Vector Quantization for Medical Image Compression Based on DCT and Fuzzy C-Means

Sookpotharom Supot[1], Pantsaena Nopparat[2], Airphaiboon Surapan[2] ,and Sangworasil Manas[2]

[1] Dept. of Electrical Engineering,
Bangkok University,
Pathumtani, Thailand 12120
Phone: +66-2-902-0299 Ext. 620, Email: supot.s@bu.ac.th

[2] The Research Center for Communications and Information Technology (ReCCIT)
Dept. of Electronics, Faculty of Engineering,
King Mongkut Institute of Technology Ladkrabang (KMITL),
Chalongkrung Rd., Ladkrabang, Bangkok, Thailand 10520
Phone: +66-2-326-9968, Email: ksamanas@kmitl.ac.th

Abstract: Compression of magnetic resonance images (MRI) has proved to be more difficult than other medical imaging modalities. In an average sized hospital, many tera bytes of digital imaging data (MRI) are generated every year, almost all of which has to be kept. The medical image compression is currently being performed by using different algorithms. In this paper, Fuzzy C-Means (FCM) algorithm is used for the Vector Quantization (VQ). First, a digital image is divided into subblocks of fixed size, which consists of 4x4 blocks of pixels. By performing 2-D Discrete Cosine Transform (DCT), we select six DCT coefficients to form the feature vector. And using FCM algorithm in constructing the VQ codebook. By doing so, the algorithm can make good time quality, and reduce the processing time while constructing the VQ codebook.

1. Introduction

MR image is commonly used in hospitals. In an average sized hospital, many tera bytes of digital imaging data are generated every year, almost all of which has to be kept. Archive this large amount of image data in the computer memory is very difficult without any compression. There are many biomedical image compression algorithms. These techniques were improved in the last ten years. The most common technique is Vector Quantization [1]. The other techniques are Interframe coding [2], Discrete Hartley Transform [3], which is the alternative to the Discrete Cosine Transform, Mixed Transform [4] and Entropy-Coded DPCM [5]. Vector Quantization algorithm can also be used to compress the biomedical images.

The VQ scheme can be divided into three parts: the codebook generation process, the encoding procedure and the decoding procedure. An identical codebook previously generated is required in both the encoding procedure and the decoding procedure in VQ scheme. In the process of vector quantization, the image to be encoded is segmented into a set of input image vectors. In the encoding procedure, the closest codeword for each input vector is chosen, and its index is transmitted to the receiver as shown in Figure 1. In the decoding procedure, a simple table look-up procedure is don to reconstruct the encoded image in the receiver. Thus, the encoded image of the original input image becomes available to the receiver. The whole compression process is accomplished when the encoded image is reconstructed with the corresponding index of each input image vector.

The most important task for the VQ scheme is to design a good codebook. A good codebook is required because the reconstructed image highly depends on the codewords in this very codebook. Of the literature concerning compression techniques so far, the generalized Lloyed clustering algorithm (referred to as the LBG algorithm) [6] proposed by Linde, Buzo, and Gray is the most popular method for codebook training.

The LBG algorithm is an iterative procedure. Starting with an initial codebook, the LBG algorithm performs the segmentation of the training set, where an exhaustive search procedure is conducted to classify each vector in the training set according to its corresponding closest codeword in current codebook. The closest codeword of one training vector is the codeword with the smallest squared Euclidean distance in the codebook. Then each training vector records the index of its closest codeword in the codebook. After completing the segmentation process for the whole training set, each codeword in the current codebook is updated with the centroid of those training vectors so that the current codeword is the closest codeword of them. The newly generated codebook is then used in the next iteration to minimize the overall averaged distortion in the codebook design procedure. Both the segmentation and the updating procedures are executed repeatedly until the overall averaged distortion between the last two successive iterations is not significantly noticeable or a certain number of iterations are reached.

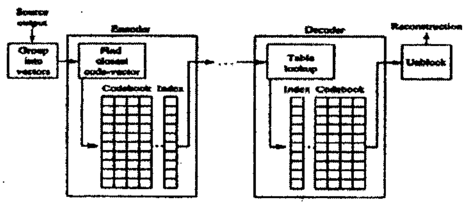


Figure 1. Block diagram of image vector quantization

Since an exhaustive codebook search is involved in the segmentation process, a large amount of computation cost is required in finding the closest codeword in the codebook for each training vector. Besides, a larger-sized training set is often required in order to generate more representative codewords. Thus, the LBG algorithm needs a large amount of computation cost for codebook training. To reduce the computation cost of the segmentation procedure in the LBG algorithm, we introduce the Fuzzy C-Means (FCM) [7-8] algorithm based on the fuzzy set theory instead of the traditional algorithm, such as LBG algorithm.

2. Fuzzy C-Mean Algorithm

Let $\Re$ be the set of real, $\Re^p$ the set in p-dimensions. $X = \{x_1, ..., x_n\}, x_i \in \Re^p$ denotes a set of raw data. c denotes number of clusters. u_{ik} denotes a membership value $\{u_{ik} = u(x_k)$, $1 \le i \le c$ and $1 \le k \le n\}$. ε is the accepted error.

$$0 \le u_{ik} \le 1 \quad \text{for all } i, k \tag{1}$$

$$\sum_{i=1}^{c} u_{ik} = 1 \quad \text{for all } k \tag{2}$$

$$0 < \sum_{k=1}^{n} u_{ik} < n \quad \text{for all } i \tag{3}$$

There are four steps in FCM as follows:

Step1. $U = [u_{ik}]$, fix $c \in \{2,3,...,n-1\}$, $m \in (1, \infty)$, and Initialized $U^{(0)}$.

Step2. At each iteration $[l, l = 0,1,2,...,]$, the cluster center is computed as.

$$v_i^{(l)} = \frac{\sum_{k=1}^{n} (u_{ik}^{(l)})^m x_k}{\sum_{k=1}^{n} (u_{ik}^{(l)})^m}, 1 \le i \le c \tag{4}$$

Step3. Update $U^{(l)} = [u_{ik}^{(l)}]$ to $U^{(l+1)} = [u_{ik}^{(l+1)}]$ using

$$u_{ik}^{(l+1)} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\left\| x_k - v_i^{(l)} \right\|}{\left\| x_k - v_j^{(l)} \right\|} \right)^{\frac{2}{m-1}}}, 1 \le i \le c, 1 \le k \le n \tag{5}$$

Step4. If $\left\| U^{(l+1)} - U^{(l)} \right\| < \varepsilon$, stop.

Otherwise, set $l = l + 1$ and go to Step2

3. Image Compression Method

A codebook has to be initialized by a fine construction using the FCM algorithm. The compression method proceeds as follows:

Step 1. Segment the training image into a nonoverlapping block of size 4x4. A block of pixels is called a training vector.

Step 2. Extract the features of image blocks using the 2-D DCT [9]. The 2-D DCT coefficient array is rearranged into a 1-D array according to the zigzag scan as shown in Figure 2. For general images, this scan method will yield and order of decreasing variance of the coefficients; ie, V(1)>V(2)>...>V(16), where V(.) denotes the variance of the zigzag scanned coefficients of 4x4 DCT. For these properties of the DCT coefficients, we choose the six coefficients D(1),D(2),D(3),D(4),D(5), and D(6) after zigzag scanning to be the feature vector. Note that, D(1) is the mean pixel value of the image block. In fact, we have tried various combinations of the DCT coefficients, and experiments showed that using the feature set of the six coefficients did obtain the best performance.

Step 3. Fix the size c of the codebook and the accepted error ϵ

Step 4. Initialize randomly the membership degree $U^{(0)} = U^{(l)}$

Step 5. Compute the codeword v_i

Step 6. Update the membership degree $U^{(l+1)}$

Step 7. If $\left\| U^{(l+1)} - U^{(l)} \right\| < \varepsilon$,stop

Otherwise, set $l = l+1$ and go to step 5.

Step 8. Defuzzification process

Step 9. Decoding

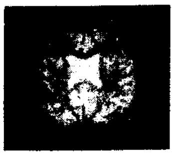
D(1)	D(2)	D(6)	D(7)
D(3)	D(5)	D(8)	D(13)
D(4)	D(9)	D(12)	D(14)
D(10)	D(11)	D(15)	D(16)

Figure 2. Zigzag scan order

4. Experimental Results

Experimental, the input image for MRI machine slice number 34 is tested. This paper uses MATLAB and CPU Pentium III for the process. We tested VQ-DCT-FCM algorithm on Proton Density (PD) image as shown in Figure 3.

(a) Original



(b) Reconstructed at 0.65 Bpp

Figure 3. Example of VQ-DCT-FCM algorithm for MRI image

As commonly used measure in a variety of coding systems, Signal-to-Noise Ratio (SNR) is adopted for evaluating objective quality. SNR is defined as

$$SNR = 10\log_{10}\frac{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[f(i,j)]^2}{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[f(i,j)-\hat{f}(i,j)\right]^2} \quad (6)$$

where $f(i,j)$ and $\hat{f}(i,j)$ are original and reconstructed pixels, and $MN = 256\text{x}256$ is the size of the image. Compression is obtained in VQ by using a codebook with relatively few codevectors compared to the number of possible image vectors. The resulting bit rate of a VQ scheme is

$$\text{Bit Rate} = \log_2 (Nc)/k \quad (7)$$

where Nc is the codebook size and k is the number pixels of the block. The SNRs for VQ-LBG and VQ-DCT-FCM are plotted in Figure 4.

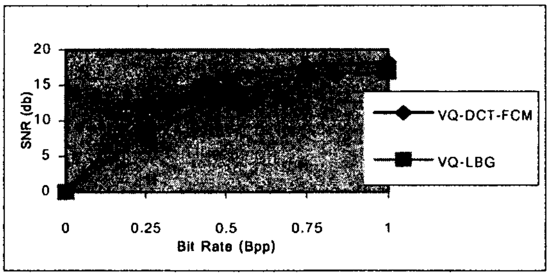


Figure 4. SNR vs. bit rate for MRI

In order to compare the relative performance of our method, the original LBG algorithm and VQ-DCT-FCM algorithm are implemented. In our simulation, the training set of proton density image is used. Simulation using different sizes of codebook for training set is conducted to evaluate the method as shown in Table 1.

Table 1. Comparative execution time (in seconds) for codebook design

Codebook size / Method	32	64	128	256
VQ-LBG	90.4	170.2	350.6	707.1
VQ-DCT-FCM	56.2	73.4	91.3	117.5

From figure 4 and table 1, it is clear that VQ-DCT-FCM method does better than VQ-LBG method and the comparative execution time between VQ-LBG method and VQ-DCT-FCM method use MR image to be training vector, the execution time by VQ-DCT-FCM method is less than by VQ-LBG method. It is because this method has done its job, as is created for, to reduce the large computation cost of VQ-LBG method.

5. Conclusions

In this paper, propose the fuzzy classification modifying VQ which is called VQ-DCT-FCM, to reduce high complexity as image coding. In VQ-DCT-FCM method, a good codebook is required both in the encoding and the decoding procedures. Therefore, the codebook design process is a crucial part in the VQ-DCT-FCM method. The reconstructed image quality of the VQ-DCT-FCM method highly depends on the codebooks generated in the codebook design process. From the experimental results, it is shown that the VQ-DCT-FCM method provides the image quality better than the traditional VQ method. In addition, the execution time needed by the VQ-DCT-FCM method is less than VQ-LBG method.

References

[1] Nasrabadi, Nasser M.& Feng, Yushu, " Image Compression Using Address Vector Quantization," *IEEE Transactions on Communication*, v. 38, pp. 2166-2173, 1990.

[2] Nosratinia, Aria & Mohsenian, Nader & Orchard, Michael T. & Liu, Bede, " Interframe Coding of Magnetic Resonance Images ," *IEEE Transactions on Medical Imaging*, v. 15, pp. 639-647, 1996.

[3] Villasenor, John D., " Alternatives to the Discrete Cosine Transform for Irreversible Tomographic Image Compression ," *IEEE Transactions on Medical Imaging*, v. 12, pp. 803-811, 1993.

[4] Ramaswamy, A. & Mikhael, W. B., " A Mixed Transform Approach for Efficient Compression of Medical Images ," *IEEE Transactions on Medical Imaging*, v. 15, pp. 343-352, 1996.

[5] Chen, Keshi & Ramabadran, Tenkasi V., " Near-Lossless Compression of Medical Images Through Entropy-Coded DPCM ," *IEEE Transactions on Medical Imaging*, v. 13, pp. 538-548, 1994.

[6] Y. Linde, A. Buzo, & R. M. Gray, " An Algorithm for Vector Quantization Design ," *IEEE Transactions on Communication*, v. COM-208, pp.84-95, 1980.
[7] G. J. Klir & B. Yuan, *Fuzzy Sets and Fuzzy Logic Theory and Applications* , Prentice-Hall Inc. Upper Saddle River, NJ, pp. 358-362, 1995.
[8] N. R. Pal & J. C. Bezdek, " On Cluster Validity for the Fuzzy C-Means Model , " *IEEE Transactions on Fuzzy System,* v. 3, pp.370-372, 1995.
[9] C. H. Hsieh, "DCT-based codebook design for vector quantization of images," *IEEE Trans. Circuits and Systems for Video Technology*, v. 2, no. 4, 1992