Image Restoration Using a Neural Network

YI-TONG ZHOU, STUDENT MEMBER, IEEE, RAMA CHELLAPPA, SENIOR MEMBER, IEEE, ASEEM VAID, AND B. KEITH JENKINS, MEMBER, IEEE

Abstract—**A new approach for restoration of gray level images degraded by a known shift-invariant blur function and additive noise is presented using a neural computational network. A neural network model is employed to represent a possibly nonstationary image whose gray level function is the simple sum of the neuron state variables. The restoration procedure consists of two stages: estimation of the parameters of the neural network model and reconstruction of images. During the first stage, the parameters are estimated by comparing the energy function of the network to a constrained error function. The nonlinear restoration method is then carried out iteratively in the second stage by using a dynamic algorithm to minimize the energy function of the network. Owing to the model's fault-tolerant nature and computation capability, a high-quality image is obtained using this approach. A practical algorithm with reduced computational complexity is also presented. Several computer simulation examples involving synthetic and real images are given to illustrate the usefulness of our method. The choice of the boundary values to reduce the ringing effect is discussed, and comparisons to other restoration methods such as the SVD pseudoinverse filter, minimum mean-square error (MMSE) filter, and modified MMSE filter using the Gaussian Markov random field model are given. Finally, a procedure for learning the blur parameters from prototypes of original and degraded images is outlined.**

I. INTRODUCTION

RESTORATION of a high-quality image from a degraded recording is an important problem in early vision processing. Restoration techniques are applied to remove 1) system degradations such as blur due to optical system aberrations, atmospheric turbulence, motion, and diffraction; and 2) statistical degradations due to noise. Over the last 20 years, various methods such as the inverse filter [1], Wiener filter [1], Kalman filter [2], SVD pseudoinverse [1], [3], and many other model-based approaches have been proposed for image restorations. One of the major drawbacks of most of the image restoration algorithms is the computational complexity, so much so that many simplifying assumptions such as wide sense stationarity (WSS), availability of second-order image statistics have been made to obtain computationally feasible algorithms. The inverse filter method works only for extremely high signal-to-noise ratio images. The Wiener filter is usually implemented only after the wide sense stationary assumption has been made for images. Furthermore, knowledge of the power spectrum or correlation matrix of the undegraded image is required. Often times, additional assumptions regarding boundary conditions are made so that fast orthogonal transforms can be used. The Kalman filter approach can be applied to nonstationary image, but is computationally very intensive. Similar statements can be made for the SVD pseudoinverse filter method. Approaches based on noncausal models such as the noncausal autoregressive or Gauss Markov random field models [4], [5] also make assumptions such as WSS and periodic boundary conditions. It is desirable to develop a restoration algorithm that does not make WSS assumptions and can be implemented in a reasonable time. An artificial neural network system that can perform extremely rapid computations seems to be very attractive for image restoration in particular and image processing and pattern recognition [6] in general.

In this paper, we use a neural network model containing redundant neurons to restore gray level images degraded by a known shift-invariant blur function and noise. It is based on the method described in [7]–[9] using a simple sum number representation [10]. The image gray levels are represented by the simple sum of the neuron state variables which take binary values of 1 or 0. The observed image is degraded by a shift-invariant function and noise. The restoration procedure consists of two stages: estimation of the parameters of the neural network model and reconstruction of images. During the first stage, the parameters are estimated by comparing the energy function of the neural network to the constrained error function. The nonlinear restoration algorithm is then implemented using a dynamic iterative algorithm to minimize the energy function of the neural network. Owing to the model's fault-tolerant nature and computation capability, a high-quality image is obtained using this approach. In order to reduce computational complexity, a practical algorithm, which has equivalent results to the original one suggested above, is developed under the assumption that the neurons are sequentially visited. We illustrate the usefulness of this approach by using both synthetic and real images degraded by a known shift-invariant blur function with or without noise. We also discuss the problem of choosing boundary values and introduce two methods to reduce the ringing effect. Comparisons to other restoration methods such as the SVD pseudoinverse filter, the minimum mean-square error (MMSE) filter, and the modified MMSE filter using a Gaussian Markov random field model are given using real images. The advantages of the method developed in this paper are: 1) WSS assumption is not required

Manuscript received February 22, 1988. This work was supported in part by AFOSR Contract F-49620-87-C-0007 and AFOSR Grant 86-0196.

The authors are with the Signal and Image Processing Institute, Department of Electrical Engineering—Systems, University of Southern California, Los Angeles, CA 90089.

IEEE Log Number 8821366.

0096-3518/88/0700-1141$01.00 © 1988 IEEE

for the images, 2) it can be implemented rapidly, and 3) it is fault tolerant.

In the above, the interconnection strengths (also called weights) of the neural network for image restoration are known from the parameters of the image degradation model and the smoothing constraints. We also consider learning of the parameters for the image degradation model and formulate it as a problem of computing the parameters from samples of the original and degraded images. This is implemented as a secondary neural network. A different scheme is used to represent multilevel activities for the parameters; some of its properties are complementary to those of the simple sum scheme. The learning procedure is accomplished by running a greedy algorithm. Some results of learning the blur parameters are presented using synthetic and real image examples.

The organization of this paper is as follows. A network model containing redundant neurons for image representation and the image degradation model is given in Section II. A technique for parameter estimation is presented in Section III. Image generation using a dynamic algorithm is described in Section IV. A practical algorithm with reduced computational complexity is presented in Section V. Computer simulation results using synthetic and real degraded images are given in Section VI. Choice of the boundary values is discussed in Section VII. Comparisons to other methods are given in Section VIII. A procedure for learning the blur parameters from prototypes of original and degraded images is outlined in Section IX, and conclusions and remarks are included in Section X.

II. A Neural Network for Image Representation

We use a neural network containing redundant neurons for representing the image gray levels. The model consists of $L^2 \times M$ mutually interconnected neurons where L is the size of image and M is the maximum value of the gray level function. Let $V = \{ v_{i,k}$ where $1 \le i \le L^2$, $1 \le k \le M \}$ be a binary state set of the neural network with $v_{i,k}$ (1 for firing and 0 for resting) denoting the state of the (i, k)th neuron. Let $T_{i,k;j,l}$ denote the strength (possibly negative) of the interconnection between neuron (i, k) and neuron (j, l). We require symmetry:

$$T_{i,k;j,l} = T_{j,l;i,k} \quad \text{for } 1 \le i, j \le L^2 \text{ and } 1 \le l, k \le M.$$

We also allow for neurons to have self-feedback, i.e., $T_{i,k;i,k} \neq 0$. In this model, each neuron (i, k) randomly and asynchronously receives inputs $\Sigma T_{i,k;j,l} v_{j,l}$ from all neurons and a bias input $I_{i,k}$:

$$u_{i,k} = \sum_{j}^{L^2} \sum_{l}^{M} T_{i,k;j,l} v_{j,l} + I_{i,k}. \tag{1}$$

Each $u_{i,k}$ is fed back to corresponding neurons after thresholding:

$$v_{i,k} = g(u_{i,k}) \tag{2}$$

where $g(x)$ is a nonlinear function whose form can be taken as

$$g(x) = \begin{cases} 1 & \text{if } x \ge 0 \\ 0 & \text{if } x < 0. \end{cases} \tag{3}$$

In this model, the state of each neuron is updated by using the latest information about other neurons.

The image is described by a finite set of gray level functions $\{x(i, j)$ where $1 \le i, j \le L\}$ with $x(i, j)$ (positive integer number) denoting the gray level of the pixel (i, j). The image gray level function can be represented by a simple sum of the neuron state variables as

$$x(i, j) = \sum_{k=1}^{M} v_{m,k} \tag{4}$$

where $m = (i - 1) \times L + j$. Here the gray level functions have degenerate representations. Use of this redundant number representation scheme yields advantages such as fault tolerance and faster convergence to the solution [10].

By using the lexicographic notation, the image degradation model can be written as

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{N} \tag{5}$$

where $\boldsymbol{H}$ is the "blur matrix" corresponding to a blur function, $\boldsymbol{N}$ is the signal independent white noise, and $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the original and degraded images, respectively. Furthermore, $\boldsymbol{H}$ and $\boldsymbol{N}$ can be represented as

$$\boldsymbol{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,L^2} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,L^2} \\ \vdots & \vdots & \cdots & \vdots \\ h_{L^2,1} & h_{L^2,2} & \cdots & h_{L^2,L^2} \end{bmatrix} \tag{6}$$

and

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{N}_1 \\ \boldsymbol{N}_2 \\ \vdots \\ \boldsymbol{N}_L \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{L^2} \end{bmatrix},$$

$$\boldsymbol{N}_i = \begin{bmatrix} n(i, 1) \\ n(i, 2) \\ \vdots \\ n(i, L) \end{bmatrix} = \begin{bmatrix} n_{(i-1)\times L+1} \\ n_{(i-1)\times L+2} \\ \vdots \\ n_{i\times L} \end{bmatrix} \tag{7}$$

respectively. Vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ have similar representations. Equation (5) is similar to the simultaneous equations solution of [10], but differs in that it includes a noise term.

The shift-invariant blur function can be written as a convolution over a small window, for instance, it takes

the form

$$h(k,l) = \begin{cases} \frac{1}{2} & \text{if } k = 0,\ l = 0 \\ \frac{1}{16} & \text{if } |k|,\ |l| \le 1,\ (k,l) \ne (0,0); \end{cases} \tag{8}$$

accordingly, the "blur matrix" H will be a block Toeplitz or block circulant matrix (if the image has periodic boundaries). The block circulant matrix corresponding to (8) can be written as

$$H = \begin{bmatrix} H_0 & H_1 & \mathbf{0} & \cdots & \mathbf{0} & H_1 \\ H_1 & H_0 & H_1 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ H_1 & \mathbf{0} & \mathbf{0} & \cdots & H_1 & H_0 \end{bmatrix} \tag{9}$$

where

$$H_0 = \begin{bmatrix} \frac{1}{2} & \frac{1}{16} & 0 & \cdots & 0 & \frac{1}{16} \\ \frac{1}{16} & \frac{1}{2} & \frac{1}{16} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{1}{16} & 0 & 0 & \cdots & \frac{1}{16} & \frac{1}{2} \end{bmatrix},$$

$$H_1 = \begin{bmatrix} \frac{1}{16} & \frac{1}{16} & 0 & \cdots & 0 & \frac{1}{16} \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{1}{16} & 0 & 0 & \cdots & \frac{1}{16} & \frac{1}{16} \end{bmatrix}, \tag{10}$$

and $\mathbf{0}$ is null matrix whose elements are all zeros.

III. Estimation of Model Parameters

The neural model parameters, the interconnection strengths, and bias inputs can be determined in terms of the energy function of the neural network. As defined in [7], the energy function of the neural network can be written as

$$E = -\frac{1}{2}\sum_{i=1}^{L^2}\sum_{j=1}^{L^2}\sum_{k=1}^{M}\sum_{l=1}^{M} T_{i,k;j,l}v_{i,k}v_{j,l} - \sum_{i=1}^{L^2}\sum_{k=1}^{M} I_{i,k}v_{i,k}. \tag{11}$$

In order to use the spontaneous energy-minimization process of the neural network, we reformulate the restoration problem as one of minimizing an error function with constraints defined as

$$E = \frac{1}{2}\left\| \boldsymbol{Y} - H\hat{\boldsymbol{X}} \right\|^2 + \frac{1}{2}\lambda \left\| D\hat{\boldsymbol{X}} \right\|^2 \tag{12}$$

where $\| \boldsymbol{Z} \|$ is the L_2 norm of $\boldsymbol{Z}$ and λ is a constant. Such a constrained error function is widely used in the image restoration problems [1] and is also similar to the regularization techniques used in early vision problems [11]. The first term in (12) is to seek an $\hat{\boldsymbol{X}}$ such that $H\hat{\boldsymbol{X}}$ approximates $\boldsymbol{Y}$ in a least squares sense. Meanwhile, the second term is a smoothness constraint on the solution $\hat{\boldsymbol{X}}$. The constant λ determines their relative importance to achieve both noise suppression and ringing reduction.

In general, if H is a low-pass distortion, then D is a high-pass filter. A common choice of D is a second-order differential operator which can be approximated as a local window operator in the 2-D discrete case. For instance, if D is a Laplacian operator

$$\nabla = \frac{\partial^2}{\partial i^2} + \frac{\partial^2}{\partial j^2} \tag{13}$$

it can be approximated as a window operator

$$\frac{1}{6}\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}. \tag{14}$$

Then D will be a block Toeplitz matrix similar to (9).

Expanding (12) and then replacing x_i by (4), we have

$$\begin{aligned} E &= \frac{1}{2}\sum_{p=1}^{L^2}\left(y_p - \sum_{i=1}^{L^2} h_{p,i}x_i\right)^2 + \frac{1}{2}\lambda\sum_{p=1}^{L^2}\left(\sum_{i=1}^{L^2} d_{p,i}x_i\right)^2 \\ &= \frac{1}{2}\sum_{i=1}^{L^2}\sum_{j=1}^{L^2}\sum_{k=1}^{M}\sum_{l=1}^{M}\sum_{p=1}^{L^2} h_{p,i}h_{p,j}v_{i,k}v_{j,l} \\ &\quad + \frac{1}{2}\lambda\sum_{i=1}^{L^2}\sum_{j=1}^{L^2}\sum_{k=1}^{M}\sum_{l=1}^{M}\sum_{p=1}^{L^2} d_{p,i}d_{p,j}v_{i,k}v_{j,l} \\ &\quad - \sum_{i=1}^{L^2}\sum_{k=1}^{M}\sum_{p=1}^{L^2} y_p h_{p,i}v_{i,k} + \frac{1}{2}\sum_{p=1}^{L^2} y_p^2. \end{aligned} \tag{15}$$

By comparing the terms in (15) to the corresponding terms in (11) and ignoring the constant term $\frac{1}{2}\Sigma_{p=1}^{L^2} y_p^2$, we can determine the interconnection strengths and bias inputs as

$$T_{i,k;j,l} = -\sum_{p=1}^{L^2} h_{p,i}h_{p,j} - \lambda\sum_{p=1}^{L^2} d_{p,i}d_{p,j} \tag{16}$$

and

$$I_{i,k} = \sum_{p=1}^{L^2} y_p h_{p,i} \tag{17}$$

where $h_{i,j}$ and $d_{i,j}$ are the elements of the matrices H and D, respectively. Two interesting aspects of (16) and (17) should be pointed out: 1) the interconnection strengths are independent of subscripts k and l and the bias inputs are independent of subscript k, and 2) the self-connection $T_{i,k;i,k}$ is not equal to zero which requires self-feedback for neurons.

From (16), one can see that the interconnection strengths are determined by the shift-invariant blur function, differential operator, and constant λ. Hence, $T_{i,k;j,l}$ can be computed without error provided the blur function

is known. However, the bias inputs are functions of the observed degraded image. If the image is degraded by a shift-invariant blur function only, then $I_{i,k}$ can be estimated perfectly. Otherwise, $I_{i,k}$ is affected by noise. The reasoning behind this statement is as follows. By replacing y_p by $\Sigma_{i=1}^{L^2} h_{p,i}x_i + n_p$, we have

$$I_{i,k} = \sum_{p=1}^{L^2}\left(\sum_{i=1}^{L^2} h_{p,i}x_i + n_p\right)h_{p,i} = \sum_{p=1}^{L^2}\sum_{i=1}^{L^2} h_{p,i}x_i h_p + \sum_{p=1}^{L^2} n_p h_{p,i}. \quad (18)$$

The second term in (18) represents the effects of noise. If the signal-to-noise ratio (SNR), defined by

$$\text{SNR} = 10 \log_{10} \frac{\sigma_s^2}{\sigma_n^2} \quad (19)$$

where σ_s^2 and σ_n^2 are variances of signal and noise, respectively, is low, then we have to choose a large λ to suppress effects due to noise. It seems that in the absence of noise, the parameters can be estimated perfectly, ensuring exact recovery of the image as error function E tends to zero. However, the problem is not so simple because the restoration performance depends on both the parameters and the blur function when a mean-square error or least square error such as (12) is used. A discussion about the effect of blur function is given in Section X.

IV. Restoration

Restoration is carried out by neuron evaluation and an image construction procedure. Once the parameters $T_{i,k;j,l}$ and $I_{i,k}$ are obtained using (16) and (17), each neuron can randomly and asynchronously evaluate its state and readjust accordingly using (1) and (2). When one quasi-minimum energy point is reached, the image can be constructed using (4).

However, this neural network has self-feedback, i.e., $T_{i,k;i,k} \neq 0$. As a result, the energy function E does not always decrease monotonically with a transition. This is explained below. Define the state change $\Delta v_{i,k}$ of neuron (i, k) and energy change ΔE as

$$\Delta v_{i,k} = v_{i,k}^{\text{new}} - v_{i,k}^{\text{old}} \quad \text{and} \quad \Delta E = E^{\text{new}} - E^{\text{old}}.$$

Consider the energy function

$$E = -\tfrac{1}{2}\sum_{i=1}^{L^2}\sum_{j=1}^{L^2}\sum_{k=1}^{M}\sum_{l=1}^{M} T_{i,k;j,l}v_{i,k}v_{j,l} - \sum_{i=1}^{L^2}\sum_{k=1}^{M} I_{i,k}v_{i,k}. \quad (20)$$

Then the change ΔE due to a change $\Delta v_{i,k}$ is given by

$$\Delta E = -\left(\sum_{j=1}^{L^2}\sum_{l=1}^{M} T_{i,k;j,l}v_{j,l} + \mathrm{I}_{i,k}\right)\Delta v_{i,k} - \tfrac{1}{2}T_{i,k;i,k}(\Delta v_{i,k})^2 \quad (21)$$

which is not always negative. For instance, if

$$v_{i,k}^{\text{old}} = 0, \quad u_{i,k} = \sum_{j=1}^{L^2}\sum_{l=1}^{M} T_{i,k;j,l}v_{j,l} + I_{i,k} > 0$$

and the threshold function is as in (3), then $v_{i,k}^{\text{new}} = 1$ and $\Delta v_{i,k} > 0$. Thus, the first term in (21) is negative. But

$$T_{i,k;i,k} = -\sum_{p=1}^{L^2} h_{p,i}^2 - \lambda \sum_{p=1}^{L^2} d_{p,i}^2 < 0$$

with $\lambda > 0$, leading to

$$-\tfrac{1}{2}T_{i,k;i,k}(\Delta v_{i,k})^2 > 0.$$

When the first term is less than the second term in (21), then $\Delta E > 0$ (we have observed this in our experiment), which means E is not a Lyapunov function. Consequently, the convergence of the network is not guaranteed [12].

Thus, depending on whether convergence to a local minimum or a global minimum is desired, we can design a deterministic or stochastic decision rule. The deterministic rule is to take a new state $v_{i,k}^{\text{new}}$ of neuron (i, k) if the energy change ΔE due to state change $\Delta v_{i,k}$ is less than zero. If ΔE due to state change is > 0, no state change is affected. One can also design a stochastic rule similar to the one used in stimulated annealing techniques [13], [14]. The details of this stochastic scheme are given as follows.

Define a Boltzmann distribution by

$$\frac{p_{\text{new}}}{p_{\text{old}}} = e^{-\Delta E/T}$$

where p_{new} and p_{old} are the probabilities of the new and old global state, respectively, ΔE is the energy change, and T is the parameter which acts like temperature. A new state $v_{i,k}^{\text{new}}$ is taken if

$$\frac{p_{\text{new}}}{p_{\text{old}}} > 1 \text{ or if } \frac{p_{\text{new}}}{p_{\text{old}}} \leq 1 \text{ but } \frac{p_{\text{new}}}{p_{\text{old}}} > \xi$$

where ξ is a random number uniformly distributed in the interval $[0, 1]$.

The restoration algorithm is summarized as below.

Algorithm 1:

1) Set the initial state of the neurons.

2) Update the state of all neurons randomly and asynchronously according to the decision rule.

3) Check the energy function; if energy does not change, go to step 4); otherwise, go back to step 2).

4) Construct an image using (4).

V. A Practical Algorithm

The algorithm described above is difficult to simulate on a conventional computer owing to high computational complexity, even for images of reasonable size. For instance, if we have an $L \times L$ image with M gray levels, then L^2M neurons and $\frac{1}{2}L^4M^2$ interconnections are required and L^4M^2 additions and multiplications are needed

at each iteration. Therefore, the space and time complexities are $O(L^4M^2)$ and $O(L^4M^2K)$, respectively, where K, typically 10–100, is the number of iterations. Usually, L and M are 256–1024 and 256, respectively. However, simplification is possible if the neurons are sequentially updated.

In order to simplify the algorithm, we begin by reconsidering (1) and (2) of the neural network. As noted earlier, the interconnection strengths given in (16) are independent of subscripts k and l and the bias inputs given in (17) are independent of subscript k; the M neurons used to represent the same image gray level function have the same interconnection strengths and bias inputs. Hence, one set of interconnection strengths and one bias input are sufficient for every gray level function, i.e., the dimensions of the interconnection matrix T and bias input matrix I can be reduced by a factor of M^2. From (1), all inputs received by a neuron, say the (i, k)th neuron, can be written as

$$u_{i,k} = \sum_{j}^{L^2} T_{i,\cdot;j,\cdot} \left(\sum_{l}^{M} v_{j,l} \right) + I_{i,\cdot} = \sum_{j}^{L^2} T_{i,\cdot;j,\cdot} x_j + I_{i,\cdot} \tag{22}$$

where we have used (4) and x_j is the gray level function of the jth image pixel. The symbol "$\cdot$" in the subscripts means that the $T_{i,\cdot;j,\cdot}$ and $I_{i,\cdot}$ are independent of k. Equation (22) suggests that we can use a multivalue number to replace the simple sum number. Since the interconnection strengths are determined by the blur function, the differential operator, and the constant λ as shown in (16), it is easy to see that if the blur function is local, then most interconnection strengths are zeros and the neurons are locally connected. Therefore, most elements of the interconnection matrix T are zeros. If the blur function is shift invariant taking the form in (8), then the interconnection matrix is block Toeplitz so that only a few elements need to be stored. Based on the value of inputs $u_{i,k}$, the state of the (i, k)th neuron is updated by applying a decision rule. The state change of the (i, k)th neuron in turn causes the gray level function x_i to change:

$$x_i^{\text{new}} = \begin{cases} x_i^{\text{old}} & \text{if } \Delta v_{i,k} = 0 \\ x_i^{\text{old}} + 1 & \text{if } \Delta v_{i,k} = 1 \\ x_i^{\text{old}} - 1 & \text{if } \Delta v_{i,k} = -1 \end{cases} \tag{23}$$

where $\Delta v_{i,k} = v_{i,k}^{\text{new}} - v_{i,k}^{\text{old}}$ is the state change of the (i, k)th neuron. The superscripts "new" and "old" are for after and before updating, respectively. We use x_i to represent the gray level value as well as the output of M neurons representing x_i. Assuming that the neurons of the network are sequentially visited, it is straightforward to show that the updating procedure can be reformulated as

$$u_{i,k} = \sum_{j}^{L^2} T_{i,\cdot;j,\cdot} x_j + I_{i,\cdot} \tag{24}$$

$$\Delta v_{i,k} = g(u_{i,k}) = \begin{cases} \Delta v_{i,k} = 0 & \text{if } u_{i,k} = 0 \\ \Delta v_{i,k} = 1 & \text{if } u_{i,k} > 0 \\ \Delta v_{i,k} = -1 & \text{if } u_{i,k} < 0 \end{cases} \tag{25}$$

$$x_i^{\text{new}} = \begin{cases} x_i^{\text{old}} + \Delta v_{i,k} & \text{if } \Delta E < 0 \\ x_i^{\text{old}} & \text{if } \Delta E \geq 0. \end{cases} \tag{26}$$

Note that the stochastic decision rule can also be used in (26). In order to limit the gray level function to the range 0–255 after each updating step, we have to check the value of the gray level function x_i^{new}. Equations (24), (25), and (26) give a much simpler algorithm. This algorithm is summarized below.

Algorithm 2:

1) Take the degraded image as the initial value.

2) Sequentially visit all numbers (image pixels). For each number, use (24), (25), and (26) to update it repeatedly until there is no further change, i.e., if $\Delta v_{i,k} = 0$ or energy change $\Delta E \geq 0$; then move to the next one.

3) Check the energy function; if energy does not change anymore, a restored image is obtained; otherwise, go back to step 2) for another iteration.

The calculations of the inputs $u_{i,k}$ of the (i, k)th neuron and the energy change ΔE can be simplified furthermore. When we update the same image gray level function repeatedly, the input received by the current neuron (i, k) can be computed by making use of the previous result

$$u_{i,k} = u_{i,k-1} + \Delta v_{i,k} T_{i,\cdot;i,\cdot} \tag{27}$$

where $u_{i,k-1}$ is the inputs received by the $(i, k-1)$th neuron. The energy change ΔE due to the state change of the (i, k)th neuron can be calculated as

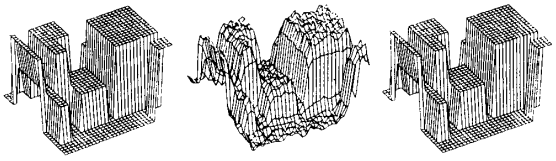
$$\Delta E = -u_{i,k} \Delta v_{i,k} - \tfrac{1}{2} T_{i,\cdot;i,\cdot} (\Delta v_{i,k})^2. \tag{28}$$

If the blur function is shift invariant, all these simplifications reduce the space and time complexities significantly from $O(L^4M^2)$ and $O(L^4M^2K)$ to $O(L^2)$ and $O(ML^2K)$, respectively. Since every gray level function needs only a few updating steps after the first iteration, the computation at each iteration is $O(L^2)$. The resulting algorithm can be easily simulated on minicomputers for images as large as 512×512.

VI. Computer Simulations

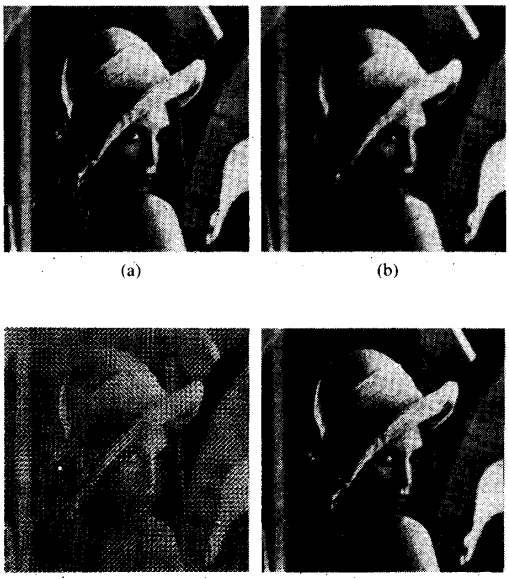
The practical algorithm described in the previous section was applied to synthetic and real images on a Sun-3/160 Workstation. In all cases, only the deterministic decision rule was used. The results are summarized in Figs. 1 and 2.

Fig. 1 shows the results for a synthetic image. The original image shown in Fig. 1(a) is of size 32×32 with three gray levels. The image was degraded by convolving with a 3×3 blur function as in (8) using circulant boundary conditions; 22 dB white Gaussian noise was added after convolution. A perfect image was obtained after six



(a) (b) (c)

Fig. 1. Restoration of noisy blurred synthetic image. (a) Original image. (b) Degraded image. (c) Result after six iterations.



(a) (b)

(c) (d)

Fig. 2. Restoration of noisy blurred real image. (a) Original girl image. (b) Image degraded by 5 × 5 uniform blur and quantization noise. (c) The restored image using inverse filter. (d) The restored image using our approach.

iterations without preprocessing. We set the initial state of all neurons to equal 1, i.e., firing, and chose $\lambda = 0$ due to the well conditioning of the blur function.

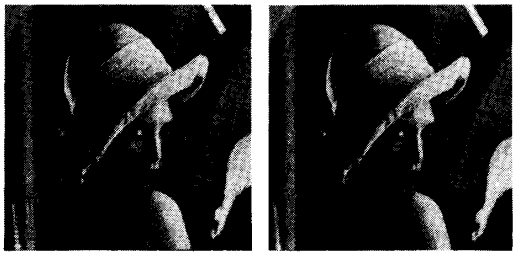
Fig. 2(a) shows the original girl image. The original image is of size 256 × 256 with 256 gray levels. The variance of the original image is 2797.141. It was degraded by a 5 × 5 uniform blur function. A small amount of quantization noise was introduced by quantizing the convolution results to 8 bits. The noisy blurred image is shown in Fig. 2(b). For comparison purpose, Fig. 2(c) shows the output of an inverse filter [15], completely overridden by the amplified noise and the ringing effects due to the ill-conditioned blur matrix H. Since the blur matrix H corresponding to the 5 × 5 uniform blur function is not singular, the pseudoinverse filter [15] and the inverse filter have the same output. The restored image by using our approach is shown in Fig. 2(d). In order to avoid the ringing effects due to the boundary conditions, we took 4 pixel wide boundaries, i.e., the first and last four rows and columns, from the original image and updated the interior region (248 × 248) of the image only. The noisy blurred image was used as an initial condition for accelerating the convergence. The constant λ was set to zero because of small noise and good boundary values. The restored image in Fig. 2(d) was obtained after 213 iterations. The square error (i.e., energy function) defined in (12) is 0.02543 and the square error between the original and the restored image is 66.5027.

VII. Choosing Boundary Values

As mentioned in [16], choosing boundary values is a common problem for techniques ranging from deterministic inverse filter algorithms to stochastic Kalman filters. In these algorithms, boundary values determine the entire solution when the blur is uniform [17]. The same problem occurs in the neural network approach. Since the 5 × 5 uniform blur function is ill conditioned, improper boundary values may cause ringing which may affect the restored image completely. For example, appending zeros to the image as boundary values introduces a sharp edge at the image border and triggers ringing in the restored image even if the image has zero mean. Another procedure is to assume a periodic boundary. When the left (top) and right (bottom) borders of the image are different, a sharp edge is formed and ringing results even though the degraded image has been formed by blurring with periodic boundary conditions. The drawbacks of these two assumptions for boundary values were reported in [16], [2], [18] for the 2-D Kalman filtering technique. We also tested our algorithm using these two assumptions for boundary values; the results indicate the restored images were seriously affected by ringing.

In the last section, to avoid the ringing effect, we took 4 pixel wide borders from the original image as boundary values for restoration. Since the original image is not available in practice always, an alternative to eliminate the ringing effect caused by sharp false edges is to use the blurred noisy boundaries from the degraded image. Fig. 3(a) shows the restored image using the first and last four rows and columns of the blurred noisy image in Fig. 2(b) as boundary values. In the restored image, there still exists some ringing due to the naturally occurring sharp edges in the region near the borders in the original image, but not due to boundary values. A typical cut of the restored image to illustrate ringing near the borders is shown in Fig. 4. To remove the ringing near the borders caused by naturally occurring sharp edges in the original image, we suggest the following techniques.

First, divide the image into three regions: border, subborder, and interior region as shown in Fig. 5. For the 5 × 5 uniform blur case, the border region will be 4 pixels wide due to the boundary effect of the bias input $I_{i,k}$ in (17), and the subborder region will be 4 or 8 pixels wide. In fact, the width of the subborder region will be image dependent. If the regions near the border are smooth, then the width of the subborder region will be small or even zero. If the border contains many sharp edges, the width will be large. For the real girl image, we chose the width



(a) (b)

(c)

Fig. 3. Results using blurred noisy boundaries. (a) Blurred noisy boundaries. (b) Method 1. (c) Method 2.

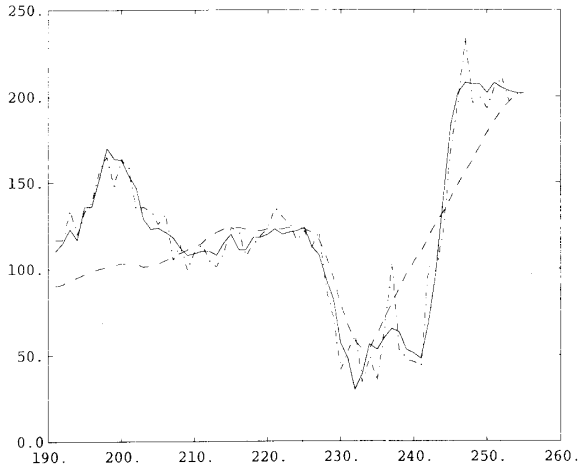


Fig. 4. One typical cut of the restored image using the blurred noisy boundaries. Solid line for original image, dashed line for blurred noisy image, and dashed and dotted line for restored image.

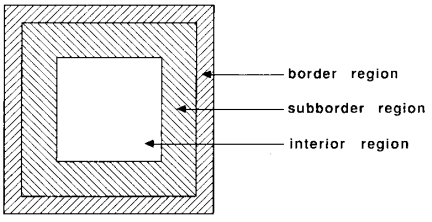


Fig. 5. Border, subborder, and interior regions of the image.

of the subborder region to be 8 pixels. We suggest using one of the following two methods.

Method 1: In the case of small noise, such as quantization error noise, the blurred image is usually smooth. Therefore, we restricted the difference between the restored and blurred image in the subborder region to a certain range to reduce the ringing effect. Mathematically, this constraint can be written as

$$\|\hat{x}_i - y_i\| \leq T \quad \text{for } i \in \text{subborder region} \quad (29)$$

where T is a threshold and $\hat{x}_i$ is the restored image gray value. Fig. 3(b) shows the result of using this method with $T = 10$.

Method 2: This method simply sets λ in (12) to zero in the interior region and nonzero in the subborder region, respectively. Fig. 3(c) shows the result of using this method with $\lambda = 0.09$. In this case, D was a Laplacian operator.

Owing to checking all restored image gray values in the subborder region, Method 1 needs more computation than Method 2. However, Method 2 is very sensitive to the parameter λ, while Method 1 is not so sensitive to the parameter λ. Experimental results show that both Methods 1 and 2 reduce the ringing effect significantly by using the suboptimal blurred boundary values.

VIII. Comparisons to Other Restoration Methods

Comparing the performance of different restoration methods needs some quality measures which are difficult to define owing to the lack of knowledge about the human visual system. The word "optimal" used in the restoration techniques usually refers only to a mathematical concept, and is not related to response of the human visual system. For instance, when the blur function is ill conditioned and the SNR is low, the MMSE method improves the SNR, but the resulting image is not visually good. We believe that human objective evaluation is the best ultimate judgment. Meanwhile, the mean-square error or least square error can be used as a reference.

For comparison purposes, we give the outputs of the inverse filter, SVD pseudoinverse filter, MMSE filter, and modified MMSE filter using the Gaussian Markov random field (GMRF) model [19], [5].

A. Inverse Filter and SVD Pseudoinverse Filter

An inverse filter can be used to restore an image degraded by a space-invariant blur function with high signal-to-noise ratio. When the blur function has some singular points, an SVD pseudoinverse filter is needed; however, both filters are very sensitive to noise. This is because the noise is amplified in the same way as the signal components to be restored. The inverse filter and SVD pseudoinverse filter were applied to an image degraded by the 5×5 uniform blur function and quantization noise (about 40 dB SNR). The blurred and restored images are shown in Fig. 2(b) and (c), respectively. As we mentioned before, the outputs of these filters are completely overridden by the amplified noise and ringing effects.

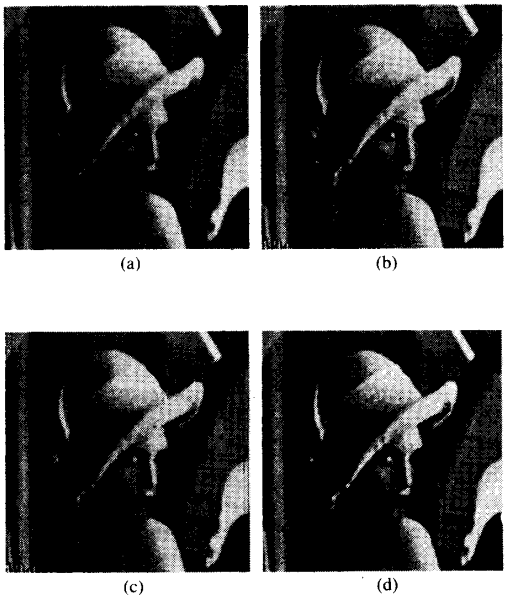


Fig. 6. Comparison to other restoration methods. (a) Image degraded by 5 × 5 uniform blur and 20 dB SNR additive white Gaussian noise. (b) The restored image using the MMSE filter. (c) The restored image using the modified MMSE filter. (d) The restored image using our approach.

B. MMSE and Modified MMSE Filters

The MMSE filter is also known as the Wiener filter (in the frequency domain). Under the assumption that the original image obeys a GMRF model, the MMSE filter (or Wiener filter) can be represented in terms of the GMRF model parameters and the blur function. In our implementation of the MMSE filter, we used a known blur function, unknown noise variance, and the GMRF model parameters estimated from the blurred noisy image by a maximum likelihood (ML) method [19]. The image shown in Fig. 6(a) was degraded by 5 × 5 uniform blur function and 20 dB SNR additive white Gaussian noise. The restored image is shown in Fig. 6(b).

The modified MMSE filter in terms of the GMRF model parameters is a linear weighted combination of a Wiener filter with a smoothing operator (such as a median filter) and a pseudoinverse filter to smooth the noise and preserve the edge of the restored image simultaneously. Details of this filter can be found in [5]. We applied the modified MMSE filter to the same image used in the MMSE filter above with the same model parameters. The smoothing operator is a 9 × 9 cross shape median filter. The resulting image is shown in Fig. 6(c).

The result of our method is also shown in Fig. 6(d). The D we used in (12) was a Laplacian operator as in (13). We chose $\lambda = 0.0625$ and used 4 pixel wide blurred noisy boundaries for restoration. The total number of iterations was 20. The improvement of mean-square error between the restored image and the original image for each method is shown in Table I. In the table, the "MMSE (o)" denotes that the parameters were estimated from the original image. The restored image using "MMSE (o)" is very similar to Fig. 6(a). As we mentioned before, the comparison of the outputs of the different restoration methods is a difficult problem. The MMSE filter visually gives the worst output which has the smallest mean-square error for the MMSE (o) case. The result of our method is smoother than that of the MMSE filter. Although the output of the modified MMSE filter is smooth in flat regions, it contains some artifacts and snake effects at the edges due to using a large sized median filter.

TABLE I
MEAN-SQUARE ERROR IMPROVEMENT

Method	MMSE	MMSE (o)	Modified MMSE	Neural Network
Mean-square error	1.384 dB	2.139 dB	1.893 dB	1.682 dB

IX. Parameter Learning for Linear Image Blur Model

Apart from fine-grain parallelism, fast (and preferably automatic) adaptation of a problem-solving network to different instances of a problem is a primary motivation for using a network solution. For pattern recognition and associative memory applications, this weight training is done by distributed algorithms that optimize a distance measure between sample patterns and network responses. However, in feedback networks, general problems that involve learning higher order correlations (like the exclusive OR) or combinatorial training sets (like the Traveling Salesperson problem) are difficult to solve and may have exponential complexity. In particular, techniques for finding a compact training set do not exist.

A. Learning Model

For model-based approaches to "neural" problem solving, the weights of the main network are computed from the parameters of the model. The learning problem can then be solved by a parallel, distributed algorithm for estimating the model parameters from samples of the inputs and desired outputs. This algorithm can be implemented on a secondary network. An error function for this "learning" network must be constructed, which will now be problem-dependent.

For the linear shift-invariant blur model (5), the problem is that of estimating the parameters corresponding to the blur function in a $K \times K$ small window centered at each pixel. Rewrite (5) as

$$y(i,j) = \boldsymbol{z}(i,j)^t \boldsymbol{h} + n(i,j) \qquad i,j = 1, 2, \cdots, L \tag{30}$$

where t denotes the transpose operator and $\boldsymbol{z}(i,j)$ and $\boldsymbol{h}$ are $K^2 \times 1$ vectors corresponding to original image samples in a $K \times K$ window centered at (i,j) and blur function, respectively.

For instance, for $K = 3$, we have

$$\boldsymbol{h} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_9 \end{bmatrix} = \begin{bmatrix} h(-1,-1) \\ h(-1,0) \\ h(-1,1) \\ \vdots \\ h(1,1) \end{bmatrix} \tag{31}$$

and

$$\boldsymbol{z}(i,j) = \begin{bmatrix} z(i,j)_1 \\ z(i,j)_2 \\ z(i,j)_3 \\ \vdots \\ z(i,j)_9 \end{bmatrix} = \begin{bmatrix} x(i-1,j-1) \\ x(i-1,j) \\ x(i-1,j+1) \\ \vdots \\ x(i+1,j+1) \end{bmatrix}. \tag{32}$$

We can use an error function for estimation of $\boldsymbol{h}$, as in the restoration process, because the roles of data $\{x(i,j)\}$ and parameter $\boldsymbol{h}$ are simply interchanged in the learning process. Therefore, an error function is defined as

$$E = \sum_{(i,j)\in S} \left[y(i,j) - \boldsymbol{h}^t \boldsymbol{z}(i,j) \right]^2 \tag{33}$$

where S is a subset of $\{(i,j), i,j = 1, 2, \cdots, L\}$ and $y(i,j)$ and $\boldsymbol{z}(i,j)$ are training samples taken from the degraded and original images, respectively. The network energy functions is given by

$$E = -\sum_{k=1}^{K^2} \sum_{l=1}^{K^2} w_{kl} h_k h_l - \sum_{k=1}^{K^2} \theta_k h_k \tag{34}$$

where h_k are the multilevel parameter activities and w_{kl} and θ_k are the symmetric weights and bias inputs, respectively. From (33) and (34), we get the weights and bias inputs in the familiar outer-product forms:

$$w_{kl} = -\sum_{(i,j)\in S} z(i,j)_k \, z(i,j)_l \tag{35}$$

$$\theta_k = 2 \sum_{(i,j)\in S} z(i,j)_k \, y(i,j). \tag{36}$$

A greedy, distributed neural algorithm is used for the energy minimization. This leads to a localized multilevel number representation scheme for a general network.

B. Multilevel Greedy Distributed Algorithm

For a K^2 neuron second-order network, we choose Γ discrete activities $\{f_i, i = 0, 1, \cdots, \Gamma - 1\}$ in any arbitrary range of activities (e.g., $[0, 1]$) where we shall assume without loss of generality that $f_i > f_{i-1}$ for all i. Then, between any two activities f_m and f_n for the kth neuron, we can locally and asynchronously choose the one which results in the lowest energy given the current state of the other neurons because

$$E_{h_k = f_m} - E_{h_k = f_n} = \left[\theta_k - \zeta_k - (f_m + f_n) w_{k,k} \right] \cdot \left[f_m - f_n \right] \tag{37}$$

where

$$\zeta_k = \sum_{i, i \neq k}^{K^2} w_{i,k} h_i$$

is the current weighted sum from the other neuron activities. Thus, we choose level m over n for $m > n$ if

$$\zeta_k > \theta_k - (f_m + f_n) w_{k,k}. \tag{38}$$

Some properties of this algorithm follow.

1) Convergence is assured as long as the number of levels is not decreasing with time (i.e., assured if coarse to fine).

2) Self-feedback terms are included as level-dependent bias input terms.

3) The method can be easily extended to higher order networks (e.g., based on cubic energies). Appropriate lower order level-dependent networks (like the extra bias input term above) must then be implemented.

The multilevel lowest energy decision can be implemented by using variations of feedforward min-finding networks (such as those summarized in [20]). The space and time complexity of these networks are, in general, $O(\Gamma)$ and $O(\log \Gamma)$, respectively. However, in the quadratic case, it is easy to verify from (38) that we need only implement the decision between all *neighboring* levels in the set $\{f_i\}$; this requires exactly Γ neurons with level-dependent inputs. The best activity in the set is then proportional to the sum of the Γ neuron outputs so that the time complexity for the multilevel decision can be made $O(1)$. This means that this algorithm is similar in implementation complexity (e.g., the number of problem-dependent global interconnects required) to the simple sum energy representation used in [10] and in this paper. Also, in the simple sum case, visiting the neurons for each pixel in sequence will result in conditional energy minimization. Otherwise, from the implementation point of view, the two methods have some properties that are complementary. For example, we have the following.

1) The simple sum method requires asynchronism in the update steps for each pixel, while the greedy method does not.

2) The level-dependent terms arise as *inputs* in the greedy method as compared to *weights* in the simple sum method.

C. Simulation Results

The greedy algorithm was used with the weights from (35) and (36) to estimate the parameters from original and blurred sample points. A 5×5 window was used with two types of blurs: uniform and Gaussian. Both real and synthetic images were used, with and without additive Gaussian noise.

TABLE II
RESULTS FOR PARAMETER LEARNING. THE NUMBER Γ OF DISCRETE ACTIVITIES IS 256 FOR ALL TESTS. *A*: ARBITRARY CHOICE OF PIXELS FROM IMAGE. *L*: PIXELS CHOSEN FROM THRESHOLDED LAPLACIAN

Image	Noise	Blur	Samples	Methods	Iterations	MSE
Synthetic		Gaussian	68	*A*	49	0.000023
Synthetic		Uniform	100	*A*	114	0.000011
Real		Uniform	50	*A*	94	0.00353
Real		Uniform	100	*L*	85	0.00014
Real	20 dB	Uniform	100	*A*	72	0.00232
Real	20 dB	Uniform	100	*L*	83	0.00054

The estimated parameters for all types of blur matrices were numerically very close to the actual values when synthetic patterns were used. The network took longest to converge with a uniform blur function. The levels chosen for the discrete activity set $\{f_i\}$ were 128–256 equally spaced points in [0, 1] with 50–100 sample points from the image. Results for various cases are summarized in Table II.

When the sample pixels were randomly chosen, the errors increased by two orders of magnitude for a real image [Fig. 2(b)] as compared to synthetic ones. This is due to the smooth nature of real images. To solve this problem, sample points were chosen so as to lie close to *edges* in the image. This was done by thresholding the Laplacian of the image. Using sample points above a certain threshold for estimation improved the errors by an order of magnitude. The results were not appreciably degraded with 20 dB noise in the samples.

X. CONCLUSION

This paper has introduced a new approach for the restoration of gray level images degraded by a shift-invariant blur function and additive noise. The restoration procedure consists of two steps: parameter estimation and image reconstruction. In order to reduce computational complexity, a practical algorithm (Algorithm 2), which has equivalent results to the original one (Algorithm 1), is developed under the assumption that the neurons are sequentially visited. The image is generated iteratively by updating the neurons representing the image gray levels via a simple sum scheme. As no matrices are inverted, the serious problem of ringing due to the ill-conditioned blur matrix H and noise overriding caused by inverse filter or pseudoinverse inverse filter are avoided by using suboptimal boundary conditions. For the case of a 2-D uniform blur plus small noise, the neural network-based approach gives high-quality images compared to some of the existing methods. We see from the experimental results that the error defined by (12) is small, while the error between the original image and the restored image is relatively large. This is because the neural network decreases energy according to (12) only. Another reason is that when the blur matrix is singular or ill conditioned, the mapping from $\boldsymbol{X}$ to $\boldsymbol{Y}$ is not one to one; therefore, the error measure (12) is not reliable anymore. In our experiments, when the window size of a uniform blur function is 3×3, the ringing effect was eliminated by using blurred noisy boundary values without any smoothing constraint. When the window size is 5×5, the ringing effect was reduced with the help of the smoothing constraint and suboptimal boundary conditions. We have also shown that a smaller secondary network can effectively be used for estimating the blur parameters; this provides a more efficient learning technique than Boltzman machine learning on the primary network.

REFERENCES

[1] H. C. Andrews and B. R. Hunt, *Digital Image Restoration*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[2] J. W. Woods and V. K. Ingle, "Kalman filtering in two dimensions: Further results," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 188–197, Apr. 1981.

[3] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 1978.

[4] R. Chellappa and R. L. Kashyap, "Digital image restoration using spatial interaction models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 461–472, June 1982.

[5] H. Jinchi and R. Chellappa, "Restoration of blurred and noisy image using Gaussian Markov random field models," in *Proc. Conf. Inform. Sci. Syst.*, Princeton Univ., Princeton, NJ, 1986, pp. 34–39.

[6] N. H. Farhat, D. Psaltis, A. Prata, and E. Paek, "Optical implementation of the Hopfield model," *Appl. Opt.*, vol. 24, pp. 1469–1475, May 15, 1985.

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

[8] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, pp. 2554–2558, Apr. 1982.

[9] S.-I. Amari, "Learning patterns and pattern sequences by self-organizing nets of threshold elements," *IEEE Trans. Comput.*, vol. C-21, pp. 1197–1206, Nov. 1972.

[10] M. Takeda and J. W. Goodman, "Neural networks for computation: Number representations and programming complexity," *Appl. Opt.*, vol. 25, pp. 3033–3046, Sept. 1986.

[11] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature*, vol. 317, pp. 314–319, Sept. 1985.

[12] J. P. LaSalle, *The Stability and Control of Discrete Processes*. New York: Springer-Verlag, 1986.

[13] N. Metropolis *et al.*, "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1091, 1953.

[14] S. Kirkpatrick *et al.*, "Optimization by stimulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[15] W. K. Pratt *et al.*, "Visual discrimination of stochastic texture fields," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, pp. 796–814, Nov. 1978.

[16] J. W. Woods, J. Biemond, and A. M. Tekalp, "Boundary value problem in image restoration," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 692–695.

[17] M. M. Sondhi, "The removal of spatially invariant degradations," *Proc. IEEE*, vol. 60, pp. 842–853, July 1972.

[18] J. Biemond, J. Rieske, and J. Gerbrand, "A fast Kalman filter for images degraded by both blur and noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1248–1256, Oct. 1983.

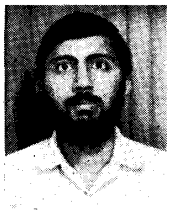
[19] R. Chellappa and H. Jinchi, "A nonrecursive filter for edge preserv-

ing image restoration," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 652–655.
[20] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, pp. 4–22, Apr. 1987.

Yi-Tong Zhou (S'84) received the B.S. degree in physics from the East China Normal University, Shanghai, China, and the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, in 1982 and 1983, respectively.

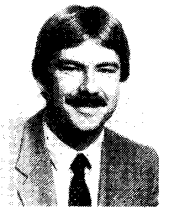
He is currently a Research Assistant at the Signal and Image Processing Institute, University of Southern California, Los Angeles, and is working toward the Ph.D. degree in electrical engineering. His research interests include image processing, computer vision, neural network algorithms, optical computing, and biomedical signal processing. He has published about a dozen technical papers in these areas.

Rama Chellappa (S'78–M'79–SM'83), for a photograph and biography, see this issue, p. 1066.



Aseem Vaid was born in Jammu, India, on January 8, 1963. He received the B.Tech. degree in electrical engineering in May 1985 from the Indian Institute of Technology, New Delhi.

Currently he is working on the Ph.D. degree at the University of Southern California, Los Angeles. His research interests are in neural networks and optical computing.



B. Keith Jenkins (M'85) received the B.S. degree in applied physics from the California Institute of Technology, Pasadena, in 1977, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1979 and 1984, respectively.

He was employed at Hughes Aircraft Company, El Segundo, CA, from 1977 to 1979 where he worked on holography for use in head-up displays. From 1984 to 1987 he was a Research Assistant Professor in the Department of Electrical Engineering, University of Southern California, where he presently is Assistant Professor of Electrical Engineering. He has also participated in advisory panels to government and industry, and has been a consultant to JPL, TRW, and Odetics. His research has included work in the areas of optical digital computing, neural networks and their optical implementation, learning algorithms, optical interconnection networks, parallel computation models and complexity, computer-generated holography, and optical 3-D position sensing systems.

Dr. Jenkins is a member of the Optical Society of America and the Association for Computing Machinery. He was awarded The Northrop Assistant Professor of Engineering at USC in 1987, and is a recipient of the 1988 NSF Presidential Young Investigator Award.