Circuits Syst Signal Process (2008) 27: 171–182
DOI 10.1007/s00034-008-9025-z



# Genetic Watermarking for Zerotree-Based Applications

**Shu-Chuan Chu · Hsiang-Cheh Huang · Yan Shi · Ssu-Yi Wu · Chin-Shiuh Shieh**

Received: 5 May 2007 / Revised: 16 September 2007 / Published online: 27 February 2008


**Abstract** An optimized scheme for watermarking based on zerotrees is proposed in this paper. Digital watermarking is an effective means for protecting copyrights with three of the most fundamental requirements: watermark imperceptibility, watermark robustness, and watermark capacity. Conventional techniques in the literature mainly perform watermark embedding and extraction processes in the transform domain, including the discrete Fourier transform, discrete cosine transform, and discrete wavelet transform domains. The three watermarking requirements above are in conflict with each other; therefore, finding a way to obtain a trade-off among them is the major purpose of this paper. We first perform watermarking in the wavelet domain. Next, we properly select zerotrees in a wavelet transform with the genetic algorithm. Our simulation results not only demonstrate better performances of the watermarked images after optimization, but also reveal the robustness of the extracted watermarks under common attacks.

**Keywords** Watermarking · Zerotree · Wavelet transform · Genetic algorithm (GA) · Optimization

S.-C. Chu
Department of Information Management, Cheng Shiu University, Kaohsiung, Taiwan, ROC

H.-C. Huang (✉)
Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung, Taiwan, ROC
e-mail: huang.hc@gmail.com

Y. Shi
School of Information Science, Kyushu Tokai University, Kumamoto, Japan

S.-Y. Wu · C.-S. Shieh
Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan, ROC

## 1 Introduction

In response to the increasing demand of distributing multimedia clips over the Internet, watermarking technology has received considerable attention in the past few years. Aimed at copyright protection, arbitration, and authentication, watermarking is the process of embedding extra information into a media clip. Major implementations for digital watermarking are focused on transform domains, especially the discrete Fourier transform (DFT) [10, 11], discrete cosine transform (DCT) [6, 10, 17], and discrete wavelet transform (DWT) [5, 8] domains.

There are requirements and constraints in designing effective watermarking algorithms. An invisible and robust watermark may be the most difficult challenge among all the types of watermarks. Considering all the requirements for watermarking research and applications, the three most fundamental ones are (i) *watermark imperceptibility*, or the quality of the watermarked image, (ii) *watermark robustness*, or the capability to resist malicious image processing, called attacks, and (iii) *watermark capacity* or the number of bits for embedding [2, 7]. With these requirements, people often perform image watermarking in the transform domain to embed the watermark bits into certain transform coefficients.

The three watermarking requirements above are in conflict with each other [1, 11]. Heuristically speaking, to ensure the robustness and imperceptibility while retaining a reasonable capacity, many researchers proposed to embed the watermark bits into the "middle frequency bands" of the transform coefficients [6, 16]. In this paper, we propose a systematic way for designing a wavelet-based watermarking algorithm by taking the requirements into account. Wavelet-based scalable-coded multimedia applications, including MPEG scalable video coding (SVC) [15] and JPEG2000 [18], are getting more and more attention nowadays. With the experience that watermarking has been applied to DCT-coded multimedia, we can expect that its counterpart for wavelet-coded multimedia should be an important application for research. We employ the genetic algorithm (GA) [3] to select appropriate zerotrees in the wavelet transform [5, 8] to pursue both the watermarked image quality, and the robustness of an extracted watermark under planned attacks. The proposed algorithm can be extended to the application of scalable-coded multimedia [1].

Taking practical applications into account, we choose to keep the watermark capacity fixed, while optimizing the requirements of imperceptibility and robustness. We do this because most algorithms in the literature [9, 12] use a fixed size for watermark embedding. The two remaining requirements, namely, imperceptibility and robustness, are in conflict with each other, and they need to reach a trade-off to be optimized by the GA. By properly selecting the fitness function, we are able to optimize both the imperceptibility of the watermarked image quality, measured by the peak signal-to-noise ratio (PSNR), and the robustness of the watermarking algorithm, measured by the bit correct rate (BCR), between the watermark extracted from the attacked watermarked image and the embedded counterpart. Simulation results demonstrate the effectiveness of the proposed scheme.

This paper is organized as follows. We briefly describe the fundamental concepts of wavelet transform and zerotrees in Sect. 2. Section 3 discusses the background of the GA. Section 4 describes the data embedding and extraction schemes employed in

this paper. Section 5 depicts the watermarking algorithm with genetic zerotree selection. We differentiate the trade-off between image quality and watermark robustness in Sect. 6. Next, Sect. 7 illustrates the simulation results, and we also show the superiority of our scheme over the results proposed by other researchers in this section. We conclude this paper in Sect. 8.

## 2 Wavelet Transform and Zerotrees

A wavelet transform decomposes signals based on certain basis functions. With the wavelet transform, people can inspect a signal from different aspects and obtain valuable insights. Among other basis functions, Haar functions have been used intensively in the realm of image processing. The Haar function-based wavelet transform recursively applies low-pass and high-pass filtering along the horizontal and vertical directions. In this paper, we implement with Haar functions to verify the usefulness of the proposed schemes, and they can be extended to other basis functions in wavelet transforms.

There exists an inherent parent-child relationship among the coefficients between successive decomposition levels in the wavelet domain. As illustrated in Fig. 1, those marked coefficients from different levels form a *quadtree*. A quadtree is defined to be a *zerotree* if all the coefficients have values below a certain threshold. In some sense, a zerotree represents a visually less significant region in the original image, and it is therefore an excellent candidate for watermark embedding. A watermark bit is embedded by adding or subtracting a pre-defined value to or from the coefficients of a selected zerotree.

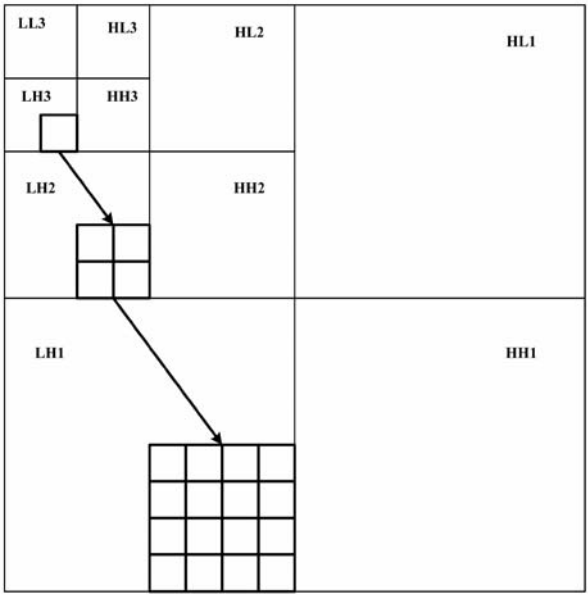


**Fig. 1** Illustration of a zerotree structure in the DWT domain

## 3 Background of Genetic Algorithm

Conventional search techniques are often incapable of optimizing nonlinear functions with multiple variables. One scheme called the "genetic algorithm" (GA) [3], based on the concept of natural genetics, is a directed random search technique developed in 1975. In the GA, parameters are represented by an encoded binary string, called the *chromosome*. The elements (or bits) in the binary strings, or the *genes*, are adjusted to minimize or maximize the fitness value. The fitness function generates its fitness value, which is composed of multiple variables to be optimized by the GA. For every iteration in the GA, a pre-determined number of individuals will correspondingly produce fitness values associated with the chromosomes.

The GA begins by defining the optimization parameters, the fitness function, and consequently the fitness value, and it ends by testing for convergence. Three major building blocks in the GA include selection, crossover, and mutation. They are briefly described as follows:

- Selection: In the training process, a large portion of the low fitness individuals is discarded through this natural selection step.
- Crossover: Two individuals are chosen from the mating pool of $N_{\text{good}}$ individuals, meaning those with larger fitness values or those with a better chance for survival, to produce two new offsprings. A crossover point is selected between the first and last chromosomes of the parent individuals. Then the fraction of each individual after the crossover point is exchanged and concatenated.
- Mutation: This step can introduce traits not found in the original individuals and keeps the GA from converging too fast. The simplest way to do this is intentionally flip some randomly selected bits in the chromosome. Generally speaking, by following the findings in genetics, the probability for mutation is supposed to be low.

According to the applications for optimization, designers need to carefully define the necessary elements for training with the GA. Then, the fitness function in addition to the terminating criteria is evaluated with the natural selection, crossover, and mutation operations [3]. We will describe in detail the relationships of optimizing a zerotree-based watermarking algorithm with the GA in Sect. 5.

## 4 Data Embedding and Extraction with Zerotrees

Let the input image be $\boldsymbol{X}$ with size $M \times N$. Our goal is to embed a robust watermark in the DWT domain, and to have a watermarked reconstruction $\boldsymbol{X}'$. We assume that the binary-valued watermark to be embedded is $\boldsymbol{W}$, which has the capacity of $M_W \times N_W$ bits.

First, the input image $\boldsymbol{X}$ is divided into $8 \times 8$ blocks $\boldsymbol{b}_{i,j}$,

$$\boldsymbol{X} = \bigcup_{i=1}^{M/8} \bigcup_{j=1}^{N/8} \{\boldsymbol{b}_{i,j}\}. \tag{1}$$

We perform a DWT independently to every block $\boldsymbol{b}_{i,j}$, $1 \le i \le \frac{M}{8}$, $1 \le j \le \frac{N}{8}$,

$$\boldsymbol{W}_{i,j}(k) = \mathrm{DWT}(\boldsymbol{b}_{i,j}), \tag{2}$$

where $k$ denotes the frequency coefficients in the wavelet domain.

The selection of zerotrees plays a major role in embedding the watermark. First, a threshold value is chosen to aid the zerotree selection. For one block $\boldsymbol{b}_{i,j}$, we first calculate the maximum value, $M_{i,j}$, among all the wavelet coefficients $\boldsymbol{W}_{i,j}(k)$ within such a block,

$$M_{i,j} = \max\bigl(\boldsymbol{W}_{i,j}(k)\bigr). \tag{3}$$

Next, we choose a weighting factor $\alpha$ to determine the threshold,

$$T_{i,j} = \alpha \cdot M_{i,j}, \tag{4}$$

and we set $\alpha = 0.5$ to be the initial value for the threshold in this paper.

Now, with the determined thresholds, the zerotrees are ready to be selected. The whole image can be changed into an $M \times N$ binary matrix such that

$$\boldsymbol{B} = \bigcup_{i=1}^{M/8} \bigcup_{j=1}^{N/8} \bigcup_{p=1}^{8} \bigcup_{q=1}^{8} \{B_{8(i-1)+p,8(j-1)+q}\}, \quad \text{and} \tag{5}$$

$$B_{p,q} = \begin{cases} 0, & \text{if } W_{i,j}(k) < T_{i,j}; \\ 1, & \text{if } W_{i,j}(k) \ge T_{i,j}. \end{cases} \tag{6}$$

Based on the $\boldsymbol{B}$ matrix, the zerotrees are ready to be selected. The number of zerotrees should be four times the capacity of the watermark, $4 \times M_W \times N_W$. Thus, the weighting factor $\alpha$ should be adaptively adjusted. If the number of zerotrees is larger than four times the watermark capacity, the threshold should be decreased; otherwise, it should be increased. Hence, we propose an equation to adaptively adjust the threshold value, which is a modification of (4),

$$T_{i,j} \leftarrow \begin{cases} 0.5 \cdot T_{i,j}, & \text{if too many zerotrees generated;} \\ 1.5 \cdot T_{i,j}, & \text{otherwise.} \end{cases} \tag{7}$$

This procedure is adjusted repeatedly until the correct number of zerotrees, or four times the watermark capacity, is generated.

As we know, the watermark is embedded by systematically modifying certain coefficients, for instance, the zerotrees in the wavelet domain in this paper. Here we set the watermarking strength $P$ for watermark embedding. We use direct replacement of coefficient values for watermark embedding based on one of the two rules below.

- If the embedded bit is 1, the value $P$ is employed to replace the existing zerotree.
- If the embedded bit is 0, the value $-P$ is used to replace the existing zerotree.

We can see that the watermarking strength $P$ plays an important role to balance the watermark imperceptibility and watermark robustness. If $P$ is too small, the robustness may be degraded while we reach a good level for imperceptibility. If $P$ is

too large, imperceptibility will be sacrificed but the robustness may improve. Based on our experiments, we set $P = 10$ to obtain a trade-off for imperceptibility and robustness.

After embedding the watermark, the inverse DWT (IDWT) is performed, and we have a watermarked reconstruction $\boldsymbol{X}'$. According to the definitions in statistics, the mean-squared error (MSE) between the original and watermarked images is defined by

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \big(X(i,j) - X'(i,j)\big)^2, \tag{8}$$

where $X(i,j)$ and $X'(i,j)$ denote the pixel value at position $(i,j)$ of the original image $\boldsymbol{X}$ and the watermarked reconstruction $\boldsymbol{X}'$, respectively. Consequently, the watermarked image quality is represented by the PSNR between $\boldsymbol{X}$ and $\boldsymbol{X}'$, formulated by

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right) (\mathrm{dB}). \tag{9}$$

As we know, the larger the PSNR value, the more imperceptible the watermarked image. This value can be regarded as the measure for imperceptibility.

Next, we study the watermark robustness. The watermarked image is supposed to be delivered to the receiver. During transmission, we can expect that the watermarked image may face intentional or unintentional image processing, called attacks. A watermarking algorithm is considered robust if some data relating to the embedded watermark can be extracted from the attacked images. A commonly used measure for watermarking robustness is the BCR, which can be represented by

$$\mathrm{BCR} = \left(1 - \frac{1}{M_W \times N_W} \sum_{t=1}^{M_W \times N_W} \big(W_t \oplus W_t'\big)\right) \cdot 100\%, \tag{10}$$

where $W_t$ and $W_t'$ represent the embedded and extracted watermark bits, $M_W \times N_W$ denotes the watermark size, and $\oplus$ means the exclusive-or (XOR) operation. We can see that BCR is the percentage of correctly extracted bits to the embedded watermark capacity. When BCR reaches 100%, all the embedded bits are correctly extracted.

## 5 Genetic Zerotree Selection

Although previous works [5, 8] on zerotree-based watermarking had achieved promising results, we anticipate that better performances in terms of imperceptibility and robustness can be expected if we judiciously select the zerotrees by considering the impact of watermarked image quality and potential attacks. Obtaining the required number of zerotrees from $N$ given candidates, while at the same time taking into account the effect of possible attacks, is a difficult optimization program with a complexity of order $O(N!)$. We have developed a new scheme, named *genetic zerotree watermarking*, to solve this challenging problem. The proposed algorithm uses the GA as its core mechanism for zerotree selection.

Let the input image be $\boldsymbol{X}$ with size $M \times N$, and the watermark be $\boldsymbol{W}$, with a capacity of $M_W \times N_W$ bits. The algorithm is outlined below in detail.

*Step 1 (Wavelet transformation)* The carrier image is partitioned into $8 \times 8$ blocks. Each block is subject to wavelet transform to obtain their wavelet coefficients, hence, there are $\frac{M}{8} \times \frac{N}{8}$ blocks.

*Step 2 (Candidate generation)* The zerotree threshold value determines how many zerotrees a block can possess. An interval-halving method is employed to decide an appropriate threshold value, which is shown in (4).

*Step 3 (Chromosome encoding)* The first issue in applying the GA is to design an adequate coding scheme to represent potential solutions in the form of chromosomes. We use binary string coding in our algorithm. In our simulations, each chromosome is a binary string of length $4 \times M_W \times N_W$. Each bit reflects the selection status of its corresponding candidate zerotree.

*Step 4 (Fitness evaluation)* We are now going to evaluate the imperceptibility and robustness of the zerotree configuration encoded in each chromosome, with the fitness function

$$f_c = \mathrm{PSNR}_c + \lambda \cdot \mathrm{BCR}_c, \tag{11}$$

where the subscript $c$ denotes the current iteration for training, $f_c$ is the fitness score, and $\lambda$ denotes the weighting factor to balance the effects between imperceptibility and robustness.

Each chromosome is decoded and used to guide the watermark embedding process by following schemes in [8]. Watermarked images are then subjected to planned attacks. There is a watermark attacking benchmark, called "Stirmark" [14], to evaluate the robustness of the watermarking algorithms. Moreover, transmission of watermarked media can also be regarded as an alternative means of attack [9, 12]. Not all watermarking applications require robustness to all possible signal processing operations. In addition, the watermarked image after attack needs to be worthy of being used or transmitted by others; therefore, an attack like image cropping is not employed in our GA training procedure. In this paper, we consider three major attacking schemes from Stirmark, namely, low-pass filtering (LPF) attack [4], median filtering (MF) attack [4], and JPEG attack with a quality factor of 80% [13]. The watermark extracted from the attacked carrier image, by employing schemes in [8], is compared with the original watermark to evaluate the robustness of the corresponding zerotree configuration. Objective measures, such as PSNR in (9) for representing imperceptibility, and BCR in (10) for representing robustness, can be employed for the evaluations.

*Step 5 (Selection and crossover)* Rank-based selection is adopted in our system with a chosen selection probability $p_s$. In our simulations, we set $p_s = 0.2$. That is, those with ranks within the top 20% of individuals, or the 20% of individuals with the larger fitness values, are used in the next iteration.

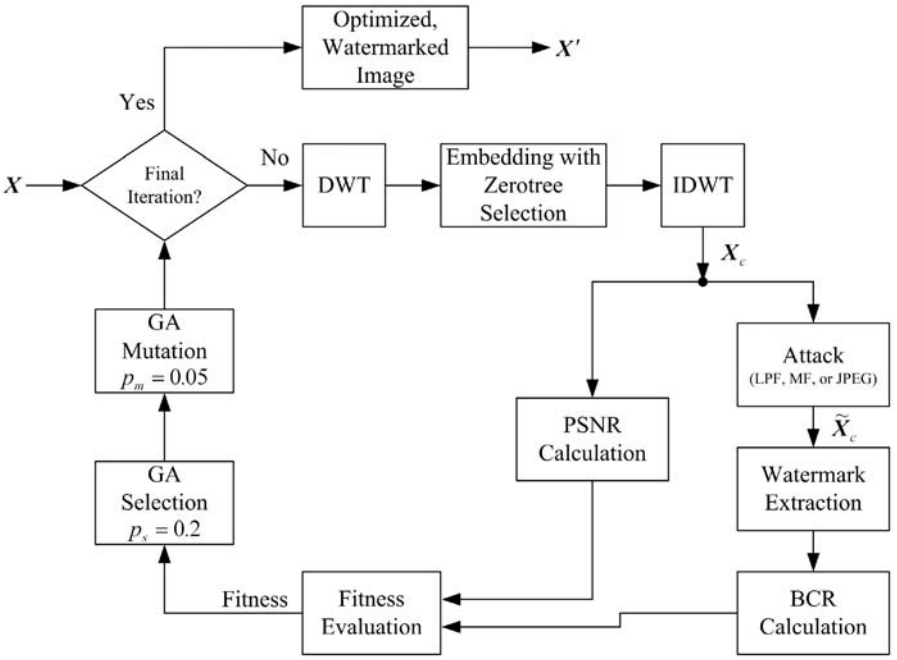


**Fig. 2** Flowchart for zerotree-based watermarking with GA optimization

*Step 6 (Mutation)* Except for the best chromosome, all the other chromosomes are subject to mutation in order to generate new solutions with a small mutation probability $p_m$. In our simulations, we set $p_m = 0.05$, meaning that 5% of the bits in the chromosomes are randomly selected and intentionally flipped.

*Step 7 (Termination check)* Repeat Steps 4 through 7 until a pre-determined number of iterations is reached.

The flowchart of the proposed schemes is depicted in Fig. 2.

## 6 The Trade-off between Image Quality and Watermark Robustness

Corresponding to Step 4 in Sect. 5, we consider both the received image quality and the watermark robustness for optimization by fixing the watermark capacity. Equation (11) is rewritten for convenience:

$$f_c = \mathrm{PSNR}_c + \lambda \cdot \mathrm{BCR}_c,$$

where $f_c, \mathrm{PSNR}_c$, and $\mathrm{BCR}_c$ denote the fitness value, PSNR value of the watermarked image, and BCR value of the extracted watermark in the $c$th iteration, or the current iteration, in GA, respectively.

It is found that watermarked PSNR values are generally larger than 35 dB based on DWT in our simulations, while BCR values lie between 0 and 1, thus we include

**Table 1** Comparisons of PSNR and BCR values with or without optimization, with $\lambda = 30$ in the fitness function

| Different attacks | With GA | | Without GA | |
|---|---|---|---|---|
| | PSNR | BCR | PSNR | BCR |
| LPF | 45.13 dB | 81.54% | 44.56 dB | 35.45% |
| MF | 44.95 dB | 85.06% | | 39.26% |
| JPEG | 45.35 dB | 96.29% | | 47.27% |

the weighting factor $\lambda$ in (11) to balance the effects caused by both image quality and robustness. By doing so, we are able to take both the conflicting factors into account by optimizing with the GA.

## 7 Simulation Results

In our simulations, we take the well-known test image, Lena, with size $256 \times 256$, as the original source. We have the embedded watermark with size $32 \times 32$, hence, the watermark capacity is 1024 bits. The original source is divided into $8 \times 8$ blocks for wavelet transform. We employ the BCR between the extracted watermark and the embedded one, in addition to the PSNR of the received, watermarked image, for evaluating the effectiveness of our algorithm. Three sets of experiments are conducted to test the robustness under LPF, MF, and JPEG attacks. We also make comparisons to show the superiority and usefulness of the GA-optimized results with those in [8].

Simulation results with the existing scheme and our schemes are presented in Figs. 3, 4, and Table 1 after training 150 iterations in the GA. We set the weighting factor $\lambda = 30$ in (11) to balance the effects from the watermarked PSNR and extracted BCR under three different attacks. After optimization, regarding watermark imperceptibility objectively, our results have higher PSNR values. To subjectively make comparisons with watermarked image quality and watermark imperceptibility, Fig. 3(a) shows the original image with size $256 \times 256$, while Fig. 3(b)–(e) demonstrate the watermarked ones. We find that the watermark is imperceptibly embedded into the original image.

In addition, to measure the watermark robustness objectively, we obtain much higher BCR values in the extracted watermark with our schemes. From a subjective point of view, the characters in the watermark, **KUAS**, are easily recognizable in Fig. 4(b), (d), and (f), hence, the copyright can be protected. On the contrary, without GA training, the extracted watermarks in Fig. 4(c), (e), and (g) are incomprehensible.

To sum up, results with GA training outperform those without optimization. This verifies the effectiveness of the proposed algorithm with the GA.

## 8 Conclusion

We proposed an optimized scheme for zerotree-based image watermarking in this paper. Our work contributes to this by pioneering the idea of genetic selection of

(a) Original

(b) Optimized with LPF attack

(c) Optimized with MF attack

(d) Optimized with JPEG attack

(e) With schemes in [8]

**Fig. 3** Comparisons among the original and watermarked images. (**a**) Original Lena. (**b**) Watermarked image for optimizing with LPF attack. PSNR = 45.13 dB. (**c**) Watermarked image for optimizing with MF attack. PSNR = 44.95 dB. (**d**) Watermarked image for optimizing with JPEG attack. PSNR = 45.35 dB. (**e**) Watermarked image with schemes in [8]. PSNR = 44.56 dB

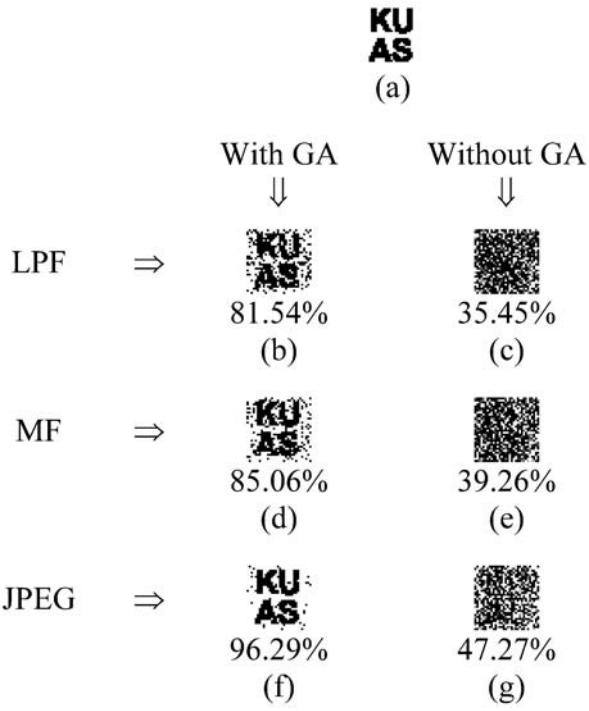


**Fig. 4** The embedded watermark (**a**), and extracted ones (**b**)–(**g**). The numbers denote the corresponding BCR values. (**a**) $32 \times 32$ binary watermark for embedding. (**b**), (**c**) Extracted watermark under LPF attack with/without GA. (**d**), (**e**) Extracted watermark under MF attack with/without GA. (**f**), (**g**) Extracted watermark under JPEG attack with/without GA

niches for watermark embedding. With the GA, we can search for both optimized, watermarked image quality, and better robustness of the proposed algorithm. Therefore, applying optimization techniques into watermarking algorithms is practical and effective in designing and implementing watermarking systems and applications. Finally, the idea proposed in this paper can be a general philosophy, not limited to zerotree-based watermarking, but also applicable to other architectures of digital watermarking.

**Acknowledgement** This work was supported by the National Science Council (Taiwan, ROC) under grant no. NSC 95-2221-E-390-034.

## References

1. Chang, F.-C., Huang, H.-C., Hang, H.-M.: Layered access control schemes on watermarked scalable media. *J. VLSI Signal Process. Syst. Signal Image Video Technol.* 49(3), 443–455 (2007)
2. De Vleeschouwer, C., Delaigle, J.-F., Macq, B.: Invisibility and application functionalities in perceptual watermarking—an overview. *Proc. IEEE* 90(1), 64–77 (2002)
3. Gen, M., Cheng, R.: *Genetic algorithms and engineering design*. Wiley, New York (1997)
4. Gonzalez, R.C., Woods, R.E.: *Digital image processing*. Addison-Wesley, Reading (1992)
5. Hsieh, M.S., Tseng, D.C.: Image subband coding using fuzzy inference and adaptive quantization. *IEEE Trans. Syst. Man Cybern. B* 33(3), 509–513 (2003)
6. Hsu, C.-T., Wu, J.-L.: Hidden digital watermarks in images. *IEEE Trans. Image Process.* 8(1), 58–68 (1999)
7. Huang, H.-C., Wang, F.H., Pan, J.S.: A VQ-based robust multi-watermarking algorithm. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* E-85A(7), 1719–1726 (2002)

8. Inoue, H., Miyazaki, A., Yamamoto, A., Katsura, T.: A digital watermark technique based on the wavelet transform and its robustness on image compression and transformation. *IEICE Trans. Fundam.* 82-A(1), 2–10 (1999)
9. Pan, J.S., Hsin, Y.C., Huang, H.-C., Huang, K.C.: Robust image watermarking based on multiple description vector quantisation. *Electron. Lett.* 40(22), 1409–1410 (2004)
10. Pan, J.S., Huang, H.-C., Jain, L.C. (editors): *Intelligent watermarking techniques*. World Scientific, Singapore (2004)
11. Pan, J.S., Huang, H.-C., Jain, L.C., Fang, W.C. (editors): *Intelligent multimedia data hiding: new directions*. Springer, Berlin (2007)
12. Pan, J.S., Sung, M.T., Huang, H.-C., Liao, B.Y.: Robust VQ-based digital watermarking for the memoryless binary symmetric channel. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* E-87A(7), 1839–1841 (2004)
13. Pennebaker, W.B., Mitchell, J.L.: *JPEG: still image data compression standard*. Van Nostrand Reinhold, New York (1993)
14. Petitcolas, F.A.P.: Image watermarking—Stirmark, http://www.petitcolas.net/fabien/watermarking/stirmark/ (2004)
15. Radha, H.M., van der Schaar, M., Chen, Y.: The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP. *IEEE Trans. Multimedia* 3(1), 53–68 (2001)
16. Shieh, C.S., Huang, H.-C., Wang, F.H., Pan, J.S.: An embedding algorithm for multiple watermarks. *J. Inf. Sci. Eng.* 19(2), 381–395 (2003)
17. Shieh, C.S., Huang, H.-C., Wang, F.H., Pan, J.S.: Genetic watermarking based on transform domain techniques. *Pattern Recognit.* 37(3), 555–565 (2004)
18. Taubman, D., Marcellin, M.: *JPEG2000: image compression fundamentals, standards and practice*. Kluwer Academic, Boston (2001)