

# VQ Compression Enhancer with Huffman Coding

Chin-Feng Lee<sup>1(✉)</sup>, Chin-Chen Chang<sup>2</sup>, and Qun-Feng Zeng<sup>3</sup>

<sup>1</sup> Chaoyang University of Technology, No. 168, Jifeng E. Rd.,  
Wufeng District, Taichung 41349, Taiwan, R.O.C.  
lcf@cyut.edu.tw

<sup>2</sup> Feng Chia University, No. 100, Wenhwa Rd., Seatwen,  
Taichung 40724, Taiwan, R.O.C.  
alan3c@gmail.com

<sup>3</sup> National Chung Cheng University, No.168, Sec. 1, University Rd.,  
Minhsiung, Chiayi 62102, Taiwan, R.O.C.  
qf801002@gmail.com

**Abstract.** Vector quantization (VQ) is an effective and important compression technique with high compression efficiency and widely used in many multimedia applications. VQ compression is a fixed-length algorithm for image block coding. In this paper, we employ the Huffman Coding technology to enhance VQ compression rate and get a better compression performance due to the reversibility of the Huffman Coding. The proposed method exploits the correlation between neighboring VQ indices with similarity. The similarity draws a large number of small differences from the current index with that of its adjacent neighbors; thereby, increasing the compression ratio due to the great quantity of small differences. The experimental results reveal that the proposed combination technique adaptively provides better compression ratios at high compression gains than that of VQ compression. The proposed method is superior in smoother pictures with the compression gains greater than 100%; even for the complex images the compression gain can be increased more than 25%. Therefore, the VQ-Huffman method can really enhance the efficiency of VQ compression.

**Keywords:** Vector quantization (VQ) · Huffman coding · Compression ratio

## 1 Introduction

The development of scientific and technological information bringing computer into family, computer becomes a part of human life audio-visual entertainment. But the current development of software design requires a huge amount of hard disk space. Moreover, with the progress of digital technology and the vigorous development of the Internet, a variety of including image, video, audio and other multimedia application innovation. As multimedia data storage needs increased, the data compression is more important and cannot be avoided. Good compression ability in data compression not only decreases the storage space but also great reduces transmission time. About data transmission security, since the data have been compressed and processed, the

information is not the original format, can only restore by decompressed operations while greatly increasing the information security.

Multimedia compression is divided into two categories: one is lossless compression also called reversal compression, which allows the original data to be perfectly reconstructed from the compressed data. This type of compression is generally used for text files, executive files, medical image and important information. The other is lossy compression also called irreversible compression. The reconstructed results of such data are not exactly the same as the data before they are being processed. But most of the information and features of the original data are still included in the results of the recovered data. Lossy compression has a much more compression rate than lossless compression; therefore, lossy compression is often used for image compression (such as VQ compression, JPEG compression and MPEG compression) and sound compression (such as MP3) which require high compression rate and allow partial distortion.

Vector quantization (VQ) is an effective and important compression technique, which can achieve very high compression efficiency; therefore VQ is widely used in many different situations, such as multimedia systems, high-definition television, telex systems, image data base management, and so on. VQ compression is a lossy data compression method based on the principle of block coding and is a fixed-to-fixed length algorithm. We use VQ compression as a basis due to the high compression rate and fast implementation of VQ. Then we employ the Huffman Coding technology to enhance VQ compression rate and get a better compression performance due to the reversibility of the Huffman Coding. The proposed VQ-Huffman method can exactly restore the VQ index table.

VQ compression is often used in multimedia including, sounds, images, text [1] and so on. Also VQ compression can add to the information hiding, for example, combined with Search ordering coding (SOC) [2, 3] or locally adaptive coding (LAC) [4, 5]. If we can have a better compression ratio on the basis of VQ, then we can transmit faster, also there will be more space to hide more secret data. This paper uses the difference between current index and surrounding index [6, 7], and then use Huffman Coding for further compression, hoping to achieve better compression and enhance VQ compression efficacy.

The rest of this paper is organized as follows. Section 2 introduces the related work. The proposed VQ-Huffman method is presented in Sect. 3. The experimental results are shown in Sect. 4. Conclusions make recommendations for applying the results in the last section.

## 2 The Related Work

Image compression using vector quantization is introduced in Sect. 2.1 and then the principle method of Huffman coding is summarized in Sect. 2.2.

### 2.1 Vector Quantization

In 1980, Linde, Buzo, and Gray (LBG) proposed a VQ algorithm to design a good codebook for data compression. A vector quantizer is composed of two parts, encoder

and decoder. An encoder will compare each input vector with every codevector (also called a codeword) in the codebook and generate the index of best-matched codeword which has the minimum distortion between the input vector and the codevector.

The decoded of VQ is to reconstruct the vector after quantization. A decoder takes the index corresponding to each image block to search the codebook and to locate the codevector in that codebook and generates the output vector. We repeat the decoding steps until every index is processed, and the collection of all the result vectors can be used to recovery an image similar to the original one by the quantization reconstruction.

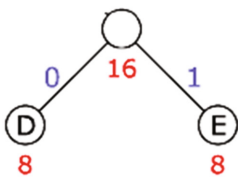
### 2.2 Huffman Coding

In 1952, Huffman proposed an encoding algorithm [9] which permits error-free data reconstruction. Huffman coding represents data by variable-length codes, with more frequent data being represented by shorter codes.

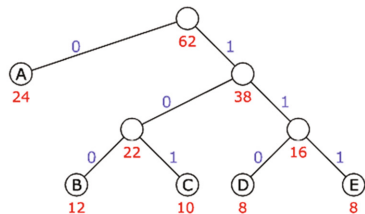
We use a simple example to show how a Huffman's Tree is built. Let the character set  $Sym = (A, B, C, D, E)$  for corresponding frequency set of symbol occurrences  $Freq = (24, 12, 10, 8, 8)$ . The technique works by creating a binary tree of nodes. Initially, all nodes are leaf nodes, which contain the symbol itself and the frequency of appearance of the symbol.

The process essentially begins with a new node whose children are the 2 nodes with smallest frequencies, such that the frequency of new internal node is equal to the sum of the children's frequency as shown in Fig. 1(a). As a common convention, bit "0" represents following the left child and bit "1" represents following the right child. With the previous 2 nodes merged into one internal node and with the new node being now considered, the procedure is repeated until only one node remains as shown in Fig. 1(b), the Huffman tree. Finally, the symbol  $A$  will be encoded by "0"; the symbol  $B$  will be encoded by "100"; the symbol  $C$  will be encoded by "101"; the symbol  $D$  will be encoded by "110", and the symbol  $E$  will be encoded by "111", respectively.

Huffman constructs a binary code tree from the bottom up, that omits unused symbols to produce the most optimal code lengths. Though the codes are of different bit lengths, they can be uniquely decoded.



(a) An internal node with two leaf nodes



(b) A Huffman's tree

**Fig. 1.** The process of a Huffman's tree

### 3 The Proposed Method

Each natural image mostly possesses the local spatial correlation which makes the neighboring positions in image pixels have the similar intensity. When we train a codebook by using VQ technique, we also sort the codewords  $CW_i = (cw_{i1}, cw_{i2}, \dots, cw_{ik})$  according to  $S_i$  in increasing order, where  $S_i = \sum_{j=1}^k cw_{ij}$ . Afterwards, we encode an input image by the sorted codebook and we can exploit the correlation between neighboring VQ indices with similarity. The similarity draws a large number of small differences from the current index with that of its adjacent neighbors; thereby, increasing the compression rate due to the great quantity of small differences.

The seed blocks are at the first column and the first row and the other blocks are called residual blocks as shown in Fig. 2. Indices of the seed blocks are kept unshorten but the indices for residual blocks are encoded with shorter bit streams. Figure 3 displays the layout of the current to-be-processed block  $X$  as well as its upper and left neighbors,  $U$  and  $L$  respectively. The encoding procedure will encode each residual block by two parts. The first part called leading indicators (LI) and the second part is codes for the difference values between the indices of ( $X$  and  $U$ ) or ( $X$  and  $L$ ). Exploiting Huffman encoding, there will be seven cases generated as shown in Fig. 4. The VQ-Huffman encoding with reversibility can further help how to encode each residual block and enhance the compression efficiency of VQ compression.

Case 1: The leading indicator is set as LI = "00" if the indices of upper block  $U$  and current block  $X$  are the same.

While the indices of upper block  $U$  and current block  $X$  are different, i.e., the difference value  $Ue \neq 0$ , we will go the following six cases.

Case 2: The leading indicator is set as LI = "01" when the different value  $Le = 0$  because the left block  $L$  and the current block  $X$  have same index.

Case 3: While  $Ue \neq 0$  and  $1 \leq |Le| \leq 4$ , the leading indicator is set as "100". Since there are eight difference values, we encode each difference with 3 bits such that the difference 1 can be encoded as "001", 2 as "010", 3 as "011", 4 as "100", -1 as "101", -2 as "110", -3 as "111", -4 as "000".

Case 4: While  $Ue \neq 0$  and  $5 \leq |Le| \leq 8$ , the leading indicator is set as "101". There also exists eight differences so we encode the difference 5 as "001", 6 as "010", 7 as "011", 8 as "100", -5 as "101", -6 as "110", -7 as "111", -8 as "000".

Case 5: While  $Ue \neq 0$  and  $9 \leq |Le| \leq 12$ , we set the leading indicator LI = "110". With nine difference values in this case, we will encode the difference 9 as "001", 10 as "010", and so on.

Case 6 has the condition  $Ue \neq 0$  and  $13 \leq |Le| \leq 28$  while Case 7 has the condition that  $Ue \neq 0$  and  $|Le| \geq 29$ . Case 6 sets the leading indicator LI = "1110" while Case 7 has LI = "1111".

In Case 6, there are 32 difference values, that is to say, the difference values from 13 to 28 and -13 to -28 are encoded by "00001", "00010", "00011", and so on. As for

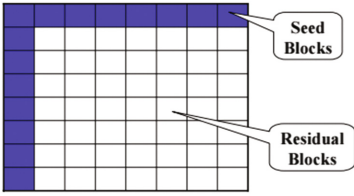


Fig. 2. Seed blocks and residual blocks

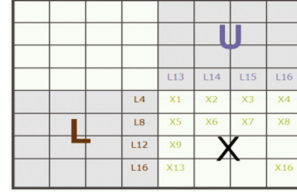


Fig. 3. Current Block  $X$ , upper block  $U$  and left block  $L$

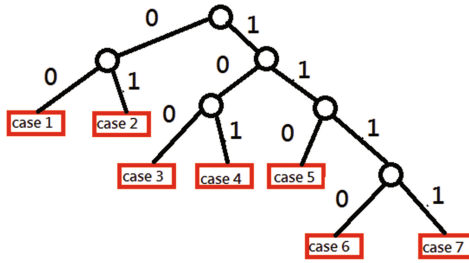


Fig. 4. Seven cases of VQ-Huffman Coding

the seventh case, we directly encode the current block  $X$  with the leading indicator and its own index value in binary representation.

### 4 Experimental Results

We used eight standard grayscale test images of size  $H \times W$  where  $H$  and  $W$  respectively stand for the height and width of an image;  $H = W = 512$  as shown in Fig. 5 for our experiment. Compression Ratio (CR) [10] is a very logical way of measuring how well a compression algorithm compresses a given digital image. The ratio of the number of bits required to represent the image before compression into the number of bits required to represent the image after compression.

In the experiment, the CR is computed by dividing the size of the original image block in bits over the size of the compressed image block. Thus Eq. 1 gives the VQ compression ratio because each grayscale image block has  $h \times w$  pixels and each pixel uses 8 bits to represent it. Therefore,  $L_x$  is constant because each image block is encoded by an 8-bit index using VQ compression. So the compression ratio by VQ is  $(8 \times 16)/8 = 16$ .

$$CR = (8 \times h \times w)/L_x \tag{1}$$

From Eq. 1, we know that the smaller  $L_x$  the larger CR. That is the reason why we exploit Huffman Coding to lower the size of the compressed image block, for creating



Fig. 5. Eight standard grayscale test images, size  $512 \times 512$  pixels

higher compression ratio. The compression ratio of proposed VQ-Huffman method can be redefined as Eq. 2 below. According to Eq. 2 the CR will become higher and so does the compression effect.

$$CR = \frac{8 \times h \times w}{L_v}, \text{ where } L_v = \sum_{i=1}^{\frac{h}{8} \times \frac{w}{8}} L_i \quad (2)$$

Table 1 provides the performance measured by VQ compression in comparison with the proposed VQ-Huffman method. Our proposed method has the average compression ratio of up to 24.857 for eight grayscale images. For the smoother pictures such as Tiffany and Toys, the compression gains can reach greater than 100%; even for the complex images such as Lena, Peppers, Gold, Barbara and F-16, the compression gain can be increased more than 25%. The results of Table 1 show that our method has a significant lift in compression gains.

Table 1. Comparisons of different encoding techniques with codebook size 256

| Compression rate | VQ method   | Propose method | Compression gain |
|------------------|-------------|----------------|------------------|
|                  | (A)         | (B)            | (B - A)/A (%)    |
| Toys             | 16.0        | 33.336         | <b>108.35</b>    |
| Tiffany          | 16.0        | 32.520         | <b>103.25</b>    |
| F-16             | 16.0        | 25.472         | 59.20            |
| Peppers          | 16.0        | 23.320         | 45.75            |
| Lena             | 16.0        | 22.096         | 38.10            |
| Girl             | 16.0        | 21.112         | 31.95            |
| Gold             | 16.0        | 21.000         | 31.25            |
| Barbara          | 16.0        | 20.000         | 25.00            |
| Avg.             | <b>16.0</b> | <b>24.857</b>  | <b>55.36</b>     |

Table 2 displays the number of blocks after VQ-Huffman method is conducted to each test image. As mentioned earlier, each image block can be encoded into a 8-bit index by VQ compression. Rather than fixed-length encoding, the proposed method encoded each image block into 2 bits by Cases 1 and 2; 6 bits by Cases 3, 4, and 5; 9 bits by Cases 6 and 12 bits by Cases 7. Therefore, we exploit our method to encode the image block by first, two cases, or else try to make the block encoded in the third, fourth, fifth cases so as to really enhance the compression ratios. From Table 2, most blocks are indeed encoded by Cases 1 or 2, which makes the VQ-Huffman method superior to that of VQ compression.

**Table 2.** Number of blocks for each case by VQ-Huffman method

| # of blocks | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 | Case 7 |
|-------------|--------|--------|--------|--------|--------|--------|--------|
| Toys        | 8267   | 2931   | 1934   | 618    | 451    | 856    | 1072   |
| Tiffany     | 7793   | 1900   | 3431   | 1284   | 589    | 792    | 340    |
| F-16        | 5292   | 3061   | 2743   | 996    | 582    | 1287   | 2168   |
| Peppers     | 4945   | 2145   | 1993   | 1799   | 1105   | 1835   | 2307   |
| Lena        | 5019   | 1444   | 2107   | 1768   | 1143   | 1891   | 2757   |
| Girl        | 3863   | 1644   | 2178   | 1977   | 1537   | 2380   | 2550   |
| Gold        | 3026   | 2328   | 2217   | 1950   | 1461   | 2765   | 2382   |
| Barbara     | 3540   | 1905   | 1861   | 1724   | 1182   | 2568   | 3349   |

## 5 Conclusions

Our paper proposed a combination of VQ compression with Huffman coding compression for enhancing image compression and exactly restoring to the VQ index table. Experimental results show that the proposed combination technique adaptively provides better compression ratios at high compression gains than that of VQ compression.

The overall superior performance offered by our method encourages the method to be applied directly in some applications such as digital libraries or online multimedia transmission. Moreover, it also can be applied to some interdisciplinary research topics such as information hiding to help prevent unauthorized access, use, disclosure, disruption, modification, or inspection of information. Since the proposed method contributes to completely restore the VQ index table without data loss, the VQ-Huffman method can provide to embed a large amount of data into VQ images for information hiding applications.

**Acknowledgments.** This research was partially supported by the Ministry of Science and Technology of the Republic of China under the Grants MOST 105-2221-E-324-014 and MOST 105-2221-E-035-051.

## References

1. Korycki, R.: Authenticity examination of compressed audio recordings using detection of multiple compression and encoders' identification. *Forensic Sci. Int.* **238**, 33–46 (2014)
2. Chang, C.C., Chen, G.M., Lin, M.H.: Information hiding based on search-order coding for VQ indices. *Pattern Recogn. Lett.* **25**, 1253–1261 (2004)
3. Lin, C.C., Liu, X.L., Yuan, S.M.: Reversible data hiding for VQ-compressed images based on search-order coding and state-codebook mapping. *Inf. Sci.* **293**, 314–326 (2015)
4. Yang, C.H., Lin, Y.C.: Fractal curves to improve the reversible data embedding for VQ-indexes based on locally adaptive coding. *J. Vis. Commun. Image Represent.* **21**, 334–342 (2010)
5. Chang, C.C., Nguyen, T.S., Lin, C.C.: A reversible data hiding scheme for VQ indices using locally adaptive coding. *J. Vis. Commun. Image Represent.* **22**, 664–672 (2011)
6. Chang, C.C., Kieu, T.D., Wu, W.C.: A lossless data embedding technique by joint neighboring coding. *Pattern Recogn.* **42**, 1597–1603 (2009)
7. Wang, J.X., Lu, Z.M.: A path optional lossless data hiding scheme based on VQ joint neighboring coding. *Inf. Sci.* **179**, 3332–3348 (2009)
8. Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer design. *IEEE Trans. Commun.* 702–710 (1980)
9. Huffman, D.A.: A method for the construction of minimum-redundancy codes. *Proc. Inst. Radio Eng.* **40**(9), 1098–1101 (1952)
10. Telagarapu, P., Naveen, V.J., Prasanthi, A.L., Santhi, G.V.: Image compression using DCT and wavelet transformations. *Int. J. Signal Process. Image Process. Pattern Recogn.* **4**, 61–70 (2011)