

# Image Restoration Using Chaotic Simulated Annealing

*Leipo Yan and Lipo Wang*

School of Electrical and Electronic Engineering  
Nanyang Technological University  
Block S1, Nanyang Avenue, Singapore 639798  
E-mail: elpwang@ntu.edu.sg

**Abstract**—Both the stochastic chaotic simulated annealing and the deterministic chaotic simulated annealing are used to restore gray level images degraded by a known shift-invariant blur function and additive noise. The neural networks are modeled to represent the image whose gray level function is the simple sum of the neuron state variables. The restoration consists of two stages: parameter estimation and image reconstruction. During the first stage, parameters are estimated by comparing the energy function of the neural network to a constraint error function. The neural networks are then updated. Experiments show that noisy chaotic neural network could get good results in relatively shorter time compared to Hopfield neural network and better results compared to transiently chaotic neural network.

**Index Terms**—Image restoration, neural network, simulated annealing.

## I. INTRODUCTION

The increasing use of imaging systems in scientific, industrial, medical and commercial environments has made image processing an important field in signal processing. As most imaging systems are imperfect and tend to produce noisy blurred images, image restoration becomes more and more important. Image restoration seeks to recover the true image using objective criteria and some prior knowledge about the original image. 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.

Blurring is present in any imaging system that uses electromagnetic radiation (for example, visible light and X-rays). Diffraction limits the resolution of an imaging device to features on the order of the illuminating wavelength. Scattering of light between the target object and imaging system (for example, by the atmosphere) introduces additional blurring. Lenses and mirrors cause blurring because they have limited spatial extent and optical imperfections. Discretization results in yet more blurring because devices such as CCDs average illumination over regions rather than sampling it at discrete points.

Various methods have been proposed for image restoration, such as inverse filter, Wiener filter [1], Kalman filter [2]. 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 stationary (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. 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 in general. Zhou et al [3] proposed a method using a neural network model containing redundant neurons to restore gray level images degraded by a known shift-invariant blur function and noise.

Simulated annealing is a general method for obtaining approximate solutions of combinatorial optimization problems. Since the optimization processing is undertaken in a stochastic manner, it is also called stochastic simulated annealing (SSA). In recent years, a large amount of work has been done in chaotic simulated annealing (CSA). CSA can search efficiently because of its reduced search spaces. Chen and Aihara [4] proposed a transiently chaotic neural networks (TCNN) which adds a large negative self-coupling with slow damping in the Euler approximation of the continuous Hopfield neural network so that neurodynamics eventually converge from strange attractors to an equilibrium point. The TCNN showed good performance in solving traveling salesman problem (TSP). Wang and Tian [5] generalized CSA by adding a decreasing random noise into the neural inputs in the TCNN to combine the best features of both SSA and CSA, thereby obtaining stochastic chaotic simulated annealing (SCSA).

A measurement of image quality is required to compare restoration results. One commonly used measurement is peak

signal-to-noise ration. Another image quality measurement  $m1$  is extracted from video quality measurements based on human perception [6]. The equation for  $m1$  is given by

$$m1 = \left| \frac{SI[g'] - SI[g]}{SI[g']} \right| \quad (1)$$

where  $SI$  is the spatial information defined by

$$SI[F] = \text{STD}_{\text{space}} \{ \text{Sobel}[F] \} \quad (2)$$

$\text{STD}_{\text{space}}$  is the standard deviation operator over the horizontal and vertical spatial dimensions in a frame, and Sobel is the sobel filter operation. We will use these two measurements to qualify our experimental results.

This paper is organized as follows. Section II reviews neural network representation for an image. In Section III the NCNN is reviewed. The restoration algorithm is presented in Section IV. Our experimental results appear in Section V, followed by concluding remarks.

## II. NEURAL NETWORK REPRESENTATION FOR IMAGE

We use the neural network proposed by Zhou et al [3], which contains redundant neurons to represent the image. 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  $V = \{v_{i,k} \text{ where } 1 \leq i \leq L^2, 1 \leq k \leq M\}$  be a state set of the neural network with  $v_{i,k}$  denoting the state of the  $(i, k)$ th neuron.  $U = \{u_{i,k} \text{ where } 1 \leq i \leq L^2, 1 \leq k \leq M\}$  denotes the internal state of neuron  $\{i, k\}$ . 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:

$$T_{i,k,j,l} = T_{j,l,i,k} \text{ for } 1 \leq i, j \leq L^2 \text{ and } 1 \leq l, k \leq M. \quad (3)$$

We also allow for neurons to have self-feedback, i.e.,  $T_{i,k,i,k} \neq 0$ . In this model, each neuron  $(i, k)$  is randomly and asynchronously updated. The image is described by a finite set of gray level functions  $\{x(i, j) \text{ where } 1 \leq i, j \leq L\}$  with  $x(i, j)$  denoting the gray level of the pixel  $(i, j)$ . The image gray level function can be represented by a simple sum of the neural state variables as

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

where  $m = (i-1)L + j$ . By using the lexicographic notation, the image degradation model can be written as  $Y = HX + N$ , where  $H$  is the blur matrix corresponding to a blur function,  $N$  is the signal independent white noise, and  $X$  and  $Y$  are the original and degraded images, respectively.

The shift-invariant blur function can be written as a convolution over a small window. For instance, it takes the form

$$h(k, l) = \begin{cases} \frac{1}{2} & \text{if } k = 0, l = 0 \\ \frac{1}{16} & \text{if } |k|, |l| \leq 1, (k, l) \neq (0, 0) \end{cases} \quad (5)$$

The blur matrix  $H$  will be a block Toeplitz or block circulant matrix. The block circulant matrix corresponding to above  $h$  can be written as:

$$H = \begin{bmatrix} H_0 & H_1 & 0 & \cdots & 0 & H_1 \\ H_1 & H_0 & H_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ H_1 & 0 & 0 & \cdots & H_1 & H_0 \end{bmatrix} \quad (6)$$

where

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

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

## III. NOISY CHAOTIC NEURAL NETWORK

The noisy chaotic neural network is defined as follows [5]:

$$v_{ik}(t) = \frac{1}{1 + e^{-u_{ik}(t)/\varepsilon}} \quad (7)$$

$$u_{ik}(t+1) = ku_{ik}(t) + \alpha \left( \sum_{j=1}^{L^2} \sum_{l=1}^M T_{ikjl} v_{jl} + I_{ik} \right) - z(t)(v_{ik}(t) - I_o) + n(t) \quad (8)$$

$$z(t+1) = (1-\beta) z(t) \quad (9)$$

$$A[n(t+1)] = (1-\beta) A[n(t)] \quad (10)$$

where

$v_{ik}$ : output of neuron  $(i, k)$ ;

$u_{ik}$ : internal state of neuron  $(i, k)$ ;

$$T_{ikjl} = T_{jl ik}; \quad \sum_{j=1, j \neq i}^{L^2} \sum_{l=1, l \neq k}^M T_{ikjl} v_{jl} + I_{ik} = -\partial E / \partial x_{ik} \quad :$$

connection weight from neuron  $(j, l)$  to neuron  $(i, k)$ ;

$I_{ik}$ : input bias of neuron  $i, k$ ;

$k$ : damping factor of nerve membrane ( $0 \leq k \leq 1$ );

$\alpha$ : positive scaling parameter for inputs;

$z(t)$ : self-feedback connection weight or refractory strength ( $z(t) \geq 0$ );

$I_o$ : positive parameter;

$\varepsilon$ : steepness parameter of the output function ( $\varepsilon > 0$ );

$n(t)$ : random noise injected into the neurons;

$A[n]$ : the noise amplitude. If  $n(t) = 0$ , for all  $t$ , then the noisy chaotic neural network reduces to the transiently chaotic neural network.

Note when we set  $n(t)=0$  in (8), the noisy chaotic neural network becomes transiently chaotic neural network. As the “temperature”  $z(t)$  tends toward zero with time evolution,

equations (7)-(9) eventually stabilize. Noisy chaotic neural networks have some added in noise compared to transiently chaotic neural network. With the noise NCNN continues to search for optimal solution after the disappearance of chaos.

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

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

As image degradation model can be written as  $Y=HX+N$ , image restoration seeks to recover the true image  $X$  from the degraded image  $Y$ . Let  $\hat{X}$  be the restored image, then image restoration tries to approximate  $\hat{X}$  to  $X$ . In order to use the spontaneous energy-minimization process of the neural network, Zhou et al [3] reformulated the restoration process as one of the minimizing an error function with constraints defined as

$$E = \frac{1}{2} \|Y - H\hat{X}\|^2 + \frac{1}{2} \lambda \|D\hat{X}\|^2 \quad (12)$$

where  $\|Z\|$  is the  $L_2$  norm of  $Z$  and  $\lambda$  is a constant. Such a constrained error function is widely used in the image restoration problems. In general  $D$  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,  $D$  may be a Laplacian operator.

The first term in (12) is to seek an  $\hat{X}$  such that  $H\hat{X}$  approximates  $Y$  in a least squares sense. Meanwhile, the second term is a smoothness constraint on the solution  $\hat{X}$ . The constant  $\lambda$  determines their relative importance.

$\lambda=\infty$ :  $\hat{X}$  is smooth, but we ignore the data;

$\lambda=0$ :  $\hat{X}$  approximate  $X$ , but  $\hat{X}$  may not be smooth.

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

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

By comparing the terms above to the corresponding terms

in (11) and ignoring the constant term  $\frac{1}{2} \sum_{p=1}^{L^2} y_p^2$ , we can

determine the interconnection strengths and bias inputs as

$$T_{i,k;j,l} = -\sum_{p=1}^{L^2} h_{p,i} h_{p,j} - \lambda \sum_{p=1}^{L^2} d_{p,i} d_{p,j} \quad (13)$$

$$I_{i,k} = \sum_{p=1}^{L^2} y_p h_{p,i} \quad (14)$$

where  $h_{ij}$  and  $d_{ij}$  are the elements of the matrices  $H$  and  $D$  respectively. Two interesting aspects of (13) and (14) 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  $T_{i,k;i,l}$  is not equal to zero which requires self-feedback for neurons.

For an  $L \times L$  image with  $M$  gray levels,  $L^2 M$  neurons and  $1/2 L^4 M^2$  interconnections are required to represent the image. The space complexity is  $O(L^4 M^2)$ , which is difficult to represent on a conventional computer. However simplification is possible if the neurons are sequentially updated. As noted earlier, the interconnection strengths given in (13) are independent of subscripts  $k$  and  $l$  and the bias inputs given in (14) 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, the dimensions of the interconnection matrix  $T$  and bias input matrix  $I$  can be reduced [3].

#### IV. RESTORATIONS

Restoration is carried out by neural evaluation and image construction procedure. Once the parameters  $T_{i,k;j,l}$  and  $I_{i,k}$  are obtained using (13) and (14), each neuron can randomly and asynchronously evaluate and update its state using (8) and (7) accordingly. When one quasi-minimum energy point is reached, the image can be constructed using (4).

The energy of the neural network does not decrease monotonically. To successfully restore the image the neural network has to find an optimal solution, which corresponding to the minimum energy of the network.

Define the state change  $\Delta v_{i,k}$  of the neuron  $(i,k)$  and energy change  $\Delta E$  as

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

Consider the energy function in (11), the change  $\Delta E$  due to a change  $\Delta v_{i,k}$  is given by

$$\Delta E = - \left( \sum_{j=1}^{L^2} \sum_{l=1}^M T_{i,k;j,l} v_{j,l} + I_{i,k} \right) \Delta v_{i,k} - \frac{1}{2} T_{i,k;i,k} (\Delta v_{i,k})^2 \quad (15)$$

At any one time only one neuron is updating. The new state  $v_{i,k}^{new}$  of neuron  $(i,k)$  is calculated according to (7)-(10), and the energy change  $\Delta E$  is calculated according to (15). If

$\Delta E \leq 0$ , then new state of neuron (i,k)  $v_{i,k}^{new}$  is accepted, and the network with new neural state is used as the starting point of the next step. Then case  $\Delta E \geq 0$  is treated probabilistically: then probability that the new neural state  $v_{i,k}^{new}$  is accepted is  $P(\Delta E) = \exp(-\Delta E / k_B T)$ . Random numbers uniformly distributed in the interval (0,1) are a convenient means of implementing then random part of the algorithm. One such number is selected and compared with  $P(\Delta E)$ . If it is less than  $P(\Delta E)$ , then new state is retained; if not, the original state is used.

The algorithm is summarized as below:

- 1) Set the initial state of the neurons.
- 2) Sequentially visit all numbers (image pixels). For each number, use (7)-(10) to update it repeatedly until there is no further change.
- 3) Check the energy function; if the energy variation is less than a limit, go to step 4); otherwise go back to step 2).
- 4) Construct the image using (4).

## V. EXPERIMENTAL RESULTS

Experimental results presented here demonstrate the performance of noisy chaotic neural network on image restoration. Performances of Hopfield neural network and transiently chaotic neural network are also shown.

We used Lena image of size  $128 \times 128$  with 256 gray levels. The image was degraded by a  $5 \times 5$  blur function:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 8 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} / 32$$

The image was further degraded by adding a small amount of quantization noise. The high pass filter we use to impose the smoothing constraint is

$$D = \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} / 6$$

The neural network starts from three different initial states. And for each initial state the noisy chaotic neural network was simulated three times as there were random noise added in it. The other two types of neural networks were simulated once for every initial state.

The simulation results are shown in Figure 1 and Table 1. The PSNR and m1 of the degraded image are 24.7036 and 0.2882, respectively.

All the three neural networks improve the degraded image. The restoration results starting from the three random initial states are consistent. HNN takes more than 100 iterations to converge to a stable point, TCNN takes around 20 iterations

to stabilize, and NCNN takes around 70 iterations to become stable. TCNN can improve the PSNR from 25 dB to 37 dB, HNN to 41 dB and NCNN to around 42 dB. The other measurement m1 shows consistent result with PSNR.

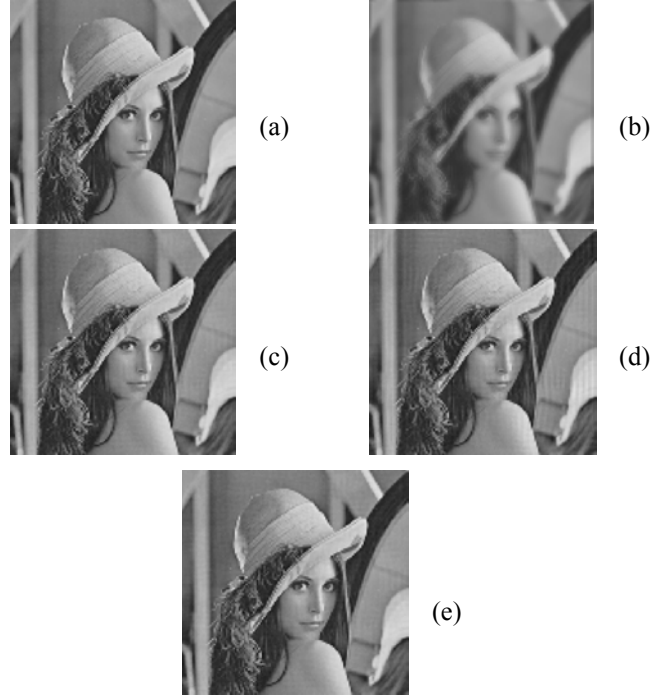


Fig. 1. Restoration of noisy blurred real image. (a) Original image. (b) Blurred image. (c)-(d) Restored image using Hopfield neural network, transiently chaotic neural network and noisy chaotic neural network, respectively.

Table 1. Restoration results in terms of PSNR and m1

## VI. CONCLUSIONS

In this report three types of neural networks have been implemented for image restoration, namely Hopfield neural network, transiently chaotic neural network and noisy chaotic

neural network. The restoration procedure consists of two steps: parameter estimation and image reconstruction. The image is generated iteratively by updating the neurons representing the image gray levels via a simple sum scheme. The networks were updated in stochastic manners. From the simulation results we see that Hopfield neural network could get good restoration result with long restoration time. Transiently chaotic neural network could get relatively poor restoration result in very short restoration time. Noisy chaotic neural network could get very good results in relatively shorter time.

## 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] Y.T. Zhou, R. Chellappa, and B.K. Jenkins, "Image Restoration Using a Neural Network", IEEE Trans. Acoust., Speech, Sign. Proc., vol. 36, pp. 38-54, 1988
- [4] L. Chen and K. Aihara, "Chaotic simulated annealing by a neuralnetwork model with transient chaos," Neural Networks, vol. 8, no. 6, pp. 915--930, 1995
- [5] L. Wang and F. Tian, "Noisy chaotic neural networks for solving combinational optimization problems", Proc. International Joint Conference on Neural Networks, IJCNN2000
- [6] A. A. Webster, Coleen T. Jones, Margaret H. Pinson, Stephen D. Voran and Stephen Wolf, "An objective video quality assessment system based on human perception", SPIE 1913, Human Vision, Visual Processing and Digital Display, pp. 15-26, 1993
- [7] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," Biol. Cybern., vol. 52, pp. 141-152, 1985

		Iterations	PSNR	m1
Random initial state1	HNN	107	41.4912	0.0132
	TCNN	24	37.1758	0.0309
	NCNN	34	40.672	0.0153
	NCNN	70	41.8275	0.0101
	NCNN	74	41.9423	0.0093
Random initial state2	HNN	106	41.483	0.0132
	TCNN	21	37.1433	0.0313
	NCNN	34	40.6966	0.0152
	NCNN	60	41.8202	0.0103
	NCNN	67	41.9289	0.0095
Random initial state3	HNN	106	41.4678	0.0133
	TCNN	26	37.2336	0.0309
	NCNN	69	41.8238	0.0102
	NCNN	78	41.8848	0.0096
	NCNN	71	41.9558	0.0092