



A competitive continuous Hopfield neural network for vector quantization in image compression

J.-S. Lin *, S.-H. Liu

Department of Electronic Engineering, National Chin-Yi Institute of Technology, Taichung, Taiwan, Republic of China

Received 1 March 1997; accepted 1 October 1998

Abstract

Vector quantization is a system in which a distortion function is minimized for multidimensional optimization problems. The purpose of such a system is data compression. In this paper, a parallel approach using the competitive continuous Hopfield neural network (CCHNN) is proposed for the vector quantization in image compression. In CCHNN, the codebook design is conceptually considered as a clustering problem. Here, it is a kind of continuous Hopfield network model imposed by the winner-take-all mechanism, working toward minimizing an objective function that is defined as the average distortion measure between any two training vectors within the same class (within-class). It also forward maximizes an objective function defined as the average distortion measure between any two training vectors in separate classes (between-class). For an image of n training vectors and c objects of interest, the proposed CCHNN would consist of $n \times c$ neurons. Each neuron (or training vector) occupies $l \times l$ components of a training vector. In the experimental results, the proposed method shows more promising results after convergence than the generalized Lloyd algorithm. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Image compression; Vector quantization; Hopfield neural networks; Competitive learning; Scatter matrices

1. Introduction

The idea of the vector quantization algorithm, which shows that a better compression performance can always be completed by coding vectors instead of scales, comes from Shannon's rate-distortion theory in information theory. Vector quantization is an approach that maps analog signals or discrete vectors into a sequence of digital signals for communication or storage in a channel. The goal of vector quantization is the creation of a codebook, for which the average distortion generated by approximating a training vector by a codevector in the codebook is minimized. The minimization of the average distortion measure is generally achieved by a gradient-descent-based iterative procedure which is called the generalized Lloyd algorithm (GLA) (Lloyd, 1982). In accordance with the cluster center in the previous iteration and the nearest-neighbor rule, the GLA performs a positive improvement, to update the codebook at every iteration.

A great many vector-quantization approaches have been proposed for image compression by several researchers—Linde et al. (1980), Gray (1984), Chang et al. (1988), Netravali (1988), Gersho and Gray (1992), Riskin et al. (1990), Yair et al. (1992), Zeger et al. (1992), and Andrew and Palaniswami (1996). In addition, neural-network-based techniques have been used to address vector quantization. Yair et al. (1992) presented vector-quantizer techniques incorporating stochastic relaxation with a competitive learning strategy and a 'soft' competitive scheme. Instead of the GLA, which is a batch method, competitive learning is an on-line approach, to update the codebook whenever a training vector is presented. Another vector-quantized design algorithm, which incorporates simulated annealing and the Lloyd iteration, was proposed by Zeger et al. (1992). In their algorithm, training vectors move randomly between neighboring clusters until thermal equilibrium is reached. However, reaching thermal equilibrium at a low temperature might take a very long time in the simulated annealing (SA) algorithm. In addition, the annealing network is a difficult

* Corresponding author. Tel.: +886-4-3924505 ext. 7311; fax: 886-4-3926610; e-mail: jslin@chinyi.nctit.edu.tw.

technique, because several penalty terms need to be adjusted for the purpose of generating good solutions.

In this paper, a vector-quantizer design approach, based on the continuous Hopfield neural-network model with competitive learning for image compression, is presented. The whole image is divided into n blocks (a block represents a training vector which occupies $l \times l$ components) and mapped on to a two-dimensional Hopfield neural network which consists of a grid of $n \times c$ neurons, with each row representing a training vector, and each column representing a codevector of the codebook. The energy function used for the CCHNN is formulated on the basis of within-class or between-class scatter matrices, a concept widely used in pattern classification (Fukunaga, 1972). That is, the energy function, to be called the ‘scatter energy function’, is formed by a within-class (intra-class) or between-class (inter-class) scatter energy function. In order to achieve the best classification, the scatter energy function should be minimized in such a fashion that either the intra-class scatter energy is minimized, or the inter-class scatter energy is maximized. With these structures, the image compression based on vector quantization can then be regarded as an optimization problem, defined as the scatter energy function depicted in this network. The training vectors are then partitioned into different feasible classes when the scatter energy function has converged. Then, the codebook is just updated with the centers of the classes after the last iteration.

This paper is organized as follows. Section 2 demonstrates the general properties for VQ design in image compression. In Section 3, the properties of a vector-quantizer design approach using the CCHNN algorithm based on within-class and between-class scatter matrices, along with the competitive learning, are described. Section 4 presents several experimental results. Section 5 gives the discussion and conclusions. Finally, the minimized intra-class Euclidean distance is equivalent to maximizing the inter-class Euclidean distance, and the convergence of the CCHNN based on within-class and between-class scatter matrices is also proved in the Appendix.

2. Vector quantization

A vector quantizer is a technique that maps a Euclidean $l \times l$ -dimensional space $\mathbf{R}^{l \times l}$ into a set $\{\mathbf{Y}_x, x = 1, 2, \dots, n\}$ of points in $\mathbf{R}^{l \times l}$, called a codebook. A vector quantizer approximates a training vector from $\mathbf{R}^{l \times l}$ with as little distortion as possible by one of the codevectors in the codebook. Suppose an image is divided into n blocks (vectors of pixels) and each block occupies $l \times l$ pixels. The performance of a system by an average distortion $E[d(\mathbf{X}_x, \mathbf{Y}_x)]$ between an

input sequence of training vectors $\{\mathbf{X}_x, x = 1, 2, \dots, n\}$ and an output sequence of codevectors is defined as

$$D_{x,y} = E[d(\mathbf{X}_x, \mathbf{Y}_x)] = \frac{1}{n} \sum_{x=1}^n d(\mathbf{X}_x, \mathbf{Y}_x) \quad (1)$$

The distortion measure $d(\mathbf{X}_x, \mathbf{Y}_x)$, the squared Euclidean distance between vectors, is defined as

$$d(\mathbf{X}_x, \mathbf{Y}_x) = \|\mathbf{X}_x - \mathbf{Y}_x\|^2 = \sum_{k=1}^{l \times l} (x_k - y_k)^2 \quad (2)$$

A vector quantizer is optimal if the average distortion converges to a minimum value. The nearest neighbor for the clustering condition and the centroid condition for the codevectors, used by Linde et al. (1980) to generalize the Lloyd algorithm (Lloyd, 1982), are necessary conditions on the encoder and decoder of a vector quantizer. The GLA iteratively decreases the average distortion for a given training vector. At each iteration, the GLA produces a new quantizer that satisfies the necessary centroid condition and obtains an average distortion that is less than or equal to that obtained in the previous iteration. In the following section, the vector quantization using the proposed algorithm will be discussed.

3. Vector quantizer using competitive continuous Hopfield neural network

The Hopfield neural network has occupied the attention of several investigators—Hopfield (1982), Hopfield and Tank (1982), Cheng et al. (1996), Chung et al. (1994), Yang and Dillon (1994), Tsai et al. (1993), Washizawa (1993), Steck and Balakrishnan (1994), Amatur et al. (1992), and Kim et al. (1992), with its features of a simple architecture and the potential for parallel implementation, has been studied extensively during the past decade. An application of the competitive Hopfield neural network to medical image segmentation was described by Cheng et al. (1996). Polygonal approximation using a competitive Hopfield neural network was demonstrated by Chung et al. (1994). Cheng et al. (1996) and Chung et al. (1994) used the winner-take-all rule to adopt in the two-dimensional discrete Hopfield neural network, to eliminate the need for finding weighting factors in the energy function. The network consists of $n \times c$ neurons, which are fully interconnected neurons. Let $V_{x,i}$ denote the output of neuron (x, i) and $T_{x,i,y,j}$ be the interconnection weight between the neuron (x, i) and the neuron (y, j) . A neuron (x, i) receives each neuron (y, j) with $T_{x,i,y,j}$ and a bias $I_{x,i}$ from exterior. The two-dimensional fully connected Hopfield neural network consists of massive interconnected neurons in the application of

optimization problem. The network minimizes a continuous-model energy function defined by

$$E = -\frac{1}{2} \sum_x \sum_y \sum_i \sum_j T_{x,i;y,j} V_{x,i} V_{y,j} + \frac{1}{\lambda} \sum_x \sum_i \frac{1}{R_{x,i}} \int_0^{V_{x,i}} g^{-1}(\xi) d\xi - \sum_x \sum_i I_{x,i} V_{x,i} \quad (3)$$

where λ is a constant called the gain parameter, and g is the sigmoid function. An image is divided into n blocks (a block represents a training vector that occupies $l \times l$ components) and mapped to a two-dimensional Hopfield neural network. In this case, for an image of n training vectors and c classes of interest (the class centroid is mapped into appropriate codevectors in the codebook), the proposed CCHNN would consist of $n \times c$ neurons. In the application of the classification procedure, the intraset (within-class) distance should be small, whereas the interset (between-class) distance should be large. These two criteria satisfy the nearest-neighbor condition. Instead of iteratively updating the codevectors in the codebook using GLA, the proposed CCHNN updates the codevectors just once after the last iteration, when the energy function has converged. The two-dimensional Hopfield network structures based on within-class and between-class scatter matrices are discussed below.

3.1. The within-class scatter energy

Using the within-class scatter matrix criteria, a two-dimensional Hopfield neural network can be modified to a partially interconnected structure that is only fully interconnected between neurons in the same column (the same class). The energy function, defined in Eq. (3), can be modified as

$$E^w = -\frac{1}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^c T_{x,i;y,i} V_{x,i} V_{y,i} + \frac{1}{\lambda} \sum_{x=1}^n \sum_{i=1}^c \frac{1}{R_{x,i}} \int_0^{V_{x,i}} g^{-1}(\xi) d\xi - \sum_{x=1}^n \sum_{i=1}^c I_{x,i} V_{x,i} \quad (4)$$

where n is the number of training vectors, and c is the number of classes. The dynamic equation for the neurons is modified as

$$C_{x,i} \frac{dU_{x,i}}{dt} = -\frac{U_{x,i}}{R_{x,i}} + \sum_{y=1}^n T_{x,i;y,i} V_{y,i} + I_{x,i} \quad (5)$$

All of the terms in $T_{x,i;y,i}$ contain arbitrary constants,

and $I_{x,i}$ can be adjusted to any desired value. The input to each neuron can be iteratively updated by

$$U_{x,i}(t+1) = U_{x,i}(t) + \frac{dU_{x,i}(t)}{dt} \quad (6)$$

The input–output function for the x th row is given by

$$V_{x,j}(t+1) = \begin{cases} 1, & \text{if } i = \arg\{\max_k(U_{x,i}(t))\} \\ 0, & \text{if } j \neq i \end{cases} \quad (7)$$

Amatur et al. (1992) pointed out that the monotonicity property of the maximum neuron is equivalent to a MacCulloch–Pitts neuron with a dynamic threshold equal to $U_{x,i}$. Because T is symmetric ($T_{x,i;y,i} = T_{y,i;x,i}$), the partial derivative $\partial E^w / \partial V_{x,i}$ may be calculated by

$$\frac{\partial E^w}{\partial V_{x,i}} = -\sum_{y=1}^n T_{x,i;y,i} V_{y,i} + \frac{1}{R_{x,i}} U_{x,i} - I_{x,i} = -C_{x,i} \frac{dU_{x,i}}{dt} \quad (8)$$

The objective function for describing the vector quantizer of the image compression based upon the average squared Euclidean distance between any two training vectors in the same class is defined as

$$E^w = \frac{A}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^c \frac{1}{\sum_{h=1}^n V_{h,i}} V_{x,i} D_{x,y} V_{y,i} + \frac{B}{2} \sum_{x=1}^n \sum_{i=1}^c \sum_{j=1}^c V_{x,i} V_{x,j} + \frac{C}{2} \left[\left(\sum_{x=1}^n \sum_{i=1}^c V_{x,i} \right) - n \right]^2 \quad (9)$$

where $\sum_{h=1}^n V_{h,i}$ is the number of training vectors in class i , and $D_{x,y}$, defined in Eq. (1), indicates the average distortion function between training vectors \mathbf{X} and \mathbf{Y} . The first term of Eq. (9) is the average distortion over c classes. The second term makes sure that no training vector can be assigned to more than one class in the final classification. The third term guarantees that the n training vectors can only be distributed among these c classes. More specifically, the last two penalty terms impose constraints on the objective function, and the first term serves to minimize average intra-class distortion from one training vector to another in one class. As mentioned by Chung et al. (1994), the quality of the classification is very sensitive to the weighting factors A , B and C . Searching for optimal values of these weighting factors is expected to be time-consuming and laborious. In order to alleviate this problem, a continuous Hopfield neural network with a competitive strategy is proposed, so that the penalty terms can be handled in a very efficient manner. All the neurons on the same row compete with

one another to determine which neuron is the winner in receiving the maximum input. According to Eq. (7), the winning neuron sets its output to 1, and all the other neurons on the same row are set to 0. In other words, only the winning neuron has output 1, and the outputs of the rest of neurons are set to 0. The competitive learning guarantees that each row has only one neuron with value 1, and this is the winning neuron. It also ensures that only n training vectors will be classified into these c classes. The competitive continuous Hopfield neural network, CCHNN, enables the scatter energy function to converge rapidly to a minimum value. By including Eq. (7) in the objective energy function, the within-class scatter energy and the weights of the connections can be further simplified as

$$E^w = \frac{1}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^c \frac{1}{\sum_{l=1}^n V_{l,i}} V_{x,i} D_{x,y} V_{y,i} \quad (10)$$

and

$$T_{x,i;y,i} = -\frac{1}{\sum_{l=1}^n V_{l,i}} D_{x,y} \quad (11)$$

By inserting Eq. (10) into Eq. (8) to simplify the mapping, Eq. (8) can be modified as follows:

$$\frac{dU_{x,i}}{dt} = -\frac{1}{2} \sum_{y=1}^n \frac{1}{\sum_{l=1}^n V_{l,i}} D_{x,y} V_{y,i} \quad (12)$$

The CCHNN-based vector quantizer with a within-class scatter matrix is thus performed as follows:

1. Initialize the input of neurons and codevectors in the codebook to random values.
2. Apply the input-output relation given in Eq. (7) to obtain the new output for each neuron.
3. Iteratively calculate the dynamic Eqs. (6) and (12) for each neuron.
4. Go back to step 2 until the network converges.
5. Update the codevectors using the centroids of the c classes.

3.2. The between-class scatter energy

A two-dimensional partially interconnected Hopfield neural network, in which the neurons are fully interconnected except for the neurons in the same column, is used with between-class matrix criteria. The energy function, described in Eq. (3), can also be rewritten as

$$\begin{aligned} E^b = & -\frac{1}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^c \sum_{\substack{j=1 \\ j \neq i}}^c T_{x,i;y,j} V_{x,i} V_{y,j} \\ & + \frac{1}{\lambda} \sum_{x=1}^n \sum_{i=1}^c \frac{1}{R_{x,i}} \int_0^{V_{x,i}} g^{-1}(\xi) d\xi \\ & - \sum_{x=1}^n \sum_{i=1}^c I_{x,i} V_{x,i} \end{aligned} \quad (13)$$

In Eq. (13), $T_{x,i;y,j}$ is also symmetric (i.e. $T_{x,i;y,j} = T_{y,j;x,i}$). The dynamic equation, and $\partial E^b / \partial V_{x,i}$ based on the between-class scatter matrix for the neurons, are

$$C_{x,i} \frac{dU_{x,i}}{dt} = -\frac{U_{x,i}}{R_{x,i}} + \sum_{y=1}^n \sum_{\substack{j=1 \\ j \neq i}}^c T_{x,i;y,j} V_{y,j} + I_{x,i} \quad (14)$$

and

$$\begin{aligned} \frac{\partial E^b}{\partial V_{x,i}} = & -\sum_{y=1}^n \sum_{\substack{j=1 \\ j \neq i}}^c T_{x,i;y,j} V_{y,j} + \frac{1}{R_{x,i}} U_{x,i} - I_{x,i} \\ = & -C_{x,i} \frac{dU_{x,i}}{dt} \end{aligned} \quad (15)$$

respectively. The objective function for describing the vector quantizer in image compression based upon the average squared Euclidean distance between any two training vectors in different classes is defined as

$$\begin{aligned} E^b = & -\frac{A}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^c \sum_{\substack{j=1 \\ j \neq i}}^c \frac{1}{\sum_{l=1}^n V_{l,j}} V_{x,i} D_{x,y} V_{y,j} \\ & + \frac{B}{2} \sum_{x=1}^n \sum_{i=1}^c \sum_{j=1}^c V_{x,i} V_{x,j} \\ & + \frac{C}{2} \left[\left(\sum_{x=1}^n \sum_{i=1}^c V_{x,i} \right) - n \right]^2 \end{aligned} \quad (16)$$

In the same way as was described in the previous paragraph, the between-class scatter energy and the weights of connections can be further simplified as

$$E^b = -\frac{1}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^c \sum_{\substack{j=1 \\ j \neq i}}^c \frac{1}{\sum_{l=1}^n V_{l,j}} V_{x,i} D_{x,y} V_{y,j} \quad (17)$$

and

$$T_{x,i;y,j} = \frac{1}{\sum_{l=1}^n V_{l,j}} D_{x,y} \quad (18)$$

By inserting Eq. (17) into Eq. (15), Eq. (15) can be modified as follows:

$$\frac{dU_{x,i}}{dt} = \frac{1}{2} \sum_{y=1}^n \sum_{\substack{j=1 \\ j \neq i}}^c \frac{1}{\sum_{l=1}^n V_{l,j}} D_{x,y} V_{y,j} \quad (19)$$

Thus, the CCHNN-based vector quantizer with a between-class scatter matrix is performed as follows:

1. Initialize the input of neurons and codevectors in the codebook to random values.
2. Apply the input-output relation given in Eq. (7) to obtain the new output for each neuron.
3. Iteratively calculate the dynamic Eqs. (6) and (19) for each training vector.
4. Go back to step 2 until the network converges.
5. Update the codevectors using the centroids of the c clusters.

This describes a fact, shown in the Appendix and pointed out by Fukunaga (1972), that minimizing the intra-class Euclidean distance is equivalent to maximizing the inter-class Euclidean distance, and that it is also equivalent to simultaneously minimizing the intra-class Euclidean distance and maximizing the inter-class Euclidean distance. This fact will also be proved in the experimental results. It is also shown in the appendix that the energy functions of these two CCHNNs must converge to the minimization as time progresses. Consequently, the simpler scheme for the CCHNN based on the within-class scatter matrix can be selected as the network for the vector quantizer.

4. Experimental results

In this paper the performance for the GLA and the proposed CCHNN algorithm were compared for real images. The codebook design is the primary problem in image compression based on vector quantization. The training vectors were extracted from 256×256 real images, which were divided into 4×4 and 8×8 blocks to generate 4096 and 1024 nonoverlapping 16- and 64-dimensional vectors respectively. Three codebooks, of size 64, 128 and 256, were built using this training data. Fig. 1 shows the original and reconstructed images. The original images are 256×256 pixels, with eight-bit gray levels. In this experiment the compression rate were $8/16 = 0.5$ bits and $8/64 = 0.125$ bits per pixel. The peak signal to noise ratio (PSNR), which is defined for an $N \times N$ image as follows, was evaluated in the reconstructed images.

$$\text{PSNR} = 10 \log_{10} \frac{255 \times 255}{\frac{1}{N^2} \sum_{x=1}^N \sum_{y=1}^N (f_{xy} - \hat{f}_{xy})^2} \quad (20)$$

where f_{xy} and \hat{f}_{xy} are the pixel gray levels from the original and the reconstructed images, and 255 is the peak gray level. Pictures (d)–(f) and (g)–(i) show the images reconstructed from the codebook design by the GLA algorithm and the proposed CCHNN, based on within-class or between-class scatter energies

with $k = 256$, respectively. In the ‘Lena’ image a, d and g, the PSNRs completed by the GLA and the CCHNN algorithms with a codebook of size $k = 256$ and a compression ratio $CR = 0.500$ were 26.35 and 29.63 dB, respectively. In the MR brain image c, f and i, the PSNRs completed by the GLA and the CCHNN algorithms with codebook size $k = 256$ and a compression ratio $CR = 0.500$ were 26.33 and 31.37 dB, respectively. In the boy–girl image b, e and h, the PSNRs completed by the GLA and the CCHNN algorithms with codebook size $k = 256$ and compression ratio $CR = 0.125$ were 25.17 and 30.43 dB, respectively. From the experimental results, the reconstructed images obtained from the CCHNN based on the within-class scatter energy are identical to those obtained from the CCHNN based on the between-class scatter energy. Table 1 shows the PSNRs of the images reconstructed from the various codebooks and different compression ratios using the GLA and the CCHNN algorithms. In accordance with Table 1, the proposed CCHNN algorithm consistently produces better results than those designed by the GLA algorithm, no matter what the compression ratio and codebook size are. Although the JPEG, a standard approach in real applications, can produce a better PSNR, the compression ratio is lower than in the CCHNN algorithm. In summary, from the experimental results, it is seen that the proposed algorithm could satisfactorily produce the codebook design, while the network convergence is guaranteed. In fact however, the Hopfield network may settle down to a local minimum after the network converges. This can be avoided by incorporating the technique of simulated annealing in the network (Kim et al., 1992).

From the experiments, the CCHNN usually takes 26–34 loops on average, to stabilize. Here, a loop is considered as the updating of all rows of the CCHNN.

5. Conclusion

In this paper, an approach using the competitive continuous Hopfield neural network imposed by a winner-take-all mechanism has been proposed to perform codebook design in image compression based on vector quantization. Instead of iteratively updating the codebook in a conventional algorithm such as GLA, the codebook is updated just once after the last iteration in the CCHNN. In the mathematical derivative and experimental results, it has also been demonstrated that the reconstructed images, obtained from the CCHNN based on within-class scatter energy, are identical to those obtained from the CCHNN based on the between-class scatter energy. In addition, this proposed algorithm lends itself admirably to parallel

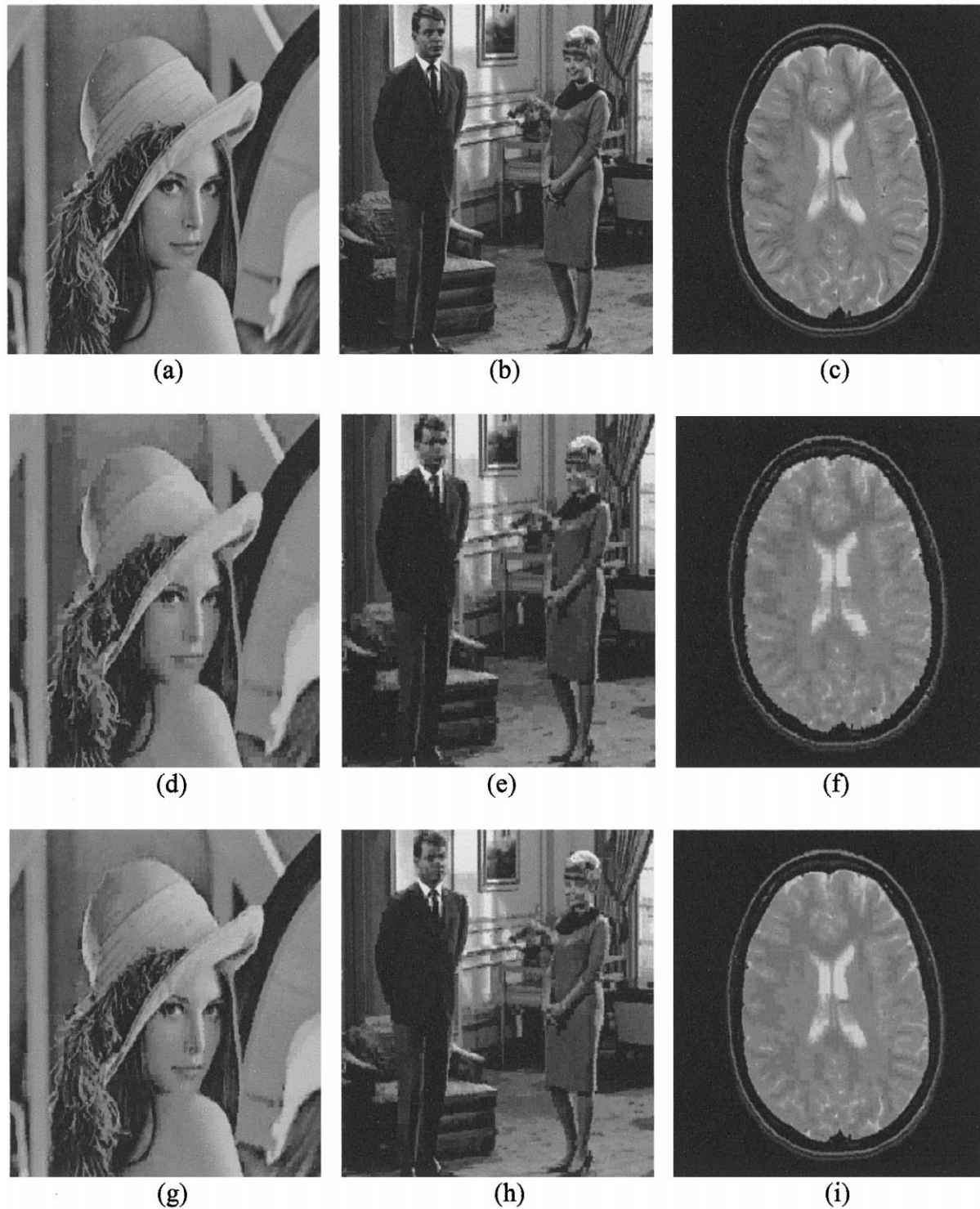


Fig. 1. Original images and reconstructed images with compression ratio = 0.500 bpp using the GLA algorithm and the CCHNN algorithm based on within-class or between-class scatter energies: (a)–(c) are the original images; (d)–(f) reconstructed images using the GLA algorithm and (g)–(i) are reconstructed images using the CCHNN algorithm from a codebook of size $k = 256$.

implementation, and has great potential in real-time applications.

Acknowledgements

This work is in part supported by the National Science Council, Republic of China, under grant no. SC87-2213-E-167-004, and in part by the image compression research group, National Chin-Yi Institute of Technology, Taichung, Taiwan.

Appendix A

Comparing the dynamic Eqs. (12) and (19), the CCHNN based on the between-class scatter matrix is so complicated that it requires enormously more computing time than those based on the within-class scatter matrix to obtain a near-optimum solution. However, Eq. (17) can be rewritten as follows:

$$\begin{aligned}
 E^b &= -\frac{1}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^c \sum_{\substack{j=1 \\ j \neq i}}^c \frac{1}{\sum_{l=1}^n V_{lj}} V_{x,i} D_{x,y} V_{y,j} \\
 &= -\frac{1}{2} \left(\sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^c \sum_{j=1}^c \frac{1}{\sum_{l=1}^n V_{lj}} V_{x,i} D_{x,y} V_{y,j} \right. \\
 &\quad \left. - \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^c \frac{1}{\sum_{l=1}^n V_{li}} V_{x,i} D_{x,y} V_{y,i} \right) \\
 &= -\frac{1}{2} \left(\frac{1}{n(n-1)} \sum_{x=1}^n \sum_{y=1}^n D_{x,y} \right. \\
 &\quad \left. - \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^c \frac{1}{\sum_{l=1}^n V_{li}} V_{x,i} D_{x,y} V_{y,i} \right) \\
 &= -\frac{1}{2} \frac{1}{n(n-1)} \sum_{x=1}^n \sum_{y=1}^n D_{x,y} + E^w \tag{A1}
 \end{aligned}$$

which implies that

$$\frac{\partial E^w}{\partial V_{x,i}} = \frac{\partial E^b}{\partial V_{x,i}} \tag{A2}$$

Therefore, the minimization of intra-class Euclidean distance is equivalent to the maximization of the inter-class Euclidean distance.

Due to the symmetric connection matrix, the energy functions Eqs. (4) and (13) are thus a Lyapunov function which guarantees that the network evolution will reach a stable state with the minimum energy as time progresses. As proof of this, taking the derivative of the energy function with respect to time produces

$$\begin{aligned}
 \frac{dE^w}{dt} &= \frac{\partial E^w}{\partial V_{1,1}} \frac{dV_{1,1}}{dt} + \frac{\partial E^w}{\partial V_{1,2}} \frac{dV_{1,2}}{dt} + \dots + \frac{\partial E^w}{\partial V_{1,c}} \frac{dV_{1,c}}{dt} \\
 &\quad + \dots + \frac{\partial E^w}{\partial V_{n,1}} \frac{dV_{n,1}}{dt} + \frac{\partial E^w}{\partial V_{n,2}} \frac{dV_{n,2}}{dt} \\
 &\quad + \dots + \frac{\partial E^w}{\partial V_{n,c}} \frac{dV_{n,c}}{dt} = \sum_x \sum_i \frac{\partial E^w}{\partial V_{x,i}} \frac{dV_{x,i}}{dt} \tag{A3a}
 \end{aligned}$$

and

$$\begin{aligned}
 \frac{dE^b}{dt} &= \frac{\partial E^b}{\partial V_{1,1}} \frac{dV_{1,1}}{dt} + \frac{\partial E^b}{\partial V_{1,2}} \frac{dV_{1,2}}{dt} + \dots + \frac{\partial E^b}{\partial V_{1,c}} \frac{dV_{1,c}}{dt} \\
 &\quad + \dots + \frac{\partial E^b}{\partial V_{n,1}} \frac{dV_{n,1}}{dt} + \frac{\partial E^b}{\partial V_{n,2}} \frac{dV_{n,2}}{dt} \\
 &\quad + \dots + \frac{\partial E^b}{\partial V_{n,c}} \frac{dV_{n,c}}{dt} = \sum_x \sum_i \frac{\partial E^b}{\partial V_{x,i}} \frac{dV_{x,i}}{dt} \tag{A3b}
 \end{aligned}$$

By inserting Eq. (8), or Eq. (15) into Eq. (A3), Eq. (A3) can be updated as follows:

$$\frac{dE^w}{dt} = - \sum_x \sum_i C_{x,i} \frac{dU_{x,i}}{dt} \frac{dV_{x,i}}{dt} \tag{A4a}$$

and

$$\frac{dE^b}{dt} = - \sum_x \sum_i C_{x,i} \frac{dU_{x,i}}{dt} \frac{dV_{x,i}}{dt} \tag{A4b}$$

Table 1

PSNR of the images reconstructed from codebooks of various sizes and different compression ratios designed by the CCHNN and the GLA algorithms

Codebook size		64			128			256		
Compression ratio	Algorithm	Lena	MR head	Boy–girl	Lena	MR head	Boy–girl	Lena	MR head	Boy–girl
0.500 bpp	CCHNN ^a	27.08	28.06	30.12	28.14	29.94	31.21	29.63	31.37	32.86
	GLA	25.73	25.47	26.45	26.03	25.66	26.89	26.35	26.33	27.72
0.125 bpp	CCHNN ^a	24.45	25.83	28.74	26.26	27.14	29.52	25.94	29.18	30.43
	GLA	23.11	23.24	24.59	23.87	23.75	24.61	24.05	24.14	25.17

^a The reconstructed images obtained from CCHNN based on within-class energy are identical to those obtained from CCHNN based on between-class energy

Using the chain rule for the differential equation, Eq. (A4) becomes

$$\begin{aligned} \frac{dE^w}{dt} &= - \sum_x \sum_i C_{x,i} \frac{dU_{x,i}}{dt} \frac{dV_{x,i}}{dU_{x,i}} \frac{dU_{x,i}}{dt} \\ &= - \sum_x \sum_i C_{x,i} \left(\frac{dU_{x,i}}{dt} \right)^2 \frac{dV_{x,i}}{dU_{x,i}} \end{aligned} \quad (\text{A5a})$$

and

$$\begin{aligned} \frac{dE^b}{dt} &= - \sum_x \sum_i C_{x,i} \frac{dU_{x,i}}{dt} \frac{dV_{x,i}}{dU_{x,i}} \frac{dU_{x,i}}{dt} \\ &= - \sum_x \sum_i C_{x,i} \left(\frac{dU_{x,i}}{dt} \right)^2 \frac{dV_{x,i}}{dU_{x,i}} \end{aligned} \quad (\text{A5b})$$

Obviously, $dE^w/dt \leq 0$ and $dE^b/dt \leq 0$, because $C_{x,i} > 0$, $(dU_{x,i}/dt)^2 > 0$ (replace $dU_{x,i}/dt$ with Eq. (12), or Eq. (19)), and $dV_{x,i}/dU_{x,i} > 0$. (Its derivative is always positive due to the monotonically increasing property of the input-output function.) No matter whether it is based on the within-class or the between-class scatter matrix, the energy function of the CCHNN must converge to the minimization as time progresses.

References

- Amatur, S.C., Piriano, D., Takefuji, Y., 1992. Optimization neural networks for the segmentation of magnetic resonance images. *IEEE Trans. Med. Imaging* 11, 215–220.
- Andrew, L.L.H., Palaniswami, M., 1996. A unified approach to selecting optimal step lengths for adaptive vector quantizers. *IEEE Trans. Commun.* 44, 434–439.
- Chang, C.Y., Kwok, R., Curlander, J.C., 1988. Spatial compression of seasat SAR images. *IEEE Trans. on Geosc. Remote Sensing* 26, 673–685.
- Cheng, K.S., Lin, J.S., Mao, C.W., 1996. The application of competitive Hopfield neural network to medical image segmentation. *IEEE Trans. Med. Imaging* 15, 560–567.
- Chung, P.C., Tsai, C.T., Chen, E.L., Sun, Y.N., 1994. Polygonal approximation using a competitive Hopfield neural network. *Pattern Recog.* 26, 1505–1512.
- Fukunaga, K., 1972. *Introduction to Statistical Pattern Recognition*. 2nd ed. Academic, New York.
- Gersho, A., Gray, R.M., 1992. *Vector Quantization and Signal Compression*. Kluwer Academic, Norwell, MA.
- Gray, R.M., 1984. Vector quantization. *IEEE ASSP Mag.* 1, 4–29.
- Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci., USA* 79, 2554–2558.
- Hopfield, J.J., Tank, D.W., 1985. Neural computation of decisions in optimization problems. *Biol. Cybern.* 52, 141–152.
- Kim, Y., Rajala, S.A., Snyder, W.E., 1992. Image segmentation using an annealed Hopfield neural network. *Proceedings of the RNNS/IEEE Symposium on Neural Informatics and Neurocomputers* 1, 311–322.
- Linde, Y., Buzo, A., Gray, R.M., 1980. An algorithm for vector quantizer design. *IEEE Trans. Commun.* COM-28, 85–94.
- Lloyd, S.P., 1982. Least squares quantization in PCM. *IEEE Trans. Inform. Theory* IT-28, 129–137.
- Netravali, A.N., 1988. *Digital Pictures: Representation and Compression*. Plenum Press, New York.
- Riskin, E.A., Lookabaugh, T., Chou, P.A., Gray, R.M., 1990. Variable rate vector quantization for medical image compression. *IEEE Trans. Med. Imaging* 9, 290–298.
- Steck, J.E., Balakrishnan, S.N., 1994. Use of Hopfield neural networks in optimal guidance. *IEEE Trans. Aerospace Electron. Systems* 30, 287–293.
- Tsai, C.T., Sun, Y.N., Chung, P.C., Lee, J.S., 1993. Endocardial boundary detection using a neural network. *Pattern Recog.* 26, 1057–1068.
- Washizawa, T., 1993. Application of Hopfield network to saccades. *IEEE Trans. Neural Networks* 4, 995–997.
- Yair, E., Zeger, K., Gersho, A., 1992. Competitive learning and soft competition for vector quantizer design. *IEEE Trans. Signal Process.* 40, 294–309.
- Yang, H., Dillon, T.S., 1994. Exponential stability and oscillation of Hopfield graded response neural network. *IEEE Trans. Neural Networks* 5, 719–729.
- Zeger, K., Vaisey, J., Gersho, A., 1992. Globally optimal vector quantizer design by stochastic relaxation. *IEEE Trans. Signal Process.* 40, 310–322.

Jzau-Sheng Lin is an Associate Professor in the Department of Electronic Engineering at The National Chin-Yi Institute of Technology, Taichung, Taiwan. He received a B.S. degree in Electronic Engineering from the Taiwan University of Science and Technology in 1980, and received M.S. and Ph.D degrees in Electrical Engineering from the National Cheng Kung University in 1989 and 1996, respectively. His research interests involve neural networks, image compression, pattern recognition, and medical image analysis.

Shao-Han Liu is a Lecturer in the Department of Electronic Engineering at the National Chin-Yi Institute of Technology, Taichung, Taiwan. He received a B.S. degree in Electronic Engineering from the Taiwan University of Science and Technology in 1983, and received an M.S. degree in Management Science from Northrop University in 1989. His research interests involve neural networks and image compression.