Proc. of the 6th WSEAS Int. Conf. on Signal Processing, Computational Geometry & Artificial Vision, Elounda, Greece, August 21-23, 2006 (pp131-134)

Visualization of Vector Quantization

Shang-Kuan Chen
Department of Computer Science and Information Engineering
Yuanpei University
Hsinchu, 300
Taiwan, R. O. C.

Abstract: **-** In this paper, a novel scheme for vector quantization (VQ) is proposed. A file called visible index file is used to record the coding result. The decompressed image reconstructed from the visible index file is the same as the one recovered using traditional VQ index file; however, the visible index file looks like the original image, and is therefore more convenient for the management of index files. Also, note that the size of the visible index file is the same as that of the traditional index file.

Key-words: vector quantization, sorted codebook, visible index files.

1 Introduction

Vector Quantization (VQ) is a well-known technique [1-7] for compressing digital images. In the encoding phase of VQ, a given image is divided into several blocks, and then each block is mapped to its closest codeword chosen from a given codebook. The indices of the codewords are kept in a file to record the mapping sequence. In the decoding phase of VQ, the indices are used for finding the corresponding codewords to reconstruct the given image. The index file can thus be viewed as the compression result of the image. In general, the index file looks like noisy data. Recently, some studies focused on the index files of VQ, but so far none of the reported works tried to create "visible" index files. In other words, all index files always look like noisy data, and can only be distinguished from one other by file names. If the index files can be visualized, the management of them will be much easier. In fact, nowadays, visualization is used in many research fields [8-11].

The remaining of the paper is organized as follows. Section 2 states the proposed method. Section 3 shows the experimental results. Finally, Section 4 presents the conclusions.

2. The proposed method

In order to produce visible index file, an auxiliary mapping table is generated first and stored in public domain so that everybody can access it. Firstly, the codewords of the VQ codebook are sorted according to the average gray value of each codeword. Each codeword with lowest average gray value has smallest index value. Then we may construct the auxiliary table, which maps each index to its corresponding binary (black-and-white) block. Without the loss of generality, assume that there are 512 indices. Then, since 512 = 2^9, each index is a 9-bit number. For visualization, the corresponding binary block can also be represented as a 3-by-3 block formed of 9 binary pixels (see Table 1).

Let {Codeword 0, Codeword 1, … , Codeword 511} be the codewords of the sorted codebook. Because Codeword 0 has smallest average gray value, to make visible index file look like the given image, binary block with index 0 is also represented as a block with smallest average brightness, i.e. all 9 pixels of the binary block are black. Then for the next $C^9_1 = 9$ indices (Index 1 – Index 9), each binary block has only one white pixel because each corresponding codeword (Codeword 1 – Codeword 9) still has small average gray value. Then, for the next $C^9_2 = 36$ indices (Index 10 – Index 45), each binary block has two white pixels. The procedure proceeds until all 2^9 indices have been mapped to all 2^9 binary blocks.

For example, the average gray value of Index 502 codeword of the sorted codebook is 228.32. The block of 9 binary pixels of index-502 is as follows,

□□□
□□□
□□■

and the average gray value of the block is (255+255+255+255+255+255+255+255+0) / 9 = 226.67. The brightness between the codeword and the block is very close. Therefore, the block not only can be used for mapping the closest codeword but also be similar to the actual codeword in brightness.

The indices of the sorted codebook	The blocks of 9 binary pixels	Remarks
0	■■■ ■■■ ■■■	no white pixel
1	□■■ ■■■ ■■■	one white pixel
2	■□■ ■■■ ■■■	
3	■■□ ■■■ ■■■	
4	■■■ □■■ ■■■	
5	■■■ ■□■ ■■■	
6	■■■ ■■□ ■■■	
7	■■■ ■■■ □■■	
8	■■■ ■■■ ■□■	
9	■■■ ■■■ ■■□	
10	□□■ ■■■ ■■■	two white pixels
11	□■□ ■■■ ■■■	
12	□■■ □■■ ■■■	
:	:	
45	■■■ ■■■ ■□□	
46	□□□ ■■■ ■■■	Three or more white pixels
:	:	
511	□□□ □□□ □□□	9 white pixels

Table 1. The auxiliary mapping table

Note that the brightness order of the sorted codewords is roughly kept, i.e., Codeword *i* can never be brighter than Codeword *i*+1, and similarly, Binary Block *i* can never be brighter than Binary Block *i*+1. This is the reason why using visible index file can roughly show a binary version of the original image, because the darker regions of the original image are still darker in the visible index file. Also, note that, from Table 1, it can be seen that each 9-bit index of the traditional index files is replaced by a 9-pixel block (each pixel is either black or white) in our visible index file. Therefore, the size of visible index file is identical to that of the traditional index file.

3. Experimental results

The quality of decoded image is often quantized by Peak-Signal-to-Noise Ratio (*PSNR*).

The *PSNR* is defined as follows:

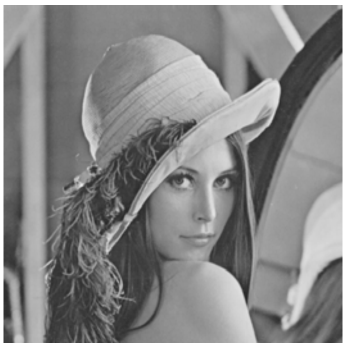
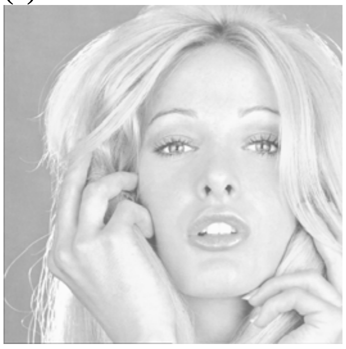
$$PSNR = 10\log_{10}\frac{255^2}{MSE} \text{-------------------------------(1)}$$

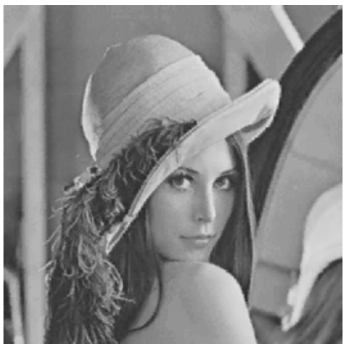
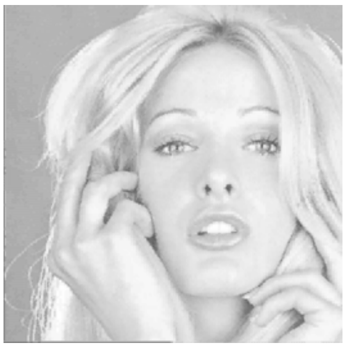
, where

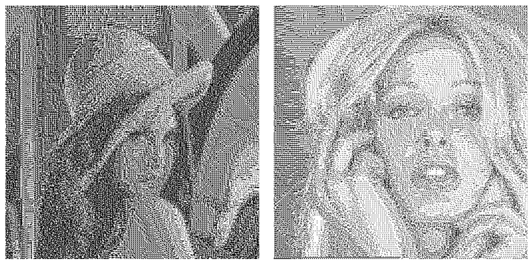
$$MSE = \left(\frac{1}{M\times N}\right)\sum_{i=1}^{M}\sum_{j=1}^{N}(x_{ij}-\hat{x}_{ij})^2 \text{.----------------(2)}$$

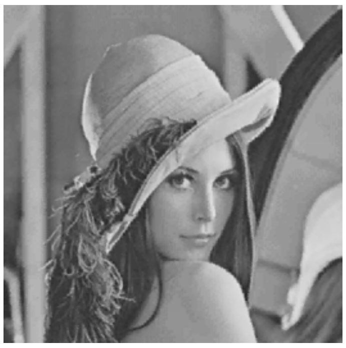
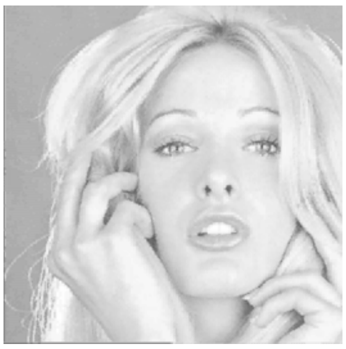
M and *N* represent the width and the height of the image, respectively. x_{ij} and $\hat{x}_{ij}$ present the original pixel value and the modified pixel value of coordinate (*i*, *j*), respectively. In general, when the value of *PSNR* is greater than 30 dB, the difference between two images is indistinguishable by human eyes.

The experimental result is shown in Figures 1-4. A universal codebook is generated by training three outside images: Milk, Baboon, and Jet. The number of codewords in the universal codebook was set to 512 and each codeword is with size 4×4. Therefore, in this case, the compression rate is the reciprocal of 9/(4×4×8) ≈ 0.07. The codebook is sorted according to the average gray value of each codeword. The sorted codebook and Table 1 (the auxiliary mapping table) are uploaded in public domain so that everyone can access them. Fig. 1(a) and Fig. 1(b) show the original images Lena and Tiff with size 512×512. The images recovered by traditional VQ index files are shown in Fig. 2(a) and Fig. 2(b). The PSNR of the recovered images are 30.86 dB and 31.37 dB, respectively. The binary images in Fig. 3(a) and Fig. 3(b) are the visible index files generated by the proposed scheme. The size of each visible index file is (9/16) × (1/8) ≈ 0.07 times smaller than that of the original image. Note that (9/16) × (1/8) is identical to the reciprocal of the compression rate; and this should be of no surprise because our visible index file and traditional index file have the same size. Finally, the images recovered by visible index file are shown in Fig. 4(a) and Fig. 4(b) which are identical to those in Fig. 2(a) and Fig. 2(b).


(a)

(b)
Figure 1. The two original images.


(a)

(b)
Figure 2. The two images recovered by traditional VQ index files (with PSNR 30.86 dB and 31.37 dB, respectively).


(a) (b)
Figure 3. The two visible index files.


(a)

(b)
Figure 4. The two images (with PSNR 30.86 dB and 31.37 dB, respectively) recovered using the files sketched in Fig. 3 were identical to those in Fig. 2.

4. Conclusions

In this paper, we have proposed a new idea and thus generated visible index files for VQ. The codebook is sorted and an indices-blocks mapping table is generated. Both the sorted codebook and the mapping table are uploaded and stored in public domain for the access with no limitation. The above steps can be viewed as a pre-processing and need be done only once. Then, for the day-to-day use in later days, each time a user wants to compress a given image by VQ, he can use the public codebook and mapping table to create a visible index file for that image. Since every image is compressed and stored as a visible index file that resembles the original image, the management of the compression results becomes very easy. No storage overhead exists for each user because the size of each visible index file is identical to that of the traditional index file. The recovered (decompressed) gray-leveled image is also identical to the one recovered using traditional index file.

References:

[1]. R.M. Gray, "Vector quantization," *IEEE ASSP Magazine*, 1984, pp. 4-29.

[2]. N. M. Nasrabadi and R.A. King, "Image coding using vector quantization: A review," *IEEE Transactions on Communication*, Vol. 36(8), 1988, pp. 957-971.

[3]. J. Mielikainen, "A novel full-search vector quantization algorithm based on the law of cosines," *IEEE Signal Processing Letters*, Vol. 9(6) , 2002, pp. 175-176.

[4]. K. L. Chung and J. Y. Lai, "An efficient law-of-cosine-based search for vector quantization", *Pattern Recognition Letters*, Vol. 25(14) , 2004, pp. 1613-1617.

[5]. D.Mukherjee and S.K. Mitra, "Successive refinement lattice vector quantization," *IEEE Transactions on Image Processing*, Vol. 11(12), 2002, pp. 1337 – 1348.

[6]. A.M. Eftekhari-Moghadam, J. Shanbehzadeh, F. Mahmoudi, H. Soltanian-Zadeh, "Image retrieval based on index compressed vector quantization," *Pattern Recognition*, Vol. 36, 2003, pp. 2635-2647.

[7]. C.C. Chang, G.M. Chen, M.H. Lin, "Information hiding based on search-order coding for VQ indices," *Pattern Recognition Letters*, Vol. 25, 2004, pp. 1253-1261.

[8]. S. Tomov, R. Bennett, M. McGuigan, A. Peskin, G. Smith, J. Spiletic, "Appication of interactive parallel visualization for commodity-based clusters using visualization APIs," *Computers & Graphics*, Vol. 28, 2004, pp. 273-278.

[9]. S. dos Santos, K. Brodile, "Gaining understanding of multivariate and multidimensional data through visualization," *Computers & Graphics*, Vol. 28, 2004, pp. 311-325.

[10]. S. Fuhrmann, "Designing a visualization system for hydrological data," *Computers & Geosciences*, Vol. 26, 2000, pp. 11-19.

[11]. C.C. Thien and J.C. Lin, "An image-sharing method with user-friendly shadow images," *IEEE Transactions on circuits and systems for video technology*, Vol. 13(12), 2003, pp. 1161-1169.