Three New Soft Computing Approaches to Two-Dimensional CT Image Reconstruction : A Comparative Study

Three New Soft Computing Approaches to Two-Dimensional CT Image Reconstruction: A Comparative Study†

Fath El Alem F. Ali*, Zensho Nakao**, & Yen-Wei Chen**

E-mail: {ali,nakao,chen}@augusta.eee.u-ryukyu.ac.jp

Keywords: CT image, Soft computing.

Abstract Recently, we developed three soft computing techniques for reconstructing two-dimensional CT images from a small number of projection data. They are genetic algorithms, simulated annealing, and back propagation reconstruction techniques. The three techniques have been developed independently of each other. In this paper we present a comparative evaluation study of these techniques. We start by introducing three new approaches one by one, and then present simulation result. Reconstruction result by a well known conventional method -*Algebraic Reconstruction Technique (ART)* - is also presented for sake of comparison. A quantitative evaluation among the four reconstruction methods is presented. A *pixel-wise error estimator* is used to calculate the overall error in the reconstructed images. The estimator reveals the effectiveness of the proposed techniques compared to the conventional ART.

1. Introduction

The problem of digitally reconstructing an image from projections has become important during the last few decades. There are many fields where the applications of the problem exist. One important field is computed tomography (CT) in the medical field. The CT image is a digitized view of the cross-section of the body. The image is a matrix of pixels. Each pixel in this image has a number, often called the *CT number* or *Hounsfield number*, determined by its x-ray attenuation. As demonstrated in Figure 1, a CT image production begins with a *scanning phase*, where the data with which the image is eventually reconstructed is obtained by measuring the transmission of multiple x-ray beams projected through the image plane. In order to obtain enough data for a full two-dimensional image, it is necessary to project x-rays through the body from many directions.

Received on: Dec 1, 1998

*Doctoral student, Depart of EEE, Univ of the Ryukyus

**Depart of EEE, Univ of the Ryukyus

† Part of the paper was presented at KES'98 (Adelaide, Australia), Apr 21 ~ 23, 1998.

The second phase of image production is known as *image reconstruction.* This is performed by the digital computer which is part of the CT system. Image reconstruction is usually a mathematical procedure that combines projections and converts them into a computed digital image. At display time, the numerical values of the individual pixels are converted to a gray scale image. Algebraic reconstruction techniques (ART)[1, 2], are well known techniques in the field. ART are iterative methods that perform step-wise updating operations on an initially uniformly-distributed estimation of an image. ART methods have merits over many other known reconstruction techniques due to the possibility of providing a reasonable solution even in case of a limited number of angular projections. With further limitation of projection directions, however, ART approximation is not always satisfactory[2, 6].

There are some other methods for reconstruction, e.g., *filtered-back projection method* [1], but most of those methods require a large number of projection directions which sometimes may not be practically possible, as with some cases of laser fusion applications[7].

Here, we are presenting three soft computing approaches to the reconstruction problem in the particular case of a limited number of projection data (*four* directions): simulated annealing (SA), back propagation (BP), and genetic algorithms (GA) reconstruction techniques. Comparative evaluation of the three new techniques as well as of the conventional ART, are given, followed by comments and future work.

2. Reconstruction by simulated annealing

The simulated annealing reconstruction system follows the structure and behavior of the Boltzmann machine[3, 4]. A grid of neuro-nodes comprises the structure of the Boltzmann machine.

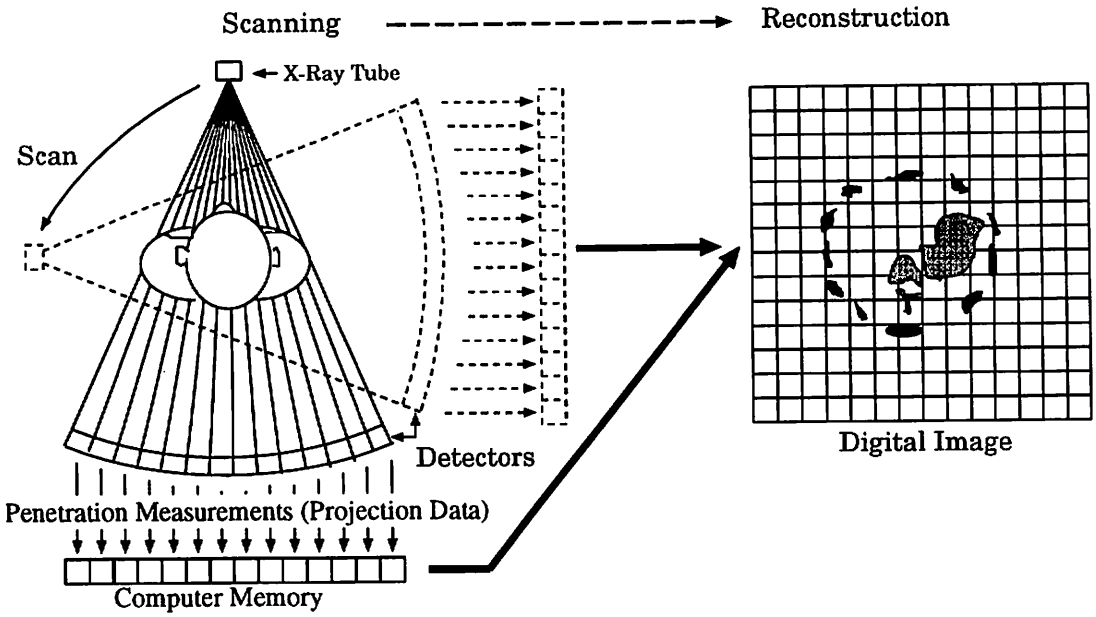


Figure 1: Scanning and reconstruction phases of CT image production

A mathematical representation of behavior of the Boltzmann machine is given below.

Network *energy* is a value attached to the current potential state of the network, and calculated using a quality criterion of some objective function. For each node of the network an *energy gap* (ΔE_i) is computed by the formula:

$$\Delta E_i = E_\alpha - E_\beta$$

where

E_α: energy of state α
E_β: energy of state β

and the network changes the output to decrease the energy with the following probability (Pi):

$$P_i = \frac{1}{1 + exp(-\Delta E_i/T)}$$

T : *temperature parameter* [1]

In the simulated annealing reconstruction system, the network consists of a grid of neuro-nodes (Figure 2). Each node maps a pixel in the plane of the image to be reconstructed, and consequently the number of the nodes correspondes to the size of the image, and furthermore the output of each node represents a pixel value.

2.1 Updating a node output

The network updates output of all nodes in a way to minimize the overall energy of the network. Based on the previous and the present condition (*energy E*) of the network, an energy gap (ΔE) is calculated.

[1]The temperature T is decreased gradually.

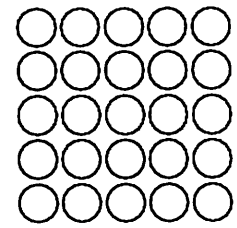


Figure 2: Network structure: a grid of nodes

$$\Delta E_t = E_{t-1} - E_t$$

An objective energy function is used to estimate the current network energy (E_t). The energy function is composed of two terms: the absolute error (δ) between projection data of the image reconstructed in the system and original projection data, and a penalty term (*Penal*):

$$E_t(energy\ \ at\ \ time\ \ t) = \delta_t + \lambda \times Penal$$

$$\delta_t = \sum_{\theta=1}^{c} \sum_{r}^{width} |P(\theta, r) - R(\theta, r)|$$

$$Penal = |f^m - f(i,j)|/S_s$$

For δ_t,

C: number of projection directions
θ : index of projection angle
$P(\theta, r)$: projection of original image
$R(\theta, r)$: projection of reconstructed image
width: width of the image
r: index of projection subdata (Figure 4)

and for *Penal*,

f(i,j): value of pixel #(i,j)
λ: scaling factor
S_s: step size for changing pixel gray level
f^m : a median pixel value in the neighborhood of pixel #(i,j) (Figure 3)

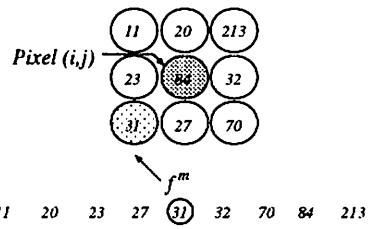


Figure 3: Neighborhood of pixel *(i,j)* and *median* pixel desgination

The penalty term, *Penal*, penalizes non-smoothness; thus, as the estimated image becomes smoother, *Penal* becomes smaller. Revising the output with probability P_i, the system behaves in such a way as to minimize the overall energy (or *error*) of the system.

2.2 Mechanics of the simulated annealing system

The system begins with reconstruction of a rough image: starting with an initial binary image $[0, 255]$, the system generates a grid of nodes randomly (Figure 4).

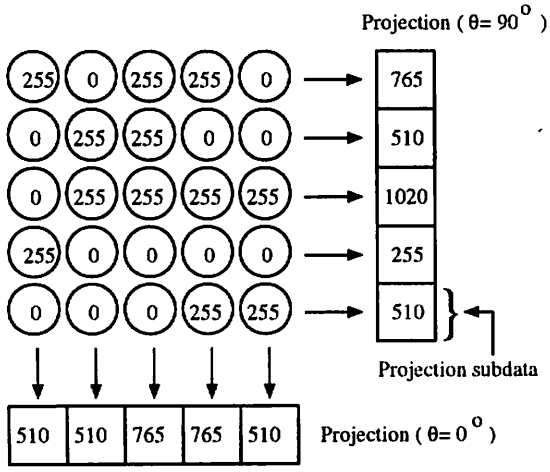


Figure 4: A binary grid and projections[2]

Next, the system uses a *step_size measure* (S_s) to update image gray level. S_s is defined as follows:

$$S_s = 2^N - 1, \qquad where \;\; N = 7, 6, .., 1$$

N is decremented when the energy of the system is trapped in what we called a *critical condition*. The *critical condition* is detected when the following inequality holds.

$$\frac{ABS(E_{t-1} - E_t)}{E_t} \quad < \quad 0.0001$$

The system updates the gray level of pixels (f), using *step-size measure* (S_s):

$$f_i = f_i \pm S_s$$

Applying the above conditioned-degrading scheme for the *step_size measure*,and the propablistic updating of pixels gray level, the initial binary image turns gradually into a fine one with gray levels spreading over the interval $[0 \sim 255]$.

The mechanics of the simulated annealing system are shown in the algorithm below.

Begin
  Set $N = 7$, $\mathcal{I}$ = *total number of nodes*,
  $\mathcal{T} = \mathcal{T}_\infty$, $t = 0$, $E_t = \infty$
  Generate an initial binary grid of nodes
  Calculate δ_t
  While $\delta_t \neq 0$ **And** $N \neq 0$ **Do**
    $i = 0$
    $t = t + 1$
    While $i \leq \mathcal{I}$ **Do**
      Calculate energy gap ΔE_t related to change of node i
      Caculate P_i for node i
      Change output of node i with P_i
      $i = i + 1$
    End
    Update projections of the grid
    Calculate δ_t
    Decrease temperature $\mathcal{T}$
    If *stationary condition* **Or** $\mathcal{T} < \mathcal{T}_0$ **Then**
      $N = N - 1$
      $S_s = 2^N - 1$
      $\mathcal{T} = \mathcal{T}_\infty$
    End
  End
End

---

3. Reconstruction by back propagation

The back-propagation algorithm[4,5] is probably the most well-known and widely used option among the several available types of neural network systems nowadays. The back-propagation algorithm uses a gradient search technique to minimize a cost function which is the mean square

[2] Only two directions are shown here for simplicity.

difference between the desired and the actual network outputs.

We developed a back-propagation network for the CT image reconstruction problem. To equip the system with the back-propagation algorithm, we need to determine a proper structure of the network as well as an adapting rule used to adapt connection weights.

3.1 Structure of the network

The proposed back-propagtion network is a linear network. It, however, consists of three layers, as depicted in Figure 5. The input layer and the output layer correspond to the projections of the original and reconstructed image, respectively, and the number of neuro-nodes in these layers is the same as the size of the projection data; the hidden layer represents the reconstructed image itself, and in its turn, the number of nodes in this layer corresponds to the number of pixels in the plane of the image to be reconstructed.

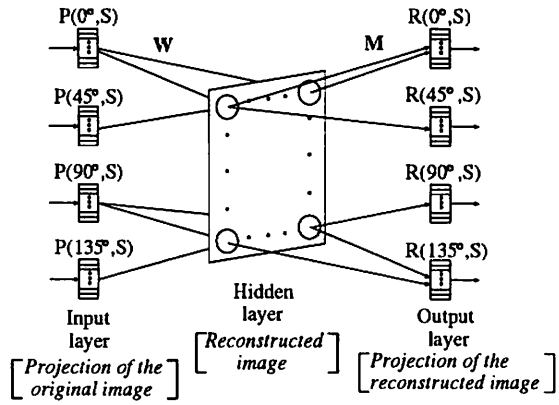


Figure 5: Structure of the network

3.2 Connection weights

There are two sets of connection weights: weights to the hidden layer (W), and weights to the output layer (M). Only connection weights to the hidden layer are adjusted through the adaptation rule. On the other hand, weights to the output layer due to functionality are kept *constant* [3]. Those constant weights are used to calculate the projection data of the image being reconsructed in the hidden layer. The set of weights to be modified (W) is initialized to small values [0.0, 1.0], while the hidden-to-output layer connection weights (M) remain *constant*. The latter (M) are determined by the area ratio of pixels to projection nodes when generating projection data, using the usual rotation formula:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

where (x', y') are the coordinates of pixel $P(x, y)$ after rotation by angle θ.

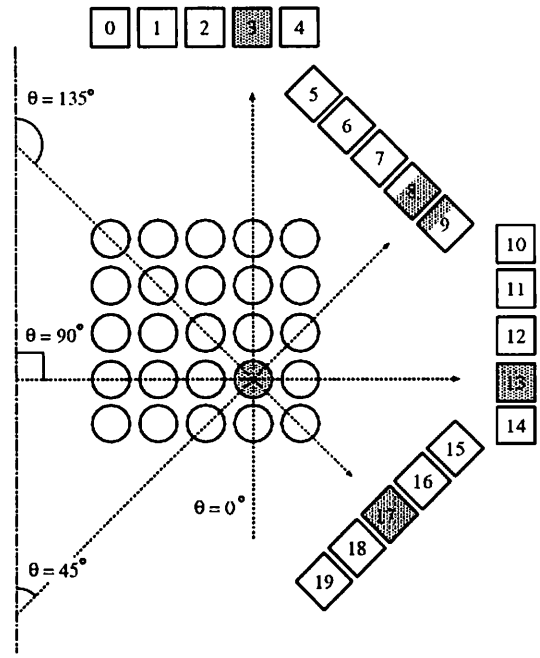


Figure 6: Weight initialization

An example of the weights initialization is shown in Figure 6. As the projection data are drawn, the pixel marked will be treated as follows:

- At angle 0°, the value of the pixel goes to projection node 3.
- At angle 45°, the value of the pixel is divided between node 8, and 9, in a 7:3 ratio[4].
- At angle 90°, the value of the pixel goes to projection node 13.
- At angle 135°, the value of the pixel goes to projection node 17.

and accordingly,

- The weights of the corresponding node in the middle layer to nodes 3, 8, 9, 13, 17 of the output layer would be, respectively, 1, 0.7, 0.3, 1, 1; and others are set to be zero.

3.3 Learning rule

[3] That is the reason we use the *delta rule* and not the *generalized delta rule*.

[4] This ratio can be precisely calculated using the rotation formula.

The back propagation network adopts the well known delta rule for adaptation of weights. The network learns to minimize the error between the projections of the original unknown image, given as the teaching signal, and of the reconstructed image, or the output from the output layer. A sigmoid transfer function of the form

$$f(\alpha) = \frac{1}{1 + exp(-(\alpha - \theta))}$$

is used to calculate the activation of each node in the hidden layer.

Suppose that y_i is the output of the i-th node in the input and w_{ji} is the connection weight between nodes i and j, then the output y_j of the j-th node in the hidden layer is given by

$$y_j = \frac{1}{1 + exp(-x_j)}, \quad x_j = \sum_i y_i w_{ji} - \theta_j$$

where θ_j and x_j are the threshold and the input of the node j in the hidden layer, respectively.

Now, let m_{lk} be the connection weight between k-th node in the hidden layer and the l-th node in the output layer. The output y_l of the l-th node in the output layer is

$$y_l = x_l, \quad x_l = \sum_k y_k m_{lk}$$

the corresponding error function E is given by

$$E = \frac{1}{2} \sum_l (y_l - d_l)^2$$

and

$$\frac{\partial E}{\partial w_{ji}} = \sum_k \left[\sum_l (y_l - d_l) m_{lk} \right] y_k (1 - y_k) m_{kj} y_j (1 - y_j) y_i$$

$$\frac{\partial E}{\partial w_{kj}} = \left[\sum_l (y_l - d_l) m_{lk} \right] y_k (1 - y_k) y_j$$

where d_l is the teaching signal, and

$$\Delta w_n = -\alpha \frac{\partial E_n}{\partial w_n} + \beta \Delta w_{n-1}$$

where α and β are constants. Then

$$w_{n+1} = w_n + \Delta w_n$$

where w_{n+1} and w_n, respectively, represent the *new* and the *old* values of the connection weight.

3.4 Mechanics of the back-propagation system

The steps below show the mechanics of the back-propagation system.

- **Step1. Initialize connection weights**
  Initialize the input-to-hidden weights to small random values. Determine the values for the set of the constant hidden-to-output weights. Set all threshold values to zero.
- **Step2. Present input and desired output**
  Present the given projection data as a sequence of input vector $a_0, a_1, ..., a_{n-1}$. The desired output vector $d_0, d_1, ..., d_{n-1}$, is a vector equal to the input vector.
- **Step3. Calculate activation for hidden layer neurons**
  Using the sigmoid transfer function, calculate the activation of each neuron in the hidden layer
- **Step4. Calculate actual output**
  Calculate the actual output vector b_0, b_1, ..., b_{n-1}.
- **Step5. Adjust input-to-hidden connection weights**
  Adjust input-to-hidden weights using the delta rule.
- **Step6. Repeat by going to step 2, for desired number of iterations, or till error (E) becomes less than a specified tolerance.**

4. Reconstruction by genetic algorithms

To encode an image in a genetic chromosome, we represent each pixel in the plane of the image by one allele in a genetic chromosome. A *float-bit* representation [0.0, 1.0] for a string allele values is made (Figure 7).

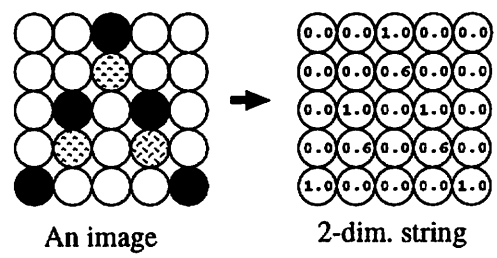


Figure 7: Image encoding

4.1 Fitness measure

A formula that compares the original image projection data to those of the reconstructed image

is used to return a chromosome fitness. A Laplacian constraint term, that penalizes non-smooth images, is included in the fitness function(F) which comes below.

$$F = \frac{1}{1 + \lambda \times (E + \kappa \times \| Lap \|^2)}$$

where

$$Lap = \nabla^2 f(i,j) = [f(i+1,j) + f(i-1,j) + f(i,j+1) + f(i,j-1) - 4 \times f(i,j)]$$

$$E = \frac{1}{c} \sum_{\theta=1}^{c} \frac{1}{width} \sum_{r}^{width} [P(\theta,r) - R(\theta,r)]^2$$

and

C: number of projection directions
θ : index of projection angle
$P(\theta,r)$: projection of original image
$R(\theta,r)$: projection of reconstructed image
width: width of the image
r: index of projection subdata (Figure 4)
f(i,j): value of pixel(i,j)
λ: an integer scaling factor
κ: a real scaling factor

In the fitness function F, E measures the mean square error between the given projection and the calculated projection of the reconstructed image, and *Lap* is the Laplacian operator.

4.2 Genetic operations

Chromosomes of an initail population are generated *randomly*: for each allele, its value is selected randomly from the unit interval [0.0, 1.0].

In the reproduction operation of a new generation, an *elitist* selection scheme is used for selection of parents, where strings with high fitness are exempted undertaking genetic operations and given chance to survive to the next generation.

A *uniform* crossover operation is applied to parents to produce offspring for the next generation. The uniform crossover generates *an offspring* from two randomly-selected parents (Figure 8).

Following the crossover, mutation operations take place. Two mutation operators are applied: *one-step uniform* mutation and a *median* mutation. The first mutation is defined as follows: if $M_v = < \cdots, v_ij, \cdots >$ is an arbitrary chromosome, then the mutative process takes place on each element v_{ij} with an equal chance. The gray level of the allele selected for mutation, will take the nearest upper/lower gray level value, ie., the new allele value becomes one step smaller or larger than the old value in its gray level.

The second mutation (median mutation) changes the pixel value (v_t) to the mean value (v^m), among the neighborhood pixels (Figure 9).

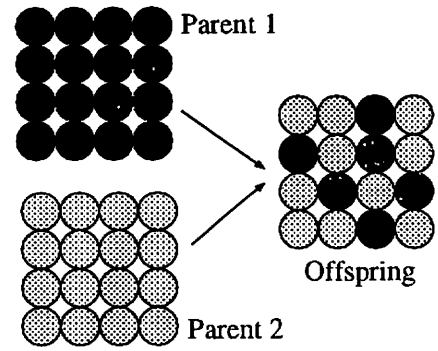


Figure 8: Uniform crossover

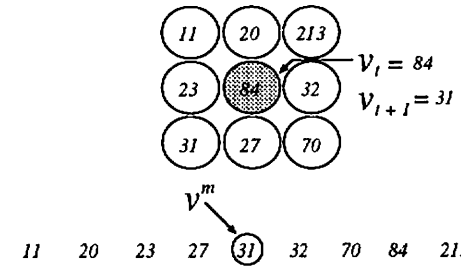


Figure 9: Median mutation

The above genetic operations are applied to each generation of population of chromosomes till a satisfactory chromosome is generated or a specified number of generations is reached. The reconstruction system starts with a small size chromosomes, and then increases the size exponentially (when certain conditions are satisfied) over generations till the actual size of the image is reached. The expansion takes place at occasions when the fitness of the best chromosome remains the same for a certain number of successive generations.

The whole process of the genetic algorithm is shown in the following algorithm.

Begin
  $t = 0$
  chromosome_size $= 8 \times 8$
  Initialize *P(t)*
  Evaluate *P(t)*
  While (**Not** *termination_condition*) **Do**
    Select $\dot{P}(t)$ from *P(t)*
    Select $\ddot{P}(t)$ from *P(t)*
    Crossover $\ddot{P}(t)$
    Mutate $\ddot{P}(t)$
    Evaluate $\ddot{P}(t)$
    $P(t+1) = \dot{P}(t) \cup \ddot{P}(t)$
    $t = t + 1$
    If *expand_condition* **Then**
      Expand *chromosome_size*

End
End
End

In this algorithm, *P(t)* denotes a population of μ individuals at generation t, $\dot{P}(t)$ is a special set of λ ($\lambda < \mu$) elite individuals, and $\ddot{P}(t)$ is a population of $(\mu-\lambda)$ randomly selected individuals from among *P(t)* .

5. Digital simulation[5]

Tables 1, 2, 3 below show parameters and simulation condition for the proposed algorithms.

Table 1. Image & projection specifications	
Angles of projections	0°,45°, 90°, 135°
Image size	32 x 32

Table 2. Simulated Annealing (SA) System	
Initial temperature T_∞	1.0 degree
Minimum temperature T_o	0.005 degree
Factor for decreasing T	0.999
Penal scaling factor λ	16.0

Table 3. Back Propagation (BP) System	
Gain factor α	0.4
Scaling factor β	0.05

Table 4. The Genetic Algorithm (GA) System	
Population size	100
Elite selection rate	0.2
Uniform mutation rate	0.01
median mutation rate	0.01
Scaling factor (λ), in the fitness function	10
Penality term scaling factor (κ)	10

As a base for a quantitative comparison among ART, SA, BP, & GA reconstruction techniques, a pixel-wise error estimator (δ_e) is used to estimate an over all error in the reconstructed image. This is based on the fact that the original image is known in the experiments.

$$\delta_e = \sqrt{\frac{\sum_{x=1}^{Width}\sum_{y=1}^{Width}(f_{org}(x,y) - f_{rec}(x,y))^2}{\sum_{x=1}^{Width}\sum_{y=1}^{Width} f_{org}(x,y)^2}} \times 100.0$$

where

$f_{org}(x,y)$: value of pixel(x,y) in the original image

$f_{rec}(x,y)$: value of pixel(x,y) in the reconstructed image

Figure 10 shows experimental results for the three algorithms: SA, BP, and ART, side by side, for two images.

6. Comments and future work

The three proposed soft computing algorithms (SA, BP & GA) yielded results which challenge ART, one of the well known conventional methods in the field, in the case of a limited number of projection angles.

In case of the simulated annealing reconstruction system, the system shows, so far, superiority over the other methods, in terms of quality of reconstruction. Parallelization of the application of the system is considered at the present time for speed up and better quality. Other representations of energy gap are also being tested.

In case of the back propagation reconstruction system, with the structure and mechanics proposed, the *delta rule* succeeded well in minimizing the cost function, and led to fairly good estimation of the unknown original images. The system has moderate time complexity. Further modification of the system is considered at present by incorporating some *a priori* information, say, information on the *uniformity* of the unknown image in the adaptation scheme.

In case of the genetic algorithm system, good estimations of the reconstructed images is obtained. The system suffers from high time complexity; the algorithm takes long computational time before the program converges to a satisfactory solution, and that necessitates parallelization of the algorithm. A merit of the genetic algorithm is the ease of incorporation of constraints or *a priori* information about the original image in the reconstruction process. A novel feature of our GA is usage of dynamic chromosome size, which led to faster and better quality of reconstruction.

A novel feature and advantage of the new methods over the conventional ART, is the possiblity of incorporation of constraints within the reconstruction algorithm. However, the fact we declared about the superiority of the proposed methods over ART has been confirmed only for rather simple and relatively small size images. Therefore, in order to derive general conclusions and declare robust results, further efforts should be made to apply those algorithms to real CT images.

[5]IBM(Pentium) machines with GNAT(Ada95) compiler were used.

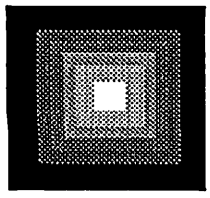
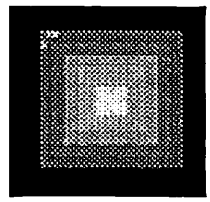
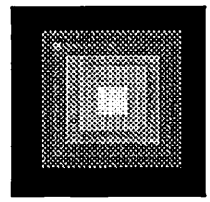
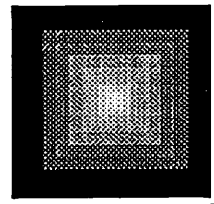
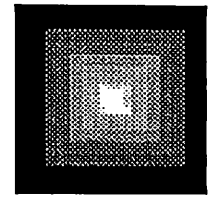
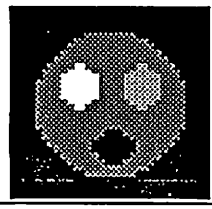
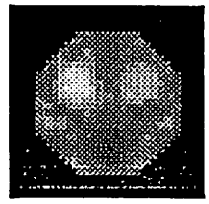
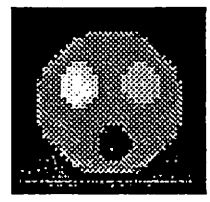
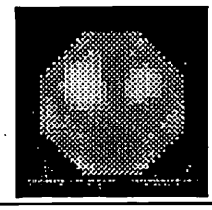
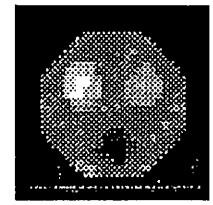
Original Image	ART Image	SA Image	BP Image	GA Image
				
δ_e	*6.75*	*3.58*	*5.38*	*5.77*
				
δ_e	*21.45*	*9.10*	*21.09*	*18.70*

Figure 10: Results of reconstruction by ART, SA, BP, & GA for two images

References

[1] A. Rosenfeld, & A. Kak, *Digital Picture Processing* (2nd ed., Academic Press, Inc., 1982).

[2] Z. Nakao, Y. W. Chen, & Y. Kina, ART estimation of color images from projections, *Joint Technical Conference on Circuits/Systems, Computers and Communications* (Kumamoto, Japan), pp.217-220, 1995.

[3] W. Smith, H. Barrett, & R. Paxman, Reconstruction of objects from coded images by simulated annealing, *Optics Letters*, vol.8, No.4, pp.199-201, 1983.

[4] R. Beale, & T. Jackson, *Neural Computing: An Introduction* (IOP Publishing Ltd, 1990).

[5] Z. Nakao, Y. W. Chen, S. Tobaru, F. F. Ali, & T. Tengan, CT image reconstruction by back propagation, *IASTED International Conference on Artificial Intelligence, Expert Systems and Neural Networks*, Honolulu, Hawaii, USA, pp.285-286, 1996.

[6] F. F. Ali, Reconstruction of CT Images by Back Propagation & Genetic Algorithms, *Unpublished M.E. Thesis*, Graduate College, Univ of the Ryukyus, Okinawa, Japan, 1997.

[7] Y.W. Chen, N. Miyanaga, M. Yamanaka, M. Nakai, K. Tanaka, K. Nishihara, T. Yamanaka, S. Nakai, Three-dimensional imaging of laser imploded targets, *J. Appl.Phys.* 68(4), August 1990.