shows the switching configuration corresponding to Fig. 3(b). Tables I and II summarize the simulation results, where the average number of iteration steps required for the convergence and the convergence frequency, and the average number of demands in the local minimum solutions are compared in seven models. Note that for each model, 100 simulation runs were performed from different initial values of U_{ij}.

We conclude the simulation results as follows:

1) The comparisons of cases #1–#4 show that the decay term disturbs the convergence of the neural network to solutions. Although cases #2 and #3 are superior in the average number of iteration steps for the convergence to case #4, they are inferior in the frequency of the local minimum convergence and the solution quality to case #4. The decay term seems to make the local minimum deeper, so some initial states of U_{ij} can be quickly converged to the global minima.

2) The comparisons of cases #4 and #5 show that the McCulloch–Pitts neuron model and the sigmoid neuron model have similar performance in terms of the average number of iteration steps for the convergence and the convergence frequency. However, because of the exponential calculation in the sigmoid neuron model, it requires much longer computation time than the McCulloch–Pitts neuron model on a digital computer. The simple McCulloch–Pitts neuron model is superior to the sigmoid neuron model for practical uses.

3) The comparisons of cases #4–6 show that the hysteresis McCulloch–Pitts neuron model is superior to the McCulloch–Pitts neuron model and the sigmoid neuron model in terms of the frequency of the global minimum convergence.

4) The comparisons of cases #6–7 show that the two heuristics increase the frequency of the global minimum convergence, reduce the number of iteration steps for the convergence, and improve the solution quality. The hysteresis McCulloch–Pitts neuron model without the decay term and with the two heuristics provides the best performance among the seven models. We have observed similar behavior in other instances.

V. Conclusion

This paper presents performance comparisons of seven neural network models on traffic control problems in multistage interconnection networks. The simulation results show that 1) the decay term in the motion equation disturbs the convergence, 2) with less computation time on a digital computer, the McCulloch–Pitts neuron model achieves the same performance as the sigmoid neuron model, and 3) the hysteresis McCulloch–Pitts neuron model and the two heuristics greatly improve the performance of the neural network computation.

References

[1] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.

[2] A. Marrakchi and T. Troudet, "A neural net arbitrator for large crossbar packet-switches," *IEEE Trans. Circuits Syst.*, vol. 36, no. 7, pp. 1039–1041, July 1989.

[3] T. X. Brown and K. H. Liu, "Neural network design of a Banyan network controller," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 1428–1438, Oct. 1990.

[4] Y. Takefuji and K. C. Lee, "Artificial neural networks for four-coloring map problems and K-colorability problems," *IEEE Trans. Circuits Syst.*, vol. 38, no. 3, pp. 326–333, Mar. 1991.

[5] ——, "An artificial hysteresis binary neuron: A model suppressing the oscillatory behaviors of neural dynamics," *Biol. Cybern.*, vol. 64, pp. 353–356, 1991.

[6] ——, "A parallel algorithm for tiling problems," *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp. 143–145, Mar. 1990.

[7] N. Funabiki and Y. Takefuji, "A parallel algorithm for spare allocation problems," *IEEE Trans. Reliability*, vol. 40, no. 3, pp. 338–346, Aug. 1991.

[8] ——, "A parallel algorithm for solving 'Hip' games," *Neurocomputing*, vol. 3, pp. 97–106, 1991.

[9] K. C. Lee, N. Funabiki, and Y. Takefuji, "A parallel improvement algorithm for the bipartite subgraph problem," *IEEE Trans. Neural Networks*, vol. 3, no. 1, pp. 139–145, Jan. 1992.

[10] W. S. McCulloch and W. H. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, p. 115–133, 1943.

[11] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proc. 1st. Annu. Int. Symp. Comput. Architecture*, Dec. 1973, pp. 21–28.

[12] T. Y. Feng, "Data manipulating functions in parallel processors and their implementations," *IEEE Trans. Comput.*, vol. C-23, pp. 309–318, Mar. 1974.

[13] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. Comput.*, vol. C-24, no. 12, pp. 1145–1155, Dec. 1975.

[14] K. E. Batcher, "The flip network in STARAN," in *Proc. Int. Conf. Parallel Processing*, 1976, pp. 65–71.

[15] M. C. Pease, "The indirect binary n-cube microprocessor array," *IEEE Trans. Comput.*, vol. C-26, no. 5, pp. 458–473, May 1977.

[16] H. J. Siegel and S. D. Smith, "Study of multistage SIMD interconnection networks," in *Proc. 5th Annu. Int. Symp. Comput. Architecture*, Apr. 1978, pp. 223–229.

[17] C. L. Wu and T. Y. Feng, "On a class of multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-29, no. 8, pp. 694–702, Aug. 1980.

[18] D. M. Dias and M. Kumar, "Packet switching in N log N multistage networks," in *Proc. IEEE GLOBECOM '84*, Nov. 1984, pp. 114–120.

A Neural Network Implementation of the Moment-Preserving Technique and Its Application to Thresholding

Shyi-Chyi Cheng and Wen-Hsiang Tsai

***Abstract*—A neural-network implementation of the moment-preserving technique which is widely used for image processing is proposed. The moment-preserving technique can be thought of as an information transformation method which groups the pixels of an image into classes. The variables in the so-called moment-preserving equations are determined iteratively by a recurrent neural network and a connectionist neural network which work cooperatively. Both of the networks are designed in such a way that the sum of square errors between the moments of the input image and those of the output version is minimized. The proposed neural network system is applied to automatic threshold selection. The experimental results show that the system can threshold images successfully. The performance of the proposed method is also compared with those of four other histogram-based multilevel threshold selection methods. The simulation results show that the proposed technique is at least as good as the other methods.**

***Index Terms*—Connectionist neural networks, gradient descent, image thresholding, moment-preserving principle, recurrent neural networks.**

I. Introduction

Recently, a new image processing technique called *moment preservation* has been successfully applied to many image processing

Manuscript received January 22, 1991; revised August 26, 1991 and April 22, 1992. This work was supported in part by the National Science Council, Republic of China under Contract NSC80–0404–E009–13.

S.-C. Cheng is with the Telecommunications Laboratories, Ministry of Transportation and Communications, Hsinchu, Taiwan, R.O.C.

W.-H. Tsai is with the Department of Computer and Information Science, National Chiao Tung University, Taiwan, R.O.C.

IEEE Log Number 9204393.

tasks, such as thresholding, image compression, feature detection, image sharpening [1]–[9]. The moments of an image inherit the important characteristic information of the image. Thus, the difference between the moments of an image and those of a processed version of the image can be used as a *goodness* criterion for an image operation. Furthermore, certain important parameters of an image can be extracted from the image by preserving the moments of the input image in the output version. Unfortunately, the moment-preserving technique involves solving a set of nonlinear equations which is especially computationally intensive for the higher-order moment-preserving case. The moment-preserving equations can be solved by a sequence of deterministic computing steps [1]. It is possible to derive analytic solutions for the cases of preserving lower-order moments; however, some iterative numerical methods must be used to compute the solutions for the cases of preserving higher-order moments. The high computational complexity of the moment-preserving technique limits its applicability to many image processing tasks.

Most of the previous use of the moment-preserving techniques in image processing are based on the lower-order moment-preserving principle [3]–[7]. This does not imply the higher-order moment-preserving principle is not needed. In fact, the higher-order moment-preserving techniques are useful in several important applications. For example, in [15] Delp and Mitchell develops a moment-preserving quantization method based on the higher-order moment-preserving principle. The higher-order moment-preserving principle is also powerful in multilevel threshold selection.

In this correspondence, a neural-network system consisting of a recurrent network and a connectionist network is proposed to implement the moment-preserving technique. Its application to image thresholding is also demonstrated. The thresholding results can be used to check the correctness of the proposed neural network system. The system accomplishes the estimation of the threshold values by minimizing an energy function defined as the square errors between the moments of an input image and those of its thresholded version. An iterative algorithm is developed for the system to minimize its energy function. The proposed neural network approach can be thought as a generalization of Tsai's method [1].

The remainder of this correspondence is organized as follows. The moment-preserving principle is reviewed first in Section II. The flowchart of the proposed neural network system to solve the moment-preserving equations is described in Section III. Section IV includes detailed descriptions of the neural network system and its theoretical foundation. Some experimental results are presented in Section V to support the validity of the proposed approach for multilevel thresholding. Finally, in Section VI some conclusions are drawn.

II. Review of Moment-Preserving Principle

Given an image f with n pixels whose gray value at pixel (x, y) is denoted by $f(x, y)$, the ith moment m_i of f is defined as

$$m_i = \frac{1}{n}\sum_x\sum_y f(x,y)^i, \qquad i = 0, 1, 2, 3, \cdots. \tag{1}$$

Moments can also be computed from the histogram of f in the following way:

$$m_i = \frac{1}{n}\sum_j n_j(Z_j)^i = \sum_j P_j(Z_j)^i \tag{2}$$

where n_j is the total number of the pixels in f with gray values Z_j and $P_j = n_j/n$. It follows from (1) that $m_0 = 1$. The moment-preserving principle for image processing is to transform an image into another form by preserving the moments of the original image. Without loss of generality, it can be said that the moment preserving transformation aims to group the gray values of the pixels in an image into a number of classes and represent all the gray values of the pixels in each class with a single gray value.

Let g be the result of applying the N-class moment-preserving transformation to an image f. Assume that Z_i denotes the representative gray value of the ith pixel class and that P_i denotes the fraction of the pixels in the ith class. By preserving the first $2N-1$ moments of f and using the fact that the sum of all the values of P_i is equal to 1, the following set of $2N$ equations can be obtained [1]:

$$\begin{aligned} P_1Z_1^0 + P_2Z_2^0 + P_3Z_3^0 + \cdots + P_NZ_N^0 &= m_0,\\ P_1Z_1^1 + P_2Z_2^1 + P_3Z_3^1 + \cdots + P_NZ_N^1 &= m_1,\\ &\vdots\\ P_1Z_1^{2N-1} + P_2Z_2^{2N-1} + P_3Z_3^{2N-1} + \cdots + P_NZ_N^{2N-1} &= m_{2N-1}; \end{aligned} \tag{3}$$

which can then be solved to get all P_i and $Z_i, i = 1, 2, 3, \cdots, N$. The neural network system proposed to solve (3) above will be described in the next two sections. Once all P_i and $Z^i, i = 1, 2, 3, \cdots, N$, are determined, the desired analysis for various applications can be performed. For convenience, the equations described by (3) will be called the moment-preserving equations. For $N \leq 4$, analytic solutions to (3) can be derived [1]. Unfortunately, for $N > 4$, no closed-form solution to (3) exists. As one applies the moment-preserving principle to the segmentation of a complicated image, N less than 5 may not be enough. For N no less than 5, an iterative numerical method may be used to solve the moment-preserving equations, but the computation time would be long. The neural network approach described in Section III can be used to solve the moment-preserving equations for abitrary values of N with no additional effort. Thus, the idea behind the proposed system can be regarded as a generalization of Tsai's method. The massive-parallelism characteristic of the proposed neural network system is suitable for solving the equations quickly. So, the long computation time problem may also be avoided.

III. Proposed Moment-Preserving Neural Network System

The neural network system proposed to implement the moment-preserving technique is shown in Fig. 1(a). The goal of the system is to solve the moment-preserving equations. An input image is first preprocessed to compute its moments. Next, recall that the effect of performing the moment-preserving operation is just to group the gray values of the pixels of the image into several classes and replace all the gray values in each class with an identical gray value. This problem can be thought as a clustering problem when a suitable distance function or similarity measure is defined. The concept of k-means clustering [11], well known in the area of pattern recognition, can be used here. Fig. 2 illustrates this concept more concisely, where Z_i is the representative gray value of the ith pixel class and the fraction value P_i defines a border of that class. Accordingly, given a new representative value Z_i for the ith class, the moment-preserving equations can be used to define a criterion function to compute the new border P_i of that class. Furthermore, a new set of values of P_i can be used to refine the class representative gray values Z_i, and then to redistribute the pixels of the image into new pixel classes. In this successive steps of estimating the values of P_i and Z_i, it is proposed to minimize the value of the following criterion function:

$$E = \sum_{k=0}^{2N-1}(m_k - \tilde{m}_k)^2 \tag{4}$$

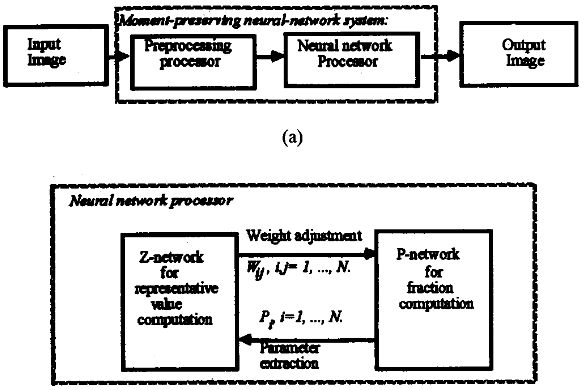


Fig. 1. Proposed moment-preserving neural network system. (a) Block diagram of the system; (b) neural network processor for solving moment-preserving equations.

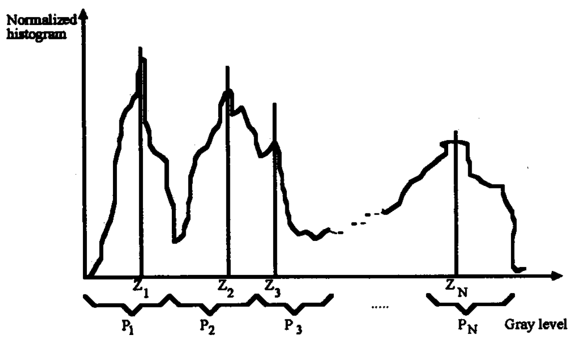


Fig. 2. Illustration of the concept to cluster the pixels of an image into N classes from the histogram of the image based on the moment-preserving principle.

where $m_k = \sum_{i=1}^{N} P_i Z_i^k$ denotes the estimated moments and $\tilde{m}_k = \sum_j \tilde{P}_j \tilde{Z}_j^k$ represents the moments of the original input image, respectively. The value $\tilde{P}_i$ specifies the fraction of the pixels in the original image with gray values $\tilde{Z}_i$. Note that $E = 0$ if and only if $m_k = \tilde{m}_k$ for all k. Thus, the goal to solve the moment-preserving equations can be achieved if the global minimum of E is obtained. Unfortunately, the above process is hard to implement by a single neural network because two types of variables, P_i and Z_i, exist in the moment-preserving equations and should be solved simultaneously. To overcome this difficulty, the proposed system is designed to solve P_i and Z_i successively using a neural network processor shown in Fig. 1(b), which includes two subnetworks, one, called Z-network, performing the refinement of the representative gray values Z_i for each pixel class, and the other, called P-network, performing the moment-preserving operation to redistribute the pixels of the image into pixel classes with the new values of Z_i. A computation algorithm for the system is as follows.

Algorithm: moment-preserving neural network system flow.

Input: histogram of an input image.

Output: K pixel classes.

Method:

1. Choose K initial representative gray values $Z_1, Z_2, \cdots, Z_K$ and set up the configurations of both the P-network and the Z-network. The initial values for Z_i can be chosen arbitrarily, but are suggested to be uniformly distributed in the gray level range.
2. Activate the P-network to estimate the values of P_i according to the gradient of the defined criterion function E.
3. Activate the Z-network to refine the representative gray values Z_i for all pixel classes. The resulting values of Z_i not only depend on the values of P_i, which are obtained in the previous step, but also reduce the value of the criterion function E.
4. If both of the P-network and Z-network are stable or if the iteration count reaches a preselected limit, then the process is terminated. Otherwise, reset the weights as well as the inputs for both networks and go to Step 2.

Because both of the P-network and Z-network try to reduce the value of the criterion function E, the convergence of the neural network system can be expected. The question about how to design the two networks such that the criterion function value gradually decreases monotonically is discussed next.

IV. Proposed Moment-Preserving Neural Network Processor

The P-network in the proposed moment-preserving neural network processor is a recurrent network used to extract the parameters $P_j, j = 1, 2, \cdots, N$, in the moment-preserving equations, while the Z-network has a connectionist architecture and is used to modify the representative gray values Z_j. The criterion function E defined in (4) is used as the energy function and seeking the minimum value of E is the common goal of both networks.

The design of the P-network is discussed first. Taking the partial derivatives of E in (4) with respective to P_i (with m_k and $\tilde{m}_k$ defined in Section III), for all $i = 1, 2, \cdots, N$, we get a family of equations:

$$\frac{\partial E}{\partial P_i} = I_i + \sum_{j=1}^{N} w_{ij} P_j, \qquad i = 1, 2, 3, \cdots, N \tag{5}$$

where $I_i = -2\sum_l \tilde{P}_l \sum_{k=0}^{2N-1} (\tilde{Z}_l Z_i)^k$, and $w_{ij} = 2\sum_{k=0}^{2N-1} (Z_j Z_i)^k$. By regarding P_j as the values of the neurons of the P-network, w_{ij} can be considered as the weights of the network and I_i as the external inputs to the neurons. To minimize E, a gradient descent method is used here. Let

$$P_i^{(k+1)} = P_i^{(k)} - \lambda \frac{\partial E}{\partial P_i}, \qquad i = 1, 2, 3, \cdots, N \tag{6}$$

where λ is a positive variable (not a constant). The initial values of the neurons can be chosen arbitrarily. The undefined parameter λ above is called the gradient descent gain. The larger the value of λ, the higher the convergence rate. However, the energy function E will not always decrease at each iteration and so make the dynamic characteristic of the system unstable, if an unsuitable value of λ is chosen. The range of λ selected in this study [20] to assure monotonic decrease of the energy function is $0 < \lambda < 2\nabla_P^t \nabla_P / \nabla_P^t W \nabla_P$, where $W = \{w_{ij}\}$ $i, j = 1, 2, 3, \cdots, N$, is a matrix and $\nabla_P = [\partial E/\partial P_1, \partial E/\partial P_2, \cdots, \partial E/\partial P_N]$ is a vector.

The proposed architecture for the P-network is shown in Fig. 3(a). There are two different types of neurons in the network, namely, P-type and ∇_P-type neurons. Each P-type neuron performs the update function defined by (6), while the output value of each of ∇_P-type neuron is the derivative of the energy function E defined by (5). According to (5), the external inputs I_i work as *seeds* which indicate a way for the network to estimate $P_i, i = 1, 2, \cdots, N$, such that the moments of the input image are preserved. The information of the moments of the original image is included in each external input I_i. It is possible for the outputs of the summation functions in the P-type neurons to exceed the feasible interval [0,1] which is the range of fraction values, so a suitable activation function should be chosen to clamp the outputs of the P-type neurons into the interval [0,1].

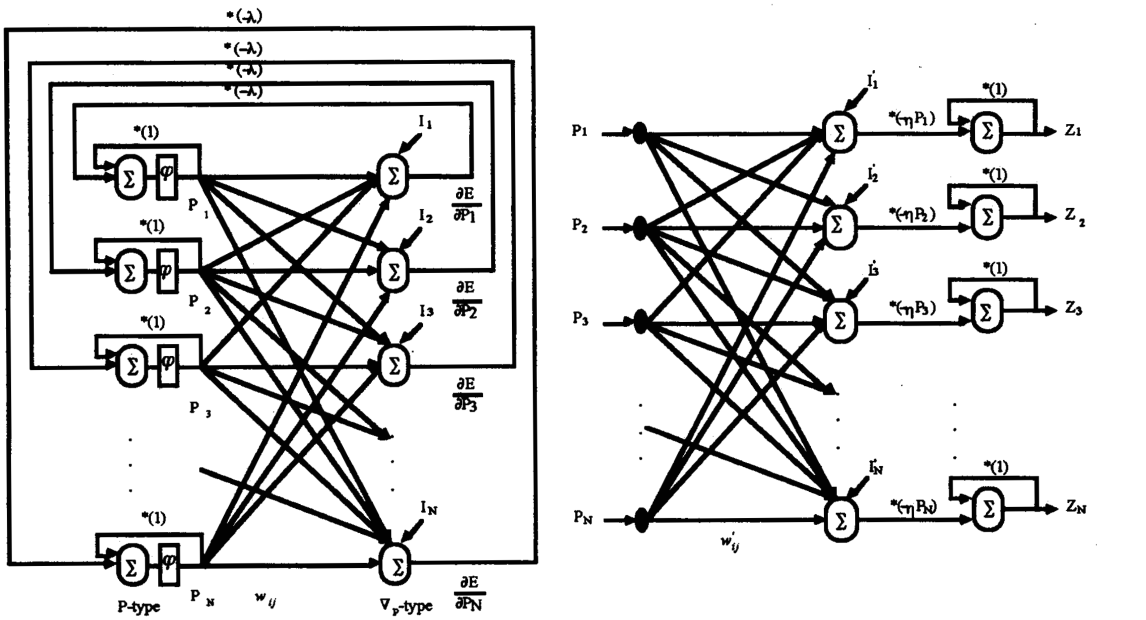


Fig. 3. Neural network architecture for order-N moment preserving. (a) P-network design; (b) Z-network design (notes: $w'_{ij} = \partial w_{ij}/\partial Z_i, I'_i = \partial I_i/\partial Z_i$).

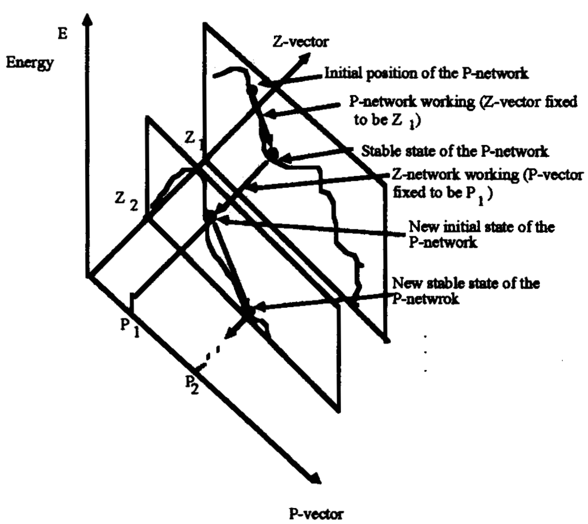


Fig. 4. Minimal energy value search by cooperating the P-network with the Z-network. The notations P-vector and Z-vector are parameters to be estimated by the moment-preserving equations.

The activation function of the P-type neuron used in the simulation of this study is defined as follows:

$$\varphi(x^{(k+1)}) = \begin{cases} x^{(k+1)} & \text{if } 0 < x < 1, \\ x^{(k)} & \text{otherwise,} \end{cases} \tag{7}$$

which can inhibit the values of P_i from crossing the boundary of its feasible interval [0,1].

When a stable state of the P-network is reached, the P-network stops working. The definition of a stable state is defined as $|P_i^{(k+1)} - P_i^{(k)}| < \xi$ for all i in this study, where the value of ξ is very small and positive. The Z-network performs the minimization of the energy function E after the P-network becomes stable. The design procedure for the Z-network is similar to that for the P-network. Taking partial derivatives of E with respective to Z_i, we get a family of equations:

$$\frac{\partial E}{\partial Z_i} = \left(\frac{\partial I_i}{\partial Z_i} + \sum_{j=1}^{N} \frac{\partial w_{ij}}{\partial Z_i} P_j\right)$$
$$P_i = \left(I'_i = \sum_{j=1}^{N} w'_{ij} P_j\right) P_i \tag{8}$$

where $I'_i = \partial I_i/\partial Z_i$ and $w'_{ij} = \partial w_{ij}/\partial Z_i$. Once more, let $\Delta Z_i = -\eta\, \partial E/\partial Z_i$. If ΔZ_i is sufficiently small, that is, if η is a very small positive constant, then

$$\Delta E \approx \sum_i \frac{\partial E}{\partial Z_i} \Delta Z_i = -\eta \sum_i \left(\frac{\partial E}{\partial Z_i}\right)^2 < 0. \tag{9}$$

The parameter η controls the refinement step of each new value of Z_i. Although the exact range of η is hard to determine theoretically, a feasible value for η can be chosen to be within [0,1]. Our experiments show that the resulting computing convergence speed is acceptable. The design of the Z-network architecture is based directly on the above equations. There are two external inputs to the Z-network, one being the values of P_i which are obtained by the P-network, and the other the values of I'_i which include the information of the moments of the original image. The architecture of the Z-network is shown in Fig. 3(b).

After the Z-network completes the job of updating the values of Z_i, the P-network takes over the energy function minimization task again. Note that, the weights and external inputs of both networks are dependent on the values of Z_i. So it is necessary to tune the weights and external inputs for both networks before a new energy minimization cycle starts again. In fact, the energy function value minimized by the P-network falls on a surface obtained by projecting the state space of the energy function E on a fixed Z-vector plane.

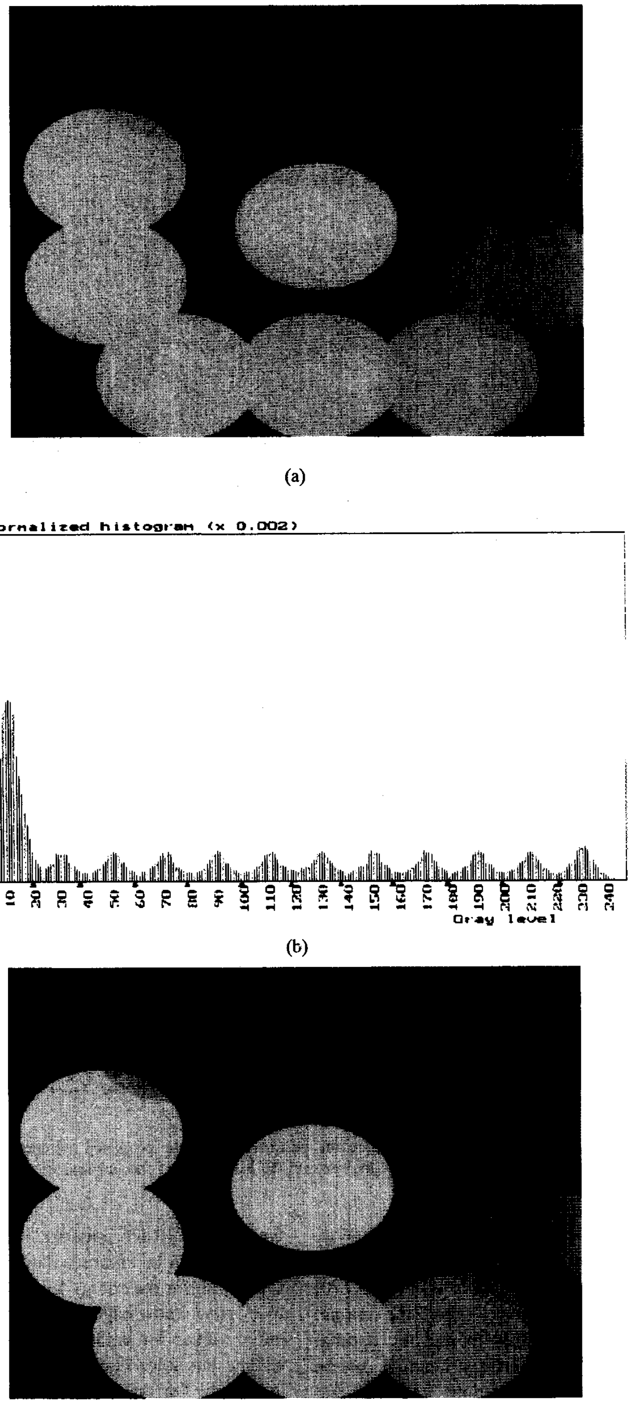
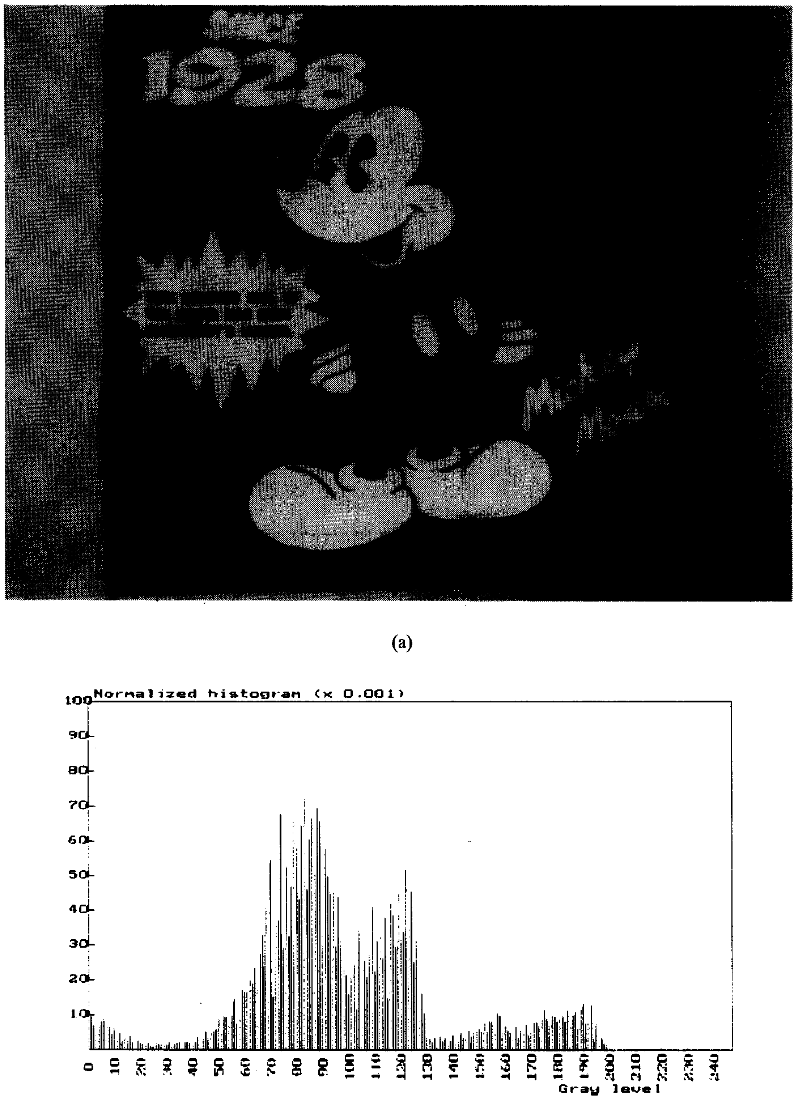


Fig. 5. 12-level thresholding of a synthetic image RING. (a) Original image; (b) histogram of the image with computed threshold values marked by the symbols Δ appearing on the axis of gray levels; (c) thresholding result.

The fact that the P-network becomes stable implies that the state of the P-network has been trapped in a local minimum of this surface. On the other hand, the action of the Z-network actually is to bring the state of the P-network to a new fixed state-space surface at a new fixed Z-vector. The P-network acts then to search a new stable state with a lower energy function value. Thus, an effect of the Z-network is to help the P-network to jump out of a local minimum of the energy function E. To tune the weights and external inputs of both networks



(a)

(b)

Fig. 6. The 5-color tested image MICKEY MOUSE used to compare the performance of the proposed technique with four other multilevel thresholding selections methods. (a) Original image; (b) histogram of the original image.

is equivalent to constructing a new state-space surface of the energy function E with a new fixed Z-vector. Fig. 4 is given for a more concise illustration. Because the energy function E is convex [20] for a fixed Z-vector, it is possible to exhaustively search the optimal Z-vector such that the P-network can be used to find the optimal P-vector. Then the global minimum of the energy function can be achieved. Unfortunately, this is an exponential-time algorithm and so impractical. The cooperation of the P-network and the Z-network in fact offers a fast search method such that the final solution of the moment-preserving neural network processor approximates the optimal solution.

The number of neurons used in the proposed system is small. The order is $O(N)$ where N is the number of pixel classes to be transformed. Thus, the system can be implemented by present VLSI fabrication technology. To complete the description of the neural network processor, both subnetworks work to decrease the energy function E at each updating iteration. Thus, the convergence of the processor can be assured. Because of the gradient descent method, it is possible for the energy function to be trapped on a local minimum. If *a priori* knowledge is given in advance to set up more suitable initial values for both subnetworks, good estimations of P_i and Z_i will be possible. This reduces the possibility of resulting in a local minimum. On the other hand, to jump out of the local minima of the energy function, some well-known optimization techniques, such as the simulated-annealing algorithm, can be used to improve the performance of the processor.

V. Experimental Results

The proposed approach has been tried on a lot of images. Each image is of size 512 by 480. All the simulations were done on a SUN/4 SPARC station. One result is shown in Fig. 5, which is the

TABLE I
COMPUTED RESULTS OF APPLYING THE FIVE MULTILEVEL THRESHOLDING ALGORITHMS TO THE TEST IMAGE

Methods	Threshold				CPU Time (second)
	t_1	t_2	t_3	t_4	
RANDC	42	79	104	146	1.1
FLOYD	38	83	106	147	1.1
MET	20	41	107	135	2733.3
ENTROPY	31	94	129	165	4100.3
MPNNP	65	92	122	160	18.1

result of 12-level thresholding of a synthetic image RING with 12 colors. A 5-color image MICKEY MOUSE, as shown in Fig. 6 is used to compare the performance of the proposed technique with those of four other histogram-based multilevel thresholding selections methods [16]–[19]. The algorithms compared are the following: 1) the Ridler and Calvard method [16] [RANDC]; 2) the Floyd method [17] [FLOYD]; 3) the Minimum error thresholding [18] [MET]; 4) the Entropy method [19] [ENTROPY]; 5) the proposed Moment-preserving neural network processor [MPNNP]. The final results are shown in Table I. It is difficult to define a good criterion to compare the performances of these methods. If such a criterion could be easily developed, one could design an algorithm to minimize (or maximize) the criterion. And then, the thresholding problem is completely solved. To compare the performances of these methods, the histogram of the original image might be useful. If the concavities of a histogram are the best suitable threshold candidates, it was observed from the experimental data that the performance of the method ENTROPY is the best, and the proposed MPNNP ranks second. However, if the computation time is considered, the MPNNP method is far superior to the entropy method. In order to compare the operation speeds of these methods, the programs written to simulate the MET and ENTROPY methods have been carefully optimized. All their floating point operations were precalculated and stored in the form of tables. The table look-up method speeds up greatly the performances of both the MET and the ENTROPY methods, otherwise the CPU time amounts for both methods would be far larger than the values listed in Table I. The CPU time listed in Table I for the proposed MPNNP method is the simulation time on a sequential machine.

VI. CONCLUSIONS AND DISCUSSION

In this correspondence, we have proposed a neural network approach to implementing the moment-preserving technique which has been proved to be of wide use [1]–[9]. An application of the proposed approach to thresholding has also been presented. The proposed method has been shown feasible for solving the moment-preserving equations, and the solution can be thought as a generalization of that derived by Tsai [1]. There is no difference between the effort required for solving lower-order moment-preserving equations using the proposed system and that for solving higher-order ones. Furthermore, the neural network development procedure and the theoretic derivations can also be applied to other image operations. The inherent parallelism of the proposed neural network system offers a possible way to achieve the goal of real-time image processing. Further studies can be directed to other applications of the proposed system in image processing, such as image segmentation and compression.

REFERENCES

[1] W. H. Tsai, "Moment-preserving thresholding: A new approach," *Comput. Vision, Graphics, Image Processing,* vol. 29, pp. 377–393, 1985.
[2] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Commun.,* vol. COM-27, pp. 1335–1342, 1979.
[3] A. J. Tabatabai and O. R. Mitchell, "Edge location to subpixel values in digital imagery," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-6, pp. 188–201, 1984.
[4] L. H. Chen and W. H. Tsai, "Moment-preserving sharpening: A new approach to digital picture deblurring," *Comput. Vision, Graphics, Image Processing,* vol. 41, pp. 1–13, 1988.
[5] ——, "Moment-preserving line detection," *Pattern Recognition,* vol. 21, pp. 45–53, 1988.
[6] ——, "Moment-preserving curve-detection," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-18, pp. 148–158, 1988.
[7] L. H. Chen and S. S. Wang, "A new approach for edge detection," in *Proc. IEEE Int. Conf. Image Processing,* Singapore, 1989.
[8] S. T. Liu and W. H. Tsai, "Moment-preserving clustering," *Pattern Recognition,* vol. 22, pp. 433–447, 1989.
[9] ——,"Moment-preserving corner detection," *Pattern Recognition,* vol. 23, pp. 441–460, 1990.
[10] A. Rosenfield and A. C. Kak, *Digital Picture Processing, Vol. II.* New York: Academic, 1982.
[11] Devijver and Kittler, *Pattern Recognition: A Statistical Approach.* London, England: Prentice-Hall International, 1982.
[12] J. J. Hopfield, "Neural networks and physical system with emergent collective computational abilities," *Proc. Nat. Acad. Sci., U.S.,* vol. 79, pp. 2554–2558, Apr. 1982.
[13] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.
[14] D. W. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-16, pp. 533–541, May 1986.
[15] E. J. Delp and O. R. Mitchell, "Moment-preserving quantization," *IEEE Trans. Commun.,* vol. COM-39, pp. 1549–1558, Nov. 1991.
[16] T. W. Ridler and S. Calvard, "Picture thresholding using an iterative selection method," *IEEE Trans. Syst., Man., Cybern.,* vol. SMC-8, pp. 630–632, Aug. 1978.
[17] D. E. Floyd, "Automatic target classification using moment invariants of image shapes," Farnborough, U.K., Rep. RAE IND AW126, Dec. 1985.
[18] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition,* vol. 19, pp. 41–47, 1986.
[19] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Compt. Vision, Graphics, Image Processing,* vol. 29, pp. 273–285, 1985.
[20] H. S. Hsu, I. H. Kuo, S. C. Cheng, and W. H. Tsai, "A study of high-speed and high-rate image data compression," National Science Council Technique Rep. NSC80–0404–E009–13, National Chiao Tung Univ., Taiwan, Hsinchu, Republic of China.