Digital Images Watermarking by Vector Quantizatio

Yih-Chuan Lin
Department of Information Management
SHU-TE Institute of Technology, YenChau,
Kaoshiung, Taiwan, R.O.C.
E-mail: yclin@mail.stit.edu.tw

Chuen-Ching Wang
Directorate General of Telecommunications
Ministry of Transportation & Communications
Taipei, Taiwan, R.O.C.
E-mail: ccwang@dgt.gov.tw.

Abstract

This paper presents a new watermarking method based on the vector quantization algorithm. The work to insert a watermark into an image is accomplished by altering the way of doing image vector quantization. The method reduces the vector quantization resolution in the search space, and thus marks the host image in that it may not produce the truly best codevectors according to the watermarking bits. Various experiments have been conducted and shown that the proposed vector quantization watermarking method is of tampering resistance and is robust to a variety of common image processing operations, including channel noise, loss compression, lowpass filtering, rescaling, and cropping.

Keywords: Watermarking, Image Processing, Vector Quantization, Copyright Protection

1. Introduction

Digital images watermarking that embeds a digital watermark signal into a digital image and detects the watermark later offers a means of protecting the intellectual property of digitized visual contents that have been explosively exchanged in the digital world. Some of the early works rely on identifying the redundant properties of an image and modifying the recognized redundant properties for hiding message [1]. For example, there exists a simple watermarking technique [2-3] that alters the least significant bits of some pixels in an image to encode the watermark information. The watermark inserted in the image would be imperceptible because of the insensitivity of human visual system (HVS) to the random alterations in the least significant bits. While it is very much simple, the watermark embedded in the image would be easily removed when simply applying a lossy compression operation (such as the Joint Photographic Experts Group JPEG [4]) to it. In [3,5], the block-mean approach was proposed. The image should be divided into blocks. The mean intensity of each block is incremented by a certain value to encode bit '1' of watermark information, or decremented to encode bit '0'. This method is susceptible to the attack of collusion, that is, a number of independent watermarked copies of the image would be used to detect or remove most of the watermarked information by comparing the different copies. Randomizing the block size or location for the watermarking of each image can alleviate this weakness. Although encoding watermark by modifying the block mean is an intuitively robust strategy to most of the lowpass oriented signal operations; the alteration of block mean would possibly create visible artifacts.

The Patchwork [1] that was developed by the MIT Media Laboratory identifies the fact that the intensity difference of any two pixels randomly chosen from an image follows a Gaussian distribution with a mean of zero. The Patchwork encodes the watermark by shifting the distribution of the identified statistic. Therefore, to verify the watermarked image, the decoding algorithm can employ a hypothesis testing that involves merely a threshold to make a decision as to whether the image is watermarked or not, without resort to the original unencoded image. This work is robust to the non-geometric transformations, but is only capable of embedding one-bit watermark. There is a similar approach as the Patchwork having been proposed in

[6]. This technique can embed a watermark in a rate of one bit per pixel. The watermarking task is accomplished by means of modifying the intensity of individual pixels to be marked. The detection algorithm is similar to the Patchwork, but the statistic used by the hypothesis testing is differently defined as the difference between the mean intensity of the marked pixels and that of the unmarked pixels.

The above mentioned watermarking techniques are usually categorized into the same class, referred to as *spatial domain watermarking.* To see more other spatial domain techniques, refer to the literature [7 -8] that has made a thorough review on watermarking strategies. Most of the above stated techniques that embed a watermark by modifying the spatial characteristics, such as the brightness of individual pixels, usually introduce the h igh-frequency watermarking noise into the marked images. Thus, they are susceptible to the attack of JPEG lossy compression or other lowpass oriented operations. To alleviate this weakness, there have been a number of *transform domain watermarking* techniques developed in [9-13], which insert watermark signal by modifying some transform coefficients of the image to be marked. The idea behind the transform techniques is that the noise of watermarking in the transform coefficients would spread over the whole spatial extent of the image [9], and thus is more robust to some geometric transformations. In order for resistance to JPEG, the techniques reported in [11-12] embed the watermark in the 8x8 DCT domain instead of the transform domain of whole image. The techniques based on the JPEG model usually insert watermark code in the middle frequency region of 8x8 DCT coefficients for both robust and imperceptible purposes. However, to select the appropriate middle frequency coefficients for modification is not an easy task because the amplitude of most coefficients in the middle frequency region is so small that any information modulated in them would be vanished by the quantization process of JPEG compression. In order to survive the quantization process, the coefficients nearby the low frequency region could be employed to carry watermark information. However, the artifacts in the marked image would appear apparentl .

In this paper, a new watermarking technique is presented that embeds watermark code into an imag e based on the vector quantization (VQ) algorithm [14]. This technique inserts watermark in a manner that alters the intensity of a group of neighboring pixels (vector) other than the individual pixels as done by the other spatial techniques. The alteration of vector pixels is not uniformly, which is done by comparing to a codebook of codevectors and replacing with the best match codevector in the codebook. The advantage of this method is that the watermarking noise can be moved away from the high frequen cy region and consequently gains more robust to the lowpass oriented operations. In the proposed algorithm, the codevectors in the codebook should be classified into two subcodebook such that all the codevectors in the same subcodebook possess a predefin ed homogeneity on their indices. The result of this classification depends on the ordering of codevectors in the codebook. The image to be marked should be divided into a sequence of vectors with a length equal to that of the watermark bit stream. For ea ch incoming vector, VQ encoder chooses one of the two subcodebooks according to the value of watermark bit and searches through the chosen subcodebook for the closest codevector. The marked images are in the VQ compressed form that comprises the codebook a nd the generated sequence of codebook indices. This is advantageous to the decoding procedure so that to read and process the compressed data is sufficient for extracting the watermark. In other words, only the VQ indices and codebook are required to decode the watermark.

By the nature of VQ, making a somewhat modification to the VQ decompressed image would not result in a large change on the corresponding bits of V indices whereas a small alteration to the index bits would lead to a significant change on the retrieved codevectors.

Therefore, the proposed technique would exhibit more robustness to most common attacks. Most VQ -based compression techniques can thus possess the ability to protecting the copyright of the compressed images. The experimental results have shown that the proposed technique is robust to a number of common signal processing operations, such as lossy compression, blurring, cropping etc.

2. Vector Quantization Algorithm

Vector quantization is a straightforward generalization of the traditional scalar quantization with a significant increase of computational complexity. Instead of the individual sample quantization, a group of successive source samples (referred to as vector) is jointly quantized and assigned to a representative vector in R^k. Formally, a vector quantization of k-dimensional and size n can be regarded as a mapping Q from a k-dimensional space R^k to a finite subset Y of R^k. That is,

$$Q : R^k \rightarrow Y$$

where $Y=\{y_i : i=1,2, \ldots, n\}$ is the codebook and n stands for the c odebook size. For an incoming vector x, vector quantization maps x to a codevector y_i in Y, i.e., $Q(x)=y_i$, such that y_i is closest to the input vector among all the codevectors in Y. The distortion measure that has often been used in most VQ systems is defined as the Euclidean distortion measure

$$d(x, y)=\sum_{m=1}^{k}(x_m - y_m)^2 \quad (1)$$

where x_m and y_m denote the mth components of the input vector x and the codevector y_i, respectively. This minimum distortion mapping can also be termed the nearest neighbor rule, which aims to select a codevector y_i such that $d(x, y_i)< d(x, y_j)$ for all $i \neq j$.

From the viewpoint of source coding, the mapping Q can be further expanded into the composition of two other mappings, the encoder E and decoder D, that is,

$$Q(x)=D(E(x)),\ E : R^k \rightarrow I,\ D : I \rightarrow Y,\ x \in R^k. \quad (2)$$

where $I=\{1,2, \ldots, 2^l\}$ is a set of codebook indices, each of which is associated to only one codevector in the codebook Y, and l is the bit length of binary code for each index, which is determined as $l = \log_2^n$. A schematic diagram of VQ coding system is shown in Figure 1. The VQ encoder E processes each input vector x and outputs an index i, which is chosen from the codebook Y in accordance with the nearest neighbor rule, to the communication channel. In short, the behavior of VQ encoder can be formulated as

$$E(x) = Arg \min_{i \in I} d(x, y_i),\ \forall x \in R^k \quad (3)$$

Upon receipt of the index i, the decoder D performs a table lookup operation to retrieve the codevector y_i as the decoded vector. Also, the decoder D can be described as

$$D(i) = y_i,\ \forall i \in I \quad (4)$$

It is apparent that the decoder D needs no arithmetic operation. Data compression is achieved by means of transmitting or storing the indices of codevectors other than the original input vectors. The quality distortion incurred in the decoded image is accumulated from the distortion of approximating each input vector with its closest codevector.

One of the important tasks in a vector quantization is to design a good codebook using a large training set. The generalized Lloyd clustering algorithm (referred to as the LBG algorithm) proposed by Linde, Buzo and Gray [15] is most popular for designing codebooks. Starting with a randomly initial codebook, the LBG algorithm runs iteratively the task of clustering the training set until the stop condition is satisfied. Since in each iteration, an exhaustive codebook search procedure is involved and in general, a large-sized set of training vectors must be required, to generate codebooks is a time-consuming task. Fortunately, many fast algorithms for codebook generation and search have been proposed [16-20] to alleviate this problem.

3. The Proposed Watermarking Scheme

3.1 Watermarking algorithm

Let the image F to be marked (named host image) be of size $W \times H$ and the watermark S that is a visually recognized binary image have $M \times N$ pixels. Prior to embedding into the host image F, the binary image S is first permuted using the pseudorandom number generator, yielding a permuted version $\bar{S}$. For the sake of convenient presentation, we will use the integers in the set $A = \{1,2,\cdots,M \times N\}$ for indexing each input vector and the corresponding watermark bit. The works to do the proposed VQ watermarking can be broken down into the five major steps:

1. Scan the watermark $\bar{S}$ into a stream of binary bits
   $\{s_i\}_{i \in A}, s_i \in \{0,1\}$.
2. Divide the host image F into a sequence of $M \times N$ vectors
   $\{x_i\}_{i \in A}$, $x_i \in R^k$, such that
   $W \times H = k \times M \times N$.
3. Generate a codebook Y of size n using the LBG algorithm with certain training set.
4. Determine a rule to classify the codebook Y into two subcodebooks based on the indices of the codevectors, or equivalently to partition the set of indices into two disjoint subsets.
5. Perform a modified encoder E' using the data stated above as the inputs, and produce a watermarked image in the VQ compressed form.

As pointed out in the fourth step, the binary classification rule that is designed to bi-partition the set I of indices would be treated as follows

$$C : I \rightarrow \{0,1\}.$$

For each index i, the function is designed to behave as

$$C(i) = b_{i1} \oplus b_{i2} \cdots \oplus b_{il}. \qquad (5)$$

where the binary string $b_{i1}b_{i2}\cdots b_{il}$ represents the binary form of i and the symbol $\oplus$ stands for the bitwise EXCLUSIVE-OR operator. For example, the codevecto y_{127} should be classified as '1' because

$$C(127) = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1.$$

With the classification rule defined in (5), the set I can then be partitioned into

$$I_0 = \{i \in I : C(i) = 0\} \text{ and}$$
$$I_1 = \{i \in I : C(i) = 1\}.$$

When n is an integral power of 2, the size of I_0 is equal to that of I_1, that is, $|I_0| = |I_1| = \frac{n}{2}$. Therefore, the whole codebook Y can be equivalently divided into the two subcodebooks

$$Y_0 = \{y_i : i \in I_0\} \text{ and}$$
$$Y_1 = \{y_i : i \in I_1\}.$$

To insert the watermark, the sequences of $\{x_i\}_{i \in A}$ and $\{s_i\}_{i \in A}$ should be given together as the input of the modified VQ encoder E', which is

$$E' : R^k \times \{0,1\} \rightarrow I.$$

For each input data $(x_i, s_i) \in R^k \times \{0,1\}$, $E'(x_i, s_i)$ would output an index t_i that is determined b

$$t_i = Arg \min_{t \in I_{s_i}} d(x_i, y_t) \qquad (6)$$

As can be found in (6), the subcodebook that should be selected for searching depends on the value of the watermark bit s_i. If s_i =1, then $t_i \in I_1$, otherwise $t_i \in I_0$. Therefore, for each input vector x_i, only one half of the codebook is considered by the VQ search process for seeking the closest codevector to x_i while the truly closest codevector may be located in the other one half. The difference between the two closest codevectors found in the whole and one half of the codebook, respectively, represents the watermarking noise. In fact, it is possible that some input vectors do not introduce the watermarking noise if the two closest codevectors in their cases are exactly identical. Since the codevectors in the either subcodebook would spread over the whole search space, the possible watermarking noise could be controlled under a tolerant level. Table 1 illustrates the examples of watermarking noise for different codebook sizes. It can be find that the watermarking no ise would be maintained about 2dB in terms of peak signal-to-noise ratio (PSNR).

3.2 Extraction algorithm in the compressed domain

The task of extracting the watermark could be done in the compressed domain by the modified VQ decoder D' that works with involving the classification function defined in (5) for determining each of the embedded watermark bits. The encoder D' needs not only doing the lookup operation for each incoming index $t_i \in I$ but also determining what value of the watermark bit s_i should be. Thus, the encoder can be concisely represented as

$$D' : I \rightarrow R^k \times \{0,1\}.$$

For the ith incoming index t_i, $D'(t_i)$ could produce an ordered pair (y_{t_i}, s_i), in which $s_i = C(t_i)$ and y_{t_i} is retrieved from the codebook Y by a table lookup operation. All the retrieved vector $y_{t_i}, i = 1,2,\cdots M \times N$, form collectively the decompressed image F'. Figure 2 shows the block diagram of the proposed VQ watermarking system. While D' is in charge of the two tasks, it does not involve any extra arithmetic operation.

The watermark extraction/verification algorith with a lower computation burden is more suited for such applications as image database or broadcast on World Wide Web (WWW), which usually have less powerful equipment in the receivers. Moreover, designing simpler decoders would be also an important concern for the applications in the environment of mobile multimedia information system that is expected to have devices with as low power consumption as possible.

3.3 The secure considerations when publicizing the algorithms

If the proposed watermarking algorithm has to be publicized, attackers can easily decode the embedded watermark by making use of the binary classification function on the received stream of indices, and consequently can make changes on some of the received indices to remove the inserted watermark.

While this proble can be solved with resort to the original stream of indices, we will propose to use a random permutation p on the initial ordering I of codebook indices for the binary classification. In this circumstance, there are two orderings of codebook indices, p and I, available for the watermarking algorithm. For each input ordered pair (x_i, s_i), E' performs watermarking based on the ordering p, and sends out indices accordin to the ordering I. This could be described precisely b

$$E'(x_i, s_i) = Arg \min_{t \in I'_{s_i}} d(x_i, y_t),$$

where the two subsets of I is redefined as follows.

$$I'_0 = \{i \in I : C(P(i)) = 0\} \text{ and}$$

$$I'_1 = \{i \in I : C(P(i)) = 1\}.$$

The random permutation P on I is treated as a part of the secret key that should be kept secretly by the owner and need be employed when decoding the watermark embedded in the received stream of indices. For the ith received index t_i, $D'(t_i)$ could produce an ordered pair (y_{t_i}, s_i), where $s_i = C(P(t_i))$ With invol ving the random permutation P, the watermark is difficult to be decoded by the intentional attackers directly based on the received indices, and it is also not easy to remove the watermark without severely destroying the decompressed qualit .

Moreover, in order to control the individual recipient of an image, the owner of the image usually embeds a different watermark into each copy of the image before sending to the receivers. If the proposed method does such task with the fixed permutation P of indices or the fixed codebook Y, the receivers each holds one independent watermarked copy of the image can collude to partially detect the binary classification of codebook based on the comparison among the different watermarked copies of the image. Having detected how to classify the codebook, the receivers are able to remove the different watermarks that have been embedded in the copies of the image. To offer the resistance to the collusion attack, the VQ watermarking algorithm can be designed to use a different random permutation P or to regenerate a codebook for the watermarking of each copies of the image. As a result, to collusively remove the watermarks becomes difficult because each watermarked copy of the image is done by means of a different secret key.

3.4 Extraction algorithm in the decompressed domain

While the watermark can be recovered from the received index stream at the decoder side, it could also be extracted compatibl from the VQ decompressed image F'. This is of very importance because the intentional attackers would probably attempt to remove the watermark information through various kinds of image processing operations for the VQ decompressed image F'. Therefore, it is necessary that the proposed watermarking algorithm can provide a strong robustness against the attack of common image processing operations for the decompressed image F'.

To extract watermark in the decompressed domain, the original VQ encoder E would be employed to encode the image F' into the form of index stream $\{t_i'\}_{i\in A}$, where the codevector $y_{t_i'}$ is closest to the ith input vector x_i' of the image F' among all the codevectors in Y. Subsequentl , the watermark bit stream could be calculated based on applying the binary class ification function to the index strea $\{t_i'\}_{i\in A}$, which is

$$s_i = C(P(t_i')),\ \forall i \in A.$$

If the image F' and the original index strea $\{t_i\}_{i\in A}$ are never corrupted by the intentional or unintentional attackers, then $\{t_i'\}_{i\in A} = \{t_i\}_{i\in A}$ and the watermark can be thus exactly extracted. When some bits of the index stream have been altered, parts of the watermark information may be lost or there exists some noise in the decoded watermark. Suppose that the image F' has been processed by some kind of image processing tool, such as JPEG compression, then we would have $\{t_i'\}_{i\in A} \neq \{t_i\}_{i\in A}$, i.e., there exits at least one integer $d \in A$ such that $t_d \neq t_d'$. Thus, the watermark may be partially extracted or be disturbed by more or less noise. Although the watermark extracted from the corrupted data appears noisy, in most cases it could still be recognized visually. In other words, the noisy watermark is still qualified enough to verify the copyright of the host image F.

In the next section, a set of experiments has been conducted and would be used to explain the effectiveness of the proposed VQ watermarking algorithm.

4. Experimental Results

The computer simulation program that was written in the C program has implemented the proposed VQ watermarking algorithms and run on a personal computer with Intel Pentium II CPU to verify the performance of the proposed method. In the experiments, a high-dimensional, moderate-sized VQ coding system for digital images was concentrated on, in which the codebook was generated by the LBG algorithm with the training set composed of the vectors in the host image. The k of vector dimensions and n of codebook size are set to 16 and 256, respectively.

Figure 3 shows the considered host image F that consists of 512x512 pixels and each pixel renders 256 possible intensity levels. The watermark image S, as shown in Figure 4(a), is composed of 128x128 binary pixels. Figure 4(b) displays a randomly dispersed version $\overline{S}$ of the binary image S, in which the seed value of pseudorandom number generator for dispersing the original watermark image serves as part of the secret key for the extraction of S.

One of the advantages of rearranging the watermark bits lies in increasing the robustness to the geometric cropping operation.

Figure 5(a) displays the VQ watermarked image with a bit rate of 0.5 bpp (bits per pixel) and the fidelity measure of PSNR=28.7dB. The binary watermark image that is extracted from the VQ watermarked indices without any modification is shown in Figure 5(b). To examine the situation of modifying the bits of the VQ index stream, we have dealt with the case of noisy channel that would randomly alters the bits sent through it with the bit error rate (BER) $e=10^{-2}$ and $10^{-1.5}$. The results are shown in Figures 6(a)-(d). With the experiments, it is concluded that the watermarking scheme exhibits a considerable immunity to the channel noise to an extent of $e \leq 10^{-1.5}$. As shown in Figure 5(d), the extracted watermark is visually useful while there is a large amount of noise incurred. When increasing the BER over the value of $10^{-1.5}$, we have found that both the decompressed host image and the extracted watermark are degraded significantly in quality and thus are almost of no further usefulness. Consequently, the proposed watermarking algorithm has been proved being able to offer a high robustness to the direct attacks on the index stream.

Next, we consider the resistance to the common image processing operations, including JPEG compression, lowpass filtering, cropping and rescaling. The considered attacks are applied to the watermarked image that is reconstructed by the VQ decoder with the received codebook. The watermarked image should be converted into the form of VQ indices by the original VQ encoder with the same codebook, and the watermark would be extracted from the reproduced stream of indices. Fi gures 7(a)-(c) give the JPEG compressed, watermarked images with the compression ratios Cr=8.7, 16.3, and 21.3, respectively.The extracted binary watermarks are also shown correspondingl in Figures 7(d)-(f). When compressing the watermarked images to the extent of Cr>21.3, the watermark information would hardly be recognizable. Figure 8(a) provides the resultant image after lowpass filtering the watermarked image, where the effects of lowpass filtering was achieved by taking a convolution with the 3x3 mas k of moving average. The decoded watermark, as shown in Figure 8(b), has proved that the proposed algorithm possesses the robustness to the lowpass filtering. Figure 9 reveals that if a certain region in the watermarked image is cropped and then is replaced with the constant black intensity, as shown in Figure 9(a), the extracted watermark information is still useful enough to identify the origin of the watermarked image. Consider the results shown in Figure 10. The rescaled image was generated in the manner that first reduces down the watermarked image to one fourth of its original size by averaging on every fou neighboring pixels, and then scales up to the original size by interpolating the pixels in the reduced image. With this rescaling operation, a lot of detail information appeared in the watermarked image would be lost, and is hard to recover. Also, we can find that the decoded watermark, as shown in Figure 10(b), remains being qualified to verify the ownership of the watermarked image after this lo ss transformation.

5. Conclusion

Data compression has been an essential part of current digital media systems that require transmitting and storing a large amount of digital media. On the other hand, digital watermarking can offer a mechanism of protecting the copyright of the digital media. Therefore, it should be desirable to joint a watermarking mechanism into the data compression techniques without introducing extra bit rate. In this paper, a general image vector quantization system has been evolved towards having the capabilities of both image compression and watermarking. From the point of view of source coding, the proposed VQ watermarking method does not incur any extra bit rate after inserting the copyright code into the source image. In addition, the watermark extraction procedure inherits the simple complexity from the VQ decoder, and does not need the resort to the original source image. This feature makes the proposed method suited for the applications that are equipped with the so-called *thin client* devices.

A set of experimental results has been conducted and shown that the proposed VQ watermarking algorith exhibits significant robustness to most common image processing operations. Also, the attack of noisy communication channel where the transmitted bits of indices may be altered in some range of bit error rate is also considered. The result indicates that the embedded watermark can survive the disturbance of channel noise even in the level of wireless channel. Besides, the proposed method has been proved being secure and resistant to the attacks of tampering and collusion even when the watermarking algorithm is demanded to publicize.

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] L. F. Turner, "Digital data security system," Patent WO 89/08915, 1989.

[3] C. Dautzenberg and F. M. Boland, "Watermarking images," Technique Report, Department of Electronic and Electrical Engineering, Trinity College Dublin, 1994.

[4] G. K. Wallace, "The JPEG still picture compression standard," Commun. ACM vol. 34, no. 4, pp. 30-40, 1991.

[5] G. Caronni, "Assuring ownership rights for digital images," in Proc.Reliable IT Systems, VIS'95, Vieweg Publishing Company, 1995.

[6] N. Nikolaidis, and I. Pitas, "Robust image watermarking in the spatial domain," Signal Processing, vol. 66, pp. 385-403, 1998.

[7] I. J. Cox, and M. L. Miller, "A review of watermarking and the importance of perceptual modeling," in Proc. SPIE Int. Conf. Human Vision and Electronic Imaging-II, pp. 92-99, 1997.

[8] J. J. K. O Ruanaidh, W. J. Dowling, and F. M. Boland, "Watermarking digital images for copyrigh protection," IEE Proc.-Vis. Image Signal Process. Vol. 143, no. 4, pp. 250-256, 1996.

[9] I. J. Cox, J. Kilian, T. Lighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Trans. on Image Processing, vol. 6, no. 12, pp. 1673-1687, 1997.

[10] M. Barni, F. Bartolini, V. Cappellini, A. Piva, "A DCT-domain system for robust image watermarking," Signal Processing, vol. 66, pp. 357-372, 1998.

[11] E. Koch, and J. Zhao, "Towards robust and hidden image copyright labeling," in Proc. IEEE Nonlinear Signal and Image Processing, pp. 452-455, 1995.

[12] C. T. Hsu, and J. L. Wu, "Hidden digital watermarks in images," IEEE Trans. on Image Processing vol. 8 no. 1, pp. 58-68, 1999.

[13] J. J. K. O. Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," in Proc. IEEE Int. Conf. Image Processing, pp. 239-242, 1996

[14] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression. Boston MA: Kluwer, 1992.

[15] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantization design," IEEE Trans. on Commun. vol. 28, pp. 84-95, 1980.

[16] C. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," IEEE Trans. on Commun., vol. 33, pp. 1132-1133, 1985.

[17] Y. C. Lin and S. C. Tai, "A fast Linde-Buzo-Gray algorithm in image vector quantization," IEEE Trans. on Circuits and Systems-II, vol. 45, pp. 432-435, 1998.

[18] S. C. Tai, C. C. Lai, and Y. C. Lin, "Two fast nearest neighbor searching algorithms for image vector quantization," IEEE Trans. on Commun., vol. 44, pp. 1263-1268, 1996.

[19] W. H. Equitz, "A new vector quantization clustering algorithm," IEEE Trans. on Acoust., Speech, Signal Processing, vol. 37, pp. 1568-1575, 1989.

[20] C. M. Huang and R. W. Harris, "A comparison of several vector quantization codebook generation approaches," IEEE Trans. on Image Processing, vol. 2, pp. 108-112, 1993.

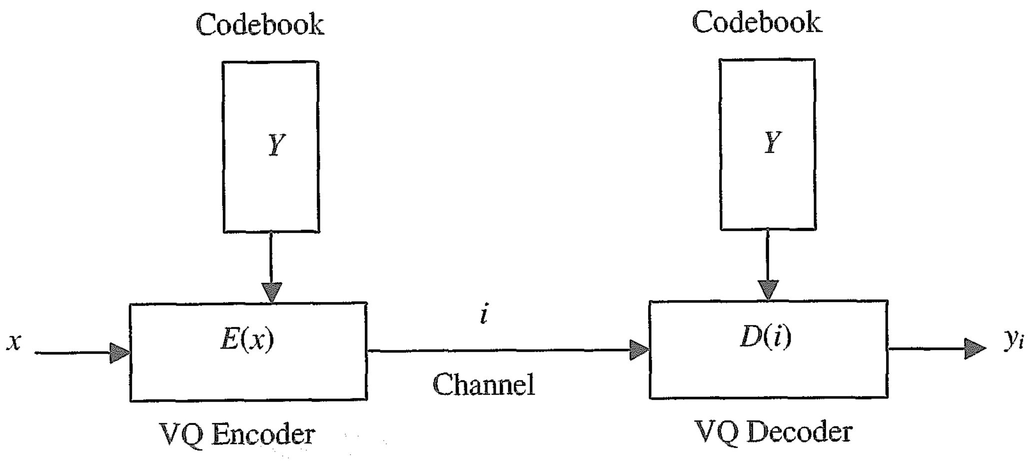


Figure 1. The block diagram of VQ coding system.

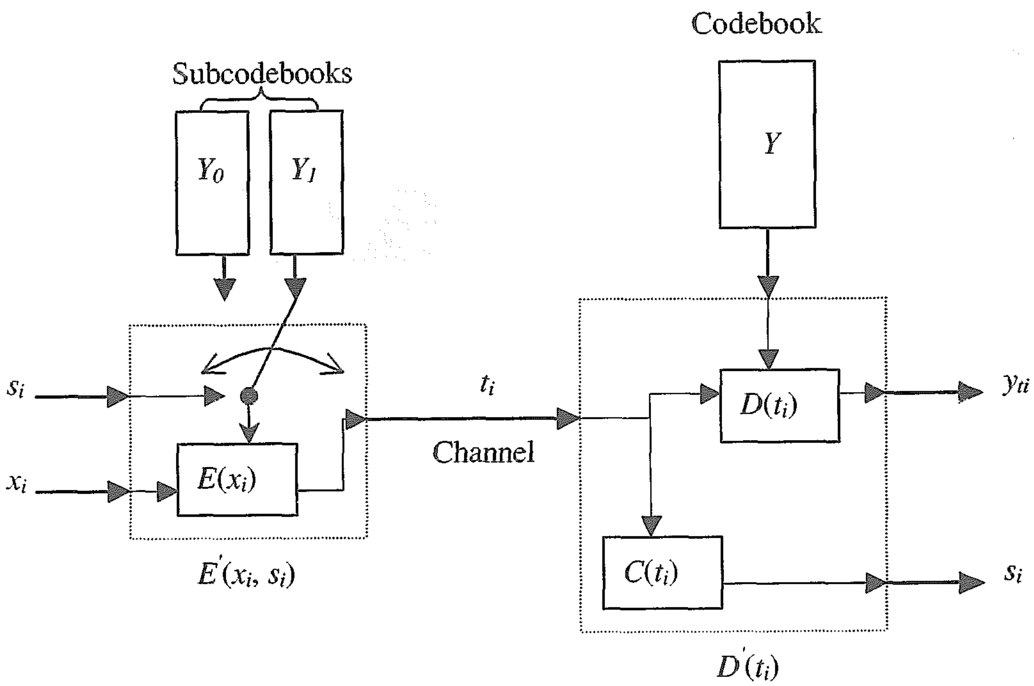


Figure 2. The schematic diagram of the proposed VQ watermarking scheme.

Figure 3. The host image F.

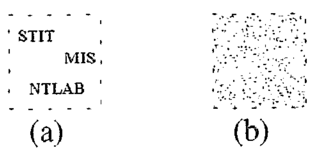


Figure 4. The watermark images (a) before random dispersion; (b) after random dispersion.

(a) (b)

Figure 5. (a) The watermarked image with bit rate 0.5 bpp and PSNR=28.7; and (b) The extracted watermark.


(a) (b)

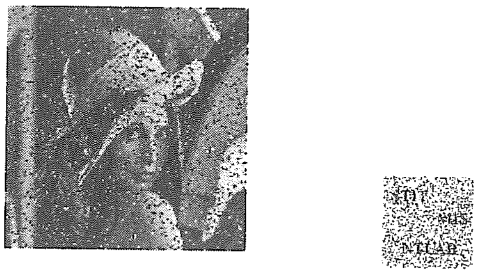

(c) (d)

Figure 6. The watermarked images that are corrupted by the channel noise with the bit error rate (a) BER=10^{-2}; (c) BER=$10^{-1.5}$ and the decoded watermark (b) and (d) from the corresponding received erroneous indices.

(a) (d)

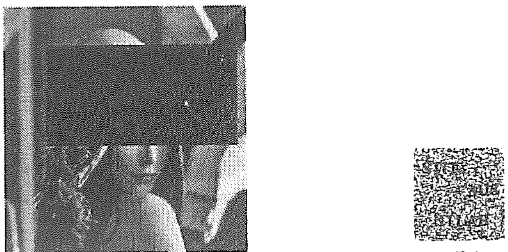

(b) (e)


(c) (f)

Figure 7. The results of JPEG compression attacks with (a) *Cr*=8.7 (b) *Cr*=16.3 (c) *Cr*=21.3; th extracted watermarks (d)-(f), each of which is decoded from the image shown at its left side.

(a) (b)

Figure 8. (a) the result after lowpass filtering the watermarked image; (b) the watermark image extracted from the image shown in (a).



(a) (b)

Figure 9. (a) A cropped, watermarked image; (b) the extracted watermark image from the image shown in (a).

(a) (b)

Figure 10. (a) the rescaled, watermarked image; (b) the watermark decoded from the image shown in (a).

Table 1. Illustration of VQ watermarking noise in terms of PSNR.

Codebook size	With watermark	Without watermark	Watermarking noise
128	27.4 dB	29.4 dB	2.0 dB
256	28.7 dB	30.4 dB	1.7 dB
512	29.3 dB	31.2 dB	1.9 dB
1024	30.0 dB	32.0 dB	2.0 dB