An Improved Digital Watermarking Scheme in the Domain of Vector Quantization

Chin-Shiuh Shieh[1, 2,*], Chao-Chin Chang[2], Wei Kuang Lai[1], and Shu-Chuan Chu[3]

[1] Department of Computer Science and Engineering, National Sun Yat-sen University
Kaohsiung 804, Taiwan, ROC
csshieh@cc.kuas.edu.tw

[2] Department of Electronic Engineering, National Kaohsiung University of Applied Sciences
Kaohsiung 807, Taiwan, ROC

[3] Department of Information Management, Cheng Shiu University
Kaohsiung County 833, Taiwan, ROC
scchu@csu.edu.tw

Received 20 March 2006; Revised 18 May 2006 ; Accepted 10 June 2006

Abstract. Digital watermarking had been well recognized as an effective measure for the copyright protection of multimedia data. Numerous schemes in the domain of vector quantization had been proposed to achieve the desired goal. A new scheme, also in the domain of vector quantization, is developed and presented in this article. The proposed approach use genetic algorithms to reassign the indices of code words. As a result, embedded information will be diffused more evenly across the image to be protected, and therefore possible security leakage can be avoided. Experimental results reveal that the proposed scheme is free from the potential limitations in previous approaches, while maintaining the robustness against various kinds of attacks.

Keywords: digital watermarking, genetic algorithm, index assignment, information leakage

1 Introduction

Data secrecy had become an important issue as communication networks getting commoditization and widely spread, especially with the blooming of the Internet. Among others technologies, watermarking technology has received considerable attention in recent years for their theoretical and practical significance. Aimed at copyright protection, arbitration, and authentication, watermarking is the process of embedding extra information into a media clip. There have been a vast number of established methods [1]-[3]. However, it is still far from trivial to make the embedded watermark robust. Various criteria are addressed in judging a watermarking technique, such as perceptibility, security, embedding rate, whether original clip is required for extraction, robustness to common signal processing or intentional attacks, and so on.

According to the hiding domain, digital watermarking can be roughly classified into spatial-domain based methods [4] and transform-domain based methods. Transform-domain approaches have been intensively studied, such as discrete cosine transform (DCT) [5], discrete Fourier transform (DFT) [6], discrete wavelet transform (DWT) [7], and the Chirp-Z transform [8].

To reduce the space requirement for storage and the bandwidth requirement for communication, wide variety of compression techniques had been developed [9]. In applications regarding image, audio, and video, human sensory system is sophisticated enough to filter out limited data loose during the process of encoding and decoding. In this kind of applications, vector quantization (VQ) [10]-[11] had received considerable attention for its high compression rate and its essential role in various compression applications. VQ and its descendents try to maintain high compression rate while retaining essential information carried in media clips. As an extension to scalar quantization, vector quantization works on vectors of raw data. A vector can be fixed number of consecutive samples of audio data or a small block of image/video data, for example, the gray-level values of a 4×4 pixel image block forms a 16-dimentional vector. Fig. 1 gives an illustration of the operation of vector quantization compression.

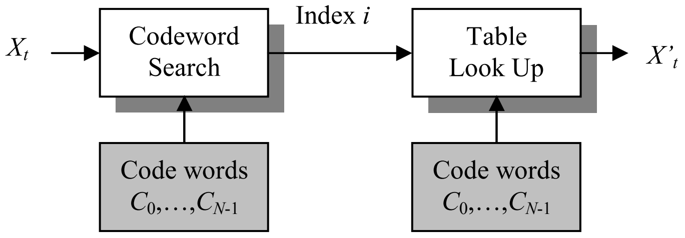
[*] Correspondence author

In the sender (compression) end, the codeword search process looks for a “nearest” codeword, C_i , from the codebook for the given input vector X_t. Euclidean distance was used in the search process to measure the distance between two vectors as indicated in Equation (1).

$$i = \arg\min_j D(X_t, C_j), j = 0,..., N-1 \ . \qquad \textbf{(1)}$$

where $D(V_1,V_2) = \sqrt{\sum_{k=1}^{K}(V_1^k - V_2^k)^2}$ is the Euclidean distance between two *K*-dimensional vectors V_1 and V_2, and V^k is the *k*-th component of vector *V*.

The index of selected codeword is then transmitted to the receiver end. With the same codebook, the decompression process can easily reconstruct vector $X_t^{'}$ by simple table look-up. Of course, there will be distortion introduced by the compression-decompression process, since $X_t^{'}$ is only an approximated version of the original X_t. If we work on 8-bit gray-level image, using a block size of 4×4 pixel and a codebook of 256 code words, then the compression ratio is up to $\frac{4\times 4\times 8}{\log_2 256} = 16$.



X_t
Codeword Search
Index *i*
Table Look Up
X'_t
Code words $C_0,\ldots,C_{N\text{-}1}$
Code words $C_0,\ldots,C_{N\text{-}1}$

Fig. 1. A block diagram for vector quantization

The codebook plays an essential role in vector quantization. The codebook size, i.e. the number of code words in a codebook, is a tradeoff between compression quality and compression rate. The code words in the codebook decide the resultant compression distortion. A dedicated procedure is required for the generation of appropriate codebook. One may regards the problem of codebook generation as a problem of finding *N* most representative vectors, $C_i, i = 0,..., N-1$, from *M* given training vectors, $X_j, j = 0,..., M-1$. The located C_i’s serve as code words used to partition the *M* given data into *N* mutual exclusive clusters, $S_i, i = 0,..., N-1$. A given vector X_j is considered to belong to cluster S_i if C_i is the nearest codeword to X_j. Among other alternatives, LBG algorithm [12] is widely used in various applications. The following pseudo code illustrates the operation of LBG algorithm:

```
Pseudo Code for LBG Algorithm
Randomly pick up N vectors from M training vectors as initial code words
DO
  Conduct clustering using the current code words
  Calculate new centers $C_i$ for each cluster $S_i$
UNTIL Improvement falls bellow threshold $\varepsilon$
```

Vector quantization had been distinguished for its high compression rate in lossy data compression applications. To be of practical significance, a digital watermarking technique should take into account the effect of vector quantization compression. Recently, several researchers pay much attention to explore the hiding scheme based on vector quantization [13]-[15]. In Section 2, we will review these representative works, and point out their relative merits and potential limitations. We will discuss the details of using genetic algorithms for index assignment in Section 3. Such an approach will reduce the possibility of information leakage. An improved scheme based on the results of Sections 3 is presented in Section 4. Experimental results are given in Section 5, followed by some conclusions in Section 6.

2 Related Works

Huang *et al.* [13] had developed a robust digital watermarking algorithm in the domain of vector quantization, from which various variants were developed. To facilitate our discussion, a brief review of their work is given here. In [13], the image $\mathbf{X}$ to be protected is an 8-bit gray-level image of size $M \times N$, and the watermark $\mathbf{W}$ is a binary image of size $M_W \times N_W$. The $\mathbf{X}$ is discomposed into non-overlapping blocks of size $M/M_w \times N/N_w$. $x(m,n)$ is used to denote the block located at (m,n). For a given code book $\mathbf{C}$, the set of indices $\mathbf{Y}$ can be obtained by regular search.

$$\mathbf{Y} = VQ(\mathbf{X}) = \{VQ(x(m,n))\} = \{y(m,n)\} \ . \tag{2}$$

With $y(m,n)$, $\sigma^2(m,n)$ is defined to be the variance of index values of those 3×3 blocks centered around (m,n). The polarity set $\mathbf{P} = \{p(m,n)\}$ can then be evaluated according to

$$p(m,n) = \begin{cases} 1, \text{if } \sigma^2(m,n) \geq T \\ 0, \text{otherwise} \end{cases} . \tag{3}$$

where threshold T is set to half of the code book size.

Two secret keys, namely Key_1 and Key_2 were used in the algorithm. Key_1 served to be the random seed for a random sequence, which is used to disturb $\mathbf{W}$ and result in $\mathbf{W_p}$. During the embedding phase, $\mathbf{Key_2}$ was obtained by applying an exclusive-OR operation on $\mathbf{W_p}$ and $\mathbf{P}$. For watermark extraction, polarity set $\mathbf{P}'$ can be evaluated according to the reconstructed $\mathbf{X}'$. $\mathbf{W_p'}$ can then be obtained with the help of $\mathbf{Key_2}$. Finally, $\mathbf{W}'$ can be arrived at by an inverse disturbing based on Key_1.

The algorithm in [13] had successfully achieved the desired robustness. However, there are some potential problems with it. Lu *et al.* [14] had pointed out two possible negative effects associated with the algorithm in [13]. Firstly, it is possible to extract the watermark from a non-watermarked image. Secondly, the code book $\mathbf{C}$ must be kept secret. Otherwise, third parties can use it to embed their own watermarks. Moreover, Charalampidis [15] also discovered an abnormal phenomenon with the algorithm in [13]. With all keys available, the watermark $\mathbf{W_f}$ extracted from a non-watermarked image $\mathbf{X_n}$ might closely resemble the true watermark $\mathbf{W}$. This phenomenon will certainly cause a major issue in copyright arbitration.

Two countermeasures were employed in Charalampidis [15] as remedies to the above problems. One of them is redefining the threshold values as follows:

$$T_m = medium\{\sigma^2(m,n)\} \ . \tag{4}$$

T_m also serves to be the third secret key Key_3. The other countermeasure is to reassign indices to the code words in order to smooth out their statistic characteristics. The new indices are assigned according to the following pattern:

$$I_{i,k} = \begin{cases} 0, & \text{if } c_{i,k} < \left(\max_i(c_{i,k}) + \min_i(c_{i,k})\right)/2 \\ 1, & \text{otherwise} \end{cases} . \tag{5}$$

where $I_{i,k}, k = 1, \cdots, k$ is the *k*-th bit in *i*-th index value; $c_{i,k}$ is the *k*-th element of *i*-th code word.

After that, a procedure was carried out to reduce the number of bits required to encode the new indices. In idea case, it should take $\log_2(L)$ bits for a code book of size L.

Variance $\sigma^2_{av}(m,n)$ was then defined, based on the new indices, according to:

$$\sigma^2_{av}(m,n) = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{9}\sum_{i=m-1}^{m+1}\sum_{j=n-1}^{n+1} y^2(i,j)_k - \left(\frac{1}{9}\sum_{i=m-1}^{m+1}\sum_{j=n-1}^{n+1} y(i,j)_k\right)^2\right) . \tag{6}$$

Following the same embedding and extraction procedures, with the newly defined T_m and $\sigma^2_{av}(m,n)$, the work of Charalampidis [6] did resolve the problem it intended. However, according to our study, the code length reducing procedure could be problematic in general cases. The procedure in [15] might terminate with a coding length larger than $\log_2 L$, which violates the original appeal of vector quantization. This observation motivated the development of our scheme on index assignment using genetic algorithms. The proposed scheme is capable of retaining the virtue of vector quantization without scarifying the robustness against common attacks.

3 Index Assignment by Genetic Algorithms

Inspired by Darwin's theory of evolution, J. Holland [16] introduced the genetic algorithm [17] as a powerful computational model for optimization. Genetic algorithms work on a population of potential solutions, in the form of chromosomes, and try to locate a best solution through the process of artificial evolution, which consists of repeated artificial genetic operations, namely evaluation, selection, crossover, and mutation. Genetic algorithms maintain a population of candidate solutions and conduct stochastic search via information selection and exchange. With genetic algorithms, near-optimal solutions can be obtained within justified computation cost. A pseudo code for the operation of genetic algorithms is given bellows:

```
Pseudo Code for Genetic Algorithms
Randomize initialize population $P(0)$
WHILE terminate condition is not met
  Evaluate $P(t)$ using object function
  Select $P(t+1)$ from $P(t)$ based on fitness value
  Conduct genetic operators, crossover and mutation, on $P(t+1)$
```

Although the operation of genetic algorithms is quite simple, it does have some important characteristics providing robustness: 1. Search from a population of points rather than a single point. 2. Use object function directly, not their derivative. 3. Use probabilistic transition rule, not deterministic one, to guide the search toward promising region. In effect, genetic algorithms maintain a population of candidate solutions and conduct stochastic search via information selection and exchange. It is well recognized that, with genetic algorithms, near-optimal solutions can be obtained within justified computation cost. However, it is difficult for genetic algorithm to pin point the global optimum. In practice, hybrid approach is recommended by incorporating gradient-based or local greedy optimization techniques. In such integration, genetic algorithms act as course-grain optimizers and gradient-based method as fine-grain optimizers.

Genetic algorithm is used in our scheme to find out a suitable index assignment such that nearby indices are assigned to similar code words. To encourage the formation of chromosomes fulfilling these criteria, the object function is defined as follows:

$$f = \sum_{i=0}^{L-1} \sum_{j=i-s}^{i+s} D(c_i, c_j) \,. \tag{7}$$

where $D(c_i, c_j)$ is the Euclidian distance between code words c_i and c_j; s is the range of comparison.

The first design issue in applying genetic algorithms is to select an adequate coding scheme to represent potential solutions in the form of chromosomes. For our problem, each chromosome is a permutation of indices representing a possible index assignment. In the initial population, all chromosomes are randomly set. During the course of evolution, each chromosome is subject to fitness evaluation according to Equation (7). Fitness values are then used in the process of roulette wheel selection to generate next population. The crossover operator plays an essential role in combing good genes. A modified version of the crossover operator, as illustrated in Fig. 2, is developed to ensure the legality of newly generated chromosomes. Mutation operator acts as a background operator against the premature phenomenon. In our work, mutation is done by swapping the indices of two randomly selected code words in a chromosome.

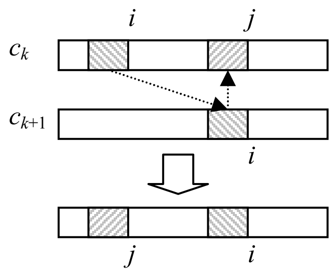


Fig. 2. Modified crossover operator

4 Watermark Embedding and Extraction

As the genetic algorithm terminates, we will arrive at a new index assignment $\mathbf{I}' = \{i'_0, i'_1, \cdots, i'_{L-1}\}$, with which a new code book $\mathbf{C}' = \{c'_0, c'_2, \cdots, c'_{L-1}\}$ can be constructed. Therefore, $\mathbf{I}'$ can be regarded as a third secret key $\mathbf{Key}_3$. The new code book $\mathbf{C}'$ will be used for internal operation during the processes of watermark embedding and extraction. With the help of the newly generated code book, statistical characteristics of the image to be protected will be diffused evenly across the entire image. As a result, the risk of possible information leakage can be effectively suppressed. Fig. 3 illustrates the process diagram for watermark embedding.

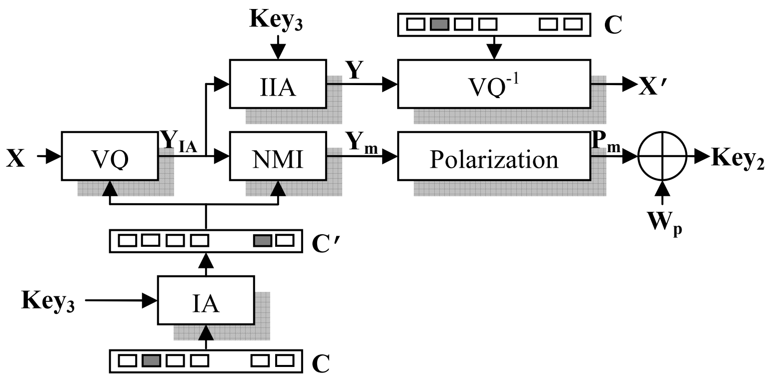


Fig. 3. Process diagram of watermark embedding

In our scheme, the vector quantization is conducted using the new code book $\mathbf{C}'$. We denote the index set for the image to be protected as $\mathbf{Y_{IA}} = \{y_{IA}(m,n)\}$. The mean index value is defined, based on $\mathbf{Y_{IA}}$, as follows:

$$\mathbf{Y_m} = \left\{ y_m(m,n) = \frac{1}{K} \sum_{k=1}^{K} NI_k\left(c_{y_{IA}(m,n)}\right) \right\} . \quad \textbf{(8)}$$

where $NI_k(c)$ is the index of the k-th nearest code word to c, and K is a parameter controlling the size of neighborhood.

We can then evaluate the variance of mean index value $\mathbf{\Sigma_m} = \{\sigma_m^2(m,n)\}$ according to:

$$\sigma_m^2(m,n) = \left(\frac{1}{9} \sum_{i=m-1}^{m+1} \sum_{j=n-1}^{n+1} y_m^2(i,j) - \left(\frac{1}{9} \sum_{i=m-1}^{m+1} \sum_{j=n-1}^{n+1} y_m(i,j) \right)^2 \right) . \quad \textbf{(9)}$$

A new threshold T_{mean}, also serves as a fourth key Key_4, is defined to be:

$$T_{mean} = \left\lfloor \frac{\sum_{m=0}^{M/W_M - 1} \sum_{n=0}^{N/W_N - 1} \sigma_m^2(m,n)}{M/W_M \times N/W_N} + 0.5 \right\rfloor . \quad \textbf{(10)}$$

Finally, we arrive at the following polarity set $\mathbf{P_m} = \{p_m(m,n)\}$:

$$p_m(m,n) = \begin{cases} 1 & \text{if } \sigma_m^2(m,n) \ge T_{mean} \\ 0 & \text{otherwise} \end{cases} . \quad \textbf{(11)}$$

For watermark embedding, $\mathbf{Key}_2$ an be obtained by applying an exclusive-OR operation on $\mathbf{W_p}$ and $\mathbf{P_m}$, as follows:

$$\mathbf{Key}_2 = \mathbf{W_p} \oplus \mathbf{P_m} . \quad \textbf{(12)}$$

The code book $\mathbf{C}'$ is designed for the internal operation of the proposed scheme, and should be kept secret. For this reason, the index set $\mathbf{Y}'$ should be mapped into $\mathbf{Y}$ before released to the publics.

$$\mathbf{Y} = \{y(m,n) = i'_{Y_{IA}(m,n)}\} \ . \qquad \textbf{(13)}$$

The process diagram for watermark extraction is given in Fig. 4. For watermark extraction from a given image $\mathbf{X}''$, $\mathbf{Y}'_{\mathbf{m}}$ can be obtained from $\mathbf{Y}'_{\mathbf{IA}}$ with the help of $\mathbf{Key}_3$. We then have $\mathbf{P}'_{\mathbf{m}}$ by incorporating Key_4. $\mathbf{W}'_{\mathbf{p}}$ can be evaluated by applying an exclusive-OR operation on $\mathbf{P}'_{\mathbf{m}}$ and $\mathbf{Key}_2$.

$$\mathbf{W}'_{\mathbf{p}} = \mathbf{P}'_{\mathbf{m}} \oplus \mathbf{Key}_2 \ . \qquad \textbf{(14)}$$

Finally, $\mathbf{W}'$ can be arrived at by an inverse disturbing based on Key_1.

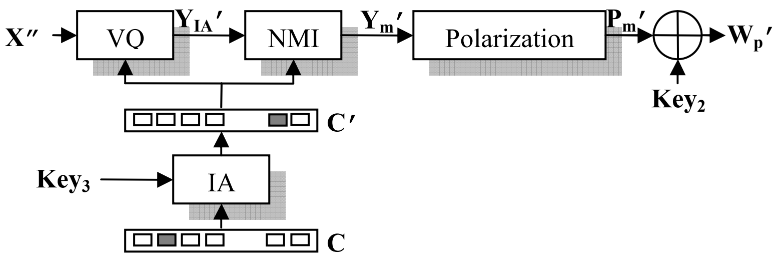


Fig. 4. Process diagram of watermark extraction

5 Experimental Results

A series of experiments was made to verify the feasibility and effectiveness of our approach. Key experiment setting is given below:

- Test image: $512 * 512$ 8-bit, gray-level Lena
- Watermark: 128×128 binary Rose
- Generation limit of GA: 10,000 generations
- Population size of GA: 20 chromosomes
- Crossover rate of GA: 70%
- Mutation rate of GA: 0.1%
- Range of comparison s: 4
- Size of neighborhood K: 6

We also implemented the algorithm by Huang *et al.* [13] for comparative study. For possible information leakage, experiments were made to extract watermark from non-watermarked images, including Baboon, Goldhill, and Peppers. The result is given in Fig. 3. It is clear that the potential problem with Huang *et al.* [13] ceases to exist with our scheme. Fig. 4 reports the watermarks extracted using our scheme under various kinds of attacks. Table 1 summarizes the bit correct rate for the two approaches under consideration. We can see that there is comparable performance between them.

6 Conclusions

A new scheme, in the domain of vector quantization, is developed and presented in this article. The proposed approach use genetic algorithms to reassign the indices of code words. As a result, embedded information will be diffused more evenly across the image to be protected, and then possible security leakage can be avoided. Experimental results reveal that the proposed scheme is free from the potential limitations in previous approaches, while maintaining the robustness against various kinds of attacks.

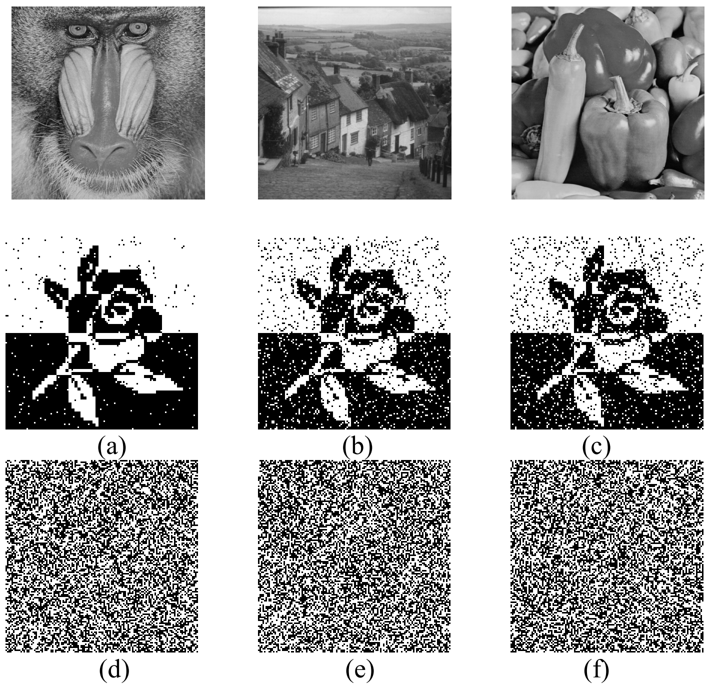


Fig. 5. Watermarks extracted from non-watermarked images. (a)-(c): Huang *et al*. [4]. (d)-(f): our scheme.

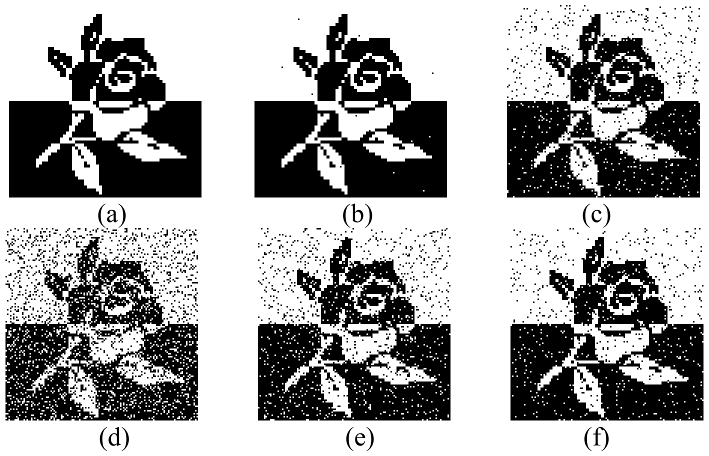


Fig. 6. Watermarks extracted under various kinds of attacks

Table 1. Comparison on bit correct rate

	Huang *et al*. [4]	Ours
(a) No Attack	1.0	1.0
(b) JPEG 90%	0.9998	0.9993
(c) JPEG 60%	0.9913	0.9559
(d) Crop 25%	0.7600	0.8437
(e) LPF	0.9984	0.9294
(f) MPF	0.9982	0.9581

7 Acknowledgement

This work is supported by the National Science Council, Taiwan, R.O.C., under the grant No. NSC 94-2213-E-151 -013.

References

[1] R. Barnett, “Digital watermarking applications, techniques, and challenges,” *IEE Electronics and Communication Engineering Journal*, Vol. 11 1999, pp. 173-183.

[2] S. Katzenbeisser and F. Petitcolas, *Information Hiding – Techniques for Stenography and Digital Watermarking*, Artech House, 2000.

[3] J.-S. Pan, H.-C. Huang, and L.C. Jain, (Eds.), *Intelligent Watermarking Techniques*, World Scientific Publishing Company, Singapore, 2004.

[4] R.G. Van Schyndel, A.Z. Tirkel, N. Mee, and C.F. Osborne, “A digital watermark,” *Proceedings of the IEEE International Conference on Image Processing*, Vol. 2, 1994, pp. 86-90.

[5] I. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, “Secure spread spectrum watermarking for multimedia,” *IEEE Transactions on Image Processing*, Vol. 6, 1997, pp. 1673-1687.

[6] J.J.K. O’Ruanaidh, W.J. Dowling, and F.M. Boland, “Phase watermarking of digital images,” *Proceedings of the IEEE International Conference on Image Processing*, Vol. 3, 1996, pp. 239-242.

[7] J.-S. Pan, S.-Y. Wu, C.-S. Shieh, and Y. Shi, “Genetic zero-tree selection for robust watermarking system,” *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics*, 2003, pp. 177-179.

[8] S. Pereira and T. Pun, “An iterative template matching al-gorithm using the Chirp-Z transform for digital image watermarking,” *Pattern Recognition*, Vol. 33, 2000, pp. 173-175.

[9] K. Sayood, *Introduction to Data Compression*, 2nd Ed., Morgan Kaufmann, 2000.

[10] R.M. Gray, “Vector quantization,” *IEEE ASSP Magazine*, 1984, pp. 4-29.

[11] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.

[12] Y. Linde, A. Buzo, and R.M. Gray, R.M., “An algorithm for vector quantizer design,” *IEEE Transactions on Communication*, Vol. 28, 1980, pp. 84-95.

[13] H.-C. Huang, F.-H. Wang, and J.-S. Pan, “Efficient and robust watermarking algorithm with vector quantisation,” *Electronic Letters*, Vol. 37, No. 13, 2001，pp. 826-828,.

[14] Z.M. Lu, D.G. Xu, and S.H. Sun, “Multipurpose image watermarking algorithm based on multistage vector quantization”, *IEEE Transactions on Image Processing*, Vol. 14, No. 6, 2005, pp. 822-831.

[15] D. Charalampidis, “Improved robust VQ-based watermarking,” *Electronics Letters*, Vol. 41, No. 23, 2005, pp. 1272-1273.

[16] J. Holland, *Adaptation In Natural and Artificial Systems*, University of Michigan Press, 1975.

[17] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA: Addison-Wesley, 1987.