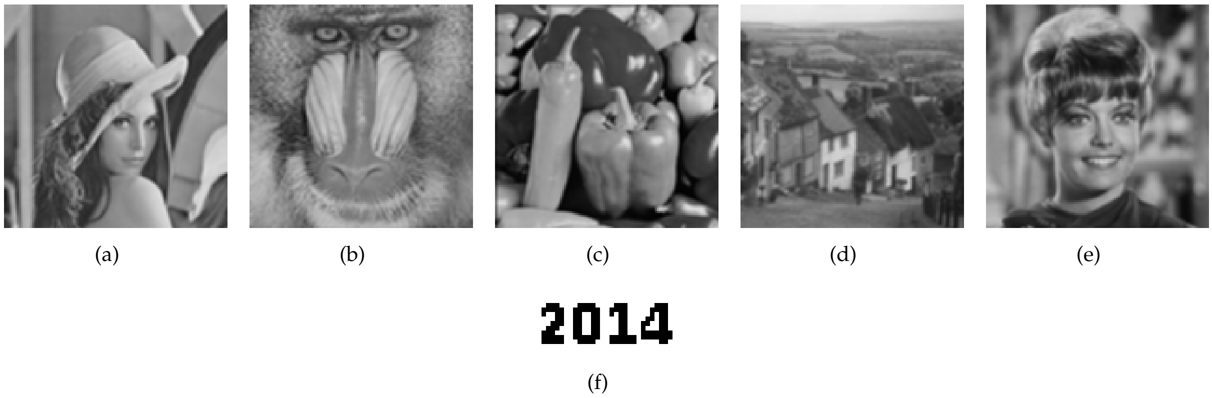


Figure 10. The original image 512 × 512: (a) Lena, (b) Baboon, (c) Pepper, (d) Goldhill, (e) Zelda and (f) The watermark 16 × 32

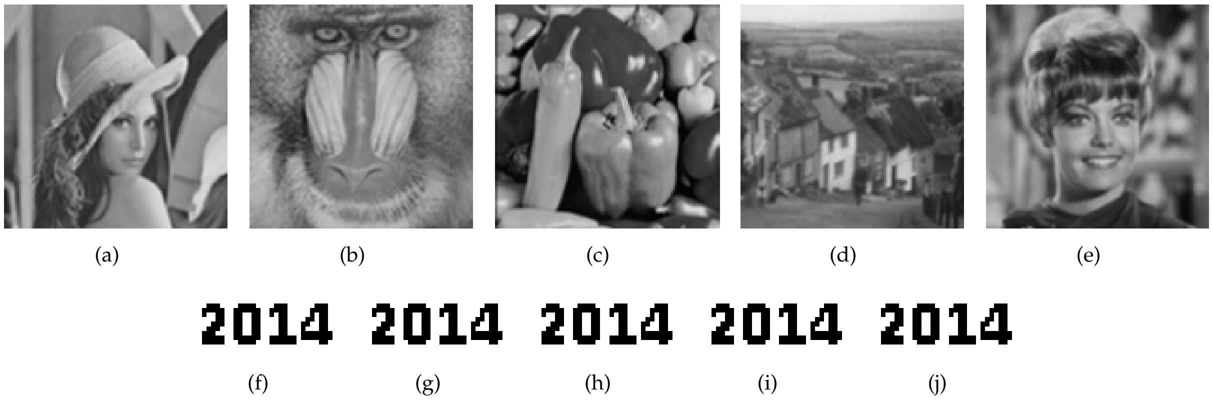


Figure 11. The embedded image 512 × 512: (a) Lena $(PSNR = 43.88dB)$, (b) Baboon $(PSNR = 48.53dB)$, (c) Pepper $(PSNR = 45.52dB)$, (d) Goldhill $(PSNR = 47.75dB)$, (e) Zelda $(PSNR = 45.99dB)$. The extracted watermark under non-attackx of: (f) Lena $(NC = 1.00)$, (g) Baboon $(NC = 1.00)$, (h) Pepper $(NC = 1.00)$, (i) Goldhill $(NC = 1.00)$, (j) Zelda $(NC = 1.00)$

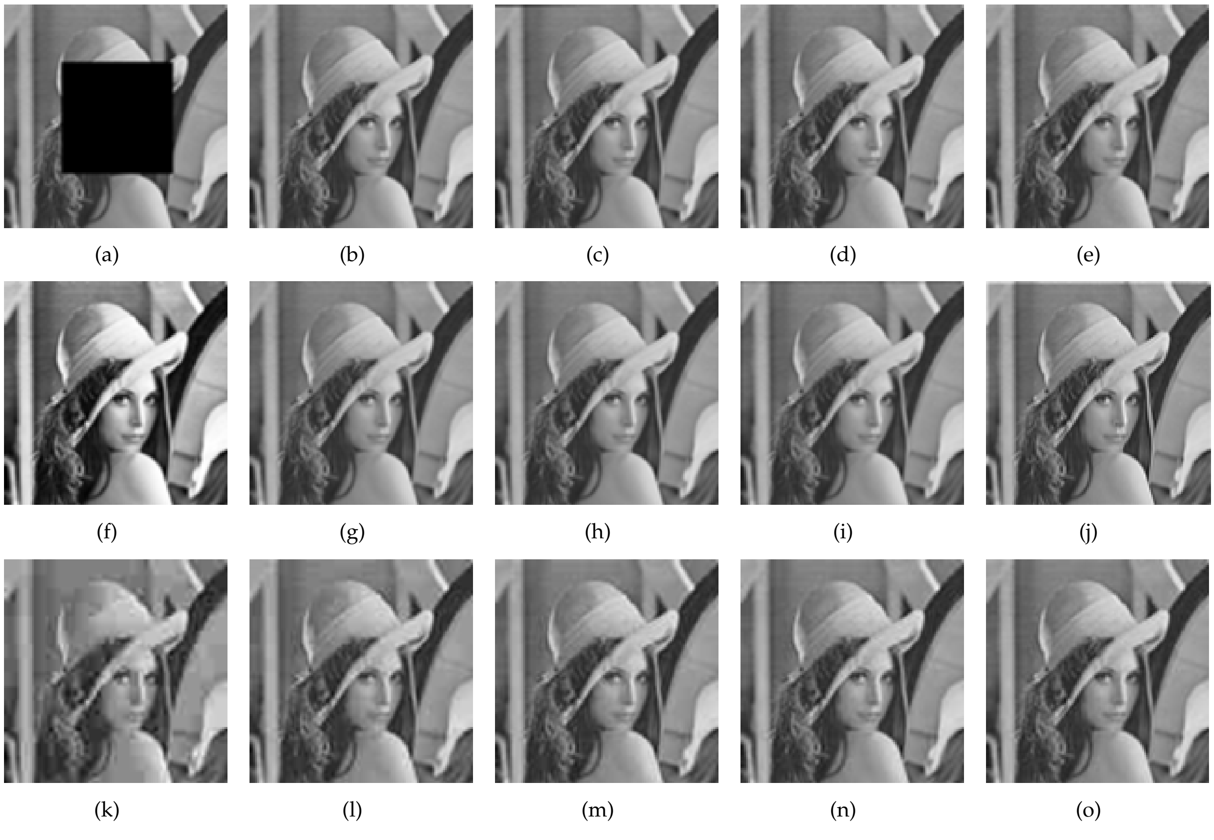


Figure 12. The embedded image under **Geometric Attacks**: (a) Cropping, (b) Scaling, (c) Rotation 0.25^0, (d) Gaussian Noise (mean=0, variance=0.001), (e) Gaussian Noise (mean=0, variance=0.002); **Non-Geometric Attacks**: (f) Histogram Equalization, (g) Gaussian Filter, (h) Median Filter, (i) Average Filter, (j) Sharpening and **JPEG Compression**: (k) QF=10%, (l) QF=20%, (m) QF=30%, (n) QF=40%, (o) QF=50%

0.25^0 and added Gaussian noise with mean = 0 and various variances. The attacked images for geometric damages can be considered in the Figure 12 (a) - (e) and the Figure 13 (a) - (e) represented the robustness of extracted watermark through NC parameter.

With non-geometric attack cases are shown in the Figure 12 (f) - (j) and Figure 13 (f) - (j), the proposed method had a good resistance for *Lena* image. As the default setting, the filters used in this simulation have mask of size (3×3): Gaussian Filter, Median Filter, Average Filter and Sharpening. The Figure 13 (f) shows the recovered watermark extracted from the host image after using global histogram equalization. Moreover, the Figure 12 (k) - (o) and Figure 12 (k) - (o) show the simulation results in JPEG lossy compression with quality factor, denoted QF, from 10 to 50. It is important to note that normalized correction of watermark achieves 0.97 when $QF = 10$. The Table I also shown the average NC of the extracted watermark under various attack types of 15 test images.

Another assessment is the comparison of two manner of determining threshold for watermark extraction process: one is the Otsu threshold value and another is the fixed value. One sample, namely Lena, is picked up to evaluate and assess the effect of this value on the extraction performance. The result of this assessment is represented in the Table II, in which the extraction algorithm is employed with the fixed threshold applying for all of attacking types. Through the result under NC parameter, the algorithm with support from Otsu threshold achieves high accuracy in most cases.

Table I
Average Result of 15 Test Images under Various Attacks

Attacking types	Normalized Correlation
Cropping	0.54
Scaling	0.97
Rotation 0.25^0	0.92
Gaussian Filter	0.99
Histogram Equalization	0.94
Sharpening	0.98
Median Filter 3 × 3	0.98
Median Filter 5 × 5	0.94
Average Filter 3 × 3	0.96
Average Filter 5 × 5	0.92
JPEG $QF = 10\%$	0.95
JPEG $QF = 20\%$	0.99
JPEG $QF = 30\%$	1.00
JPEG $QF = 40\%$	1.00
JPEG $QF = 50\%$	1.00

3.3 Comparing to Other Methods

In this section, the authors compared the proposed method to some recent methods: Lin et al. [3], Run et al. [5], Wu et al. [6]. In Table III, the experimental results of the proposed method are far better than the listed methods in the almost cases of attacking, except the cropping. Filtering wavelet-tree blocks based on the significant differences is the reason of achieving the high robustness of watermark with the minimum degradation of embedded image. It in important to note that the strength of watermarking can be dynamically controlled through the quantization value. The shortcoming of the proposed method is the limitation of the payload of the embedded information. In this paper the maximum number of bits can be embedded is 1024. However, enlarging the capacity of embedded watermark is not impossible if we utilize the 2-level or 1-level sub-bands instead of 3-level sub-bands as in our paper. The unexpected behavior in that case is attenuation of robustness. This statement is basically explained by the characteristic of wavelet decomposition in which the high-level coefficients is more rigid than lower-level ones. In additions, unlike the algorithm used to determine threshold represented in Lin [3] and Run [5] method which employed the scale parameter α to define the value of threshold y [5]:

$$y = \left[\frac{1}{N_W \times \alpha} \sum_{j=1}^{N_W \times \alpha} \varphi_j \right] \tag{14}$$

Table III
Comparing the Proposed Method to the Other Methods: Wu (2007), Lin (2008), Run (2011)

Attacking types	Wu[6]	Lin[3]	Run[5]	**Proposed**
Cropping	NA	0.70	0.68	0.54
Scaling	NA	0.86	0.86	0.97
Rotation 0.25^0	NA	0.67	0.65	0.92
Gaussian Filter	0.82	0.86	0.95	0.99
Histogram Equalization	NA	0.77	0.86	0.94
Sharpening	1.00	0.99	0.99	0.98
Median Filter 3 × 3	NA	0.88	0.93	0.98
Median Filter 5 × 5	0.79	0.74	0.84	0.94
Average Filter 3 × 3	NA	0.91	0.95	0.96
Average Filter 5 × 5	NA	0.72	0.80	0.92
JPEG $QF = 10\%$	NA	0.41	0.32	0.95
JPEG $QF = 20\%$	NA	0.68	0.68	0.99
JPEG $QF = 30\%$	0.93	0.87	0.85	1.00
JPEG $QF = 40\%$	NA	0.95	0.93	1.00
JPEG $QF = 50\%$	0.99	0.98	0.96	1.00

where $\phi = \{\max_1 - \sec_1, ..., \max_i - \sec_i\}$, for $i = 1, 2, ..., N_W$; where N_W is the number of modified blocks. In (14), α is the scale parameter, $0 < \alpha \leq 1$. In these algorithms, α used to determine how many percentages of significant difference in ϕ is used for the average. The threshold which is defined in this algorithm is able to control through the scale parameter α, however, only the constant value of α have been set for various attack types. Therefore, it can be seen that no parameter need to be tuned is the advantage of the proposed algorithm in this paper.

4 Conclusion

In this research, we proposed the blind digital watermarking for gray-scale images using the weighted-variance threshold for extracting. The binary image is used as the watermark is embedded into the host image based on super wavelet tree technique. The significant difference of each wavelet tree has been

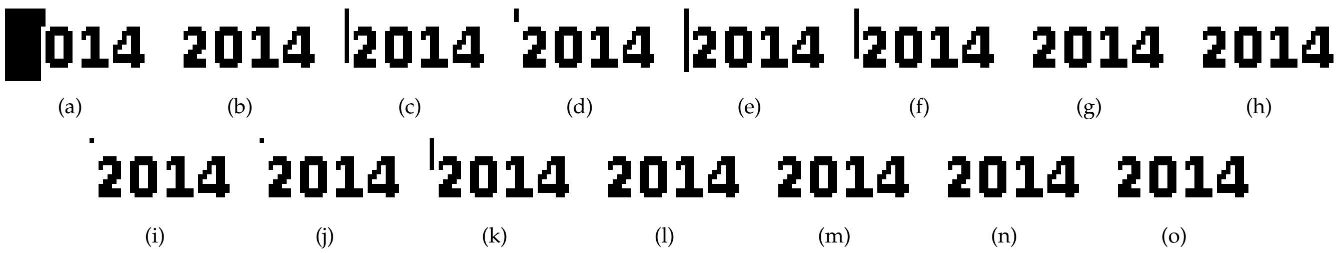


Figure 13. The extracted watermark of embedded image in Figure 12: (a) Cropping ($NC = 0.54$), (b) Scaling ($NC = 1.00$), (c) Rotation 0.25^0 ($NC = 0.95$), (d) Gaussian Noise (mean=0, variance=0.001) ($NC = 0.99$), (e) Gaussian Noise (mean=0, variance=0.002) ($NC = 0.95$), (f) Histogram Equalization ($NC = 0.96$), (g) Gaussian Filter ($NC = 1.00$), (h) Median Filter ($NC = 1.00$), (i) Average Filter ($NC = 1.00$), (j) Sharpening ($NC = 1.00$), (k) QF=10% ($NC = 0.97$), (l) QF=20% ($NC = 1.00$), (m) QF=30% ($NC = 0.99$), (n) QF=40% ($NC = 1.00$), (o) QF=50% ($NC = 1.00$)

Table II
Comparing Extraction Accuracy (NC) between Otsu and Fixed Threshold Value under Various Attacking Types

Attacking type	Otsu threshold	Fixed threshold						
		15	20	25	30	35	40	45
Cropping	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
Scaling	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Rotation 0.25^0	0.95	0.97	0.96	0.95	0.92	0.83	0.69	0.50
Gaussian.N (mean=0, variance=0.001)	0.99	0.94	0.98	1.00	0.99	0.97	0.91	0.80
Gaussian.N (mean=0, variance=0.002)	0.95	0.92	0.96	0.99	0.94	0.83	0.73	0.60
Histogram equalization	0.96	1.00	0.99	0.98	0.98	0.98	0.96	0.94
Gaussian filter (3×3)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97
Median filter (3×3)	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.84
Median filter (5×5)	0.96	0.99	0.96	0.94	0.89	0.71	0.46	0.20
Average filter (3×3)	1.00	1.00	1.00	1.00	1.00	0.98	0.84	0.46
Average filter (5×5)	0.95	0.98	0.95	0.88	0.64	0.43	0.23	0.03
Sharpening	1.00	0.98	0.99	1.00	1.00	1.00	0.99	0.97
JPEG $QF = 10\%$	0.97	0.97	0.97	0.97	0.88	0.88	0.75	0.74
JPEG $QF = 20\%$	1.00	0.94	0.98	0.99	0.99	0.92	0.85	0.84
JPEG $QF = 30\%$	1.00	0.97	1.00	1.00	1.00	0.99	0.97	0.90
JPEG $QF = 40\%$.	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97
JPEG $QF = 50\%$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98

modified by comparing to the quantization value. The improvement in embedding process is that the wavelet blocks were chosen for watermarking based on their significant differences instead of randomization. In extraction, the watermark is basically recovered by comparison of different values and the threshold. Therefore we proposed the algorithm to define the threshold based on computing the weighted within-class variance through considering the probability density function of significant difference. In experimental results, the proposed method produced the embedded images with preserving the original visualization while the extracted watermark is recovered exactly in the non-attack case. In addition, the method is also tested under various attack types, such as, JPEG compression, Histogram Equalization, Low-frequency Filter, Sharpening... Comparing with the other recent methods, no requirement of any parameters in extraction except security key can be seen the advantage of our method.

References

[1] M. Hsieh, D. Tseng, and Y. Huang, "Hiding digital watermarks using multiresolution wavelet transform," *IEEE Transactions on Industrial Electronics*, vol. 48, pp. 875–882, Oct 2001.

[2] B. K. Lien and W. Lin, "A watermarking method based on maximum distance wavelet tree quantization," *19th Conf. Computer Vision, Graphics and Image Processing*, pp. 269–276, 2006.

[3] W. H. Lin, S. J. Horng, T. W. Kao, P. Fan, C.-L. Lee, and Y. Pan, "An efficient watermarking method based on significant difference of wavelet coefficient quantization," *IEEE Transactions on Multimedia*, pp. 746–757, Aug. 2008.

[4] W. H. Lin, Y. Rau, and S. J. Horng, "A wavelet-tree-based watermarking method using distance vector of binary cluster," *Expert Systems with Applications*, pp. 9869–9878, 2009.

[5] R. S. Run, S. J. Horng, W. H. Lin, T. W. Kao, P. Fan, and M. K. Khan, "An efficient wavelet-tree-based watermarking method," *Expert Systems with Applications*, pp. 14 357–14 366, 2011.

[6] G.-D. Wu and P.-H. Huang, "Image Watermarking Using Structure Based Wavelet Tree Quantization," *6th IEEE/ACIS International Conference on Computer and Information Science*, pp. 315–319, 2007.

[7] A. Ding and S. Dong, "Algorithm of Digital Image Watermark Based on Decomposition of Wavelet Packet," *2012 Sixth International Conference on Internet Computing for Science and Engineering*, pp. 135–137, 2012.

[8] L. Fan and T. Gao, "A Novel Blind Robust Watermarking Scheme Based on Statistic Characteristic of Wavelet Domain Coefficients," *International Conference on Signal Processing Systems*, pp. 121–125, May 2009.

[9] P. Liu and Z. Ding, "A Blind Image Watermarking Scheme Based on Wavelet Tree Quantization," *Second In-*

ternational Symposium on Electronic Commerce and Security, vol. 1, pp. 218–222, May 2009.

[10] J. Huang and C. Yang, "Image Digital Watermarking Algorithm Using multiresolution Wavelet Transform," *2004 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2977–2982, Oct. 2004.

[11] J. H. Uhl, C. Koidl, and A. Uhl, "Multiple Blind Re-watermarking with Quantization-Based Embedding," *18th IEEE International Conference on Image Processing*, pp. 265–268, 2011.

[12] T. Huynh-The and T. Le-Tien, "An Efficient Blind Watermarking Method based on Significant Difference of Wavelet Tree Quantization using Adaptive Threshold," *International Journal of Electronics and Electrical Engineering*, vol. 1, pp. 98–103, June 2013.

Thien Huynh-The was born in 1988 in Bentre province, Vietnam. He received his B.Eng and M.Sc from Hochiminh City University of Technical and Education, HCMUTE, Vietnam in 2011 and 2013, respectively. Currently he is pursuing his Ph.D degree in the Kyung Hee University, Korea. His research interests are image and video processing, computer vision, and machine learning.

Thuong Le-Tien was born in 1957 in Saigon, Vietnam. He received his B.Eng and M.Sc from the Ho Chi Minh city University of Technology (HCMUT), Vietnam then Ph.D degree from the University of Tasmania, Australia. He has been a lecturer at HCMUT since 1981 and is appointed to various academic positions at the University. He has been awarded the national associate professor title since 2003 then the national distinguished lecturer since 2008 in Vietnam and the distinguished invited professor from Mannheim University of Applied Science Germany, in 2009. His academic interests include Signal Processing, Electronics and Digital Communications.

Tuan Nguyen-Thanh received the B.Eng. and M.Eng. degrees from HoChiMinh City University of Technology, Vietnam, in 2002 and 2004, respectively, both in electrical engineering and telecommunications. He has been a lecturer with the same university since 2002. Currently, he is pursuing the Ph.D. degree at University of New South Wales, Sydney, Australia, under the supervision of Prof. David Taubman. His main research interests include watermarking, digital signal processing and communication systems.