Medical Image Vector Quantizer Using Wavelet Transform and Enhanced SOM Algorithm

Kwang-Baek Kim[1], Gwang-Ha Kim[2], and Sung-Kwan Je[3]

[1] Dept. of Computer Engineering, Silla University, Busan, Korea
[2] Dept. of Internal Medicine, Pusan National University College of Medicine, Busan, Korea
[3] Dept. of Computer Science, Pusan National University, Busan, Korea
gbkim@silla.ac.kr

Abstract. Vector quantizer takes care of special image features like edges also and hence belongs to the class of quantizers known as second generation coders. This paper proposes a vector quantization using wavelet transform and enhanced SOM algorithm for medical image compression. We propose the enhanced self-organizing algorithm to improve the defects of SOM algorithm, which, at first, reflects the error between the winner node and the input vector to the weight adaptation by using the frequency of the winner node. Secondly, it adjusts the weight in proportion to the present weight change and the previous weight change as well. To reduce the blocking effect and improve the resolution, we construct vectors by using wavelet transform and apply the enhanced SOM algorithm to them. Our experimental results show that the proposed method energizes the compression ratio and decompression ratio.

1 Introduction

Computer graphics and medical 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. The vector quantization is the well-known method as a component algorithm for loss compression methods, and many loss compression methods are using LBG algorithm for the vector quantization, which was developed by Linde, Buzo, and Gray [2]. But, LBG algorithm is recursive and requires considerable times to get optimal code vectors [3]. The quantization method using the artificial neural network is well suitable to the application that the statistical distribution of original data is changing as time passes, since it supports the adaptive learning to data [4][5]. Also, the neural network has the huge parallel structure and has the possibility for high speed processing. The H/W implementation of vector quantizer using the neural network supports *O(1)*'s codebook search and doesn't require designing the extra structure for codebook.

G.I. Webb and Xinghuo Yu (Eds.): AI 2004, LNAI 3339, pp. 98–108, 2004.

The vector quantization for color image requires the analysis of image pixels for determinating the codebook previously not known, and the self-organizing map (SOM) algorithm, which is the self-learning model of neural network, is widely used for the vector quantization (VQ). However, the vector quantization using SOM shows the underutilization that only some code vectors generated are heavily used [6][7]. This defect is incurred because it is difficult to estimate correctly the center of data with no prior information of the distribution of data.

In this paper, we propose the enhanced SOM algorithm, which, at first, reflects the error between the winner node and the input vector to the weight adaptation by using the frequency of the winner node. Secondly, it adjusts the weight in proportion to the present weight change and the previous weight changes as well. By using the wavelet transform and the proposed SOM algorithm, we implement and evaluate the vector quantization. The evaluation result shows that the proposed VQ algorithm reduces the requirement of computation time and memory space, and improves the quality of the decompressed image decreasing the blocking effect.

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 [3]. 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,

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

where $Y = (x_i' : i = 1,...., N_c)$ 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. 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;

$$MSE = \frac{1}{n \times n} \sum_{k=1}^{RGB} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (x_{ij} - x'_{ij}) \tag{2}$$

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.

2.2 LBG Algorithm

LBG algorithm, which was proposed by Linde, Buzo, and Gray, is most representative among the codebook generation algorithms. This algorithm generalizes the scalar quantizer designed by Lloyd and called GLA (Generalized Llyod Algorithm)[3]. LBG algorithm is mapped to representation vector with minimum distortion in all input vectors, and calculating the average distortion degree. LBG algorithm generates the optimal codebook according to the following steps: *Step 1*. all input vectors are mapped to code vectors with minimum error in the codebook. *Step 2*. the mean square of error between input vectors and code vector is calculated. *Step 3*. the error value is compared with the given allowance. If the error value is greater than the allowance, the current codebook is adjusted by recalculating the centers of code vectors. Otherwise, the current codebook is determined to be optimal. The selection of the initial codebook is important to generate the optimal codebook in LBG algorithm. The selection methods of the initial codebook are the random selection method, the separation method that divides all learning vectors to 2 groups, and the merging method that merges two adjacent vectors repeatedly from N clusters to the N_c 's codebook. Without regard to the selection of the initialization method, LBG algorithm scans repeatedly all image vectors to generate the optimal codebook and requires the high computation time for images of large size.

2.3 Vector Quantization Using Neural Network

Self-Organizing Map (SOM) is widely applied in the vector quantization, which is the self-learning method among neural network algorithms [7][8]. The 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. The process adaptively adjusting the weight of the stored pattern in SOM algorithm is the same as the process generating dynamically a code vector in the codebook for the given input vector in the vector quantization. Therefore, the vector quantization using SOM algorithm generates the codebook dynamically for color images.

However, from practical experience, it is observed that additional refinements are necessary for the training algorithm to be efficient enough for practical applications [9]. And with no information of the distribution of training vectors, the vector quantization using SOM algorithm selects randomly the initial code vectors and progresses the adaptive learning. Therefore, this adaptive VQ algorithm generates code vectors never used after the initial codebook generation, incurring the underutilization of code vectors.

3 Medical Image Vector Quantizer Using Wavelet Transform and Enhanced SOM Algorithm

In this paper, we apply the wavelet transform and the enhanced SOM algorithm sequentially to images, generating the codebook for the compression of image as shown in Fig. 1. The vector quantization using the traditional SOM algorithm incurs the underutilization of code vectors. And, for the improvement of this defect, we propose the enhanced SOM algorithm that reflects the frequency of winner node for each class, the previous change of weight as well as the difference between input vector and winner node to the weight adaptation. The application of the enhanced SOM algorithm to the wavelet-transformed image reduces computation time and memory space for the codebook generation and lightens the blocking effect incurred by the insufficient size of codebook.

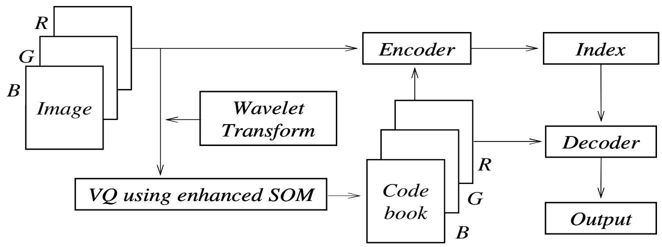


Fig. 1. The processing structure of the proposed vector quantizer

3.1 Enhanced SOM Algorithm

In this paper, we propose the enhanced SOM algorithm for the vector quantization that is able to generate the codebook in real-time and provide the high recovery quality. The generation procedure of codebook using enhanced SOM algorithm is showed in Fig. 2 and Fig. 3.

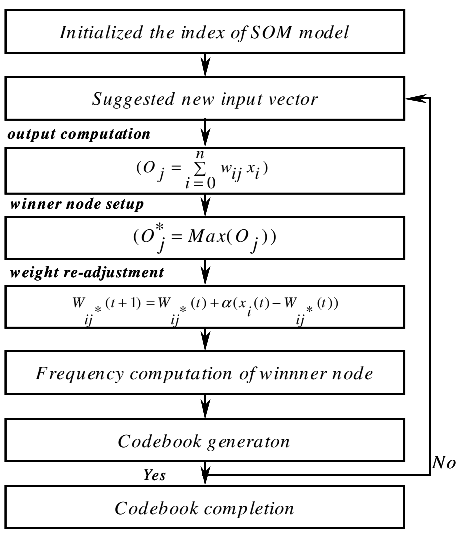


Fig. 2. The enhanced SOM algorithm for the generation of initial codebook

In this paper, we improved the SOM algorithm by employing three methods for the efficient generation of the 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.

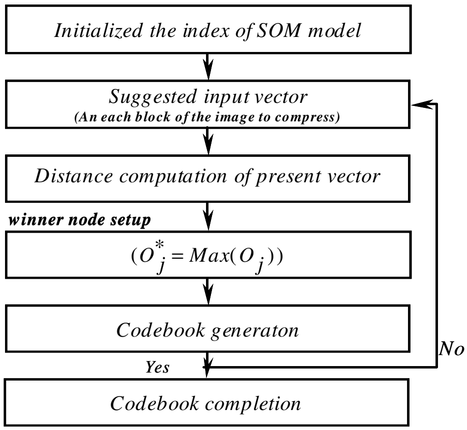


Fig. 3. The procedure of index determination and codebook generation for each block

In the proposed method, the codebook is generated by scanning 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 frequency to be selected as winner node and the distortion for the selection of the winner node in the competitive learning algorithm are needed. We use the following equation in the weight adaptation.

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

where 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 is, the more the amount for the weight adaptation is. Therefore, the weight is adapted in proportion to the size of the output error. $f(e_j)$ 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 j-th neuron, and f_j is the frequency for the j-th neuron as the winner.

The above method considers only the present change of weight and does not consider the previous change. 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 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_j 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:

$if\ i \in N_c(t)$

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

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

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

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

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

$if\ i \notin N_c(t)$

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

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.

3.2 Application of Wavelet Transform

In this paper, for the proposed SOM algorithm, we apply a wavelet transform to reduce the block effect and to improve the decompression quality. After the wavelet transforms the color image, the color image is compressed by applying the vector quantization using the enhanced SOM algorithm to each separated RGB values. That is, by applying the wavelet transforms to the image, input vectors are generated, and the enhanced SOM algorithm are applied to the input vectors. If the index of the winner node corresponding to the input vector is found, the original image vector corresponding to the transformed input vector is stored in the codebook. Wavelet transform is applied to the original image in the vertical and horizontal direction of a low frequency prior to the codebook generation. Specially, the image information of the original resolution is maintained without the down sampling used in the existing wavelet transform. Using the low frequency pass filter of wavelet emphasizes the strong areas of image and attenuates weak areas, have an equalization effect and remove the noise. Fig. 4 shows the structure of wavelet transform [10]. Fig. 5 shows the example of the filters in high frequency and low frequency.

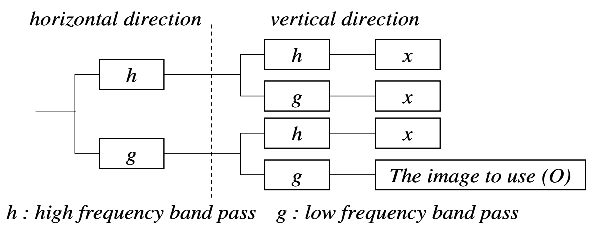


Fig. 4. The structure of wavelet transforms

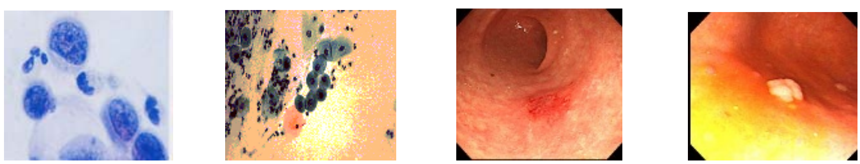
$$\begin{bmatrix} 1+\sqrt{3} & 3+\sqrt{3} \\ 3-\sqrt{3} & 1-\sqrt{3} \end{bmatrix} \times 4\sqrt{2} \qquad \begin{bmatrix} 1-\sqrt{3} & \sqrt{3}-3 \\ 3+\sqrt{3} & -1-\sqrt{3} \end{bmatrix} \times 4\sqrt{2}$$

(a) Low-frequency band pass filter (b) High-frequency band pass filter

Fig. 5. The filters used in wavelet transform

4 Simulation Results

An experiment environment was implemented on an IBM 586 Pentium III with C++ Builder. The image to be used experiment is a color bitmap images of 128×128 pixel size. The image is divided into blocks of 4×4 size and each block is represented by the vector of 16 bytes, which constitutes the codebook. In this paper, the proposed VQ algorithm and LBG algorithm are compared in performance. In the case of the codebook generation and image compression, the vector quantization using the enhanced SOM algorithm improves 5 times in the computation time than LBG algorithm and generates the codebook by scanning all image vectors only two times. This reduces the requirement of memory space. The application of the wavelet transform lightens the block effect and improves the recovery quality. Fig. 6 shows medical color images used in the experiment. Although the proposed algorithm can be applied to grayscale images, we selected various medical color images for this experiment because the proposed vector quantization algorithm is for the color medical image.



(a) Cell image (b) Cancer image (C) Endoscopic image 1 (d) Endoscopic image 2

Fig. 6. Medical image samples used for experiment

Table 1 shows the size of codebooks generated by LBG algorithm, SOM algorithm, enhanced SOM and the integration of wavelet transform and enhanced SOM for images in Fig. 6. In Table 1, the proposed integration of wavelet transform and enhanced SOM algorithm shows a more improved compression ratio than other methods. In the case of image 2 which the distribution of color is various, the compression ratio is low compared with different images. For the comparison of decompression quality, we measure the mean square error (MSE) between the original image and the recovered image, and presented in Table 2 the MSE of each image in the three algorithms.

Table 1. Size of codebook by VQ (unit: byte)

Images \ Algorithms	LBG	SOM	Enhanced SOM	Wavelet and Enhanced SOM
Cell Image	49376	52080	51648	27365
Cancer Cell Image	50357	53213	52347	30645
Endoscopic image 1	50232	54081	53649	28377
Endoscopic image 2	50125	54032	53591	28321

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

Algorithms / Images	LBG	SOM	Enhanced SOM	Wavelet and Enhanced SOM
Cell Image	14.5	11.3	10.8	9.1
Cancer Cell Image	15.1	14.1	13.2	11.2
Endoscopic image 1	14.9	13.8	12.7	10.6
Endoscopic image 2	14.8	13.6	12.4	10.3

As shown in Table 2, the integration of wavelet transform and enhanced SOM algorithm shows the lowest MSE. Also, for images shown in Fig. 7, the decompression quality of LBG algorithm is worse than the above three algorithms.

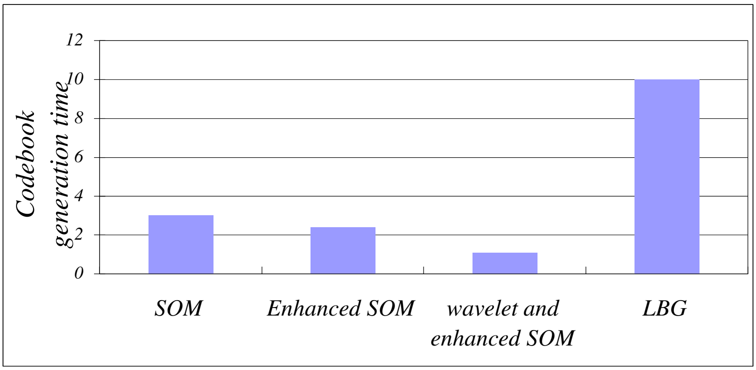
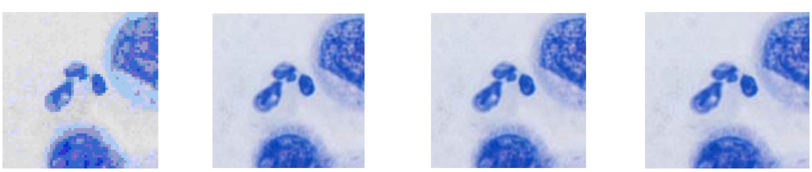


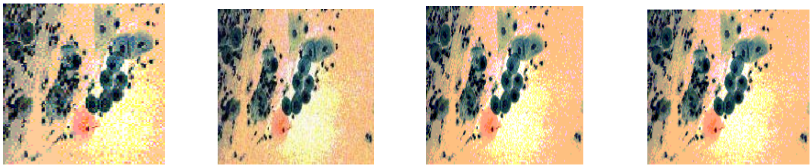
Fig. 7. Comparison of processing time for codebook generation

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. Fig.8, Fig.9, Fig.10 and Fig.11 show respectively recovered images for original images of Fig.6. The enhanced SOM algorithm improves the compression ratio and the recovery quality of images by the codebook dynamic allocation more than the conventional SOM algorithm.



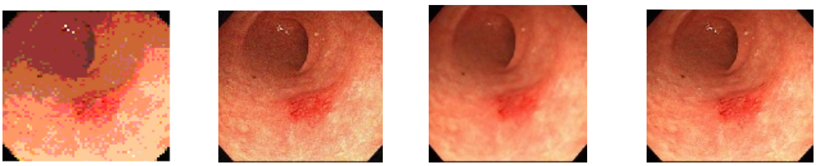
(a) LBG (b) SOM (c) Enhanced SOM (d) Wavelet and Enhanced SOM

Fig. 8. The recovered image for cell image



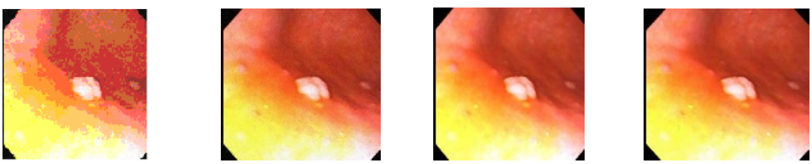
(a) LBG (b) SOM (c) Enhanced SOM (d) Wavelet and Enhanced SOM

Fig. 9. The recovered image for cancer cell image



(a) LBG (b) SOM (c) Enhanced SOM (d) Wavelet and Enhanced SOM

Fig. 10. The recovered image for endoscopic image 1



(a) LBG (b) SOM (c) Enhanced SOM (d) Wavelet and Enhanced SOM

Fig. 11. The recovered image for endoscopic image 2

5 Conclusion

The proposed method can be summarized 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 the medical color image compression produces a major improvement in both subjective and objective quality of the decompressed images. LBG algorithm is traditionally used for the codebook generation and requires considerable time especially for large size images, since the codebook is generated by repetitive scanning of the whole image. The proposed method is apt to real time application because the codebook is created by scanning the whole image only twice. The enhanced SOM algorithm performs the learning in two steps and total learning vectors are used only once in each step. In the first step, it produces the initial codebook by reflecting the distribution of learning vectors well. In the second step, it produces the optimal codebook by shifting to the current center of each code group based on the initial codebook. For reducing the memory space and the computation time for the codebook generation, we construct vectors by using wavelet transform and we apply the enhanced SOM algorithm to them. The simulation results showed that the integration of the wavelet transform and the enhanced SOM algorithm improves the defects of vector quantization such as the time and memory space caused by the complex computation and the block effect.

References

1. Rabbani M., and Jones P. W.: Digital Image Compression Technique. Spie Optical Engineering Press (1991) 144-169
2. Linde Y., Buzo A., and Gray R. M.: An Algorithm for Vector Quantizer Design. IEEE Trans. On Communications. **1(1)** (1980) 84-95
3. Gray R. M.: Vector Quantization. IEEE ASSP Magazine. (1984) 4-29
4. Kohonen T.: Improved versions of learning vector quantization. Proceedings of IJCNN. **(1)** (1990) 545-550
5. Godfrey K. R. L., and Attikiouzel, Y.: Self-Organized Color Image Quantization for Color Image Data Compression. Proc. of ICNN. **(3)** (1993) 1622-1626
6. Kim K. B., and Cha E. Y.: A Fuzzy Self-Organizing Vector Quantization for Image. Proceedings of IIZUKA. **(2)** (1996) 757-760
7. Madeiro F., Vilar R. M., Fechine J. M., and Aguiar Neto B. G.: A Self-Organizing Algorithm for Vector Quantizer Design Applied to Signal Processing. International Journal of Neural Systems. **9(3)** (1999) 219-226
8. Oehler K. L., and Gray R. M.: Combining Image Compression and Classification using Vector Quantization. IEEE Multimedia. (1997) 36-45
9. Seo S., Bode M., and Obermayer K.: Soft Nearest Prototype Classification. IEEE Trans. Neural Networks. **14(2)** (2003) 390-398
10. Strang G., and Nguyen T.: Wavelets and Filter Banks. Wellesley-Cambridge Press (1996)