Hiding Data in VQ-compressed Images Using Dissimilar Pairs

Chih-Yang Lin[1,*] and Chin-Chen Chang[2]

[1] Department of Computer Science and Information Engineering, National Chung Cheng University
Chiayi 621, Taiwan, ROC
gary@cs.ccu.edu.tw

[2] Department of Information Engineering and Computer Science, Feng Chia University
Taichung 40724, Taiwan, ROC
ccc@cs.ccu.edu.tw

Received 5 May 2006; Revised 10 June 2006 ; Accepted 18 June 2006

Abstract. Steganography is the art and science of embedding secret data in another medium to prevent the leakage of secret information. A VQ-based (vector quantization) steganographic method usually involves changes of the block values in the VQ images, which might cause serious distortion. As a result, many existing methods use closest pairs or clustering techniques to preserve an acceptable image quality. In this paper, we propose a new VQ steganographic method for preserving extremely high image quality. Instead of finding similar pairs, the proposed method uses a pair of dissimilar codewords to embed one secret bit in each image block. The property of dissimilar pairs allows the VQ image to be nearly fully recovered after extraction of the secret bits. Experimental results show that our method keeps an acceptable embedding capacity but has much better image quality than existing schemes.

Keywords: steganography, data hiding, declustering

1 Introduction

The ever-increasing digitalization of all kinds of data and the popularity of the Internet increase the importance of information security. Some information transmitted over the Internet may be considered confidential data and thus must be kept or transmitted securely from the sender to the receiver. One technique that solves this problem is called steganography [14], which embeds secret data in a cover carrier imperceptibly and prevents a hostile interceptor from discovering the concealed data. In general, the cover carrier can be any kind of digital data such as an image, text, audio, video, and so forth. In this paper, we discuss a steganographic method using digital images as the cover carrier to conceal the secret data. For a clear description, the *cover image* is referred to as the original image that is used to embed the secret message, and the *stego image* is a version of the cover image that has been modified to contain the secret message.

A simplest steganographic approach for digital images is the least-significant bit (LSB) insertion method [1, 15], which embeds secret data in the least-significant bits of the stego image. Because the LSB method is quite simple and changes values only slightly as a result of small distortions, a number of variants of the LSB method have been developed [2, 3, 16].

However, the LSB method cannot be applied directly to VQ-compressed images. VQ (vector quantization) is a lossy compression method [5, 6] based on the principle of block coding. It is a popular compression method due to its simplicity and low compression bit rate. The main concept of VQ is to use a codebook with codewords in it to represent an image. The VQ compression process starts with partitioning an image into nonoverlapping blocks, and then it maps each block to the closest codeword in the codebook. These blocks are finally represented by the indices of the codewords. If we apply the LSB method directly to a VQ-compressed image, the quality of the stego image may become worse since the distortion caused by replacing two adjacent codewords in a codebook may be large.

In recent years, several methods have been proposed to hide secret data in the VQ-compressed images. Lin and Wang [11] proposed a VQ steganographic method that partitions one codebook into two subcodebooks of equal size, so that each codeword in one subcodebook has a corresponding codeword in the other subcodebook, and the two together form a pair. To reduce the distortions in codeword replacements during embedding of secret data, the subcodebooks are rearranged by the *p*airwise *n*earest *c*lustering *e*mbedding (PNCE) method [11] so that

* Correspondence author

all pairs of codewords between the subcodebooks are as similar as possible. The two subcodebooks are responsible for indicating whether a secret bit “0” or “1” is embedded. However, the PNCE in Lin and Wang’s method cannot guarantee that each codeword finds its closest codeword, so the quality of the stego image can be degraded.

To resolve this problem, Jo and Kim [9] partitioned the codebook into three subcodebooks, where one subcodebook collects the singular codewords, and the others serve the same function as in Lin and Wang’s method. A singular codeword means that this codeword has no corresponding similar codeword under a predefined distance threshold. Because these singular codewords do not represent the secret bits, the quality of the stego image can be improved but the embedding capacity is decreased. The embedding capacity of Lin and Wang’s method can be easily improved by partitioning the codebooks into more than two subcodebooks without considering the singular codewords [13]. If the codebook is partitioned into 2^t subcodebooks, the embedding capacity of each subimage block can be increased from one secret bit to t ones. However, this method is still confronted with deterioration in quality of the stego image.

In this paper, we propose a steganographic method for VQ-compressed images based on the principle of declustering. Declustering is the opposite of clustering, in that the aim is to put dissimilar codewords together. During embedding, each block can embed one secret bit and is classified into one of the following three types: changeable, pseudo-changeable, and unchangeable. Owing to the declustering property, the changeable and unchangeable blocks can be completely recovered, and the pseudo-changeable blocks can be approximately recovered after extraction of the secret bits. Therefore, the proposed method can achieve much better image quality than the previous methods.

The remainder of this paper is organized as follows. First, some related works of VQ and Lin and Wang’s hiding scheme are presented in Section 2. Then, in Section 3, we present our new embedding scheme for VQ images. Empirical results are listed in Section 4. Finally, the conclusions are appeared in Section 5.

2 Related Works

2.1 Vector Quantization

VQ [6, 12] is a well-known lossy compression method especially designed for digital images due to its simple yet efficient encoding and decoding procedures. Fig. 1 shows the VQ encoding and decoding processes.

Before encoding a grayscale image, the image is first partitioned into nonoverlapping blocks of $r \times l$ pixels, so each block can be represented by an $r \times l$-dimensional vector. The basic function of VQ is to map each block using a mapping function Q from $r \times l$-dimensional Euclidean space $R^{r\times l}$ to a finite subset Ψ of $R^{r\times l}$; that is, Q: $R^{r\times l} \rightarrow \Psi$, where $\Psi = \{Y_1, Y_2, ..., Y_n\}$ is called the codebook and Y_i is the i-th codeword in Ψ.

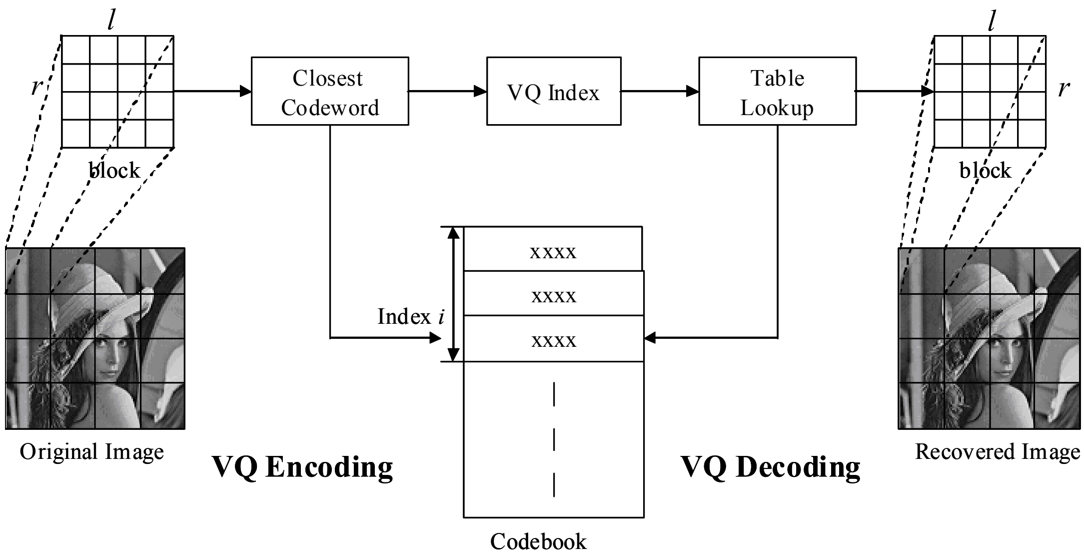


Fig. 1. VQ encoding and decoding processes

During encoding, the closest codeword in the codebook is found for each vector $X \in R^{r\times l}$ of the original image. The distance between X and a codeword Y_i, $i = 1, 2, \ldots, n$, is determined by the Euclidean distance, $d(X, Y_i)$:

$$d(X,Y_i)=\|X-Y_i\|=\left[\sum_{j=0}^{r\times l-1}\left(x_j-y_{i,j}\right)^2\right]^{1/2}. \quad \textbf{(1)}$$

where x_j and $y_{i,j}$ are the j-th elements of vectors X and Y_i, respectively. When the closest codeword Y_i of X is found, index i is used to encode vector X, with the original image eventually represented by the indices of these closest codewords.

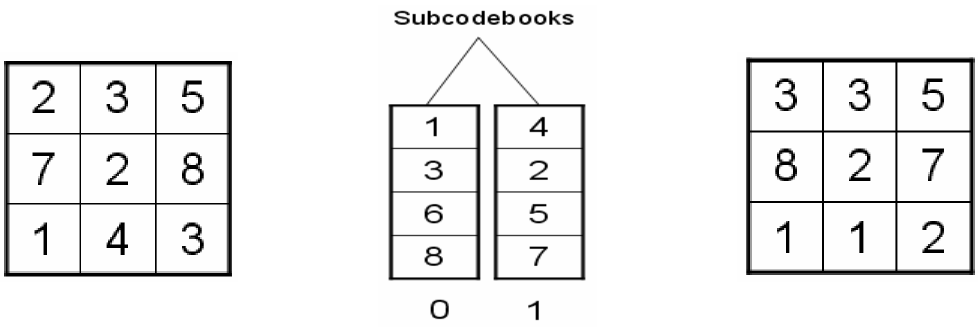
In the decoding phase, only table lookup operations are required to reconstruct the original image. The VQ decoder requires a codebook that is the same as the VQ encoder. According to the indices compressed by the VQ encoder, the VQ decoder fetches the corresponding codewords to reconstruct the image. Therefore, the VQ-compressed image quality is significantly influenced by the quality of the codebook, which can be well-designed by [7, 12].

2.2 Lin and Wang's Embedding Method

The embedding method for VQ images proposed by Lin and Wang [11] uses an LSB-like approach, where one secret bit is embedded in one image block. They partitioned a codebook into two subcodebooks of equal size that are rearranged by a pairwise nearest clustering embedding (PNCE) method [11], and each pair of codewords between the subcodebooks is as similar as possible. The PNCE is a greedy method that selects the closest pair first each time. The results of PNCE method may not be optimal. For instance, the PNCE result is shown in Fig. 2(a), but the optimal result is shown in Fig. 2(b).

(a) PNCE result (b) Optimal result

Fig. 2. An example of PNCE method



(a) Original index table (b) Pairing result (c) Embedding result

Fig. 3. An example of Lin and Wang's method

After partitioning the codebook, one of the two subcodebooks is responsible for indicating secret "0" bits and the other subcodebook is responsible for indicating secret "1" bits. Since each pair of codewords is similar to each other, the replacement with the alternative codeword in the same pair to indicate the secret bit does not cause serious distortion. A simple example of Lin and Wang's method is given as follows. Assume the secret bits are 001011001, the original VQ index table is shown in Fig. 3(a), and the partition result of the codebook is shown in Fig. 3(b). After embedding the secret bits, the final result is shown in Fig. 3(c). The modified index values in Lin and Wang's method cannot be recovered after extraction, so the pairing result significantly affects the image quality.

3 Proposed Method

The following subsections present the detailed reversible embedding scheme for VQ-compressed images.

3.1 Sorting the Codebook

Before embedding, codebook C is sorted by *p*rincipal *c*omponents *a*nalysis (PCA) [4, 8, 10]. The PCA method is widely used in data and signal analysis, such as multimedia coding and recognition. The power of PCA is that it can project a higher-dimensional input vector onto a lower-dimensional space while still preserving the maximal variances of the input vectors on the new coordinate axes. In our scheme, each codeword with sixteen dimensions is projected onto a one-dimensional space, the first principal component, according to which the codewords are sorted. The details of PCA algorithm to sort k codewords are presented as follows.

PCA Algorithm.
Input: A set of n-dimensional vectors ($cw_1, cw_2, \ldots, cw_k$).
Output: The first principal component of each input vector.
Step 1: Compute the mean vector of the input vectors m and normalize the vectors to be zero mean:
$cw_i \leftarrow cw_i - m$.
Step 2: Compute the covariance matrix Cov of the normalized vectors.
Step 3: Find all the eigenvalues and eigenvectors of the covariance matrix Cov. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues and $v_1, v_2, \ldots, v_n$ be the corresponding eigenvectors. The eigenvectors are sorted by the nondecreasing order of the corresponding eigenvalues.
Step 4: For each input vector cw_i, its first principal component y_1 can be computed by the following inner product:

$$y_1 = v_1^T cw_i . \quad (2)$$

3.2 Embedding Process

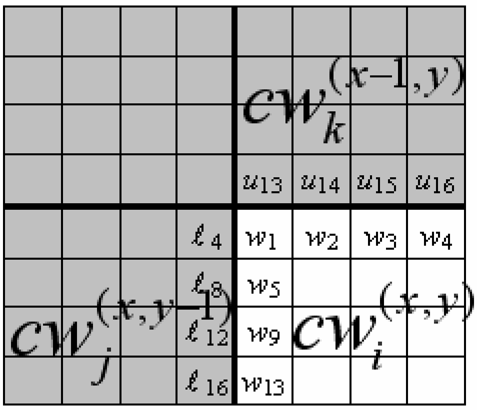


Fig. 4. Evaluating side-match distortion

After sorting the codebook, each codeword, except for the first and last, finds its dissimilar codeword by the following rule to form a dissimilar pair: (cw_i, $cw_{((k-2)/2)+i}$), where $1 \le i \le (k-2)/2$ and k is the codebook size. Assume the codeword cw_i, with its index in the codebook as i, in the position (x, y) of the VQ image denotes $cw_i^{(x,y)}$ and each codeword represents a 4×4 subimage block. The side-match distortion function SMD for an input block $cw_i^{(x,y)}$ is defined in Eq. (3), which is the Euclidean distance between the border values of $cw_i^{(x,y)}$ and their adjacent values, where w_i, ℓ_i, and u_i are the elements of codewords $cw_i^{(x,y)}$, $cw_j^{(x,y-1)}$, and $cw_k^{(x-1,y)}$, respectively, as shown in Fig. 4.

$$SMD(cw_i^{(x,y)}) = [(w_1 - \frac{u_{13}+\ell_4}{2})^2 + (w_2 - u_{14})^2 + (w_3 - u_{15})^2 + (w_4 - u_{16})^2 + (w_5 - \ell_8)^2 + (w_9 - \ell_{12})^2 + (w_{13} - \ell_{16})^2]^{1/2}. \quad \textbf{(3)}$$

A subimage block in the VQ image is called *changeable* if $SMD(cw_i^{(x,y)}) < SMD(cw_{((k-2)/2)+i}^{(x,y)})$. A changeable block can embed one secret bit by cw_i and $cw_{((k-2)/2)+i}$ to represent 0 and 1, respectively. If $cw_i^{(x,y)}$ is not changeable, we select t most similar codewords in order from the sorted codebook according to the Euclidean distance function, and verify sequentially which one is changeable to replace $cw_i^{(x,y)}$, making the codeword in the position (x, y) of the VQ image become changeable. This kind of $cw_i^{(x,y)}$ is called *pseudo-changeable*. The value of t deeply impacts the image quality after extracting the secret bits, so it should be selected carefully. If the t codewords are not all changeable, $cw_i^{(x,y)}$ still can embed one secret bit by $cw_0 \| cw_i^{(x,y)}$, $cw_{k-1} \| cw_i^{(x,y)}$ to represent 0 and 1, respectively, where $\|$ denotes the concatenation. Such $cw_i^{(x,y)}$ is called *unchangeable*. Note that in our scheme, the first and the last codewords, cw_0 and cw_{k-1}, are not used in the VQ encoding process, so the original VQ image would not contain the indices of cw_0 and cw_{k-1}. Besides, for convenience, the blocks of the first row and the first column of the VQ image do not participate in the embedding process. The algorithm of the embedding process is summarized below.

Embedding Process.
Input: A VQ-compressed image and a stream of secret bits.
Output: A stego VQ-encoded code.

Step 1. Retrieve the next image block $cw_i^{(x,y)}$ of the VQ image in a raster-scan order.

Step 2. If $cw_i^{(x,y)}$ is changeable, it remains unchanged to represent secret bit 0, and is replaced with $cw_{((k-2)/2)+i}^{(x,y)}$ to represent secret bit 1. Go to Step 5.

Step 3. If $cw_i^{(x,y)}$ is unchangeable, select t most similar codewords from the sorted codebook. If any one of t codewords is changeable, replace $cw_i^{(x,y)}$ with the most similar changeable codeword. Go to Step 2.

Step 4. If the t codewords are all unchangeable, represent secret bit 0 and 1 by $cw_0 \| cw_i^{(x,y)}$, $cw_{k-1} \| cw_i^{(x,y)}$, respectively.

Step 5: Repeat Steps 1 through 4 until all the indices are processed.

3.3 Extraction Process

The extraction process not only can extract the secret bits but also can correct the blocks of the VQ image to the original or similar ones. If $cw_i^{(x,y)}$ is changeable or unchangeable, the original $cw_i^{(x,y)}$ can be completely recovered after extraction. However, if $cw_i^{(x,y)}$ is pseudo-changeable, $cw_i^{(x,y)}$ is modified to another similar codeword after extraction. The extraction process is as follows:

Extraction Process.
Input: A stego VQ-encoded code.
Output: A VQ-compressed image and a stream of secret bits.

Step 1. Retrieve the next VQ-encoded block code in the same order as the embedding process.

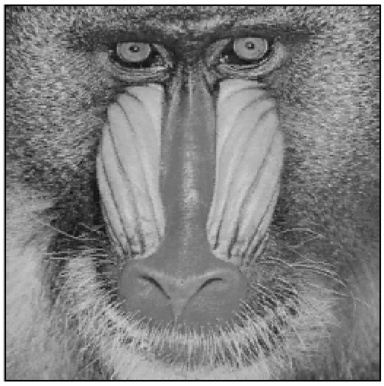
Step 2. If $cw_i^{(x,y)}$ is not unchangeable and $SMD(cw_i^{(x,y)}) < SMD(cw_{((k-2)/2)+i}^{(x,y)})$, the secret bit is 0 and $cw_i^{(x,y)}$ remains unchanged. However, if $SMD(cw_i^{(x,y)}) > SMD(cw_{((k-2)/2)+i}^{(x,y)})$, the secret bit is 1 and $cw_i^{(x,y)}$ is changed to $cw_{((k-2)/2)+i}^{(x,y)}$.

Step 3. If $cw_i^{(x,y)}$ is equal to cw_0 (or cw_{k-1}) (i.e., unchangeable case), then the secret 0 (or 1) bit can be extracted and $cw_i^{(x,y)}$ is changed to the next encoded code of cw_0 (or cw_{k-1}).

Step 4. Repeat Steps 1 through 3 until all the secret bits are pieced together.

4 Experimental Results

In this section, we show some experimental results to demonstrate the effectiveness and efficiency of our new schemes. The three standard 512×512-pixel grayscale VQ images shown in Fig. 5 were used as the cover images to hide a random bitstream produced by a random-number generator. The cover image was divided into 16384 blocks of 4×4 pixels. The codebook comprising 512 16-dimensional codewords used in the experiments was generated using the LBG (Linde-Buzo-Gray) algorithm [12], and then sorted using the PCA [4] method before embedding.



(a) Lena with a PSNR of 32.24 dB (b) Pepper with a PSNR of 31.40 dB (c) Baboon with a PSNR of 24.70 dB

Fig. 5. Three cover images

Table 1 compares the various methods. The table shows that although the embedding capacity of the proposed method is slightly lower than that of Lin and Wang's method, the image quality of the proposed method is greatly improved after the secret bits have been extracted. Jo and Kim's method [9], an improved version of Lin and Wang's method, benefits the image quality by reducing the embedding capacity. A comparison of these results shows that the proposed method has the best image quality and acceptable embedding capacity. The results of the "Lena" image using all three methods are shown in Fig. 6, and illustrate that the methods developed by Lin and Wang and Jo and Kim obviously encounter more severe block effects and distortions than the proposed method.

Table 1. The experimental results of various methods

Method	Item	Lena	Pepper	Baboon
Proposed method	Embedding capacity (bit)	16129	16129	16129
	PSNR (dB)	32.240	31.407	24.703
Lin et al's method	Embedding capacity (bit)	16384	16384	16384
	PSNR (dB)	28.831	28.226	23.092
Jo et al's method	Embedding capacity (bit)	14510	14749	10946
	PSNR (dB)	30.745	29.877	24.324

Table 2 presents the details of the proposed method under different thresholds. The threshold is used for the pseudo-changeable case. The larger threshold results in more pseudo-changeable cases but fewer unchangeable cases. The extra size in this table is the difference between the original file size and the stego file size. From this table, we can observe that the proportion of unchangeable blocks to the total image blocks is very small, so the increase in stego file size is quite limited. When the threshold is set to 200, the stego file size is equal to the original file size. Besides, "Baboon" has more most unchangeable cases than other images since "Baboon" is more complicated. The PSNRs of the images are decreased as the number of pseudo-changeable case is increased, but all the PSNRs of the proposed method are still very close to that of the original VQ images. In this table, the number of embedding bits of each cover image is independent of the image itself and the threshold, and is equal to 16129 bits.

(a) The proposed method (b) Lin and Wang's method (c) Jo and Kim's method

Fig. 6. The results after extraction of the secret bits

Table 2. Adjustable embedding approach using different thresholds

Threshold	Item	Lena	Pepper	Baboon
	Extra size (bit)	1,251	1,233	4,122
	Number of pseudo-changeable blocks	0	0	0
0	Number of unchangeable blocks	139	137	458
	PSNR (dB)	32.240	31.407	24.703
	Extra size (bit)	945	639	2,682
	Number of pseudo-changeable blocks	34	66	160
50	Number of unchangeable blocks	105	71	298
	PSNR (dB)	32.223	30.440	24.679
	Extra size (bit)	234	144	333
	Number of pseudo-changeable blocks	113	121	421
100	Number of unchangeable blocks	26	16	37
	PSNR (dB)	32.053	30.355	24.583
	Extra size (bit)	0	0	0
	Number of pseudo-changeable blocks	139	137	458
200	Number of unchangeable blocks	0	0	0
	PSNR (dB)	31.900	30.300	24.563

5 Conclusions

In this paper, we propose a new embedding method for VQ-compressed images. In traditional VQ hiding methods, similar codewords are gathered in the same groups to represent the secret bits. This approach suffers from serious distortions caused by codeword replacements. The proposed method, on the other hand, applies a declustering technique that allows the original image to be nearly fully recovered after extraction of the secret bits. Although in the proposed method, the size of the stego file may be larger than that of the original one, this can be easily resolved by adjusting a threshold to let the number of unchangeable blocks be zero. Such an adjustment creates only a tiny distortion of the original image. The experimental results demonstrate that the proposed method can completely extract the secret data and nearly recover the original image without referencing any auxiliary information. Furthermore, in comparison with other methods, the proposed method has the best image quality after extraction of the secret bits and can achieve high embedding capacity.

References

[1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, Vol. 35, No. 3&4, pp. 313-336, 1996.

[2] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, Vol. 37, No. 3, pp. 469-474, 2004.

[3] C. C. Chang, J. Y. Hsiao, and C. S. Chan, "Finding optimal LSB substitution in image hiding by dynamic programming strategy," *Pattern Recognition*, Vol. 36, No. 7, pp. 1583-1595, 2003.

[4] C. C. Chang, D. C. Lin, and T. S. Chen, "An improved VQ codebook search algorithm using principal component analysis," *Journal of Visual Communication and Image Representation*, Vol. 8, No. 1, pp. 27-37, 1997.

[5] A. Gersho and R. M. Gray, "Vector quantization and signal compression," Kluwer Academic Publishers, 1992.

[6] R. M. Gray, "Vector quantization," *IEEE ASSP Magazine*, 1984, pp. 4-29.

[7] Y. C. Hu and C. C. Chang, "A progressive codebook training algorithm for vector quantization," *Proceedings of the Fifth Asia-Pacific Conference on Communications and Fourth Optoelectronics and Communications Conference*, Beijing, China, pp. 936-939, 1999.

[8] A. Hyvarinen, J. Karhunen, and E. Oja, "Independent component analysis," New York: John Wiley & Sons, 2001.

[9] M. Jo and H. D. Kim, "A digital image watermarking scheme based on vector quantization," *IEICE Transactions on Information and Systems*, Vol. E85-D, No. 6, pp. 1054-1056, 2002.

[10] R. C. T. Lee, Y. H. Chin, and S. C. Chang, "Application of principal component analysis to multikey searching," *IEEE Transactions on Software Engineering*, Vol. SE-2, No. 3, pp. 185-193, 1976.

[11] Y. C. Lin and C. C. Wang, "Digital images watermarking by vector quantization," *Proceedings of National Computer Symposium*, Vol. 3, pp. 76-87, 1999.

[12] Y. Linde, A. Buzo, and R. M. Gary, "An algorithm for vector quantization design," *IEEE Transactions on Communications*, Vol. 28, No. 4, pp. 84-95, 1980.

[13] Z. M. Lu, J. S. Pan, and S. H. Sun, "VQ-based digital image watermarking method," *Electronics Letters*, Vol. 36, No. 14, pp. 1201-1202, 2000.

[14] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—a survey," *Proceedings of the IEEE*, Vol. 87, No. 7, pp. 1062-1078, 1999.

[15] R. G. V. Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," *Proceedings of the IEEE International Conference on Image Processing (ICIP-94)*, Austin, Texas, pp. 86-90, 1994.

[16] R. Z. Wang, C. F. Lin, and J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," *Pattern Recognition*, Vol. 34, No. 3, pp. 671-683, 2001.