1998 International Computer Symposium
Workshop on Image Processing and Character Recognition
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

A NEW COOLING SCHEDULE IN AN ANNEALED HOPFIELD NEURAL NETWORK FOR IMAGE VECTOR QUANTIZATION[1]

Jzau-Sheng Lin

Department of Electronic Engineering,
National Chin-Yi Institute of Technology, Taichung, TAIWAN, R.O.C.
E-MAIL: jslin@chinyi.ncit.edu.tw

ABSTRACT

This paper shows an unsupervised parallel approach called the Annealed Hopfield Neural Network (AHNN) with a new cooling schedule for vector quantization in image compression. The main purpose is to combine the characteristics of neural networks and annealing strategy so that on-line learning and hardware implementation for vector quantization are feasible. The idea is to cast a clustering problem as a minimization problem where the criterion for the optimum vector quantization is chosen as the minimization of the average distortion between training vectors. Although the simulated annealing method can yield the global minimum, it is very time-consuming with asymptotical iterations. In addition, to resolve the optimal problem using Hopfield or simulated annealing neural networks, designer must determine the weighting factors to combine the penalty terms. The quality of final result is very sensitive to these weighting factors, and feasible values for them are difficult to find. Using the AHNN to vector quantization, the need of finding weighting factors in the energy function formulated and based on a basic concept of the "within-class scatter matrix" principle can be eliminated and the rate of convergence is much faster than that of simulated annealing. The experimental results show that good and valid solutions can be obtained using the AHNN in image vector quantization. In addition, the convergent rate with different cooling schedule will be discussed.

Keyword: Simulated Annealing, Mean Field Annealing, Annealed Hopfield Neural Network, Image Compression.

1. INTRODUCTION

In image compression, the process of codebook design from training vectors is a very important step in vector quantization of coding process. A number of vector quantization algorithms in image compression have been demonstrated in other articles [1]-[9]. In general, a vector quantization is an approach for mapping analog signals or discrete vectors into a sequence of digital signal for communication or storage in a channel. The purpose of vector quantization is a creation of a codebook for which the average distortion generated by approximating a training vector and by a codevector in codebook is minimized. The minimization of average distortion measure is widely used by a gradient descent based iterative procedure that is called the generalized Lloyd algorithm (GLA) [1]. According the cluster center in previous iteration and nearest neighbor rule, the GLA performs a positive improvement to update the codebook iteratively.

During the past decade, the Hopfield [10]-[11] neural network has been studied extensively with its features of simple architecture and potential for parallel implementation. The Hopfield neural network is a well-known technique used for solving optimization problems based on the Lyapunov energy function. The application of competitive Hopfield neural network to medical image segmentation was described by Cheng *et al.* [12]. Polygonal approximation using a competitive Hopfield neural network was demonstrated by Chung *et al.* [13]. In [12] and [13], a 2-dimensional discrete Hopfield neural network used the winner-take-all learning to eliminate the need for finding weighting factors in the energy function. Lin *et al.* [14]-[15] proposed the segmentation of single and multispectral medical images using a fuzzy Hopfield neural network (FHNN).

Robust identification for image processing needs data compression that preserves the features in original image. Image compression is coding of transformed image using a code of fixed or variable length. Vector quantization is a significant methodology in image compression, in which blocks of divided pixels are formed as training vectors rather than individual scales. Such a method results in massive reduction of the image information in image transmission. The image is reconstructed by replacing each image block by its nearest codevector. The dimensions, of an image with $N \times N$ pixels, can be divided into n blocks (vectors of pixels) and each block occupies $\lambda \times \lambda$ ($\lambda < N$) pixels. A vector quantization is a technique that maps training vectors $\{\mathbf{X}_x, x = 1,2,...,n\}$ in Euclidean $\lambda \times \lambda$-dimensional space $R^{\lambda \times \lambda}$ into a set $\{\mathbf{Y}_x, x = 1,2,...,n\}$ of

[1] This work was supported by the National Science Council, R.O.C., under the Grant #NSC87-2213-E-167-004.

points in $R^{\lambda \times \lambda}$, called a codebook. The mapping is usually defined to minimize expected distortion measure, $E[d(\mathbf{X}_x, \mathbf{Y}_x)]$, using the mean square error (MSE) given by $d(\mathbf{x}, \mathbf{y}) = (\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})$.

Codebook design can be considered as a clustering process in which the training vectors are classified to the specific classes based on the minimization of average distortion between the training vector and codebook vectors (classes' centroids). Then the clustering algorithms perform a positive improvement to update the codebook iteratively. In addition to the neural network-based technique, the annealing technique with a new cooling schedule has also been demonstrated to address codebook design problems in this article. In AHNN, the problem of the vector quantization is regarded as a process of the minimization of a cost function. This cost function is defined as the average distortion between the training vectors in a divided image to the cluster centers represented by the codevectors in the codebook. The structure of this network is constructed as a two-dimensional fully interconnected array with the columns representing the number of codevectors (classes) and the rows representing the training vectors in the divided image. However, a training vector does not necessarily belong to one class. Instead, a certain probability belonging to proper class is associated with every vector sample. In AHNN, an original Hopfield network is modified and the annealing strategy with a new cooling schedule is added. The structure of AHNN is identical the one of mean field annealing (MFA) network. Consequently, the energy function may be quickly converged into a near global minimum in order to produce a satisfactory codebook. Compared with conventional techniques, the major strength of the presented AHNN is that it is computationally more efficient due to the inherent parallel structures. In a simulated study, the AHNN is demonstrated to have the capability for vector quantization in image compression and shown the promising results.

2. ANNEALING TECHNIQUES

Simulated annealing is a stochastic relaxation algorithm which has been used successfully to resolve the optimization problems including computer network topology problems [16], traveling salesman problems [17], circuit routing problems [18], image processing problems [19]-[20], and clustering problems [21]. Instead of the other optimization methods such as steepest descent approach used in the Hopfield neural network, the simulated annealing technique, which allows the search to move away from a local minimum, seeks the global or near global minimum of an energy function without getting trapped in local minimum. The simulated annealing technique had non-zero probability to go from one state to another, moves temporarily toward a worse state so as to escape from local traps. The probability function depends on the temperature and the energy difference between the two states. With the probabilistic hill-climbing search approach, the simulated annealing technique has a better probability to go to a higher energy state at a higher temperature.

Simulated annealing strategy was first proposed by Metropolis *et al* [22] to simulate molecular processes in 1953. Kirkpatrick *et al* [18] used the ideal as a method to resolve minimizing functions of many variables, such as NP-hard problems. Simulated annealing derives its name from an analogy between its behavior and that of a physical process of thermodynamics and metallurgy. In which, a metal is first melted at a very high temperature and then slowly cooled until it solidifies in a structure of minimum energy. At the beginning, the temperature T, used to control the probability of accepting a worsening perturbation over time, is set to a very high value; later it is multiplied by a factor T_{rate} ($0 < T_{rate} < 1$), called the annealing factor or cooling rate, after every iteration.

Although the simulated annealing method can yield the global minimum, it is very time-consuming with asymptotical iterations. The AHNN, presented to vector quantization in image compression and discussed in the following sections, which incorporate the characteristics of the annealing strategy with a new cooling schedule and the Hopfield neural network based on a modified objective function, can converge much faster than the simulated annealing. In addition, The AHNNmay also achieve the level of optimization that is comparable to that achieved by simulated annealing network.

3. VECTOR QUANTIZATION BY THE AHNN

The AHNN, combined the characteristics of annealing algorithm and the rapid convergence of Hopfield neural network, is a well-known technique used for solving optimization problems based on the Lyapunov energy function. A two-dimensional image is divided into n blocks (a block represents a training vector that captures $\lambda \times \lambda$ pixels) and mapped to a two-dimensional Hopfield neural network. Therefore, an image can consist of n training vectors and c interesting classes (The class centroid is represented a proper codevectors in codebook). If the number of codevectors c in codebook were defined in advance, then the network array in this paper would consist of n by c neurons. The AHNN can be conceived as a two-dimensional neuron array. Each vector was trained with the nearest neighbor condition and iteratively updating the neurons' weights. In this section, we will show that the vector quantization in image compression problem can be mapped onto the AHNN so that the cost function serves as the energy function of the network. The idea is to form the energy function of the network in terms of the intra-class energy function. In the pattern recognition application, the intraset (within-class) distance should be small. The proposed technique first assigns training vectors to their associated classes in such a manner that the average distortion between arbitrary vectors to their class center or codevector, referred to as the within-class assignment, is minimized in accordance with the nearest neighbor condition. In linear discriminate analysis [26], the concept of within-class scatter matrix is widely used for class separability. The iteratively updated synaptic weight between the neuronal interconnections will gradually force the network to converge into a stable state where its energy function is minimized.

The divided image can be represented by n training vectors. Using the within-class scatter matrix criteria, the optimization problem can be mapped into a two-dimensional fully interconnected Hopfield neural network with the annealing strategy for vector qauntization in image compression. Instead of using the competitive learning strategy, the AHNN uses the Boltzmann probability distribution to eliminate the need for finding weighting factors in the energy function. Henceforward, each neuron would be identified with a double index, x,i (where the index i=1, 2, ... c relates to the group, whereas the index x= 1, 2, ..., n refers to the neurons in each group), its state with $V_{x,i}$, the weight vector for neurons x,i and y,i with $\mathbf{W}_{x,i;y,i}$, and the external bias vector for neuron x,i with $\mathbf{I}_{x,i}$. According to this convention, the cost function can be modified as

$$E=\frac{1}{2}\sum_{x=1}^{n}\sum_{i=1}^{c}V_{x,i}\left|\mathbf{z}_x-\sum_{y=1}^{n}\mathbf{W}_{x,i;y,i}V_{y,i}\right|^2-\sum_{x=1}^{n}\sum_{i=1}^{c}\mathbf{I}_{x,i}V_{x,i} \quad (1)$$

Each column of this modified Hopfield network represents a codevector (class) and each row represents a training vector in a proper class. The network reaches a stable state when the modified Lyapunov energy function is minimized.

In order to generate an adequate codebook with the constraints, we define the objective function as follows:

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

where E is the total intra-class scatter energy that is accounted for the scattered energies distributed by all training vectors in the same class, and both $\mathbf{z}_x$ and $\mathbf{z}_y$ are the trained vectors at rows x and y, respectively, and the V's are continuous variables in the region [0,1] such that $V_{x,i}\approx 1$ indicates that training vector $\mathbf{z}_x$ belongs to codevector i (otherwise, $V_{x,i}\approx 0$). The first term in Eq. (2) is the within-class scatter energy that is the average distortion between training vectors to the cluster center over c clusters (codevectors in codebook). The second term attempts to insure that any training vector $\mathbf{z}_x$ does not show up on the final solution in two classes i and j. While the third term guarantees those n vectors in $\mathbf{Z}$ can only be distributed among these c classes. More specifically, the last two term, which is the penalty term, imposes constraints on the objective function and the first term minimizes the within-class Euclidean distance from a training vector to the cluster center in any given cluster. These terms are combined into a weight sum using three coefficients determined by the designer. As mentioned in references [13] and [23], the quality of classification result is very sensitive to the weighting factors, and good values for them are difficult to find when even a moderate number of training samples are considered. Searching for optimal values for these weighting factors is expected to be time-consuming and laborious. In [23]-[25], Van Den Bout indicated good values for penalty terms can easily determine using a trial and zero approach or analytical techniques in a TSP problem on the order of 10 cities. Unfortunately, these terms do not scale even as the problem grows modestly to 30 cities. Therefore, the problem of finding a feasible codevectors from n training vectors has been replaced with the problem of finding the best value of A, B, and C. In this paper, a new objective function is developed which does not require any weighting factor. Each state $V_{x,i}$ is looked upon as the probability of finding training vector $\mathbf{z}_x$ currently residing to class i undergo random thermal perturbations. The probability of the training vector $\mathbf{z}_x$ occupied by class i at a given temperature T conforms to a Boltzmann distribution

$$V_{x,i}\propto e^{-\Delta E_{x,i}/T} \quad (3)$$

As each training vector can only be occupied by one class, that is, every raw can have at most 1. In other words, the summation of states in the same row equals 1. It also ensures that only n vectors will be classified into these c codevectors. That is the network must match the following constraints

$$\sum_{i=1}^{c}V_{x,i}=1$$

and

$$\sum_{x=1}^{n}\sum_{i=1}^{c}V_{x,i}=n.$$

Therefore, the objective function of the AHNN can be further simplified as

$$E=\frac{1}{2}\sum_{x=1}^{n}\sum_{i=1}^{c}V_{x,i}\left|\mathbf{z}_x-\sum_{y=1}^{n}\frac{1}{\sum_{h=1}^{n}V_{h,i}}\mathbf{z}_yV_{y,i}\right|^2. \quad (4)$$

Then the mean field $E_{x,i}$ can be calculated from Eq. (4) to be

$$E_{x,i}=\left|\mathbf{z}_x-\sum_{y=1}^{n}\frac{1}{\sum_{h=1}^{n}V_{h,i}}\mathbf{z}_yV_{y,i}\right|^2. \quad (5)$$

The probability of training vector $\mathbf{z}_x$ occupied by class i can then be normalized as follows:

$$V_{x,i} = \frac{e^{-E_{x,i}/T}}{\sum_{j=1}^{c} e^{-E_{x,j}/T}} \qquad (6)$$

The normalization operation in (6) guarantees that each training vector will be absorbed on several classes with certain probability degrees so there will be n vectors assigned among c codevectors. By using Eq. (4), which is modified from Eq. (2), the minimization of energy E is greatly simplified since it contains only one term and hence the requirement of having to determine the weighting factors A, B and C, vanishes. Comparing Eq. (4) with the modified cost function Eq. (1), the synaptic interconnection weights and the bias input can be obtained as

$$\mathbf{W}_{x,i;y,i} = \frac{1}{\sum_{h=1}^{n} V_{h,i}} \mathbf{z}_y, \qquad (7)$$

and

$$\mathbf{I}_{x,i} = 0. \qquad (8)$$

As the temperature is reduced, the training vectors will begin to approach in feasible class that will minimize the total cost. In summary, the AHNN algorithm for image vector qunatization consists of the following steps:

1. Input a set of training vectors $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\}$, the number of classes c (number of codevectors), and randomly initialize the probabilities of neuron states for all neurons.
2. Start with an initial temperature T_0.
3. Select a training vector $\mathbf{z}_x$ randomly and calculate the mean field for training vector $\mathbf{z}_x$ at each class i using Eq. (5).
4. Calculate the normalized probability that training vector $\mathbf{z}_x$ assigned class i using Eq.(6) for each possible class.
5. Repeat the iteration with another randomly selected neuron until all neurons are trained.
6. Decrease T with the annealing factor T_k shown in Eq. (12) iteratively.
7. Repeat steps 3, 4, 5, and 6 until cost function is convergent.

4. COOLING SCHEDULE

In order to converge to a near global minimum in annealing process, a feasible cooling schedule is required. The reaching thermal equilibrium at low temperature might take a very long time. The search for adequate cooling schedules has been the subject of an active research field for several years [27]. Geman *et al.* [28] demonstrated that if the temperature is lowered at the rate:

$$T_{rate} = \frac{T_0}{\log(k+1)} \qquad (9)$$

where T_0 is a constant and k is the number of iterations, the algorithm will converge to the set of states of least energy. Jalali *et al.* [20] presented that the value of the constant T_0 for which Geman *et al.* were able to guarantee convergence is in general very high, so that the convergence time becomes impractically slow. Jalali *et al.* used a schedule very similar to that of Geman *et al.*, given in Eq. (9), but with a steeper descent at higher iterations as follows:

$$T_{rate} = \frac{T_0}{\log(k+1)^3}. \qquad (10)$$

Jalali *et al.* showed that the value of T_0 in Eq. (10) has to be kept as small as possible, so that the number of iterations can be held within a reasonable limit. Unfortunately, the cooling schedules specified by Eq. (9) and Eq. (10) with high value of T_0 are too slow to be of practical use [29]. Kirkpatrick *et. al.* [18] proposed a cooling schedule specified a finite sequence of values of the temperature and a finite number of transitions attempted at each value of the temperature. The decrement function of cooling schedule is defined by

$$T_{rate} = (\alpha)^k T_0, \quad k = 1, 2, \ldots \qquad (11)$$

where α ($0.8 \le \alpha \le 0.99$) is a constant smaller but close to unit.

In this paper, a new decrement function of cooling schedule is proposed as follows:

$$T_k = \frac{1}{\beta+1}[\beta + \tanh(\alpha)^k] T_{k-1}, \quad k = 1, 2, \ldots \qquad (12)$$

where α is a constant same as the one in Eq. (11). And β is another constant. β=4 been selected in this paper, Eq. (12) can result in a faster decrement speed than those resulted from Eq. (11). Figure 1 shows the reduction process using different decrement functions described from Eq. (9) to Eq. (12) with α=0.98, initial temperature T_0=4000, and 100 iterations. From Figure 1, we can find Eq. (12) results in the fastest decrement speed.

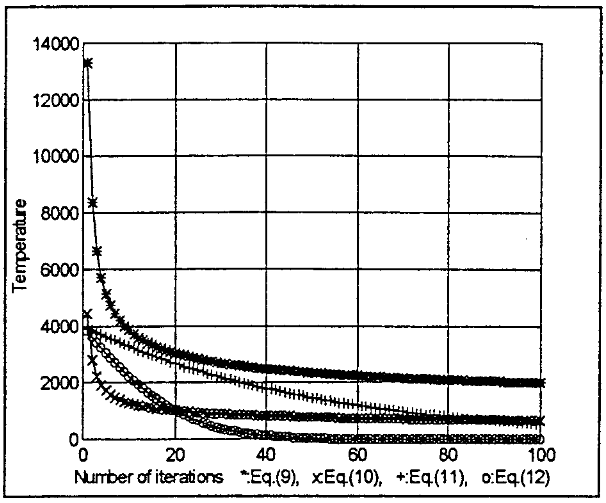


Figure 1. The reduction process using different decrement functions described from Eq. (9) to Eq. (12) with T_0=4000 and 100 iterations.

5. EXPERIMENTAL RESULTS

In this paper, the performance of images reconstructed from the designed codebooks was discussed using the AHNN with different cooling schedules. 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-dimmensional vectors respectively. Three codebooks of size 64, 128, and 256 were built using this training data. Column 1 in Figure 2 shows the training images. These images are 256×256 pixels with 8-bit gray levels. In this experiment the compression rates were 8/16 = 0.5 and 8/64=0.125 bits per pixel respectively. The peak signal to noise ratio (PSNR), that is defined for $N \times N$ image as follows, was evaluated in the reconstructed images.

$$PSNR = 10\log_{10}\frac{255\times 255}{\frac{1}{N^2}\sum_{x=1}^{N}\sum_{y=1}^{N}\left(f_{xy}-\hat{f}_{xy}\right)^2} \qquad \textbf{(13)}$$

where f_{xy} and $\hat{f}_{xy}$ are the pixel gray levels from the original and reconstructed images, 255 is the peak gray level, respectively. Columns 2 and 3 in Figure 2 show the images reconstructed from the codebooks of size $c = 64$ and 256 design by the proposed AHNN based on the decrement function demonstrated in Eq. (12). In the boy-girl image, the PSNR completed by the AHNN algorithm with 0.500 bpp and codebook of size $c = 256$ was 33.03 dB. Table 1 shows the PSNR of the images reconstructed from the various codebooks design using the AHNN with cooling schedule shown in Eq. (12). From the experimental results, the proposed cooling schedule can results in a fastest convergence and get a feasible PSNR in the test images with different codebook size after several iterations. Figure 3 shows the PSNRs of reconstructing from Lenna image with codebook size $c = 256$ and boy-girl image with codebook size $c = 128$ by 4×4 blocks using various cooling schedule described from Eq. (9) to Eq. (12) during 60 iterations. In summary, from the experiment results, the proposed algorithm and cooling schedule could satisfactorily produce the codebook design while the network convergence with finite number of iterations is guaranteed.

Chung [13] indicated that the quality of the final solution is very sensitive to the values of weighting factors A, B and C, and searching for the optimal values would be time-consuming and laborious. The problem of determining the optimal values of the weighting factors is avoided in the AHNN. It is implied that this approach is more efficient and versatile than the simulated annealing neural network for vector quantization. It is also noted that the resulting images are processed without human intervention. Thus, the experimental results can be regarded as near optimal. Generally, the AHNN approach for vector quantization needs much more computation time than conventional methods. However, due to the AHNN's highly interconnected and parallel abilities, computation time can be largely reduced by way of parallel processing with hardware implementation.

6. DISCUSSION & CONCLUSIONS

A two-dimensional AHNN neural network based on the within-class scatter matrix with a new cooling schedule for vector quantization has been presented in this paper. The energy function used for the AHNN is called the scatter energy function that is formulated and based on a widely used concept in pattern classification. The AHNN method greatly simplifies the scatter energy function so that there is no need to search for the weighting factors imposed on the original energy function. In addition, the proposed cooling schedule could satisfactorily produce the feasible codebook design while the network convergence with finite number of iterations is guaranteed. As a result, the proposed algorithm appears to converge rapidly to the near optimal solution using the proposed annealing schedule. Moreover, the designed AHNN neural-network-based approach is a self-organized structure that is highly interconnected and can be implemented in a parallel manner. It can also easily be designed for hardware devices to achieve very high-speed implementation.

REFERENCES

[1] Y. Linde, A. Buzo, and R. M. Gray et al., "An algorithm for vector quantizer design," IEEE Trans. Commun., Vol. COM-28. pp. 85-94, 1980.

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

[3] C. Y. Chang, R. Kwok, and J. C. Curlander et al., "Spatial compression of seasat SAR images," IEEE Trans. on Geoscience and Remote Sensing, Vol. 26, pp. 673-685, 1988.

[4] A. N. Netravali, "Digital Pictures: Representation and Compression," Plenum Press, New York, 1988.

[5] A. Gersho and R. M. Gray et al., "Vector Quantization and Signal Compression," Kluwer Academic Publishers, Norwell, Ma, 1992.

[6] E. A. Riskin, T. Lookabaugh, P. A. Chou, and R. M. Gray et al., "Variable rate vector quantization for medical image compression," IEEE Trans. Med. Imaging, Vol. 9. pp. 290-298, 1990.

[7] E. Yair, K. Zeger, and A. Gersho et al., "Competitive learning and soft competition for vector quantizer design," IEEE Trans. Signal Processing, Vol. 40. pp. 294-309, 1992.

[8] K. Zeger, J. Vaisey, and A. Gersho et al., "Globally optimal vector quantizer design by stochastic relaxation," IEEE Trans. Signal Processing, Vol. 40. pp. 310-322, 1992.

[9] L. L. H.Andrew and M. Palaniswami et al., "A unified approach to selecting optimal step lengths for adaptive vector quantizers," IEEE Trans. Commun., Vol. 44. pp. 434-439, 1996.

[10] J. J. Hopfield and D. W. Tank et al., "Neural computation of decisions in optimization problems,"

Biol. Cybern., Vol. 52. pp. 141-152, 1985.

[11] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," Proc. Nat. Acad. Sci., USA, Vol. 79. pp. 2554-2558, 1982.

[12] K. S. Cheng, J. S. Lin, and C. W. Mao et al., "The application of competitive Hopfield neural network to medical image segmentation," IEEE Trans. med. imaging, Vol. 15. 560-567, 1996.

[13] P. C. Chung, C. T. Tsai, E. L. Chen, and Y. N. Sun et al., "Polygonal approximation using a competitive Hopfield neural network," Pattern Recognition, Vol. 27. pp. 1505-1512, 1994.

[14] J. S. Lin, K. S. Cheng, and C. W. Mao et al., "A fuzzy Hopfield neural network for medical image segmentation," IEEE Trans. Nucl. Sci., Vol. 43. pp. 2389-2398, 1996.

[15] J. S. Lin, K. S. Cheng, and C. W. Mao et al., "Multispectral magnetic resonance images segmentation using fuzzy Hopfield neural network," *J. Biomed. Comput.*, Vol. 42. pp. 205-214, 1996.

[16] S. Pierre, M. A. Hyppolite, J. M. Bourjolly, and O. Dioume et al., "Topological design of computer communication networks using simulated annealing,", Engng Applic. Artif. Intell., Vol. 8. pp. 61-69, 1995.

[17] P. J. M. Van Larhoven, and E. H. L. Arts et al., "Simulated annealing : theory and application," D. Reidel, Hingham, MA, 1987.

[18] S. Kirkpatrick, C. D. Gelatt JR., and M. P. Vecchi et al., "Optimization by simuted annealing," Science, N.Y. Vol. 220. pp. 671-680, 1983.

[19] H. L. Tan., S. B. Gelfand, and E. J. Delp et al., "A cost minimization approach to edge detection using simulated annealing," IEEE Trans. on PAMI, Vol. 14. pp. 3-18, 1991.

[20] S. Jalali, and J. F. Boyce et al., "Determination of optimal general edge detectors by global minimization of a cost function," Image and Vision Computing, Vol. 13. pp. 683 –693, 1995.

[21] R. W. Klein, and R. C. Dubes et al., "Experiments in projection and clustering by simulated annealing," Pattern Recognition, Vol. 22. pp. 213-220, 1989.

[22] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller et al., "Equations of state calculations by fast computing machines," J. chem. phys., Vol. 21. pp. 1087-1092, 1953.

[23] D. E. Van Den Bout, and T. K. Miller III et al., "A traveling salesman objective function that works," International Conference on Neural Network, Vol. II. pp. 299-303, 1988.

[24] G. Bibro, R. Mann, T. K. Miller III, W. Snyder, D. E. Van Den Bout, and M. White et. al., "Mean field annealing and neural networks," in Advances in Neural Information Processing Systems, Morgan-Kaufmann, pp. 91-98, 1989.

[25] D. E. Van Den Bout and T. K. Miller III et al., "Graph partitioning using annealed neural networks," IEEE Trans. Neural Networks, Vol. 1. pp. 192-203, 1990.

[26] K. Fukunaga, "Introduction to statistical patter recognition," New York: Academic, 1972.

[27] Van Laarhoven, and E. H. L. Aarts et al., "Simulated Annealing: Theory and Applications," Boston, MA: Kluwer Academic Publishers, 1988.

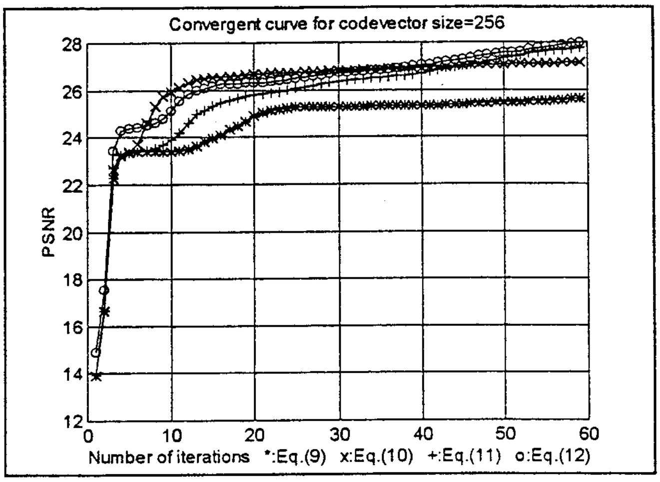
[28] S. Geman, and D. Geman et al., "Stochastic relaxation, Gibbs distributions and the Baysian restoration of images," IEEE Trans. Pattern Analy. and Mach. Intell., Vol. 6. pp. 721-741, 1984.

[29] S. Haykin, "Neural Network: A Comprehensive Foundation," Macmillan College Publlishing Company, 1994.

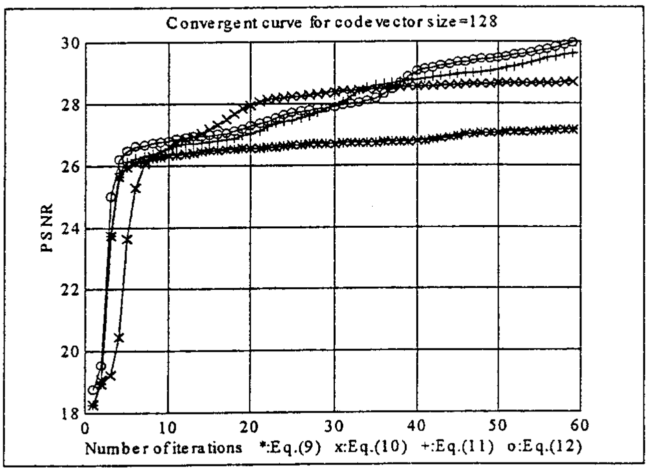
Figure 2. Test images and reconstructed images with 4x4 blocks using AHNN: (column 1) Orignal images; (column 2) reconstructed images with codevectors c=64; and (column 3) reconstructed images with codevectors c=256.

Table I. PSNR of the images reconstructed from codebooks of various sizes and different compression ratios designed by the AHNN.

Codebook Size	64				128				256			
Image	Lenna	F-16	Girl	Boy-girl	Lenna	F-16	Girl	Boy-girl	Lenna	F-16	Girl	Boy-girl
0.500bpp	27.56	25.73	29.53	30.23	28.14	26.76	30.45	31.21	29.56	28.32	31.55	33.03
0.125bpp	24.87	23.23	26.89	28.74	26.32	24.42	27.69	29.52	27.86	26.65	28.87	30.66



(a)



(b)

Figure 3. Convergent curve of the test images in 60 iterations with 4×4 blocks using different cooling schedule. (a) lenna, and (b) boy-girl images.