OPTO-ELECTRONICS REVIEW **20**(2), 187–193

DOI: 10.2478/s11772-012-0016-z

Improved vector quantization scheme for grayscale image compression

Y.-C. HU*[1], W.-L. CHEN[1], C.-C. LO[2], and J.-C. CHUANG[3]

[1]Department of Computer Science and Information Management, Providence University, 200 Chung Chi Rd., Taichung 43301, Taiwan, R.O.C.
[2]Department of Computer Science and Information Engineering, Providence University, 200 Chung Chi Rd., Taichung 43301, Taiwan, R.O.C.
[3]Department of Computer Science & Communication Engineering, Providence University, 200 Chung-Chi Rd., Shalu, Taichung County 43301, Taiwan, R.O.C.

This paper proposes an improved image coding scheme based on vector quantization. It is well known that the image quality of a VQ-compressed image is poor when a small-sized codebook is used. In order to solve this problem, the mean value of the image block is taken as an alternative block encoding rule to improve the image quality in the proposed scheme. To cut down the storage cost of compressed codes, a two-stage lossless coding approach including the linear prediction technique and the Huffman coding technique is employed in the proposed scheme. The results show that the proposed scheme achieves better image qualities than vector quantization while keeping low bit rates.

Keywords: image compression; vector quantization, LBG algorithm; linear prediction, Huffman coding.

1. Introduction

Vector quantization (VQ) was introduced by Linde, Buzo, and Gray in 1980 [1–3]. It is a commonly used scheme for a grayscale image compression. Besides, it can be applied to the compression of speech. Basically, VQ consists of three procedures: codebook design, image encoding, and image decoding. The main goal of the codebook design procedure is to generate a set of representative codewords. The set of representative codewords is also called the codebook. The codebook is then used in the image encoding/decoding procedures.

The LBG algorithm is the most popular algorithm for designing the codebook of N codewords [1]. In the LBG algorithm a set of training images is selected for a codebook design. Each training image is divided into non-overlapped image blocks of $n \times n$ pixels. Then, the initial codebook consisting of N codewords is chosen. Several rounds of the vector clustering process and the centroid updating process are executed in order to design the codebook. In the vector clustering process, each training vector is classified as a group corresponding to its closest codeword in the current codebook. In the centroid updating process, the mean vector of training vectors in each group is calculated. These calculated mean vectors form the updated codebook.

Two possible termination conditions of the LBG algorithm can be used. One condition is that the LBG algorithm is terminated when the predefined number of rounds is reached. The other is that the LBG algorithm is stopped when the designed codebooks in two successive rounds tend to be stable. The performance of the codebooks in two successive rounds is stable when the relation distortion change is less than or equal to the predefined threshold. The relation distortion change for the i^{th} round can be calculated by using the following equation

$$RDC_i = \left| \frac{D_{i-1} - D_i}{D_i} \right|. \tag{1}$$

Here, D_i and D_{i-1} denote the sum of the squared Euclidean distance incurred in $_i^{th}$ and $(i–1)^{th}$ rounds of the LBG algorithm, respectively.

In the image encoding procedure, the given grayscale image is first divided into a set of non-overlapped image blocks of $n \times n$ pixels. Each image block is sequentially processed in the left-to-right and top-to-bottom order. The closest codeword in the codebook for each image block is searched and the corresponding index is stored. In the image decoding procedure, the same codebook that was used in the image encoding procedure is stored and used. In order to rebuild each compressed block, the codeword in the codebook corresponding to extracted index is employed to recover the compressed block.

In general, a great deal of a computational cost is consumed in the LBG algorithm for a codebook design. According to the literature, some improved algorithms based on the LBG algorithm have been proposed to cut down the computational cost or to design a better codebook [4–7]. Besides, some fast codebook search algorithms for VQ have

*e-mail: ychu@pu.edu.tw, wlchen@pu.edu.tw

been proposed to accelerate the image encoding procedure of VQ without incurring any extra image distortion [8–11].

The index of each VQ-compressed block is stored in $\log_2 N$ bits with the codebook of N codewords. In other words, the required bit rate of VQ is $(\log_2 N)$/nxn bpp. To reduce the bit rate of VQ, some methods aiming to compress losslessly the index table of a VQ-compressed image have been proposed [12–15]. In addition, some methods sacrificing the quality of a VQ-compressed image to reducing the bit rate have also been proposed [16–21].

Based on the study of VQ, we observe that the image quality of a VQ-compressed image is poor when the codebook size is small. To solve this problem, an improved image coding scheme based on VQ is proposed in this paper. The rest of the paper is organized as follows. In Sect. 2, the proposed scheme based on VQ is introduced. In Sect. 3, the experimental results and discussions are presented. Finally, some conclusions are given in Sect. 4.

2. Proposed scheme

The aim/purpose of the proposed scheme is to improve the image quality of the VQ scheme with a small-sized codebook while keeping a low bit rate. To achieve this aim, the encoding of an image block by its mean value of pixels cooperates with the VQ scheme.

2.1. Image encoding procedure

Suppose the codebook $CB = cw_0, cw_1, \ldots, cw_{N-1}$of N codewords was previously designed. The codebook CB is sorted by the sum values of the codewords, so that neighbouring codewords had high degree of similarity. The flowchart of the proposed image encoding procedure is depicted in Fig. 1. The given grayscale image of $W \times H$ pixels is divided into a set of non-overlapped image blocks of $n \times n$ pixels. There are $w \times h$ image blocks to be processed where $w = W/n$ and $h = H/n$. Each image block can be viewed as a k-dimensional vector where $k = n \times n$.

In order to encode each image block x, two possible encoding rules including the VQ scheme and the use of the block mean value are tested. We need to determine the best way to encode each image block. Firstly, the closest codeword for x in the codebook is searched and its index idx is stored. Besides, the squared Euclidean distance between x and its closest codeword cw_{idx} is stored in $dist_{vq}$.

Then, we need to find out the squared Euclidean distance when x is encoded by its block mean value. The mean value $\bar{x}$ of the pixels in x is computed. Then, the squared Euclidean distance $dist_{mean}$ when x is encoded by its block mean value is calculated by the following equation

$$dist_{mean} = \frac{1}{k}\sum_{i=1}^{k}(x_i - \bar{x}). \qquad (2)$$

If $dist_{vq}$ is less than or equal to $dist_{mean}$, x is encoded by VQ and the index idx of the closest codeword in the codebook is stored. Otherwise, x is encoded by its mean value and the *mean* value mean is stored. To record the block type of each image block, 1-bit indicator *type* is stored according to the following rule

$$type = \begin{cases} 0 & \text{if } x \text{ is encoded by} VQ \\ 1 & \text{if } x \text{ is encoded by its block mean value} \end{cases}. \qquad (3)$$

When each image block is sequentially processed by the above mentioned process, the block types and the compressed codes consisting of the indices and the block mean values are generated.

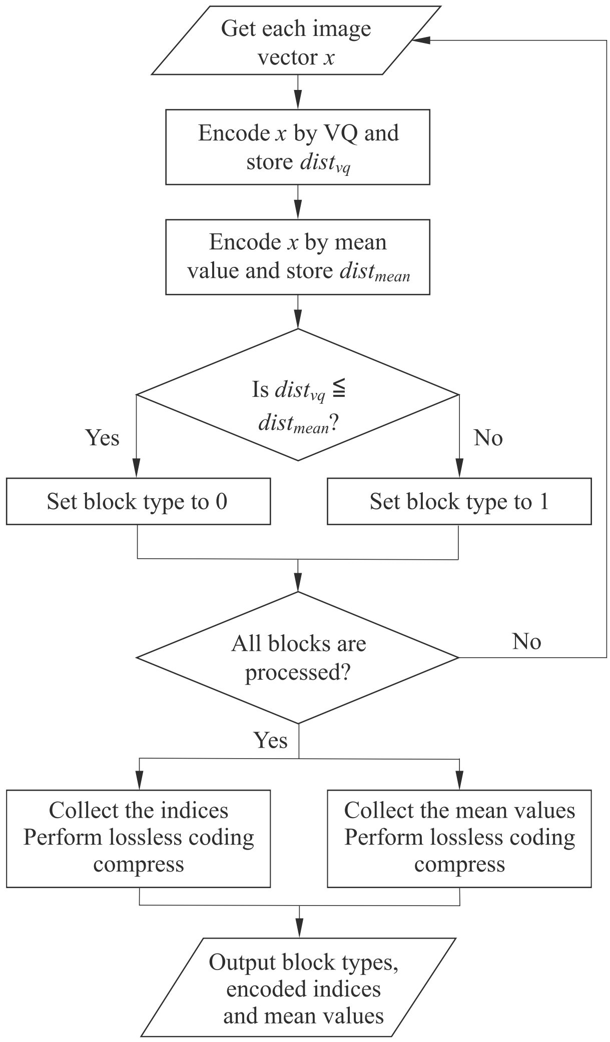


Fig. 1. Flowchart of proposed image encoding procedure.

0	0	0	1	1
0	0	1	1	0
1	1	1	0	1
1	1	0	0	1
0	1	0	1	1

(a) Block types

i_{12}	i_{13}	i_{12}	118	120
i_{10}	i_{11}	116	117	i_{18}
112	114	115	i_{19}	124
113	114	i_{20}	i_{21}	125
i_{20}	113	i_{20}	124	125

(b) Compressed codes

Fig. 2. Image encoding example of proposed scheme.

 © 2012 SEP, Warsaw

Figure 2 shows an example of the encoded results of the proposed scheme. To distinguish the index from the block mean value, the index with the value j is represented by i_j. For example, i_{12} denotes the index of the 12th codeword in the codebook.

In order to cut down the storage cost of the compressed codes further, a two-stage lossless coding approach including the linear prediction technique and the Huffman coding technique are employed. Here, the block mean values and the indices are processed separately, because they have different possible values.

We recall that the compressed code of each image block x may be either the index or the block mean value $\bar{x}$. Before a lossless coding of the indices is executed, we need to transform all the mean values in the compressed codes into the indices, so that the linear prediction technique could be started. Each block mean value in the compressed codes is replaced by the corresponding index of the codeword that has the closest sum values of pixels within it. An example of such transformation using the compressed codes as it is shown in Fig. 2(b) is listed in Fig. 3(a). The shaded indices indicate that they are translated from the block mean values.

Similarly, in order to encode the mean values in the compressed codes losslessly, we need to transform the indices found in the compressed codes into the mean values. The transformation can be easily done by replacing each index value by the block mean value of its corresponding codeword in the codebook. An example of the index for the mean value transformation using the compressed codes as it is shown in Fig. 2(b) is listed in Fig. 3(b). The shaded mean values indicate that they are translated from the indices.

i_{12}	i_{13}	i_{12}	I_{19}	i_{20}
i_{10}	i_{11}	i_{17}	I_{18}	i_{18}
i_{12}	i_{15}	i_{16}	I_{19}	i_{20}
i_{14}	i_{15}	i_{20}	I_{21}	i_{21}
i_{20}	i_{14}	i_{20}	I_{20}	i_{21}

(a) Mean value to index transformation

112	113	112	118	120
108	109	116	117	117
112	114	115	119	124
113	114	124	125	125
124	113	124	124	125

(b) Index to mean value transformation

Fig. 3. Example of compressed codes transformation.

After the two-way transformation of the indices and mean values has been done, the indices and the block mean values are processed by the linear prediction technique. Here, D denotes the data to be processed. Besides, L and U denote the adjacent left and up neighbours of D, respectively. Three prediction functions used in the proposed scheme are listed in the following

$$f_1(D) = L, \tag{4}$$

$$f_2(D) = U, \tag{5}$$

$$f_3(D) = (L + D)/2. \tag{6}$$

An example of the linear prediction technique for the block mean values is described as follows. There are 11 indices and 14 block mean values found in the example depicted in Fig. 2. For processing the block mean values by the linear prediction technique, the index of the mean value transformation is executed to generate Figure 3(b). Suppose the linear prediction function mentioned in Eq. (4) is used. The results of the linear prediction are listed in Fig. 4. There are 14 prediction errors in this example. These prediction errors can then be encoded by the Huffman coding technique to generate the compressed results.

			6	2
		7	1	
4	2	1		5
1	1			0
	–11		0	1

Fig. 4. Prediction errors of the mean values.

The compressed codes for the indices are generated after encoding losslessly the indices by the linear prediction technique and the Huffman coding technique. Similarly, the mean values are encoded by the two-stage lossless compression. The compressed result of the image consists of the block types of $w \times h$ bits, the encoded indices, and the encoded mean values.

2.2. Image decoding procedure

In order to rebuild the compressed image of $W \times H$ pixels, the same codebook of N codewords that is used in the image encoding procedure is stored. Besides, the prediction function and the Huffman coding tables for the indices and mean values are needed to recover the image. The received compressed codes consist of the block types of $w \times h$ bits, the encoded indices, and the encoded mean values. We need to recover the indices and the mean values, so that $w \times h$ image blocks could be reconstructed. The flowchart of the proposed image decoding procedure is depicted in Fig. 5.

In order to recover the indices, the encoded indices are first processed by the Huffman decoding procedure to generate prediction errors. Then, these indices' prediction errors are processed by the reverse linear prediction procedure to generate the indices. Similarly, the encoded mean values are processed by the Huffman decoding procedure and followed by the reverse linear prediction procedure.

After the indices and the mean values of all the image blocks are recovered, each image block can be rebuilt by performing the following process. First, the block type indicator *type* is extracted. If the *type* is equal to $(0)_2$, the index of $\log_2 N$ bits is extracted and the corresponding codeword in the codebook is used to rebuild the image block. If the *type* is equal to $(1)_2$, the 8-bit block mean value is extracted and each pixel in the image block is replaced by the mean value. The compressed image of the proposed scheme can be reconstructed by recovering sequentially each image block by the above-mentioned steps.

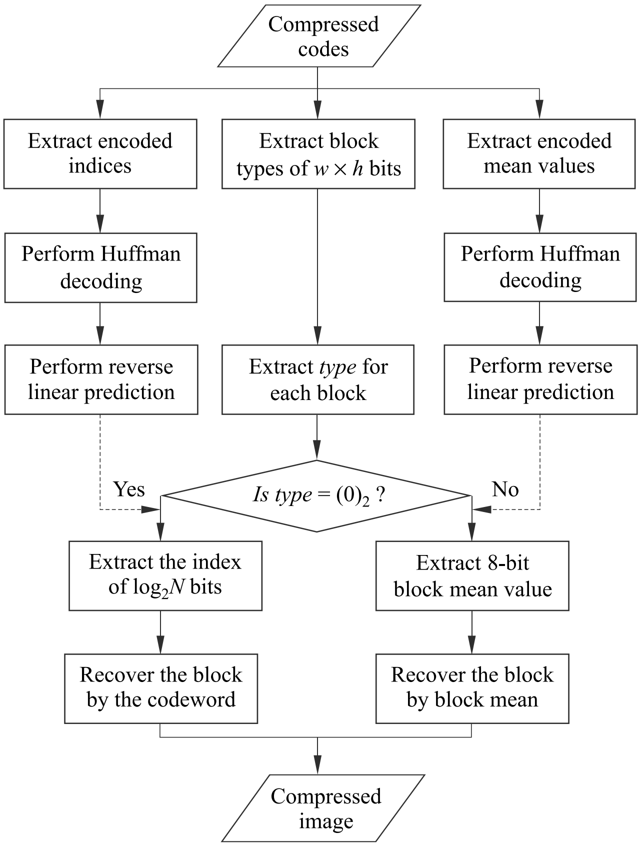


Fig. 5. Flowchart of proposed image decoding procedure.

3. Experimental results

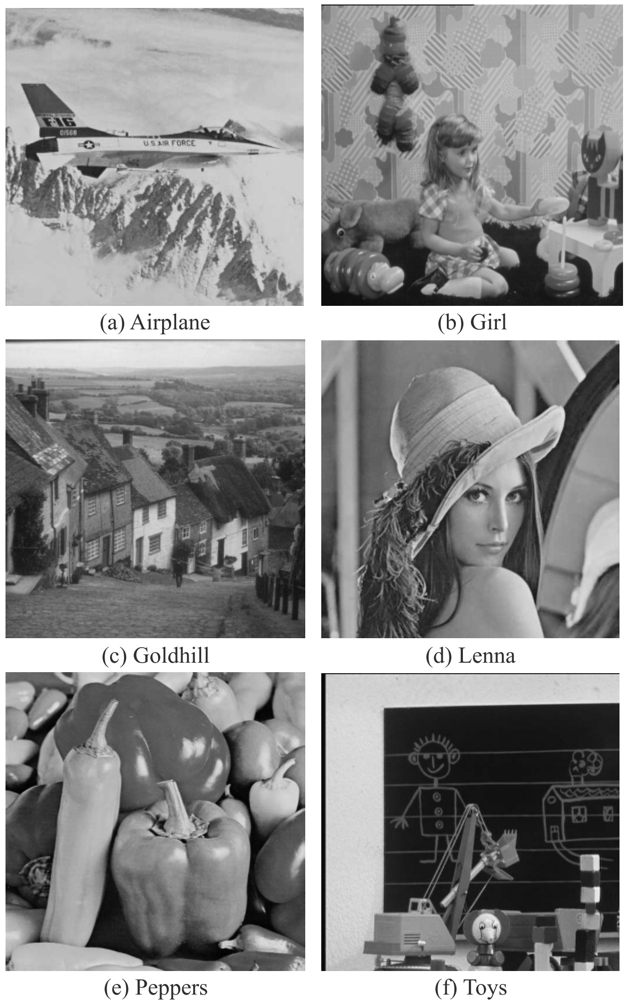
All the simulations were performed on the IBM compatible PC with a Pentium IV 3G Hz CPU and 1G RAM. Four grayscale images "Airplane", "Boat", "Goldhill" and "Toys" of 512×512 pixels were used as the training images for designing the VQ codebooks by using the accelerated LBG algorithm [10]. In the simulations, the termination threshold of the LBG algorithm was set to 0.001.

Six test images "Airplane", "Girl", "Goldhill", "Lenna", "Peppers" and "Toys" of 512×512 pixels as shown in Fig. 6 are selected to evaluate the performance of the vector quantization scheme and the proposed scheme. Three of them are inside the training set of the codebook design. In the codebook design procedure, each training image is divided into non-overlapped 4×4 image blocks. There are 16384 image blocks in each training image.

In the simulations, two image quality measurements are used. The first measurement is the peak signal-to-noise-ratio (PSNR), which is defined as

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE}. \tag{7}$$

Here, MSE denotes the mean square error (MSE) between the original and the reconstructed images of $W \times H$. Basically, *PSNR* is considered as an indication of an image quality rather than a definitive measurement. However, it is a commonly used measurement for evaluating the image quality.



(a) Airplane (b) Girl (c) Goldhill (d) Lenna (e) Peppers (f) Toys

Fig. 6. Grayscale testing images.

The second one is the structural similarity index measurement (SSIM) [22]. SSIM was proposed by Wang *et al.* in 2004 for the measuring of the perceptual similarity between two images. It is designed to improve the traditional methods like PSNR and MSE, which have been proved to be inconsistent with the human eye perception. The structural similarity index measurement of two signals *x* and *y* is defined as

$$SSIM(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}. \tag{8}$$

Here, μ_x and μ_y denote the mean intensities of *x* and *y*, respectively. Besides, σ_x and σ_y denote the standard deviation of *x* and *y*, respectively. σ_{xy} denotes the covariance coefficient which corresponds to the cosine of the angle between the vectors $x - \mu_x$ and $y - \mu_y$. In addition, c_1 and c_2 are two predefined constant values. In the simulations, the 8×8 square window is used and it moves pixel-by-pixel over the entire image to calculate SSIM.

 © 2012 SEP, Warsaw

The reconstructed image qualities of VQ using different codebook sizes are shown in Table 1. It is shown, that the image quality increases as the codebook size increases. The average image qualities of 26.502 dB, 29.205 dB, and 31.167 dB are obtained by VQ with the codebooks of size 16, 64 and 256, respectively. When the size of the codebook is less than or equal to 64, the average image quality of VQ is smaller than 30 dB.

In addition to the PSNR measurement, the SSIM value of each original image and its compressed image of VQ are also listed. According to the suggestions in Ref. 22, c_1 and c_2 used in Eq. (8) are set to $(0.01 \times 255)^2$ and $(0.03 \times 255)^2$, respectively. It is shown, that the SSIM value increases as the codebook size increases. The average SSIM values of 0.728, 0.819, and 0.871 are obtained by VQ with the codebooks of sizes 16, 64 and 256, respectively.

The PSNR values and the SSIM values of the proposed scheme using different codebook sizes are listed in Table 2. Similarly, the image quality increases as the codebook size increases in the proposed scheme. The average image qualities of 27.534 dB, 29.556 dB, and 31.321 dB are obtained by the proposed scheme with the codebooks of size 16, 64, and 256, respectively. The gains of the image quality compared to the results of VQ are 1.032 dB, 0.351 dB, and 0.154 dB when the codebook sizes equal 16, 64, and 256, respectively. The image quality gain of the proposed scheme decreases as the codebook size increases. In addition to that, the average SSIM values of 0.812, 0.846, and 0.882 were obtained by the proposed scheme with the codebook sizes 16, 64, and 256, respectively.

Table 3 lists the results of the percentage of the image blocks that are encoded by the block mean values in the proposed scheme. It is shown that 85.180%, 67.855%, and 45.349% image blocks are encoded by the block mean values in the proposed scheme with the codebooks of size 16, 64, and 256, respectively. The total number of the image blocks that are encoded by the mean values decreases as the codebook size increases in the proposed scheme.

Table 1. Results of image qualities of VQ.

N / Images	16		32		64		128		256	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Airplane	25.983	0.765	27.800	0.812	29.315	0.863	30.648	0.891	31.450	0.906
Girl	27.414	0.707	28.412	0.755	29.413	0.793	30.429	0.835	31.245	0.863
Goldhill	26.277	0.671	27.692	0.717	28.892	0.772	29.780	0.802	30.587	0.832
Lenna	26.785	0.725	27.943	0.775	29.121	0.816	30.195	0.847	30.972	0.867
Pepper	26.783	0.730	28.245	0.784	29.476	0.826	30.289	0.843	31.145	0.867
Toys	25.772	0.770	27.449	0.795	29.014	0.845	30.428	0.869	31.604	0.890
Average	26.502	0.728	27.924	0.773	29.205	0.819	30.295	0.848	31.167	0.871

Table 2. Experimental results of image qualities of proposed scheme.

N / Images	16		32		64		128		256	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Airplane	26.759	0.837	28.290	0.861	29.502	0.880	30.755	0.899	31.515	0.911
Girl	28.607	0.785	29.096	0.797	29.728	0.814	30.612	0.846	31.357	0.870
Goldhill	27.725	0.774	28.459	0.766	29.339	0.798	30.099	0.819	30.787	0.841
Lenna	27.979	0.831	28.753	0.843	29.608	0.859	30.493	0.872	31.186	0.885
Pepper	27.956	0.838	28.998	0.851	29.943	0.863	30.718	0.874	31.372	0.884
Toys	26.177	0.805	27.825	0.828	29.218	0.863	30.610	0.882	31.709	0.898
Average	27.534	0.812	28.570	0.824	29.556	0.846	30.548	0.865	31.321	0.882

Table 3. Percentage of image blocks encoded by block mean values in proposed scheme.

N / Images	16	32	64	128	256
Airplane	86.761%	79.132%	66.742%	49.573%	36.938%
Girl	85.748%	78.534%	68.994%	50.586%	38.013%
Goldhill	84.808%	77.655%	66.467%	56.219%	44.373%
Lenna	87.036%	82.599%	74.213%	65.747%	55.914%
Pepper	87.836%	81.439%	73.969%	66.711%	54.340%
Toys	78.888%	66.986%	56.744%	60.138%	42.517%
Average	85.180%	77.724%	67.855%	58.162%	45.349%

Experimental results of the bit rates of VQ and the proposed scheme with different lossless procession strategies are listed in Table 4. Here, PM-0 and PM-HC denote the proposed scheme without any post processing and with Huffman coding, respectively. In addition, PM-TS1, PM--TS2, and PM-TS3 denote the proposed scheme with the two-stage lossless coding using the linear prediction functions $f_1()$, $f_2()$, and $f_3()$ defined in Eqs. (4) to (6), respectively.

Table 4. Required bit rates of VQ scheme and proposed scheme with different lossless processing strategies.

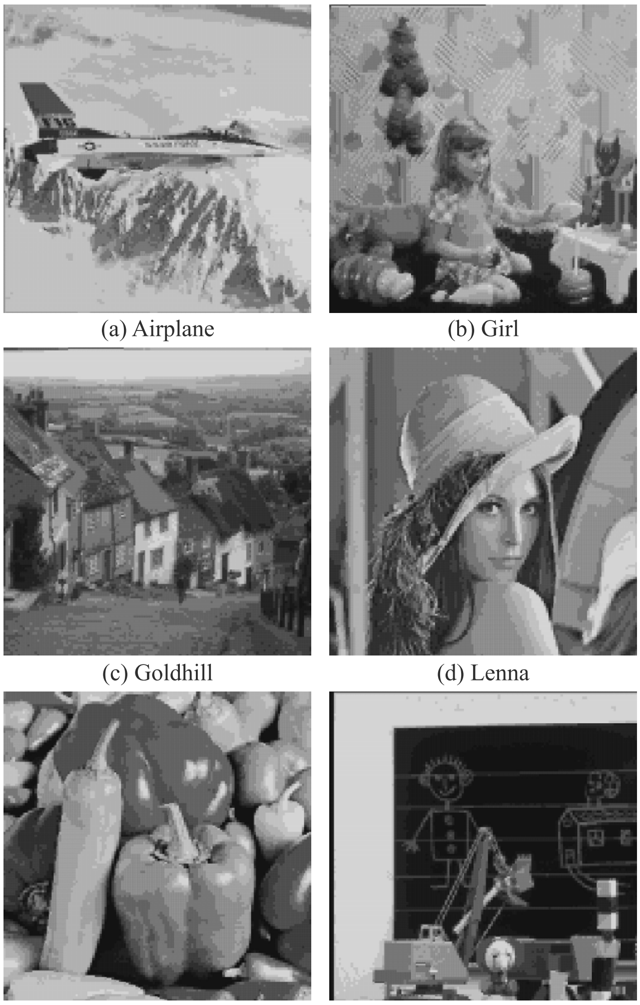
Images \ *N*	16	32	64	128	256
VQ	0.250	0.3125	0.375	0.4375	0.500
PM-0	0.525	0.522	0.522	0.536	0.563
PM-HC	0.472	0.465	0.465	0.459	0.493
PM-TS1	0.365	0.361	0.363	0.379	0.403
PM-TS2	0.363	0.358	0.359	0.376	0.401
PM-TS3	0.355	0.351	0.354	0.372	0.398

According to the results, PM-0 required the highest bit rates. The proposed scheme with Huffman coding, PM-HC, slightly reduces the bit rates compared to PM-0. But, it provides worse performance than the proposed scheme with the two-stage lossless compression. The proposed scheme with two-stage lossless compression consumes less bit rates than VQ when the codebook sizes are greater than or equal to 64. According to the results, PM-TS3 achieves the best performance. The average bit rates of 0.355 bpp, 0.354 bpp, and 0.398 bpp are needed in PM-TS3 with the codebooks of size 16, 64, and 256, respectively.

According to Tables 1 and 2, it is obvious that the proposed scheme improves the image quality of the compressed image when a small-sized codebook is used. The average image qualities of 26.502 dB and 27.534 dB are obtained by VQ and by the proposed scheme when the codebook of size 16 is used, respectively.

Compressed images of VQ and those of the proposed scheme with the codebook of 16 codewords are shown in Figs. 7 and 8, respectively, for better understanding of the visual difference between the compressed images of VQ and the proposed scheme. According to the results in Fig. 7, the false contour effect can be easily found in the VQ-compressed images. For example, the clouds in the sky in Fig. 7(a) are visually different from the original image as it is shown in Fig. 6(a). In fact, the false contour effect can be found in the backgrounds of the VQ-compressed images. It also appears in the objects of the VQ-compressed images with continuous intensities. In other words, the visual qualities of the VQ-compressed images are quite poor.

However, the false contour effect cannot be found in the compressed images of the proposed scheme even when a codebook of 16 codewords is used. Compared to the results shown in Fig. 7, the visual qualities of the compressed images of the proposed scheme are much better than those of VQ-compressed images. In other words, the proposed scheme not only improves the image qualities, but also provides better visual qualities of the compressed images compared to the VQ scheme.



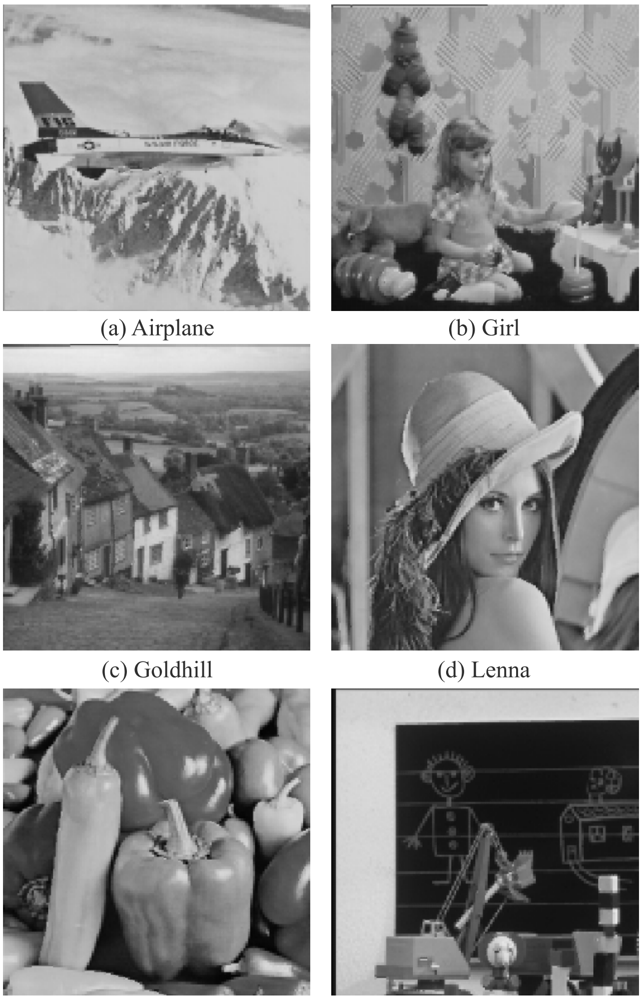
(a) Airplane (b) Girl (c) Goldhill (d) Lenna (e) Peppers (f) Toys

Fig. 7. Reconstructed images of VQ with codebook of 16 codewords.

4. Conclusions

The improved image compression scheme based on the VQ scheme is proposed in this paper. In this scheme, the block encoding using the block mean value cooperates with the VQ scheme. Besides, a two-stage lossless compression approach is designed to cut down the consumed bit rates. According to the results, more than 1-dB image quality gain is achieved in the proposed scheme when the codebook of size 16 is used. Compared to the VQ scheme, the required bit rates are less than those of the VQ scheme when the codebook sizes are greater than or equal to 64. In addition to that, the visual qualities of the compressed images by the proposed scheme are much better than those of the VQ scheme. Furthermore, the false contour effect cannot be found in the compressed images of the proposed scheme.

© 2012 SEP, Warsaw



(a) Airplane (b) Girl

(c) Goldhill (d) Lenna

(e) Peppers (f) Toys

Fig. 8. Reconstructed images of proposed scheme with codebook of 16 codewords.

References

1. A.Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design", *IEEE T. Commun.* **28**, 84–95 (1980).
2. N.M. Nasrabadi and R.A. King, "Image coding using vector quantization: a review", *IEEE. T. Commun.* **36**, 957–971 (1988).
3. A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1992.
4. C.C. Chang and Y.C. Hu, "A fast codebook training algorithm for vector quantization", *IEEE T. Consum. Electr.* **44**, 1201–1208 (1998).
5. P.Y. Tsai, Y.C. Hu, and H.L. Yeh, "Fast VQ codebook generation method using codeword stability check and finite state concept", *Fundam. Inform.* **87**, 447–463 (2008).
6. J.Z.C. Lai, Y.C. Liaw, and J. Liu, "A fast VQ codebook generation algorithm using codeword displacement", *Pattern Recogn. Lett.* **41**, 315–319 (2008).
7. C.W. Tsai, C.Y. Lee, M.C. Chiang, and C.S. Yang, "A fast codebook generation algorithm via pattern reduction", *Pattern Recogn. Lett.* **30**, 653–660 (2009).
8. Y.C. Hu and C.C. Chang, "An effective codebook search algorithm for vector quantization", *Imaging Sci. J.* **51**, 221–234 (2003).
9. Z. Pan, K. Kotani, and T. Ohmi, "Improved the-law-of-cosines-based fast search method for vector quantization by updating angular information", *Pattern Recogn. Lett.* **27**, 688–695 (2006).
10. Y.C. Hu, B.H. Su, and C.C. Tsou, "Fast VQ codebook search algorithm for grayscale image coding", *Image Vision Comput.* **26**, 657–666 (2008).
11. J.Z.C. Lai and Y.C. Liaw, "A novel encoding algorithm for vector quantization using transformed codebook", *Pattern Recogn. Lett.* **42**, 3065–3070 (2009).
12. C.H. Hsieh and J.C. Tsai, "Lossless compression of VQ index with search-order coding", *IEEE T.. Image Process.* **5**, 1579–1582 (1996).
13. Y.C. Hu and C.C. Chang, "Low complexity index-compressed vector quantization for image compression", *IEEE T.. Consum. Electr.* **45**, 1225–1233 (1999).
14. T.C. Lu and C.C. Chang, "Data compressor for VQ index tables", *Fundam. Inform.* **65**, 353–371 (2005).
15. C.C. Chang, G.M. Chen, and Y.C. Hu, "A novel index coding scheme for vector quantization", *Fundam. Inform.* **71**, 215–227 (2006).
16. Y.C. Hu and C.C. Chang, "Variable rate vector quantization scheme based on quadtree segmentation", *IEEE T. Consum. Electr.* **45**, 310–317 (1999).
17. Y.C. Hu and C.C. Chang, "Image coding schemes based on subsampling vector quantization", *Opt. Eng.* **40**, 108–114 (2001).
18. Y.C. Hu, C.C. Lin, and K.L, Chi, "Block prediction vector quantization for grayscale image compression", *Fundam. Inform.* **78**, 257–270 (2007).
19. Y.C. Hu, P.Y. Tsai, and C.C. Lo, "New bit reduction of vector quantization using block prediction and relatively addressing", *Fundam. Inform.* **87**, 313–329 (2008).
20. K. Sasazaki, S. Saga, J. Maeda, and Y. Suzuki, "Vector quantization of images with variable block size", *Appl. Soft Comput.* **8**, 634–645 (2008).
21. Y.C. Hu, J.C. Chuang, and C.C. Lo, "Efficient grayscale image compression technique based on VQ", *Opto-Electron. Rev.* **19**, 104–113 (2011).
22. Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity", *IEEE T. Image Process.* **13**, 600–612 (2004).