

## Improved Mean-Removed Vector Quantization Scheme for Grayscale Image Coding

Jun-Chou Chuang<sup>1</sup>, Yu-Chen Hu<sup>2</sup>, Chun-Chi Lo<sup>3</sup>,  
Wu-Lin Chen<sup>2</sup> and Chia-Hsien Wen<sup>2</sup>

<sup>1</sup>*Department of Computer Science and Communication Engineering,  
Providence University, 200 Chung Chi Rd., Taichung 43301, Taiwan*

<sup>2</sup>*Department of Computer Science and Information Management,  
Providence University, 200 Chung Chi Rd., Taichung 43301, Taiwan*

<sup>3</sup>*Department of Computer Science and Information Engineering,  
Providence University, 200 Chung Chi Rd., Taichung 43301, Taiwan*  
*lzchung@pu.edu.tw, ychu@pu.edu.tw, wlchen@pu.edu.tw, cclo@pu.edu.tw,  
chwen@pu.edu.tw*

### Abstract

*The mean-removed vector quantization (MRVQ) scheme achieves good reconstructed image quality, but it requires a high number of bit rates. In this paper, we propose an improved MRVQ scheme. The compressed codes of MRVQ for an image block contain the block mean and the index recording the closest residual vector in the codebook. In the proposed scheme, the block mean values are encoded by the linear prediction technique followed by the Huffman coding technique. The MRVQ indices of the residual vectors are further compressed by the Huffman coding technique. From the experimental results, it is shown that a great deal of bit rate reduction is achieved by using the proposed scheme with acceptable image quality loss.*

**Keywords:** *Image coding, vector quantization, mean-removed vector quantization, linear prediction, Huffman coding*

### 1. Introduction

Along with the rapid development of computer technology, more and more multimedia contents are distributed on the Internet. Most of the digital images on the Web are stored in compressed formats, such as JPEG and GIF, in order to save storage spaces and transmission cost. From the literature, vector quantization (VQ) [1-2] is a lossy image compression scheme for grayscale images. Because of its simple decoding structure, it is often used in the low computational power systems.

Typically, VQ consists of three procedures: codebook design, image encoding, and image decoding. The design of a good codebook is very important because the codebook plays an important role on the reconstructed image quality of VQ. Besides, the researches toward the fast codebook design algorithms [3-5] and the fast codebook search algorithm [6-8] are important. In general, the main drawback of VQ is that its required bit rate is high. To reduce the bit rate of VQ compression, the lossless index coding approach [9-12] and the lossy coding approach [13-15] had been proposed. Some VQ-based image coding schemes that improve the image qualities of the compressed images while keeping low bit rates had been proposed [16-17].

In addition, some modified image compression schemes based on VQ had been proposed. They are the finite-state vector quantization (FSVQ) [18-19], the side-match vector quantization (SMVQ) [20], the classified vector quantization (CVQ) [21-22], and the mean-removed vector quantization (MRVQ) [23-24]. Among these schemes, MRVQ is an extended version of the VQ scheme. The goal of MRVQ is to provide better image quality of the reconstructed image, but a higher bit rate is required in MRVQ compared to VQ. In VQ, each image block is compressed by the closest codeword in the codebook that was generated previously. To improve the compressed image of VQ, the block mean value of each image block is computed. Then, the residual block is generated by subtracting each pixel value from its block mean value. Then, the closest residual codeword in the residual codebook is searched. In other words, the compressed codes of MRVQ for each image block consists of the block mean value and the index recording the closest residual codeword in the residual codebook. Generally, MRVQ provides better reconstructed image quality; however, it requires a higher bit rate than VQ.

In this paper, we propose an improved image coding scheme based on MRVQ to cut down the bit rate of MRVQ while keeping a good image quality of the compressed image. In the proposed scheme, the storage cost of the block mean values is reduced by using the linear prediction technique and the Huffman coding technique. In addition, the Huffman coding technique is employed to cut down the storage cost of the indices of the residual vectors. The rest of this paper is organized as follows. We will review the vector quantization scheme and the mean-removed vector quantization scheme in Section 2. The proposed scheme will be presented in Section 3. The experimental results will be given in Section 4. Finally, conclusions will be presented in Section 5.

## 2. Related Works

In this section, we will first introduce the vector quantization (VQ) scheme for grayscale image compression. Then, the mean-removed vector quantization (MRVQ) scheme will be described.

### 2.1. The Vector Quantization Scheme

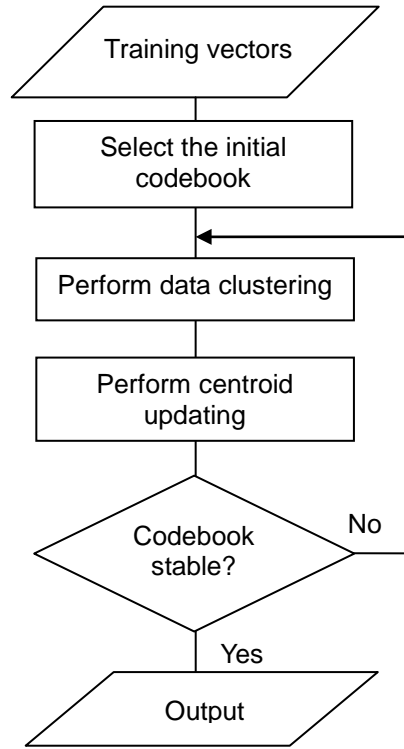
The vector quantization scheme is a lossy image compression scheme for grayscale images. In general, VQ consists of three procedures: codebook generation, image encoding, and image decoding. Basically, VQ can be defined as a mapping from a  $k$ -dimensional Euclidean space  $\mathbf{R}^k$  to a finite subset  $CB = \{cw_i \mid i = 0, 1, \dots, (N-1)\}$ . Here,  $CB$  denotes the codebook of  $N$  codewords. Besides,  $cw_i$  denotes the  $i$ -th codeword in the codebook.

The codebook generation procedure is very important in VQ. The reconstructed image quality of VQ compressed image can be achieved when a good codebook is used. From the literature, the LBG algorithm [1] is the most commonly used algorithm for codebook generation. The flowchart of the LBG algorithm is shown in Figure 1.

To generate the codebook  $CB$  of  $N$  codewords, we need to select some training images for codebook design. Suppose  $t$  training images of  $W \times H$  pixels are selected and each image is divided into non-overlapping image blocks of  $n \times n$  pixels. A total of  $t \times (W \times H) / (n \times n)$  image vectors are used for codebook design. In the LBG algorithm, we need to generate the initial codebook. The codewords of the initial codebook can be randomly selected from the training vectors. Then, several rounds of the vector clustering process and the centroid updating process are executed to generate the required codebook.

In the vector clustering process, the closest codeword in the codebook for each training vector is searched. Each training vector is then classified as the group corresponding to its

closest codeword in the codebook. After the vector clustering process is executed, all the training vectors are classified into  $N$  groups. An example of the vector clustering process is depicted in Figure 2. In this example, the training vectors numbered 0, 25, ..., and 7 are classified as group  $g(0)$ . Here,  $g(i)$  denotes the group which corresponds to the codeword  $cw_i$ . Similarly, the training vectors numbered 1, 33, 31, ..., and 46 are classified as group  $g(1)$ .



**Figure 1. Flowchart of the Codebook Design Procedure**

After the vector clustering process is executed, the centroid updating process can be executed. In the centroid updating process, the mean vector  $cw'_i$  of each group  $g(i)$  is calculated as follows:

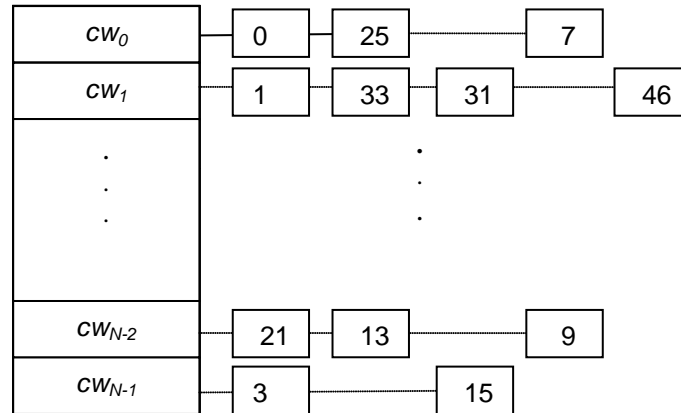
$$cw'_i = \frac{1}{|g(i)|} \sum_{x_j \in g(i)} x_j \quad (1)$$

Here,  $|g(i)|$  denotes the number of training vectors in  $g(i)$ . When all the mean vectors of these  $N$  groups are computed, the codebook of the current round is generated.

The vector clustering process and the centroid updating process are repeatedly executed for several rounds until the codebooks generated in two successive rounds are stable. Let  $D_{i-1}$  and  $D_i$  denote the total squared Euclidean distortion incurred in the  $(i-1)$ -st and  $i$ -th rounds, respectively. The LBG algorithm stops if the following condition is reached

$$\left| \frac{D_{i-1} - D_i}{D_i} \right| < \beta \quad (2)$$

where  $\beta$  is the predefined threshold.



**Figure 2. An Example of the Vector Clustering Process**

In the image encoding procedure of VQ, the original image sized  $W \times H$  pixels is divided into a set of non-overlapping image blocks of  $n \times n$  pixels, and each image block can be viewed as a  $k$ -dimensional vector, where  $k = n \times n$ . The closest codeword in the codebook for each input block  $x$  is to be searched. To find out the closest codeword in the codebook for  $x$ , the squared Euclidean distance is used. The squared Euclidean distance between  $x$  and  $cw_i$  can be computed as follows:

$$d(x, cw_i) = \sum_{j=0}^{k-1} (x_j - cw_{ij})^2, \quad (3)$$

where  $x_j$  denotes the  $j$ -th element of image vector  $x$ , and  $cw_{ij}$  denotes the  $j$ -th element of codeword  $cw_i$ . The codeword that has the minimal squared Euclidean distance from  $x$  is selected as the closest codeword and its index is recorded. Each index of the close codeword in the codebook is stored in  $\log_2 N$  bits.

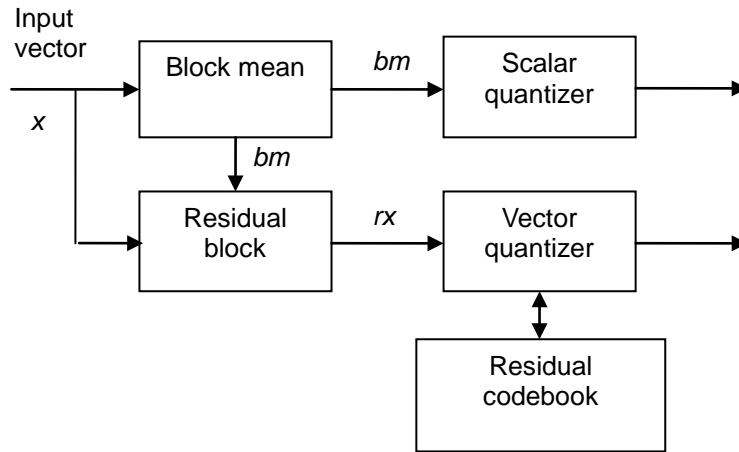
In the image decoding procedure, the same codebook of  $N$  codewords that was used in the image encoding procedure should be stored and used. To reconstruct the compressed image, the index of each compressed image block is sequentially extracted. Each index of  $\log_2 N$  bits is used to determine the closest codeword in the codebook for the current decoding block. Then, this block is recovered by the corresponding codeword. In other words, a simple table look-up operation is executed to rebuild the image block. After each image block is sequentially rebuilt by the same way, the whole compressed image can be then recovered.

## 2.2. The Mean-Removed Vector Quantization Scheme

The mean-removed vector quantization (MRVQ) is a modified version of the VQ scheme. The goal of MRVQ is to provide better image quality than VQ. To achieve the goal, the block mean value and the index of the residual vector are used to encode each image block in MRVQ. Therefore, the required bit rate of MRVQ is much higher than that of VQ.

Basically, the MRVQ scheme consists of three procedures: the residual codebook generation, image encoding, and image decoding. To generate the residual codebook RCB of  $N$  residual codewords for MRVQ, the LBG algorithm can be applied here. Instead of the training images used for VQ codebook design, the residual vectors are used for MRVQ

codebook design. Firstly, the training image blocks are generated by dividing the training images into non-overlapping image blocks of  $n \times n$  pixels. Each training image block of  $n \times n$  pixels can be viewed as a  $k$ -dimensional vector where  $k = n \times n$ . Each training vector is transformed to one residual vector by simply subtracting its block mean value from each pixel value. Then, the LBG algorithm described in Section 2.1 is employed to design the residual codebook  $RCB = \{rcw_0, rcw_1, \dots, rcw_{N-1}\}$  by using the residual vectors.



**Figure 3. The Block Diagram of the Image Encoding Procedure**

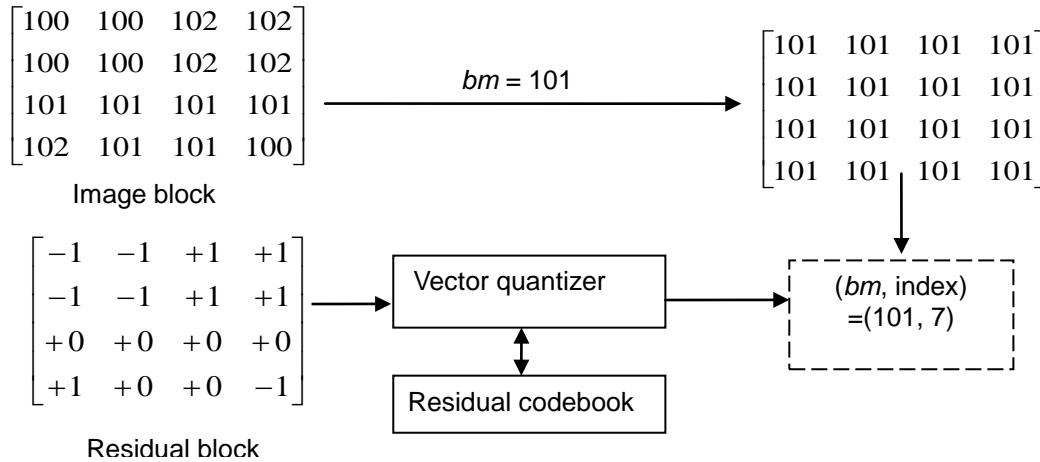
The block diagram of the image encoding procedure is depicted in Figure 3. In the image encoding procedure, the block mean value  $bm$  of each image block  $x$  is computed as follows:

$$bm = \frac{1}{k} \sum_{i=0}^{k-1} x_i \quad (4)$$

The residual vector  $rv$  is then generated by subtracting  $bm$  from each pixel value in  $x$ . The closest residual codeword in  $RCB$  for the residual vector  $rv$  is searched and the index of the closest residual codeword is recorded. In addition, the block mean value of each image block can be optionally encoded by lossy or lossless coding technique to cut down the bit rate.

The image decoding procedure is the reverse of the image encoding procedure. The residual codebook  $RCB$  that was used in the image encoding procedure is stored and used. The block mean value and the index of  $\log_2 N$  bits of each compressed image block are sequentially extracted. The extracted index is used to select the residual codeword in  $RCB$ . Then, each pixel in the recovered image block is generated by summing the corresponding residual value and the block mean. The entire compressed image can be recovered when all image blocks are sequentially rebuilt.

An image encoding example of MRVQ for the image block of  $4 \times 4$  pixels is depicted in Figure 4. The block mean value  $bm$  of the image block equals to 101. Then the residual block is generated by subtracting the block mean value from each pixel value. Suppose the residual block is encoded by 7<sup>th</sup> residual codeword in the residual codebook. The index with value 7 is stored in this example. The compressed codes of the image block are the block mean value 101 and the index numbered 7.



**Figure 4. An Encoding Example of MRVQ for the 4x4 Image Block**

### 3. The Proposed Scheme

MRVQ provides better reconstructed image quality than VQ. However, the required bit rate of MRVQ is much higher than that of VQ. In order to cut down the bit rate of MRVQ while keeping good image quality of the compressed image, an improved MRVQ scheme by using the linear predication technique and the Huffman coding technique is proposed. The mean values are compressed by the linear predication technique and then followed by the Huffman coding technique. Besides, the Huffman coding is applied to the compression of the residual indices. The proposed scheme consists of the image encoding procedure and the image decoding procedure. The flowchart of the proposed scheme is depicted in Figure 5.

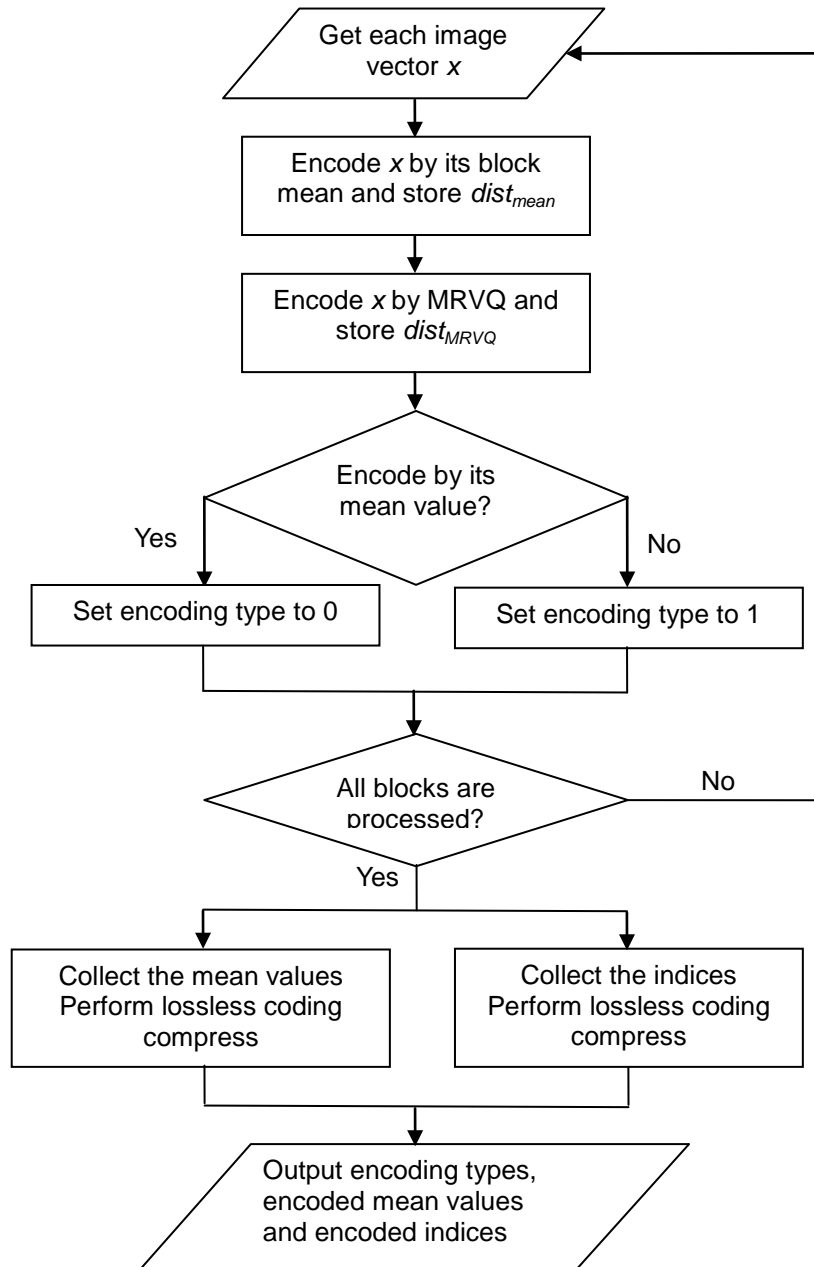
#### 3.1. Image Encoding Procedure

Suppose the residual codebook of  $N$  residual codewords had already been designed. To compress the grayscale image of  $W \times H$  pixels, it is first divided into non-overlapping blocks of  $n \times n$  pixels. Each image block  $x$  is sequentially processed in the order of left-to-right and top-to-bottom.

To compress each image block  $x$ , two possible approaches are tested. First,  $x$  is encoded by its block mean value  $bm$  and the incurred distance is stored in  $dist_{mean}$ . In addition,  $x$  is encoded by MRVQ and the incurred distance is stored in  $dist_{mrpq}$ . After the distances of these approaches are calculated, we now turn to decide the encoding type of  $x$ . The block type of  $x$  is set to 0 when either one of the following two test conditions is satisfied:

- (1) if  $dist_{mean} \leq dist_{mrpq}$ , the encoding type of  $x$  is set to 0, or
- (2) if  $(dist_{mean} > dist_{mrpq})$  and  $(dist_{mean} - dist_{mrpq} \leq TH)$ , the encoding type of  $x$  is set to 0.

Otherwise, the encoding type of  $x$  is set to 1. If the first test condition is satisfied, it indicates that the residual codeword searched in  $RCB$  is not a good candidate for encoding the residual vector. Therefore, this image block should be encoded by its block mean value. If the second test condition is satisfied, it indicates that the reconstructed image block of MRVQ is slightly better than that of the block mean value. However, the difference is not significant. To cut down the bit rates, a little image quality loss is sacrificed when the second test condition is applied.



**Figure 5. The Block Diagram of the Proposed Scheme**

To distinguish between the encoding types for each image block, 1-bit indicated is required. When each image block is sequentially processed, their encoding types are recorded. In addition to the 1-bit encoding type, the compressed codes of each image block may be either  $bm$  or  $(bm, index)$  when the image block is encoded by the block mean and the MRVQ scheme, respectively.

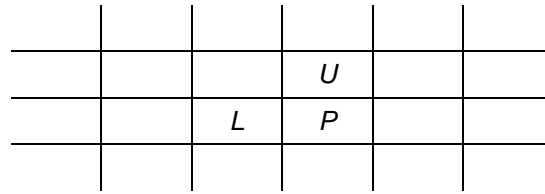
A total of 8 bits are needed to store the block mean value, and  $(8+\log_2N)$  bits are required to store the compressed codes of MRVQ. To further cut down the storage cost of the compressed codes, all the block mean values are collected together and will be further compressed by the two-stage lossless coding approach including the linear prediction

technique and the Huffman coding technique. In addition, the indices of the residual vectors for those image blocks with their encoding type equal to 1 are collected. These residual indices will be further compressed by the Huffman coding technique. Here, the block mean values and the indices are encoded separately because they have different possible values.

To losslessly compress the block mean values, the linear prediction technique is first employed to generate the prediction errors. The following prediction function is used in the proposed scheme to generate the predicted errors:

$$f(P)=(L+U) / 2. \tag{5}$$

In the prediction function, the input value  $P$  is predicted by the average value of its adjacent left value ( $L$ ) and upper value ( $U$ ). Figure 6 depicts the positional diagram for  $P$  and its adjacent neighbors.



**Figure 6. The Prediction Coding using  $U$ ,  $L$ , and  $(U+L)/2$**

There are three special cases without limiting the foregoing provision:  $P$  is located at pixel position  $(0, 0)$ ,  $P$  is in the first row, and  $P$  is in the first column. If  $P$  is located at  $(0, 0)$ , the constant value 128 is used to predict  $P$ . If  $P$  is in the first row, the left value ( $L$ ) of  $P$  is employed to predict it. Similarly, the upper value ( $U$ ) is used to predict  $P$  when  $P$  is in the first column.

An example of the linear prediction technique for the block mean values of  $4 \times 4$  is illustrated in Figure 7. Consider the pixel 126 located at  $(0, 0)$ ; its predicted value is 128 and the predicted error equals to 2 ( $128-126=2$ ). The predicted value for pixel 131 located at  $(1,1)$  equals to 119. The predicted error of this pixel equals to -12.

126	115	113	110
123	131	128	128
121	126	126	134
124	122	120	123

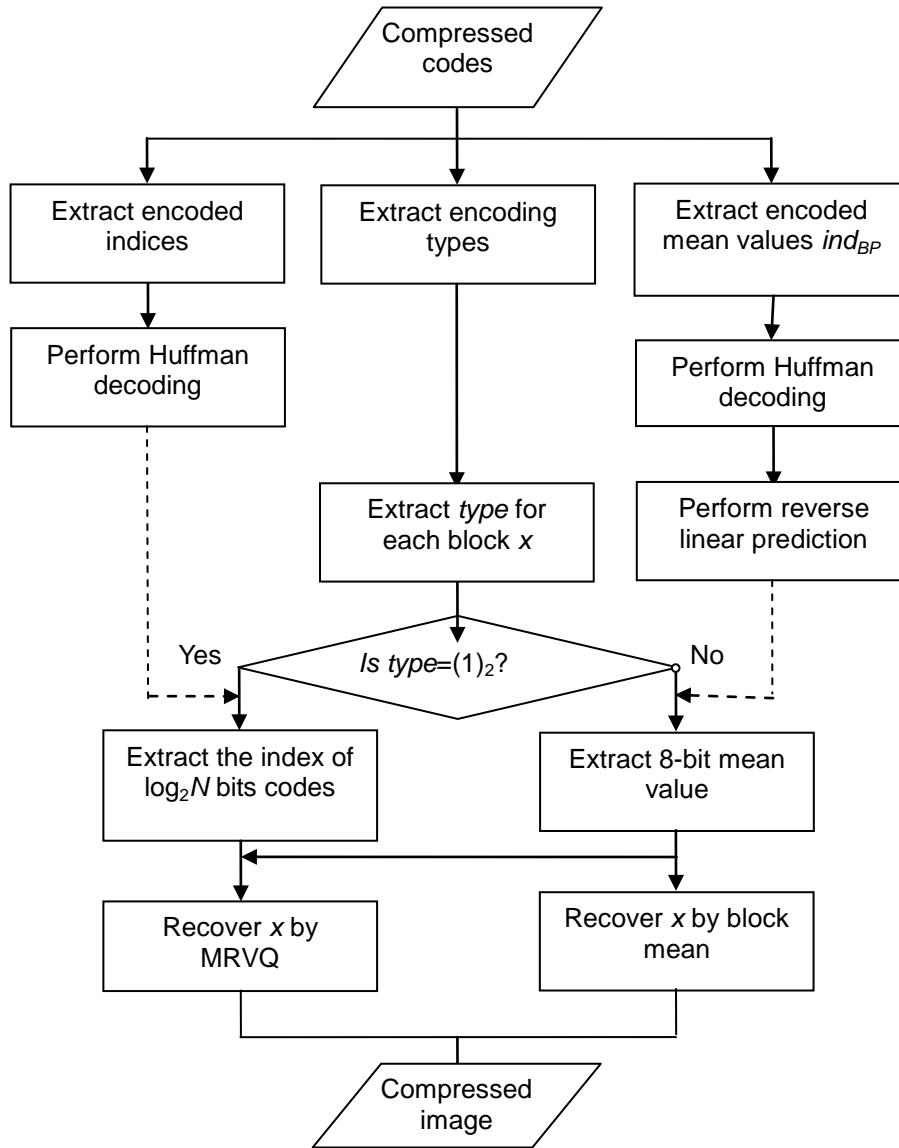
(a) Block means values

2	11	2	3
3	-12	-6	-9
2	0	1	-7
-3	3	4	4

(b) The predict errors

**Figure 7. An Example of the Linear Prediction Technique**





**Figure 8. The Block Diagram of our Decoding Procedure**

After the linear prediction technique is employed on the block mean values, the predicted errors are then generated. These predicted errors can then be encoded by the Huffman coding technique to generate the compressed results. To cut down the storage cost of the indices, the indices are collected together and compressed by the Huffman coding technique. The compressed result of the image consists of the encoding types, the encoded mean values, and the encoded indices.

### 3.2. Image Decoding Procedure

To rebuild the compressed image of  $W \times H$  pixels, the same residual codebook  $RCB$  of  $N$  codewords that was used in the image encoding procedure is stored. Besides, the prediction function and the Huffman coding tables for the mean values and the indices are needed to

recover the image. The received compressed codes consist of the block types, the encoded mean values, and the encoded indices. We need to recover the indices and the mean values so that the image blocks can be reconstructed. The flowchart of the proposed image decoding procedure is depicted in Figure 8.

To recover the mean values, the encoded mean values are processed by the Huffman decoding procedure to generate the predicted errors. Then, these predicted errors of the mean values are processed by the reverse linear prediction procedure to generate the block mean values. To recover the indices, the encoded indices are processed by the Huffman decoding procedure.

After the mean values and the indices are recovered, each image block can be rebuilt by performing the following process. First, the 1-bit encoding type for each block  $x$  is extracted. If the encoding type is equal to  $(0)_2$ , the 8-bit block mean value is extracted and each pixel in  $x$  is replaced by the mean value. If the encoding type is equal to  $(1)_2$ , the index of  $\log_2 N$  bits is extracted and the corresponding residual vector in the codebook is searched. Then, the MRVQ decoding procedure by using the residual vector and the block mean is used to recover  $x$ . By sequentially recovering each image block by the above-mentioned steps, the compressed image of the proposed scheme can be reconstructed.

#### 4. Experimental Results

Several simulations were performed to evaluate the validity of the proposed scheme. The six grayscale images of  $512 \times 512$  pixels are selected as the test images shown in Figure 9, such as “Airplane”, “Girl”, “Goldhill”, “Lenna”, “Peppers”, and “Toys”. The four training images, “Airplane”, “Boat”, “Goldhill”, and “Toys”, are used for codebook design. In the simulations, the vector dimension  $k$  of the image block is set to 16. In addition, the LBG algorithm is used to generate the VQ codebooks and the MRVQ codebooks of different sizes. In the simulations, the termination threshold of the LBG algorithm was set to 0.001.

In the simulations, the peak signal-to-noise-ratio (PSNR) measurement is used. The PSNR measurement is defined as

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \quad (6)$$

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 image quality rather than a definitive measurement; however, it is a commonly used measurement for evaluating the image quality.

**Table 1. Results of the Image Qualities of VQ**

Images \ $N$	16 (0.25 bpp)	32 (0.3125 bpp)	64 (0.375 bpp)	128 (0.4375 bpp)	256 (0.5 bpp)
Airplane	25.983	27.800	29.315	30.648	31.450
Girl	27.414	28.412	29.413	30.429	31.245
Goldhill	26.277	27.692	28.892	29.780	30.587
Lenna	26.785	27.943	29.121	30.195	30.972
Pepper	26.783	28.245	29.476	30.289	31.145
Toys	25.772	27.449	29.014	30.428	31.604
<b>Average</b>	<b>26.502</b>	<b>27.924</b>	<b>29.205</b>	<b>30.295</b>	<b>31.167</b>

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. 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.



**Figure 9. Test Images of 512×512 Pixels**

Reconstructed image qualities of MRVQ using different codebook sizes are shown in Table 2. It is shown that the image quality increases as the codebook size increases. Average image qualities of 30.543 dB, 32.375 dB, and 34.094 dB are obtained by MRVQ with the codebooks of size 16, 64 and 256, respectively. The average image qualities of MRVQ are greater than 30 dB. Nevertheless, the required bit rate of MRVQ is much higher than that of VQ when the same codebook size is used.

Reconstructed image qualities of the proposed scheme with different codebook sizes are shown in Table 3. The control threshold values are set to 0, 50, 100, ..., 300. It is shown that the average image quality decreases as the threshold value increases. Average image qualities of 30.561 dB, 30.500 dB, 30.449 dB, and 30.419 dB are obtained by the proposed scheme with the residual codebook of 16 codewords when the threshold values are 0, 100, 200, and 300, respectively.

**Table 2. Results of the Image Qualities of MRVQ**

$N$ Images	16 (0.75 bpp)	32 (0.8125 bpp)	64 (0.875 bpp)	128 (0.9375 bpp)	256 (1 bpp)
Airplane	30.770	31.945	32.839	33.764	34.515
Girl	31.037	32.249	33.067	34.057	35.177
Goldhill	30.528	31.272	32.143	32.788	33.325
Lenna	31.000	31.956	32.860	33.670	34.137
Pepper	31.372	32.382	33.166	33.985	34.450
Toys	28.550	29.664	30.175	31.012	32.958
<b>Average</b>	<b>30.543</b>	<b>31.578</b>	<b>32.375</b>	<b>33.213</b>	<b>34.094</b>

**Table 3. Results of the Image Qualities of the Proposed Scheme**

$N$ $TH$	0	50	100	150	200	250	300
16	<b>30.561</b>	30.534	30.500	30.468	30.449	30.434	30.419
32	<b>31.585</b>	31.549	31.514	31.485	31.459	31.430	31.400
64	<b>32.375</b>	32.335	32.283	32.238	32.198	32.156	32.116
128	<b>33.213</b>	33.154	33.087	33.030	32.977	32.921	32.868
256	<b>34.094</b>	34.021	33.933	33.856	33.453	33.720	33.657

Average bit rates of the proposed scheme without lossless compression are listed in Table 4. It is shown that the average bit rate decreases as the threshold value increases. That is because more image blocks are encoded by their block mean values by using the proposed scheme when the threshold value increases. The required bit rates of the proposed scheme are less than that of MRVQ when the threshold values are greater than or equal to 50.

**Table 4. Average Bit Rates of the Proposed Scheme without Lossless Compression**

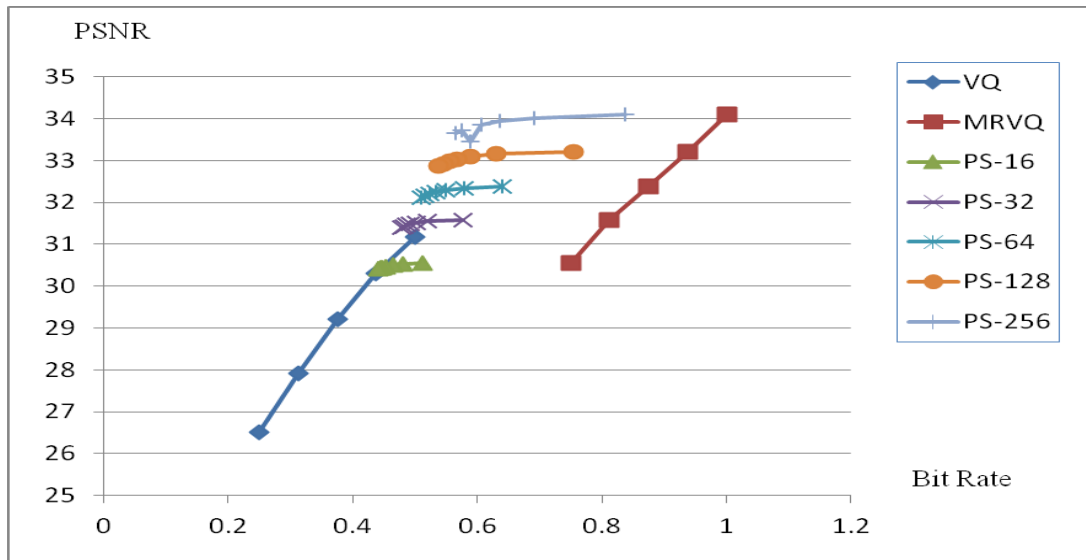
$N$ $TH$	0	50	100	150	200	250	300
16	0.739	0.680	0.645	0.632	0.626	0.623	0.620
32	0.813	0.716	0.692	0.680	0.672	0.666	0.660
64	0.870	0.778	0.714	0.723	0.711	0.702	0.649
128	0.985	0.832	0.785	0.762	0.747	0.735	0.725
256	1.050	0.889	0.832	0.803	0.783	0.768	0.757

Average bit rates of the proposed scheme with lossless compression are listed in Table 5. Similarly, the average bit rate decreases as the threshold value increases. Average bit rates of 0.512 bpp, 0.464 bpp, 0.447 bpp, and 0.441 bpp are obtained by the proposed scheme with the residual codebook of 16 codewords when the threshold values are set to 0, 100, 200, and 300, respectively. Compared to the results in Table 4, the proposed scheme with lossless compression achieves 0.189 bpp, 0.194 bpp, 0.185 bpp, 0.199 bpp, and 0.197 bpp bit rate reductions when the codebook sizes are set to 16, 32, 64, 128, and 256, respectively.

**Table 5. Average Bit Rates of the Proposed Scheme with Lossless Compression**

$N_c \backslash TH$	0	50	100	150	200	250	300
16	0.512	0.480	0.464	0.453	0.447	0.444	0.441
32	0.576	0.520	0.501	0.492	0.487	0.483	0.479
64	0.640	0.579	0.549	0.533	0.523	0.516	0.510
128	0.754	0.631	0.589	0.568	0.556	0.546	0.537
256	0.838	0.692	0.636	0.607	0.589	0.575	0.565

Comparative results among VQ, MRVQ, and the proposed scheme are listed in Figure 10. It is shown that the image quality of MRVQ is better than that of VQ and that of the proposed scheme when the same codebook size is set. However, MRVQ requires the highest bit rate of three schemes. Compared to MRVQ, 0.124 dB, 0.178 dB, 0.256 dB, 0.345 dB, 0.437 dB image quality losses are incurred by the proposed scheme with TH equals to 300 when the codebook sizes are 16, 32, 64, 128, and 256, respectively. However, 41.200%, 41.046%, 41.714%, 42.743%, and 43.500% bit rate reductions are achieved by the proposed scheme with TH equal to 300 when the codebook sizes are 16, 32, 64, 128, and 256, respectively.



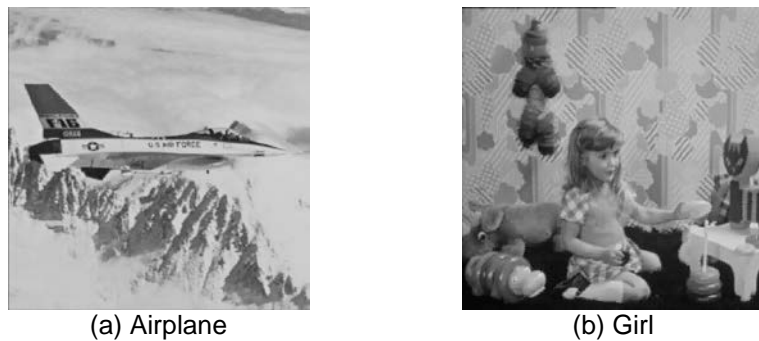
**Figure 10. Comparative Performance among VQ, MRVQ, and the Proposed Scheme**

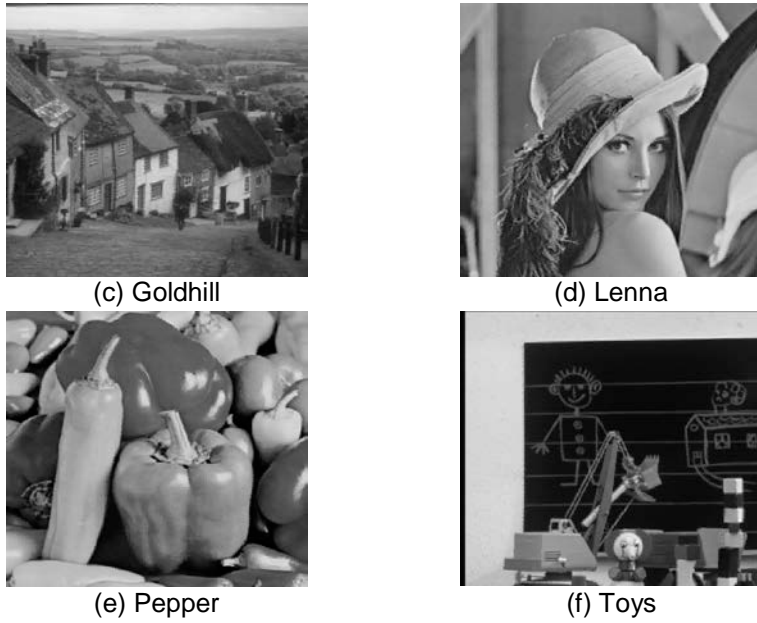
Compared to the bit rates of MRVQ with the codebook of 16 codewords, 0.238 bpp, 0.286 bpp, 0.303 bpp, and 0.309 bpp bit rate reductions are achieved by the proposed scheme when the threshold values are set to 0, 100, 200, and 300, respectively. Similarly, 0.162 bpp, 0.364 bpp, 0.411 bpp, 0.435 bpp bit rate reductions are obtained by the proposed scheme with the codebook of 256 codewords when the threshold values are set to 0, 100, 200, and 300, respectively. In other words, 16.200%, 36.400%, 41.100%, 43.500% bit rate reductions are obtained by the proposed scheme with the codebook of 256 codewords when the threshold values are set to 0, 100, 200, and 300, respectively.



**Figure 11. Reconstructed Images of VQ with the Codebook of 16 Codewords**

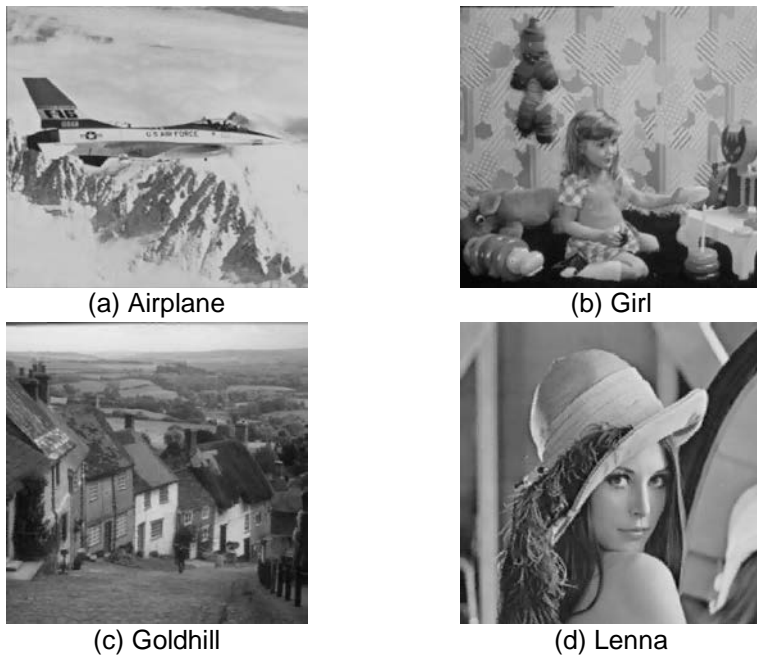
To understand the visual quality of compressed images, we listed the reconstructed images of VQ, MRVQ, and the proposed scheme in Figures 11, 12 and 13, respectively. Reconstructed images of VQ with the codebook of 16 codewords are listed. It is obvious that the visual qualities of these reconstructed images of VQ are poor because the codebook size is too small. From Table 1, average image quality of 26.502 dB is achieved by VQ when the codebook of 16 codewords is used, but the required bit rate of VQ equals to 0.25 bpp when the codebook of 16 codewords is used.

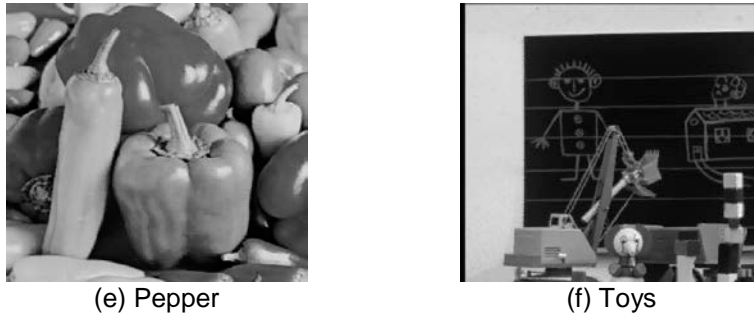




**Figure 12. Reconstructed Images of MRVQ with the Residual Codebook of 16 Codewords**

Reconstructed images of MRVQ with the codebook of 16 codewords are listed in Figure 12. From Table 2, average image quality of 30.543 dB is achieved at 0.75 bpp by MRVQ when the codebook of 16 codewords is used. The visual qualities of MRVQ reconstructed images are much better than those of VQ, but the required bit rate of MRVQ is much higher than that of VQ when the codebook size is set to 16. Compared to the original images as shown in Figure 9, it is hard to distinguish the differences between the original images and the rebuilt images of MRVQ.





**Figure 13. Reconstructed Images of the proposed Scheme with the Residual Codebook of 16 Codewords**

Reconstructed images of the proposed scheme with the codebook of 16 codewords are listed in Figure 13. In this example,  $TH$  is set to 300. From Table 3, average image quality of 30.419 dB at 0.441 bpp is achieved by MRVQ when the codebook of 16 codewords is used. Average image quality of the proposed scheme is slightly worse than that of MRVQ, but the visual qualities of MRVQ reconstructed images are much better than those of VQ. The reconstructed images of the proposed scheme are visually indistinguishable to the compressed images of MRVQ.

## 5. Conclusions

In this paper, we proposed an improved MRVQ scheme to cut down the bit rate of MRVQ while keeping acceptable reconstructed image quality. When the searched residual codewords are not suitable for encoding some image blocks, only the block mean values are used to encode such image blocks. Besides, when the use of residual codewords did not provide significant image quality improvement, the block mean values are used for these image blocks.

In addition, two lossless coding approaches are designed to compress the block mean values and the residual indices, respectively. Compared to the results of MRVQ, 41.200%, 41.046%, 41.714%, 42.743%, and 43.500% bit rate reductions are achieved by the proposed scheme with  $TH$  equal to 300 when the codebook sizes are 16, 32, 64, 128, and 256, respectively. In other words, the proposed scheme significantly cuts down the bit rate while keeping good image quality when different codebook sizes are used.

## Acknowledgements

This research was supported by Providence University, Taichung, Taiwan under contract the National Science Council, Taipei, R.O.C. under contract NSC 101-2221-E-126-014.

## References

- [1] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," IEEE Transactions on Communications, vol. 28, (1980), pp. 84-95.
- [2] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: a review," IEEE Transactions on Communications, vol. 36, no. 8, (1988), pp.957-971.
- [3] J. C. Lai and Y. C. Liaw, "Fast searching algorithm for VQ codebook generation," Journal of Visual Communication and Image Representation, vol. 7, no. 2,(1996), pp. 163-168.
- [4] Y. C. Lin and S. C. Tai, "A fast Linde-Buzo-Gray algorithm in image vector quantization," IEEE T. Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no. 3, (1998), pp. 432-435.
- [5] C. C. Chang and Y. C. Hu, "A fast codebook training algorithm for vector quantization," IEEE Transactions



- on Consumer Electronics, vol. 44, no. 4, (1998), pp. 1201–1208.
- [6] Y. C. Hu and C. C. Chang, “An effective codebook search algorithm for vector quantization,” *Imaging Science Journal*, vol. 51, no. 4, (2003), pp. 221–234.
  - [7] Y. C. Hu, B. H. Su and C. C. Tsou, “Fast VQ codebook search algorithm for grayscale image coding,” *Image and Vision Computing*, vol. 26, no. 5, (2008), pp. 657–666.
  - [8] T. C. Lu and C. C. Chang, “An improved full-search scheme for the vector quantization algorithm based on triangle inequality,” *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 6, (2009), pp. 1625–1632.
  - [9] C. H. Hsieh and J. C. Tsai, “Lossless compression of VQ index with search order coding,” *IEEE Transactions on Image Processing*, vol. 5, no. 11, (1996), pp. 1579–1582.
  - [10] Y. C. Hu and C. C. Chang, “Low complexity index-compressed vector quantization for image compression,” *IEEE Transactions on Consumer Electronics*, vol. 45, no. 1, (1999), pp. 1225–1233.
  - [11] C. C. Chang, G. M. Chen and Y. C. Hu, “A novel index coding scheme for vector quantization,” *Fundamenta Informaticae*, vol. 71, no. 2-3, (2006), pp. 215–227.
  - [12] H. M. Sun and B. H. Ku, Lossless index coding for image quantization using Huffman codes, *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 10, (2011), pp. 5961–5976.
  - [13] Y. C. Hu, C. C. Lin, and K. L. Chi, “Block prediction vector quantization for grayscale image compression,” *Fundamenta Informaticae*, vol. 78, no. 2, (2007), pp. 257–270.
  - [14] Y. C. Hu, J. C. Chuang and C. C. Lo, “Efficient grayscale image compression technique based on VQ,” *Opto-Electronics Review*, vol. 19, no. 1, (2011), pp. 104–113.
  - [15] Y. C. Hu, W. L. Chen, C. C. Lo, and J. C. Chung, “An improved vector quantization scheme for grayscale image compression,” *Opto-Electronics Review*, vol. 20, no. 2, (2012), pp. 187–193.
  - [16] Y. C. Hu, C. H. Wen, C. C. Lo and W. L. Chen, “Image vector quantization using geometric transform and lossless index coding,” *Optical Engineering*, vol. 52, no. 3, (2013), 037402.
  - [17] Y. C. Hu, W. L. Chen, C. C. Lo, C. M. Wu and C. H. Wen, “Efficient VQ-based image coding scheme using inverse function and lossless index coding,” *Signal Processing*, vol. 93, no. 9, (2013), pp. 2432–2439.
  - [18] R. F. Chang, W. T. Chen and J. S. Wang, “A fast finite-state algorithm for vector quantizer design,” *IEEE Transactions on Signal Processing*, vol. 40, no. 1, (1992), pp.221–225.
  - [19] R. F. Chang and W. M. Chen, “Adaptive edge-based side match finite-state classified vector quantization with quadtree map,” *IEEE Transactions on Image Processing*, vol. 5, no. 2, (1996), pp. 378–383.
  - [20] T. Kim, “Side-match and overlap match vector quantizer for images,” *IEEE Transactions on Image Processing*, vol. 1, no. 2, (1992), pp.170–185.
  - [21] B. Ramamurthi and A. Gersho, “Classified vector quantization of images,” *IEEE Transactions on Communications*, vol. 34, no. 11, (1986), pp. 1105–1115.
  - [22] A. Kubrick and T. Ellis, “Classified vector quantization of images: codebook design algorithm,” *IEE Proceedings I - Communications, Speech and Vision*, vol. 137, no. 6, (1990), pp.379–386.
  - [23] J. C. Lai and Y. C. Liaw, “A fast algorithm for mean-removed vector quantization using edge and texture strengths of a vector,” *Image and Vision Computing*, vol. 23, no. 8, (2005), pp.739–746.
  - [24] J. C. Lai and Y. C. Liaw, “Improvement of interpolated color filter array using modified mean-removed classified vector quantization,” *Pattern Recognition Letters*, vol. 26, no. 8, (2005), pp.1047–1058.

## Authors



**Jun-Chou Chuang** received his PhD. degree in computer science and information engineering from the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan in 2004. Currently, Dr. Chuang is an assistant professor in the Department of Computer Science and Communication Engineering, Providence University, Sha-Lu, Taiwan. His research interests include multimedia security, data hiding, digital watermarking and signal processing.



**Yu-Chen Hu** is a professor in the Department of Computer Science and Information Management, Providence University, Sha-Lu, Taiwan. He is a member of ACM and IEEE. Dr. Hu Servers as the Editor-in-Chief of International Journal of Image Processing since 2009. He joins the editorial boards of several other journals including International journal of Security and Its Applications, International Journal of Signal Processing, Image Processing and Pattern Recognition, International Journal of Digital Contents and Applications and so on. His research interests include image and signal processing, data compression, information hiding, and data engineering.



**Chun-Chi Lo** received the B.S., M.S., and Ph.D. degrees in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1989, 1991, and 1996, respectively. In 2004, he joined the faculty of the Department of Computer Science and Information Engineering, Providence University, Taiwan. Currently, he is an Assistant Professor. His research interests include wireless sensor networks, mean-field annealing, and combinatorial optimizations.



**Wu-Lin Chen** is currently an associate professor in the department of Computer Science and Information Management at Providence University. He received his M.S. and Ph.D. degrees in the School of Industrial Engineering at Purdue University in 1995, and 1999, respectively. His research interests include operations research, production management, and stochastic models.



**Chia-Hsien Wen** is currently an associate professor of the Department of Computer Science and Information Management, Providence University, Taiwan. Prior to joining Providence University in 2005, he was the director of the Computing Center at Taichung Veterans General Hospital (TCVGH). His research interests include medical image processing, medical informatics, database management, and machine learning.