JOURNAL OF KOREA MULTIMEDIA SOCIETY, VOL. 7, NO. 12, DECEMBER 2004(pp. 1737-1744)

Color Image Vector Quantization Using Enhanced SOM Algorithm

Kwang-Baek Kim[†]

ABSTRACT

In the compression methods widely used today, the image compression by VQ is the most popular and shows a good data compression ratio. Almost all the methods by VQ use the LBG algorithm that reads the entire image several times and moves code vectors into optimal position in each step. This complexity of algorithm requires considerable amount of time to execute. To overcome this time consuming constraint, we propose an enhanced self-organizing neural network for color images. VQ is an image coding technique that shows high data compression ratio. In this study, we improved the competitive learning method by employing three methods for the generation of codebook. The results demonstrated that compression ratio by the proposed method was improved to a greater degree compared to the SOM in neural networks.

Keywords: Image compression, LBG algorithm, VQ, enhanced self-organizing neural network

1. INTRODUCTION

Computer graphics and imaging applications have started to make inroads into our everyday lives due to the global spread of information technology. This has made image compression an essential tool in computing with workstations, personal computers and computer networks. Videoconferencing, desktop publishing and archiving of medical and remote sensing images all entail the use of image compression for storage and transmission of data[1]. Compression can also be viewed as a form of classification, since it assigns a template or codeword to a set of input vectors of pixels drawn from a large set in such a way as to provide a good approximation of representation[2].

A color image is composed of three primary components. The most popular choices of color primaries are (R, G, B), (Y, I, Q), and (Y, U, V). The Y component represents the brightness and IQ(UV) components represent the chrominance signals. In this paper, we considered color images in the (R, G, B) domain with a color of one pixel determined by three primary components, the red(R), green(G) and blue(B). Each component is quantized to 8 bits, hence 24 bits are needed to represent one pixel. The number of palette elements to be represented by 24 bits is 2^{24}, but all of colors are not used to represent one image. So it is possible to compress the pixel color of the real image. It is also necessary to compress the pixel color because of the limitation of disk space and the transmission channel bandwidth[3,4].

In the compression methods introduced until now, the image compression by Vector Quantization(VQ) is most popular and shows a good data compression ratio. Most methods by VQ use the LBG algorithm developed by Linde, Buzo, and Gray[3]. But this algorithm reads an entire image several times and moves code vectors into optimal position in each step. Due to the complexity of the algorithm, it takes considerable time to execute. To

※ Corresponding Author : Kwang-Baek Kim, Address : (617-736) San 1-1, Gwaebop-Dong, Sasang-Gu, Busan, Korea, TEL : +82-309-5052, FAX : +82-999-5652
E-mail : gbkim@silla.ac.kr
Receipt date : May 13, 2004, Approval date : Sep. 17, 2004
[†] Dept. of Computer Engineering, Silla Univ.
※ This work was supported by the research grant 2004 of Silla University.

overcome this time consuming constraints, we propose an enhanced self-organizing vector quantization method for color images.

2. RELATED RESEARCH

2.1 Definition of VQ

A minimum distortion data compression system or source coder can be modeled as a vector quantization(VQ), by mapping of input vectors into a finite collection of templates or reproduction code words called a codebook[5]. In VQ, the original image is first decomposed into n-dimensional image vectors. The process of mapping the decomposed image vector X into the template vector having a minimal error is called VQ. That is, VQ can be defined as a mapping Q of k-dimensional Euclidean space R^k into a finite subset Y of R^k. Thus,

$$Q: R^k \rightarrow Y,\ Y = (x'_i : i = 1, \ldots, N_c) \tag{1}$$

where $Y = (x'_i: i=1,\ldots,N_c)$ is the set of reproduction vectors, the codebook. And N_c is the number of vectors in Y, the size of the codebook. It can be seen as a combination of two functions: an encoder, which views the input vector x and generates the address of the reproduction vector specified by $Q(x)$, and a decoder, which generates the reproduction vector x' using this address.

2.2 VQ Using A Self-Organizing Feature Map

The Self-Organizing Feature Map has been found to serve as a good algorithm for codebook generation. The various properties of the artificial neural networks, such as learning property, ability to quickly form categories, parallel and distributed processing ability etc, are being exploited in these methods. The Self-Organizing Map (SOM) algorithm, which is derived from an appropriate stochastic gradient decent scheme, results in a natural clustering process in which the network performs competitive learning to perceive pattern classes based on data similarity. Smoothing of vector elements does not take place in this unsupervised training scheme. At the same time, since it doses not assume an initial codebook, the probability of getting stranded in local minima is also small. The investigations for high quality reconstructed pictures have led us to the edge preserving self-organizing map. This greatly reduces the large computational costs involved in generating the codebook and finding the closest codeword for each image vector. However, from practical experience, it is observed that additional refinements are necessary for the training algorithm to be efficient enough for practical applications[6,7]. The SOM technique can display, in a single grey level image, the most significant clustering of data in an n-dimensional feature space, without confusing clusters that are distinct in the feature space[8-10]. This is possible because points which are far apart in the feature space can map to the same grey level, quantized to the same map node, only if the distribution of data near them is very sparse[11]. Thus, the points far apart in grey level belong to different significant clusters. However, this conventional method leaves a large amount of neurons under-utilized after training[12,13]. Different types of edge vectors could get crowded into the same neuron, which is "over-represented". In this paper, we propose an enhanced self-organizing vector quantization for color images.

3. ENHANCED SOM ALGORITHM FOR COLOR IMAGE VECTOR QUANTIZATION

The proposed learning method is based on the SOM structure of Fig. 1. The structure is composed of the input and output layer. It is a bi-directional network with complete connectivity and uses the same weights in each direction. The weight adaptation is performed in proportion to the

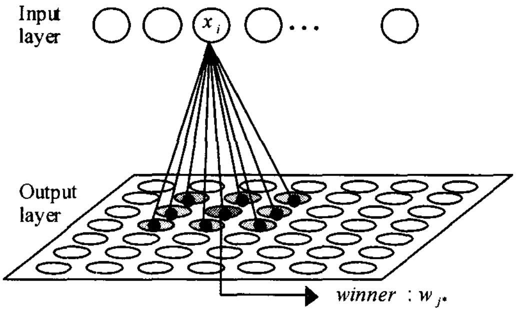


Fig. 1. Self-Organizing Feature Map Structure.

error value between the input vector and the code vector of the winner node. Moreover, the winner neuron participated in the weight adaptation remembers its frequency being a winner.

In this paper, we improved the SOM algorithm by employing three methods for the efficient generation of codebook. First, the error between the winner node and the input vector and the frequency of the winner node are reflected in the weight adaptation. Second, the weight is adapted in proportion to the present weight change and the previous weight change as well. Third, in the weight adaptation for the generation of the initial codebook, the weight of the adjacent pixel of the winner node is adapted together.

In the proposed method, the codebook is generated by presenting the entire image only two times. In the first step, the initial codebook is generated to reflect the distribution of the given training vectors. The second step uses the initial codebook and regenerates the codebook by moving to the center within the decision region.

To generate the precise codebook, it needs to select the winner node correctly and we have to consider the real distortion of the code vector and the input vector. For this management, the measure of the frequency to be selected as winner node and the distortion for the selection of the winner node in the competitive learning algorithm are needed. Generally the SOM algorithm uses the following equation for the weight adaptation.

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i - w_{ij}(t)) \tag{2}$$

α is the learning factor between 0 and 1 and is set between 0.25 and 0.75 in general. $(x_i - w_{ij}(t))$ is an error value and represents the difference between the input vector and the representative code vector. This means weights are adapted as much as the difference and it prefers to adapt the weight in proportion to the size of the difference. Therefore, we use the normalized value for the output error of the winner node that is converted to the value between 0 and 1 as a learning factor. The larger the output error, the more amount for the weight adaptation. So, the weight is adapted in proportion to the size of the output error.

Based on the above method, we use the following equation in the weight adaptation.

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \alpha(x_i - w_{ij}(t)) \\ \alpha &= f(e_j) + \frac{1}{f_j} \end{aligned} \tag{3}$$

where $f(e_j)$ is the normalization function that converts the value of e_j to the value between 0 and 1, e_j is the output error of the jth neuron, and f_j is the frequency for the jth neuron as the winner.

The above method considers only the present weight change and does not consider the previous weight change. So in the weight adaptation, we consider the previous weight change as well as the present one's. This concept corresponds to the momentum parameter of BP. We will also call this concept as a momentum factor. Based on the momentum factor, the equation for the weight adaptation is as follows:

$$w_{ij}(t+1) = w_{ij}(t) + \delta_{ij}(t+1) \tag{4}$$

$$\delta_{ij}(t+1) = \alpha(x_i - w_{ij}(t)) + \alpha\delta_{ij}(t) \tag{5}$$

In equation (5), the first term represents the effect of the present weight change and the second term is the momentum factor representing the previous weight change.

The algorithm is detailed below:

Step 1. Initialize the network. i.e., initialize weights (w_{ij}) from the n inputs to the output nodes to small random values. Set the initial neighborhood, N_c to be large. Fix the convergence tolerance limit for the vectors to be a small quantity. Settle maximum number of iterations to be a large number. Divide the training set into vectors of size $n \times n$.

Step 2. Compute the mean and variance of each training input vector.

Step 3. Present the inputs $x_i(t)$.

Step 4. Compute the Euclidean distance d_i between the input and each output node j, given by,

$$d_j = f_j \times d(x, w_{ij}(t)) \tag{6}$$

where f_j is the frequency of the jth neuron being a winner. Select the minimum distance. Designate the output node with minimum d_j to be j^*.

Step 5. Update the weight for node j^* and its neighbors, defined by the neighborhood size N_c. The weights are updated:

if $i \notin N_c(t)$

$$f_{j^*} = f_{j^*} + 1 \tag{7}$$

$$w_{ij^*}(t+1) = w_{ij^*}(t) + \delta_{ij^*}(t+1) \tag{8}$$

$$\delta_{ij^*}(t+1) = \alpha(t+1)(x_i - w_{ij^*}(t)) + \alpha(t+1)\delta_{ij^*}(t) \tag{9}$$

$$\alpha(t+1) = f(e_{j^*}) + \frac{1}{f_{j^*}} \tag{10}$$

$$e_{j^*} = \frac{1}{n}\sum_{i=0}^{n-1} |x_i(t) - w_{ij^*}(t)| \tag{11}$$

if $i \notin N_c(t)$

$$w_{ij}(t+1) = w_{ij}(t) \tag{12}$$

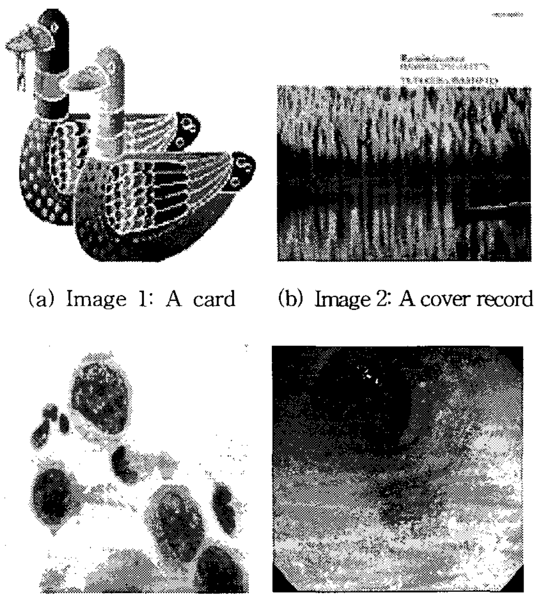
The neighborhood $N_c(t)$ decreases in size as time goes on, thus localizing the area of maximum activity. And $f(e_j)$ is normalization function.

Step 6. Repeat by going to step 2 for each input vector presented, till a satisfactory match is obtained between the input and the weight or till the maximum number of iterations are complete.

4. SIMULATION RESULTS

Simulation was performed on a personal computer using C++ builder to evaluate the proposed method. Digitized color images of the (R, G, B) domain and a resolution of 128×128 were used for the simulation. Fig. 2 shows various images used for simulation.

Based on simulation results, we can see that the proposed method makes the codebook 5 times faster than the LBG algorithm for the vector quantization. Moreover, the performance of the proposed method is better than the conventional VQ algorithm. That is, the proposed method shows higher data compression ratio than the conventional VQ algorithm. The LBG algorithm reads the entire image data several times and moves code vectors into optimal position in each step. This



(a) Image 1: A card (b) Image 2: A cover record

(c) Image 3: A cell image (d) Image 4: Endoscopic image

Fig. 2. Original images used for simulation.

repetitive process shows the block effect by which the image is recovered. Often in an image, adjacent pixels tend to have similar color and compose a color block. This means that one image is composed of such block. We adapt the weight of adjacent pixels of the winner node together in case of generating the initial codebook. In the proposed method, if an input block and the adjacent pixel have a similarity pixel, the neighboring is adapted. The equation for measuring the similarity is as follows:

$$|x_i - x_k| < Escrit \tag{13}$$

where x_k is the adjacent pixel.

That is, if the difference of an input vector from the adjacent pixel is less than *Escrit* which is a criterion for admission, then the neighboring pixel is adapted. Here, the *Escrit* was set to 0.0001. In conventional SOM, only winning neurons and their neighbors participate in learning, for a given vector. Hence, a neighborhood parameter was used in the Kohonen net. The initial values of the gain factor and the neighborhood size were given and were decreased as the input vector cycles continued. The error tolerance limit was fixed at a very low level of *Escrit* = 0.0001 and the error tolerance for all the individual vectors was accumulated. The enhanced self-organizing vector quantization for color images proposed in this paper, can decrease the number of times the entire image data is read. Table 1 show the size of the codebook file for the LBG Algorithm, the conventional SOM and the enhanced self-organizing vector quantization for color images.

Table 1. Size of codebook by VQ (byte)

Images \ Algorithms	LBG	SOM	Enhanced SOM
Image1	30064	32208	31968
Image2	30037	48816	33672
Image3	49376	52080	51648
Image4	50232	54081	53649

To Measure a degree of distortion of the reproduction vector x', a mean square (MSE) is generally used. For the color image, it is defined as follows and the dimension of image vector is an $n \times n$ blocks;

$$MSE = \frac{1}{n \times n} \sum_{k=1}^{RGB} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (x_{ij} - x'_{ij}) \tag{14}$$

where x_{ij} is the input value of the original image, x'_{ij} is the value of reproduction vector, RGB has a value of 3. Table 2, which shows the MSE values of images created by using the LBG algorithm, the conventional SOM and the enhanced SOM and from Fig. 3 through 5. Consequently, the transmission time and the memory space reduced than LBG algorithm.

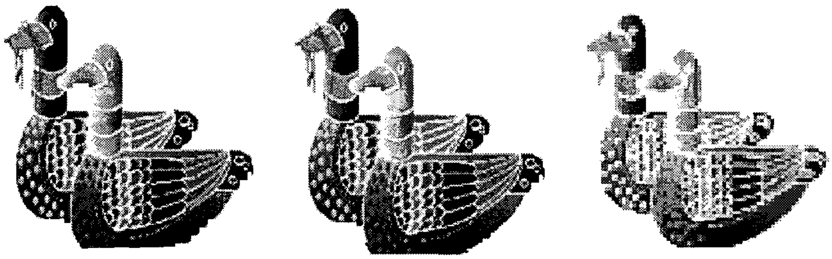
Fig. 3, Fig. 4, Fig. 5 and Fig. 6 show respectively recovered images for original images of Fig. 2. This contribution proposed an improved SOM algorithm. It improves compression and replay rate of image by the codebook dynamic allocation than the conventional SOM algorithm.

Also, for images shown in Fig. 7, the decompression quality of LBG algorithm is worse than the above two algorithms.

LBG algorithm generates 10's temporary codebooks until the creation of the optimal codebook and requires a high computation time for codebook generation. Oppositely, the proposed algorithm generates only one codebook in the overall processing and reduces greatly the computation time and the memory space required for the codebook generation.

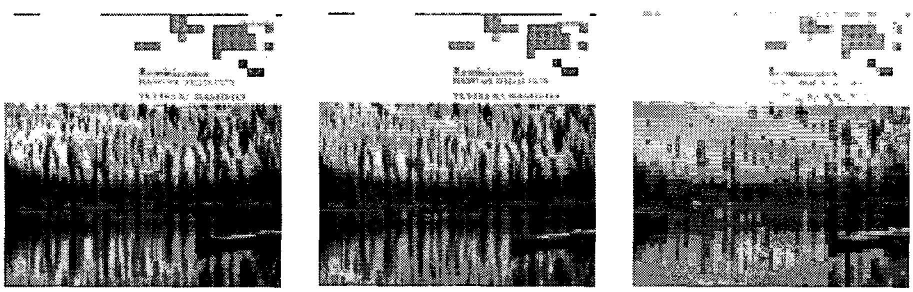
Table 2. Comparison of MSE (Mean Square Error) for compressed images

Images \ Algorithms	LBG	SOM	Enhanced SOM
Image1	15.2	13.1	11.2
Image2	15.6	14.2	13.1
Image3	14.5	11.3	10.8
Image4	15.9	13.8	12.7



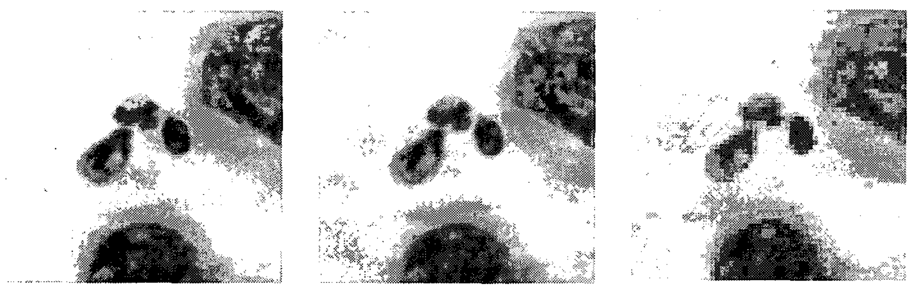
(a) SOM (b) Enhanced SOM (c) LBG Algorithm

Fig. 3. The recovered image for Image1.



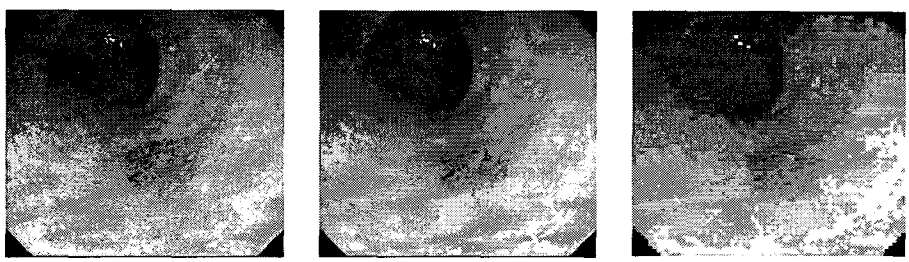
(a) SOM (b) Enhanced SOM (c) LBG Algorithm

Fig. 4. The recovered image for Image2



(a) SOM (b) Enhanced SOM (c) LBG Algorithm

Fig. 5. The recovered image for Image3



(a) SOM (b) Enhanced SOM (c) LBG Algorithm

Fig. 6. The recovered image for Image2

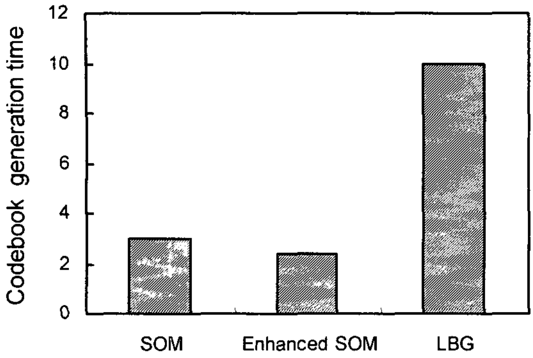


Fig. 7. Comparison of processing time for codebook generation.

5. CONCLUSIONS

The proposed method in this paper can be summarized method as follows: Using the enhanced SOM algorithm, the output error concept is introduced into the weight adaptation and the momentum factor is added. The simulation results show that the enhanced SOM algorithm for color image compression produces a major improvement in both subjective and objective quality of the decompressed images.

Traditionally used, LBG algorithm for codebook generation requires considerable time especially for large size images, since the codebook is generated by repetitive reading of the whole image. The proposed method is apt to real time application because the codebook is created by reading whole image only twice. Generally, the procreation of the codebook is difficult work in vector quantization of color image. Therefore, we propose a new method that uses enhanced SOM learning algorithm to increase the compression and replay ratio. In the future study, we plan to develop a novel VQ using the fuzzy clustering algorithm.

6. REFERENCES

[1] M. Rabbani and P. W. Jones, "Digital Image Compression Technique," Spie Optical Engineering Press, pp.144-169, 1991.

[2] R. M. Gray, Vector Quantization, IEEE ASSP Magazine, Apr., pp.4-29, 1984.

[3] Y. Linde, A. Buzo and R. M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. On Communications, Vol. Com-28, No.1, pp.84-95, Jan., 1980.

[4] A. Gersho, R. M. Gray, "Vector Quantization and Signal Compression," Kluwer Academic Oublishers, Norwell, Massachusetts, 1993.

[5] K. R. L. Godfrey and Y. Attiliuzel, "Self-Organized Color Image Quantization for Color Image Data Compression," Proceedings of IJCNN, Vol.3, pp.162-166, 1993.

[6] L. Jo and M. R. Kaimal, "Image Compression using Self-Organizing Neural Networks," Proceedings of National Seminar on Artificial Neural Networks, and Cognitive System, pp.143-147, 1998.

[7] Liza Jo and M. R. Kaimal, "A Self Organizing Neural Network for Second Generation Image Coding," Proceedings of TAINN, pp. 130-137, 1999.

[8] T. Kohonen, "Improved versions of learning vector quantization," Proceedings of IJCNN, Vol.1, pp.545-550, 1990.

[9] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1992.

[10] S. Seo, M. Bode, K. Obermayer, "Soft Nearest Prototype Classification," IEEE Trans. Neural Networks, Vol.14, No.2, pp.390-398, 2003.

[11] K. L. Oehler and R. M. Gray, "Combining Image Compression and Classification using Vector Quantization," IEEE Multimedia, pp.36-45, 1997.

[12] K. B. Kim, G. H. Kim and S. K. Je, "Medical Image Vector Quantizer using Wavelet Transform and Enhanced SOM Algorithm," Lecture Notes in Artificial Intelligence, LNAI 3339, pp.98-108, 2004.

[13] F. Madeiro, R. M. Vilar, J. M. Fechine, B. G. Aguiar Neto, "A Self-Organizing Algorithm

for Vector Quantizer Design Applied to Signal Processing," International Journal of Neural Systems, Vol.9, No.3, pp.219-226, 1999.

Kwang-Baek Kim

He received M. S. and the Ph.D. degrees in Department of Computer Science from Busan National University, Busan, Korea, in 1993 and 1999, respectively. From 1997 to present, he is an Associate Professor, Department of Computer Engineering, and Silla University in Korea. He is currently an Associate Editor for Korea Journal of Fuzzy Logic and Intelligent Systems. His research interests include Fuzzy Neural Networks and Application, Medical Imaging, Biological Signal Processing and Biomedical System.