

Wavelet-Based Initialization for VQ Codebook Generation

CHIN-CHEN CHANG^{1,2}, YI-PEI HSIEH², WEN-CHUAN WU²

¹*Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 40724, Taiwan*

E-mail: *ccc@cs.ccu.edu.tw*

²*Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi 62145, Taiwan*

E-mail: *{hsiehyp,wenn}@cs.ccu.edu.tw*

Abstract

In this paper, a wavelet-based initialization method is presented for codebook generation in vector quantization image compression. The discrete wavelet transform that keeps the maximum characteristics of blocks is put in use to produce the initial codebook of LBG algorithm and, in the meantime, to hasten the convergence rate of the final codebook. Experimental results showed that the proposed algorithm performs well when compared to the previous codebook generation method, and its final codebook is provided with representative codevectors which lead to high output image quality.

Key words: image compression, vector quantization, discrete wavelet transform

1. Introduction

Compression is an indispensable coding routine for many forthcoming digital devices, such as personal computer, digital camera, media player, and so on. This coding operation is able to minimize the data volume for low data rate transmission and storage [3], especially for digital images and videos. Considering the imperfect human vision system, moreover, images are mostly processed and compressed in a noninvertible or lossy fashion, in which redundancy among data or less perceivable information are removed and discarded. By this way, digital images can be greatly spread at a smaller size without abruptly visually distortions. Over the last decade, many lossy compression techniques have been widely investigated [2,3,4]. Vector quantization encoding algorithm [7,8], "VQ" for short, is one of these celebrated techniques. Unlike scalar quantization, which encodes the inputting data individually, VQ is an efficient block-based quantization method. That is to say, this method puts a group of k samples in source data together, treats them as a block (or vector), and encodes the data collectively. However, there are so many variations in k -sized blocks. In order to achieve a low bit-rate, its quantizer is developed by condensing the set of all possible blocks into a smaller subset that minimizes the average distortion.

In mathematics, a simple VQ encoder is defined as a mapping function Q of k -dimensional Euclidean space R^k into a finite subset C of R^k . In other words, each probable vector B in space R^k will be homologized into the most similar one in subset C such that $Q(B) = c_i$, where $C = \{c_i \mid i = 1, 2, \dots, N\}$ is the representative codebook and N is the number of codevectors in C . Certainly, a non-overlapping block x_i divided from a source image X can be replaced with the best matched vector c_{bm} that requires conforming to the following equation:

$$ED(x_i, c_{bm}) = \min_{1 \leq j \leq N} \{ED(x_i, c_j)\},$$

where $ED(A, B) = \|A - B\|^2 = \sum_{i=1}^k [a_i - b_i]^2$ is the Euclidean distance between k -dimensional vectors $A = (a_1, a_2, \dots, a_k)$ and $B = (b_1, b_2, \dots, b_k)$. Then, after applying the mapping procedure to each block in X , the encoder only keeps the indices of the optimally matched codevectors in C instead of the whole image. In such way, a source block volume of $k \times \lceil \log_2 256 \rceil$ -bit will be greatly reduced to only $\lceil \log_2 N \rceil$ bits required. Intuitively, the VQ decoder simply works by using these indices as addresses to retrieve the corresponding codevectors from the same codebook C as that in the encoder. By appropriately piecing them together, finally, another image is thereby reconstructed which is extremely similar to the original one X . Taking all things into consideration, VQ coding technique is not only simple but also provided with excellent rate-distortion performance for digital images. While the compression ability of VQ relates to the given codebook size, the quality of the reconstructed image is greatly dependent upon the representative of codevectors in C . Consequently, a good VQ quantizer performs at a good codebook, which is able to minimize the average distortion among all possible blocks in R^k . How to, but, derive such a representative codebook efficiently and promptly?

The earliest VQ codebook design algorithm was developed by Linde, Buzo, and Gray in [8], known as the LBG iterative tuning (clustering) algorithm. In order to construct an ideal codebook, LBG algorithm iteratively tunes and improves the seed vectors step by step until a local minimum is achieved. Unfortunately, this work often consumes excessive convergence time. The rate of convergence of the LBG algorithm and the resultant codebook performance greatly depend on the choice of initial seed vectors. A set of applicable initial vectors will bring less rounds in the codebook convergence, and it can also produce excellent interpretative codevectors in the final codebook [5]. In this paper, we attempted to develop a fine initialization based on wavelet for the LBG codebook generation algorithm. Because discrete wavelet transform features many advantages such as joint spatial-spectral locality, multi-resolution analysis, signal compression and clustering [9], it is suitable for most applications in image processing including feature extraction. Considering these properties of discrete wavelet transform, we further exploits it to achieve reduction of the complexity of training vectors by de-correlating the resolution and, at the same time, hold the compact energy

distribution of vectors itself for a large quantity of training data. Naturally, a smaller complicated dataset is easy to select good codevectors out as the initial one. This paper is organized as follows. Section 2 briefly reviews the Linde-Buzo-Gray codebook design algorithm. Section 3 introduces the operation of discrete wavelet transform and then describes the details of the proposed initialization algorithm based on wavelet. And further, Section 4 shows and explains the experimental results to put the proposed method to the proof. Finally, conclusions are summarized in Section 5.

2. Linde-Buzo-Gray Iterative Algorithm (LBG)

During VQ coding procedure, the adopted codebook plays a decisive role in determining the compression ratio of source data and the quality of the encoded image. In general, the VQ codebook is derived from LBG iterative clustering algorithm, which was proposed by Linde et al. [8]. The details of this codebook design algorithm are described as follows.

In order to generate a VQ codebook of N k -dimensional codevectors, firstly, a training data set with numerous k -dimensional blocks is acquired by selecting five standard natural images and dividing them into non-overlapping blocks. Secondly, this system randomly picks N blocks out from the given training set as the codevectors of the initial codebook. According to this initial codebook, the system operates to group these training blocks into N clusters by the nearest-neighbour rule. Thirdly, another N codevectors are obtained by evaluating each cluster's centroid, and then they are formed into a new codebook. By repeating this clustering process above, we can iteratively improve the N codevectors of the codebook until a small enough fractional change is achieved in distortion improvement between two consecutive clustering steps. The corresponding flowchart of the LBG clustering process mentioned above is presented in Fig. 1.

Briefly, the LBG algorithm performs an iterative optimization to the chosen initial codebook for yielding a final VQ codebook. However, the action of random guess in the initiation selection is so irregular that sometimes this system consumes a considerable time for tuning because of non-dispersed distribution. It is noted that the convergence rate of the LBG iterative algorithm greatly depends on the choice of the initial codebook [5]. A good initial codebook can reduce the number of tuning iterations that require complicated computation, and avoid falling into the locally optimal as possible. In order to show clearly the defect of random guess, here, we display two simple examples about poor initial conditions in Fig. 2. If an initial codevector is far away from the other training vectors as Fig. 2(a), such codevector will not be excluded from the N candidates according to the LBG clustering rule because all training vectors are nearer to other codevectors. At the end of iterations, it still exists in the final codebook. But, this singular codevector might be useless for the reduction of distortion errors. In another poor case shown in Fig. 2(b), the initial codevectors chosen are mostly in the clusters

with small members. This situation also makes less contribution to the reduction of distortion errors. Perhaps more tuning processes in the LBG clustering will be required. Consequently, a good initiation selection is extremely important for the future codebook generation.

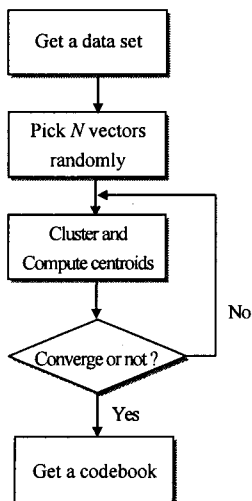
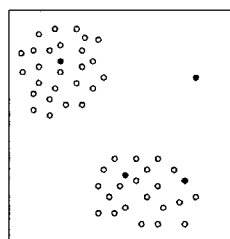
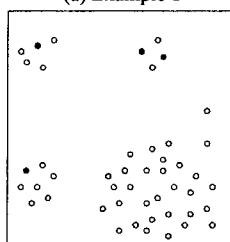


Fig. 1 Flowchart of LBG clustering algorithm



(a) Example 1



(b) Example 2

Fig. 2 Poorly initial codevectors, where the solid circles are the initial codevectors and the hollow circles are the training vectors

3. The Proposed Method

In order to improve the original LBG algorithm, in this section, we shall present a novel wavelet-based initialization method instead of random guess. The proposed method includes two phases: the wavelet feature construction and the codebook generation phases. Further details about the two phases are presented in Subsections 3.1 and 3.2, respectively.

3.1 The Wavelet Feature Construction Phase

In the first phase, the goal is to reduce dimensional complexity of all the training data for easier acquisition of the initial codebook. Because the discrete wavelet transform has the property of signal decomposition [6,10], we use it here as a tool of data simplification and feature extraction. Discrete wavelet transform is a mathematical technique that transforms signal data from the spatial domain into the frequency domain [1,6,10]. In practice, this transformation is achieved by using two kinds of signal filters and a down-sampling operation. In which, one of

the two filters is low pass filter and the other is high pass filter. For one-dimensional (1-D) wavelet transform, its low frequency coefficients can be obtained when signal data come through the low pass filter and then down-sample by two. Similarly, the high frequency coefficients can also be derived when signal data come through the high pass filter and down-sample by two.

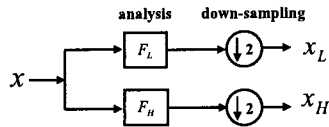


Fig. 3 The flowchart of 1-D wavelet transform

Fig. 3 shows the flowchart of 1-D wavelet transform. Assume that x is the inputting sequence of signal data, F_L represents the low pass filter, and F_H represents the high pass filter. After transforming sequence x , the final results yielded in the frequency domain are x_L and x_H , respectively. As for 2-D signal data such as digital images, they can be decomposed by applying 1-D wavelet transform to two orientations, the rows of an image and the columns of an image. Given an $m \times n$ image and let $P_{i,j}$ be the pixel value at the i -th row and the j -th column. Then, the transformed wavelet coefficient $P'_{i,j}$ would be evaluated by the following formulas.

Row: $P'_{i,j} = (P_{i,2j-1} + P_{i,2j})$, $P'_{i,\frac{n}{2}+j} = (P_{i,2j-1} - P_{i,2j})$, where $1 \leq i \leq m$ and $1 \leq j \leq n/2$.

Column: $P'_{i,j} = (P_{2i-1,j} + P_{2i,j})$, $P'_{\frac{m}{2}+i,j} = (P_{2i-1,j} - P_{2i,j})$, where $1 \leq i \leq m/2$ and $1 \leq j \leq n$.

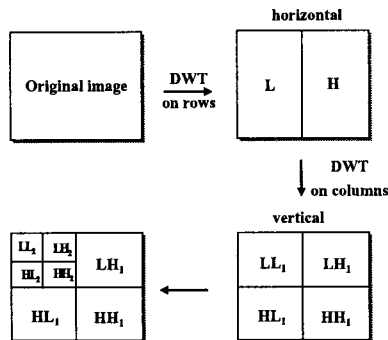


Fig. 4 The sketch map of two-level 2-D wavelet decomposition

When one-level 2-D wavelet transform is completed, an image would be decomposed into four sub-bands labeled LL_1 , LH_1 , HL_1 , and HH_1 . Here, LL_1 is the low frequency sub-band, HH_1 is the high frequency sub-band, and LH_1 as well as HL_1 are the middle frequency sub-bands. If a smaller and finer-scale wavelet sub-band is needed, the same 2-D wavelet transform can repeatedly work on the

scaling sub-band LL_1 . Fig. 4 is the diagram of two-level 2-D wavelet transform, where sub-bands LL_2 , LH_2 , HL_2 , HH_2 , LH_1 , HL_1 , and HH_1 are the two-level decomposed results of the original image.

Here, we summarily list some important properties [6] of the wavelet transform technique as below.

- (1) These sub-bands acquired from an image have different importance. The low frequency sub-band is regarded as an approximation part of the source image; contrarily, the middle and high frequency sub-bands are the detailed parts of the image. Thus, the low frequency sub-band is more important than the others.
- (2) All the energy of an image is compacted into low frequency sub-band. This fact means that the low frequency sub-band holds the maximum characteristics of an image.
- (3) After decomposing an $m \times n$ image by l -level wavelet, the size of sub-band LL_l would be $(m/2^l) \times (n/2^l)$. This sub-band size is far smaller than that of the source image.

Obviously, these properties indicate that LL_l is extremely suitable as the feature of source data. Not only the complexity of source data is simplified but also the original characteristic of data still survives. For this reason, we consider exploiting this sub-band to easily build a good initial codebook. For data fidelity, here, we strongly suggest that only one-level 2-D wavelet transform is adopted. Fig. 5 is the generation process of the proposed initial codebook based on the wavelet transform. First of all, we choose some standard image as the training one. All the training images are partitioned into a great number of non-overlapping blocks of k pixel values and each block is rearranged to form a k -dimensional vector. Second, 2-D wavelet transform is triggered to decompose each training vector into four sub-bands by using one-level operation. Here, we only adopt the LL_1 sub-band of each vector as the corresponding simplified feature. So, these low frequency sub-bands are collected together to construct a new training set, and the new set size is $k/4$ times the size of the source dataset.

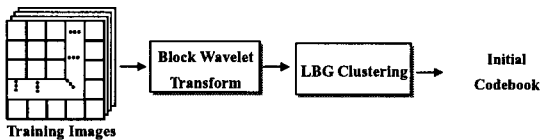


Fig. 5 The generation of the proposed initial codebook

Third, the new training set is performed to produce N codevectors by using the traditional LBG iterative clustering algorithm. Finally, the N codevectors are organized to an initial codebook. Due to that the new training dataset size is very small, the tuning cost in the LBG clustering process would be relatively less. In addition, the simplified data in $k/4$ dimensions possess the representative so that the number of clustering iterations is also small. Hence, the proposed initial

codebook generation would not consume too much time. In order to have a better understanding about data simplification, an example of one-level block wavelet transform is displayed in Fig. 6. Assume that an image block is 4×4 . In this case, one-level 2-D wavelet transform is first applied to the rows of the block and then to the columns. The frequency coefficients are finally presented in sub-bands LL_1 , LH_1 , HL_1 , and HH_1 as Fig. 6. It is noted that the coefficients in LL_1 are selected and rearranged to become a 4-dimensional vector (i.e. [68, 43, 75, 52]) to be a part of the new training dataset.

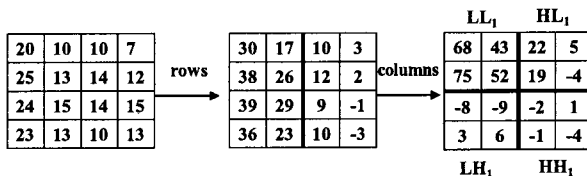


Fig. 6 An example of one-level block wavelet transform

3.2 The Codebook Generation Phase

In the second phase, the goal is to exploit the initial codebook yielded by the foregoing phase to efficiently produce the final VQ codebook, shown in Fig. 7. However, the current initial codebook is only temporary, so it is not directly applied to the LBG iterative algorithm. There are two reasons for this status. The first reason is that each codevector in the initial codebook is composed of frequency coefficients not the spatial pixels; the second reason is that each codevector dimension is $k/4$ not k . Therefore, we develop a mapping inverse wavelet transform that transforms frequency coefficients in LL_1 sub-band into spatial pixel values.



Fig. 7 The generation of the proposed final codebook

First, we must have a mapping table composed of many low frequency sub-bands. Actually, this table can be derived in the wavelet feature construction phase. Each k -dimensional training vector in the proposed first phase is decomposed to four sub-bands. At this time, the LL_1 sub-band of each vector can be collected in this mapping table. Hence, this mapping table has the same amount of samples as the original training data. After that, when a $k/4$ -dimensional codevector in the initial codebook inputs, the mapping inverse wavelet transform works to find the most similar sub-band in the aforesaid mapping table. And each sub-band in the table directs to the original un-decomposed image block. So, the found image block will be used to be the inverse one of the inputting initial codevector. By repeating this mapping process above until N $k/4$ -dimensional

codevectors in the initial codebook are all performed, finally, we will derive a formal initial codebook with N k -dimensional codevectors. To produce the final codebook, the formal initial codebook is iteratively improved by the LBG tuning clustering algorithm.

4. Experimental Results and Discussions

In this section, some experiments will be conducted to demonstrate the performance of our proposed initialization method. The experimental results are divided into two parts: the first part is to evaluate the convergence speed of the final codebook by the proposed wavelet-based initialization method, and the second part is to measure the quality of the encoded image in order to know if the resultant codebook is representative. The details of results of the two parts are described in Subsections 4.1 and 4.2, respectively.

4.1 Evaluation of the Convergence Rate of the Final Codebook

In this part, all of the experiments were performed on the Wintel platform with the programming language Matlab. We used two sets of test images for the codebook generation; one is named Set-I and the other is named Set-II. The Set-I consists of five standard gray-scale images: "Lena", "Baboon", "GoldHill", "Pepper", and "Toy" as shown in Fig. 8. The Set-II also includes five standard gray-scale images: "Jet (F14)", "Boat", "Babara", "Zelda", and "Tiffany" as demonstrated in Fig. 9.

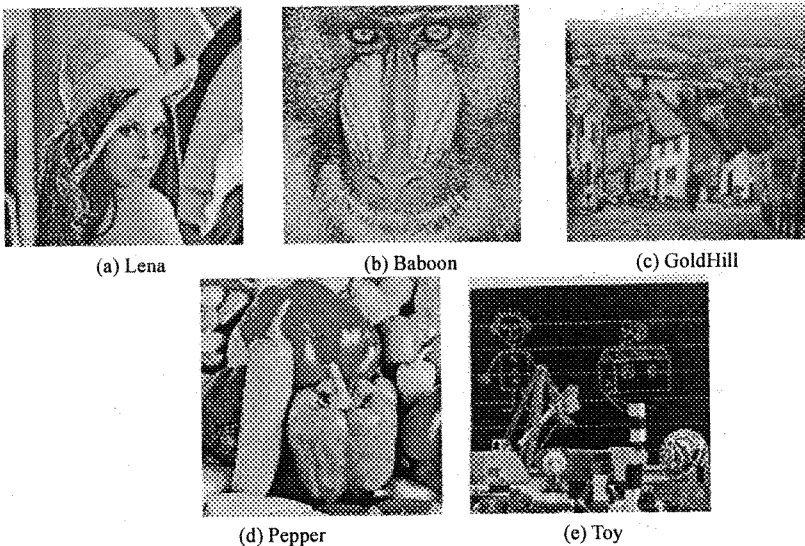


Fig. 8 Training images in Set-II

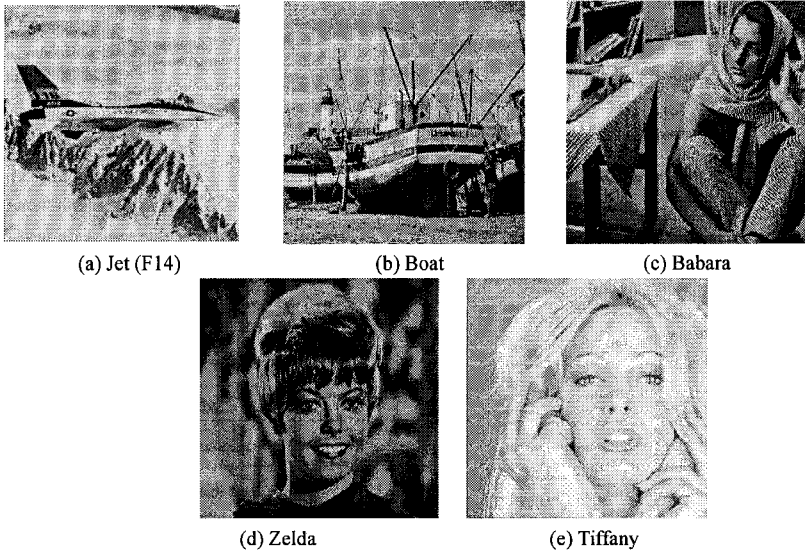


Fig. 9 Training images in Set-II

In the first simulation, the above test gray-scale images have a resolution of 256×256 pixels. Firstly, we segmented these test images into a great number of non-overlapping image blocks, each of which is 4×4 pixels (i.e. $k = 16$). Then, the LBG algorithm and our proposed initialization method, which executes one-level forward and backward wavelet transforms, were used to generate codebooks sized 256 and 512 to test the speed of convergence. The experimental results are revealed in Tables 1 and 2, where Table 1 is for Set-I and Table 2 is for Set-II. In the two tables, Phase 1 of the proposed method represents the process using 4-dimensional vectors to produce the initial codebook, and Phase 2 of the proposed method represents the process using the initial codebook to yield the final VQ codebook. From two tables, it is shown that the utilization of wavelet property in Phase 1 can reduce the computational complexity of the training blocks and generate more representative initial codebook, where the yielded initial codebook leads to a smaller number of clustering iterations for the final codebook generation of Phase 2.

Table 1 Performance comparison in the number of iterations by using training images of Set-I

Training data		Size of training images: 256×256	
		$N = 256$	$N = 512$
Methods			
	LBG algorithm	25	20
Our method	Phase1	3	3
	Phase2	17	19

Table 2 Performance comparison in the number of iterations by using training images of Set-II

Methods	Training data	Size of training image: 256×256	
		$N = 256$	$N = 512$
LBG algorithm		23	22
Our method	Phase1	3	3
	Phase2	18	17

In the second simulation, the two sets Set-I and Set-II were applied again to the LBG algorithm and the proposed method, where the used test images are in a resolution of 512×512 pixels. And the experimental results are presented in Tables 3 and 4. From the two tables, it can also be proven that the proposed wavelet-based initialization indeed facilitates the convergence rate of LBG clustering algorithm and reduces the computational cost.

Table 3 Performance comparison in the number of iterations by using training images of Set-I

Methods	Training data	Size of training image: 512×512	
		$N = 256$	$N = 512$
LBG algorithm		27	23
Our method	Phase1	3	2
	Phase2	17	20

Table 4 Performance comparison in the number of iterations by using training images of Set-II

Method	Training data	Size of training image: 512×512	
		$N = 256$	$N = 512$
LBG algorithm		26	22
Our method	Phase1	3	2
	Phase2	22	19

4.2 Quality Measurement of the Encoded Images

In this part, all of the experiments are performed on the Wintel platform with the programming language C++. In general, the criterion of peak signal to noise ratio (PSNR) can be applied to evaluate the quality of the encoded images. Here, hence, we adopted PSNR measurement to state the difference between the original image and its encoded counterpart. The equation of PSNR is listed as $PSNR = 10 \times \log_{10} (255^2/MSE)$. Here, the mean square error (MSE) for an $m \times n$ image is defined as $MSE = 1/(m \times n) \sum_{i=1}^m \sum_{j=1}^n (x_{ij} - y_{ij})^2$, where x_{ij} and y_{ij} denote the original and the encoded pixel values, respectively.

In the first simulation, the final codebooks, that were trained from the test images in Set-I, of sizes 256 and 512 are used to compress some gray-scale images by using VQ encoding algorithm [7,8]. Then, we evaluated the PSNR value

between these compressed images and their corresponding original images. The experimental results of the encoded images are listed in Tables 5 and 6, where Table 5 is for codebook 256 and Table 6 is for codebook 512. According to the two tables, the fact shows that our proposed method produces better PSNR than the LBG algorithm.

Table 5 Performance comparison in image quality (PSNR) with codebook sized 256

256 × 256 image	Toys	Sailboat	Zelda	Average
LBG codebook	29.258	25.056	31.47	28.594
Our method	29.262	25.134	31.504	28.633

Table 6 Performance comparison in image quality (PSNR) with codebook sized 512

256 × 256 image	Toys	Sailboat	Zelda	Average
LBG codebook	30.172	25.495	32.212	29.293
Our method	30.295	25.572	32.214	29.360

Table 7 Performance comparison in image quality (PSNR) with codebook sized 256

512 × 512 image	Toys	Sailboat	Zelda	Average
LBG codebook	26.295	25.041	31.372	27.569
Our method	26.444	25.096	31.461	27.667

Table 8 Performance comparison in image quality (PSNR) with codebook sized 512

512 × 512 image	Toys	Sailboat	Zelda	Average
LBG codebook	27.036	25.459	32.438	28.311
Our method	27.377	25.481	32.467	28.442

In the second simulation, we compressed some gray-scale images by using the codebooks sized 256 and 512, where the two codebooks were trained from the test images in Set-II. Next, we also calculated their PSNR value and listed the results in Tables 7 and 8, respectively. The experimental results as demonstrated in two tables show that our proposed method also acquires better PSNR value than the LBG algorithm. Based on the above experimental results, in sum, it is proven that our proposed method outperforms the LBG algorithm in terms of convergence speed for the training codebook and the PSNR performance for the encoded image quality.

5. Conclusions

A simple and efficient codebook design algorithm for vector quantization image compression is proposed in this paper. In order to speed up the iteration optimization procedure of the conventional LBG, the proposed method exploits the

wavelet properties to reduce the computational complexity of the training blocks. Then, a good initial codebook is acquired from these reduced blocks. Finally, the resultant initial codebook is iteratively improved to become an eventual codebook. According to the experimental results, it is proven that the proposed method requires less convergence cost than the conventional LBG algorithm. Moreover, its initialization holding specific characteristics of blocks brings forth the representative codebook that leads to a better PSNR performance.

References

- [1] Antonini, M., and Barlaud, M., "Image Coding Using Wavelet Transform," *IEEE Transformations on Image Processing*, Vol. 1, No. 2, Apr. 1992, pp. 205-220.
- [2] Cardinal, J., "Fast Fractal Compression of Grayscale Images," *IEEE Transactions on Image Processing*, Vol. 10, No. 1, Jan. 2001, pp. 159-164.
- [3] Chang, C. C., Chuang, J. C., and Chung, C. Y., "Quadtree-Segmented Image Compression Method Using Vector Quantization and Cubic B-Spline Interpolation," *The Imaging Science Journal*, Vol. 52, No. 2, Jun. 2004, pp. 106-116.
- [4] de Queiroz, R. L., and Fleckenstein, P., "Very Fast JPEG Compression Using Hierarchical Vector Quantization," *IEEE Transactions on Signal Processing Letters*, Vol. 7, No. 5, May 2000, pp. 97-99.
- [5] Feng, J., and Lo, K. T., "Feature-Based Initialisation for Image Vector Quantisation Codebook Generation," *Electronics Letters*, Vol. 36, No. 24, Nov. 2000, pp. 2005-2006.
- [6] Graps, A., "An Introduction to Wavelets," *IEEE Computational Science and Engineering*, Vol. 2, No. 2, 1995, pp. 50-61.
- [7] Gersho, A., and Gray, R. M., "Vector Quantization and Signal Compression," Kluwer Academic Publishers, Boston US, 1992.
- [8] Linde, Y., Buzo, A., and Gray, R. M., "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, Vol. 28, No. 1, Jan. 1980, pp. 84-95.
- [9] Martin, K., Lukac, R., and Plataniotis, K. N., "Efficient Encryption of Wavelet-Based Coded Color Images," *Pattern Recognition*, Vol. 38, No. 7, Jul. 2005, pp. 1111-1115.
- [10] Usevitch, B. E., "A Tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000," *IEEE Signal Processing Magazine*, Vol. 18, No. 5, Sept. 2001, pp. 22-35.