High-Speed Parallel Implementation of a Modified PBR Algorithm on DSP-Based EH Topology

K. Rajan, *Member, IEEE*, L. M. Patnaik, *Fellow, IEEE*, and J. Ramakrishna

Abstract— **Algebraic Reconstruction Technique (ART) is an age-old method used for solving the problem of three-dimensional (3-D) reconstruction from projections in electron microscopy and radiology. In medical applications, direct 3-D reconstruction is at the forefront of investigation. The simultaneous iterative reconstruction technique (SIRT) is an ART-type algorithm with the potential of generating in a few iterations tomographic images of a quality comparable to that of convolution-backprojection (CBP) methods. Pixel-based reconstruction (PBR) [10] is similar to SIRT reconstruction, and it has been shown that PBR algorithms give better quality pictures compared to those produced by SIRT algorithms. In this work, we propose a few modifications to the PBR algorithms. The modified algorithms are shown to give better quality pictures compared to PBR algorithms. The PBR algorithm and the modified PBR algorithms are highly compute intensive. Not many attempts have been made to reconstruct objects in the true 3-D sense because of the high computational overhead. In this study, we have developed parallel two-dimensional (2-D) and 3-D reconstruction algorithms based on modified PBR. We attempt to solve the two problems encountered by the PBR and modified PBR algorithms, i.e., the long computational time and the large memory requirements, by parallelizing the algorithm on a multiprocessor system. We investigate the possible task and data partitioning schemes by exploiting the potential parallelism in the PBR algorithm subject to minimizing the memory requirement. We have implemented an extended hypercube (EH) architecture for the high-speed execution of the 3-D reconstruction algorithm using the commercially available fast floating point digital signal processor (DSP) chips as the processing elements (PE's) and dual-port random access memories (DPR) as channels between the PE's. We discuss and compare the performances of the PBR algorithm on an IBM 6000 RISC workstation, on a Silicon Graphics Indigo 2 workstation, and on an EH system. The results show that an EH(3,1) using DSP chips as PE's executes the modified PBR algorithm about 100 times faster than an IBM 6000 RISC workstation. We have executed the algorithms on a 4-node IBM SP2 parallel computer. The results show that execution time of the algorithm on an EH(3,1) is better than that of a 4-node IBM SP2 system. The speed-up of an EH(3,1) system with eight PE's and one network controller (NC) is approximately 7.85.**

I. Introduction

THE problem of three-dimensional (3-D) image reconstruction can be described as follows. Suppose there exists an unknown density distribution function (an unknown distribution of the attenuation coefficient) within a 3-D volume and a number of cone-beam X-ray sources at positions outside the volume of interest. The cone-beam sources provide a series of two-dimensional (2-D) projections of the unknown 3-D distribution function. We wish to estimate the 3-D unknown distribution of the linear attenuation coefficient from these 2-D projections.

Computerized Tomography (CT) is an accepted methodology for medical imaging. There are basically two types of CT image reconstruction methods, namely, iterative and noniterative algorithms. A noniterative algorithm such as convolution-backprojection (CBP) consists of two main computations. One is convolution and the other one is backprojection. An iterative algorithm such as expectation maximization (EM), algebraic reconstruction technique (ART), etc., on the other hand, starts with an initial guess of the object and iteratively corrects the object according to the measured projection data. The major computations in an iterative method are forward (pseudo-projection) and backward (correction) projections.

Algorithm improvement [6], [11], [18], dedicated hardware [4], [17], [27], and parallel processing, [5], [7], [9], [21], [29], are the three general methods adopted to quicken the image reconstruction. Cho *et al.* have proposed an incremental algorithm [6] which restructures the conventional CBP algorithm and performs backprojection on a beam-by-beam (instead of pixel-by-pixel) basis. Kaufman [18] has proposed a few techniques to reduce the computational complexity of expectation maximization (EM)-based positron emission tomography (PET) reconstruction. An EM algorithm is an iterative procedure for finding the maximum likelihood estimate of parameters of a probability distribution function from an incomplete set of measurement data, where the incomplete data may be viewed as a many-to-one function of some unobserved random variable.

Feldkamp *et al.* [11] have proposed a convolution backprojection formula for direct reconstruction of a 3-D object from fan-beam projections. The procedure involves convolution and 3-D projection, and it includes the correct weighting of the data.

Wang *et al.* [28] have proposed a general cone-beam reconstruction algorithm; it allows various scanning loci, and the algorithmic efficiency is comparable to that of Feldkamp's. Schlindwein [25] has developed a twin-cone algorithm for true 3-D reconstruction based on the ART technique. A fully 3-D image reconstruction algorithm from cone-beam sources has been developed by Altschuler and Herman [2]. Defrise and Clark [8] have proposed an inversion formula written in the form of shift-variant filtered backprojection for reconstruction

Manuscript received April 24, 1996; revised September 16, 1996 and March 24, 1997.

K. Rajan is with the Department of Physics, Indian Institute of Science, Bangalore 560 012, India.

L. M. Patnaik is with the Microprocessor Applications Laboratory, I.I.Sc. Bangalore 560 012, India.

J. Ramakrishna is with the Department of Physics, I.I.Sc. Bangalore 560 012, India (e-mail: jr@physics.iisc..ernet.in).

Publisher Item Identifier S 0018-9499(97)05586-X.

from cone-beam data taken from a complete geometry. The major problems in all these reconstruction algorithms are the prohibitively high execution time.

The dedicated hardware approach has been tried in [4], [17], and [27]. Jones *et al.* have suggested a VLSI architecture to support forward and backward projection in [17].

The CBP algorithm gives good pictures, provided a large number of equi-spaced projection data are available. With a lesser number of projection data, iterative algorithms fare better. With noisy projection data, CBP-reconstructed images are found to have ring artifacts. ART-type algorithms give better pictures from noisy projection data compared to those produced by CBP algorithm. ART-type algorithms find a place in single photon emission tomography (SPECT) also. The attenuation map of the object is reconstructed from transmission data collected on one of the cameras of a three-headed SPECT system.

A parallel implementation of a 3-D reconstruction based on CBP is reported by Chen *et al.* [7]. An intel iPSC/2 hypercube system was used for the implementation. In order to exploit the functional parallelism, a two-stage pipelining approach was employed in this study [7] in which the convolution stage is followed by a backprojection stage. Though Chen *et al.* have used a hypercube system, the pseudo-binary tree embedded in the hypercube has been used for the parallel CBP implementation. The two different data flows involved in parallel CBP, namely, broadcasting and integration, were efficiently supported in a binary tree topology. However, no data on the execution time of the algorithm are presented. Shieh *et al.* [9] have developed a scheme for parallel implementation of Radon transform on a linear array using digital signal processors (DSP's). They have studied the accuracy and speed characteristics of implementation of several line integrals for the computation of Radon transform. Chen *et al.* [5] have implemented a parallel EM algorithm on a hypercube topology. No work has been reported to parallelize the 3-D reconstruction algorithm based on ART. In this work, we try to parallelize the modified PBR algorithm for a true 3-D reconstruction.

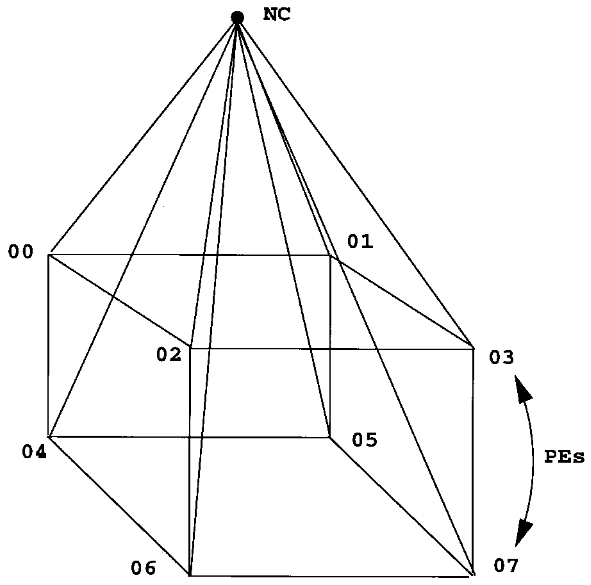


Fig. 1. Implementation of the EH.

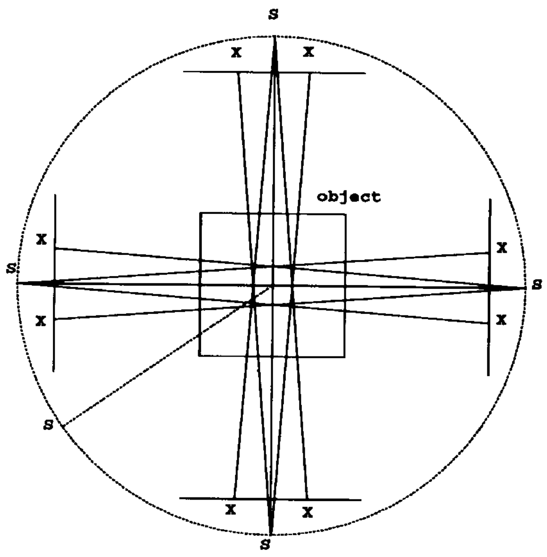


Fig. 2. Illustration of ray-group symmetry.

II. Algorithms

A. ART

ART's were proposed simultaneously by Gordon *et al.* [13] and by Hounsfield [14], in 1970, for obtaining 3-D reconstruction from projections in electron microscopy and radiology.

For an image matrix of size $N_x \times N_y$, with M projections $P_k(n)$, $0 \leq k \leq M-1$, and N_d detector elements $0 \leq n \leq (N_d - 1)$, the problem is to find $f(x, y)$ of $N_x \times N_y$ pixels such that the error between $P_k(n)$ and $R_k^{(q)}(n)$ is minimized, where $R_k^{(q)}(n)$ is the set of computer-calculated projections obtained using the same geometry used in obtaining $P_k(n)$.

Methods of ART in CT are based on the projection line integrals as discrete ray sums [15], [16]. The problem of reconstruction then becomes one of solving a system of linear equations of the form

$$\begin{aligned} P_k(n) = \sum_x \sum_y W(x, y, k, n) f(x, y) \\ k = 1, \cdots, M,\ n = 1, \cdots, N_d \\ x = 1, \cdots, N_x,\ y = 1, \cdots, N_y \end{aligned} \quad (1)$$

for the unknown density function $f(x, y)$ on a sampling grid of $N_x \times N_y$ in 2-D space. The summation coefficients W are the weights that correspond to the length of intersection of a given ray and a pixel.

The ART reconstruction is an iterative method for solving a system of (1). The new estimate $f^{(q+1)}(x, y)$ is determined from the estimate $f^{(q)}(x, y)$ after an update correction

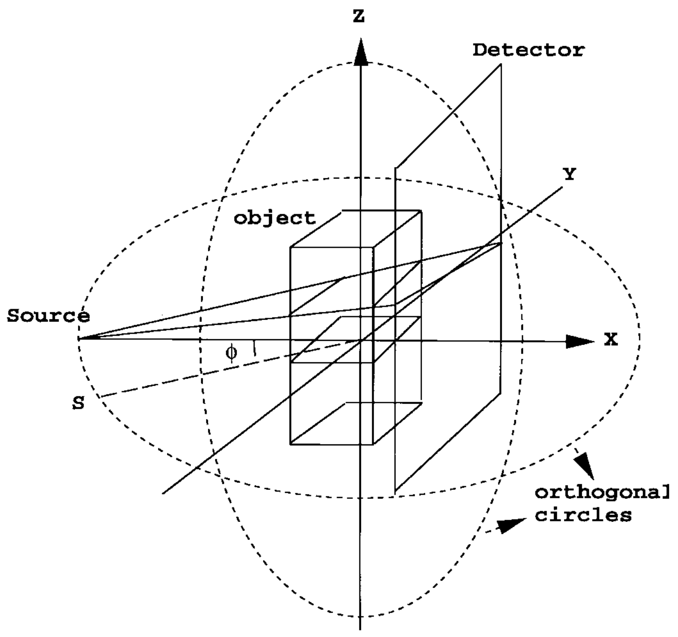


Fig. 3. Cone-beam reconstruction setup.

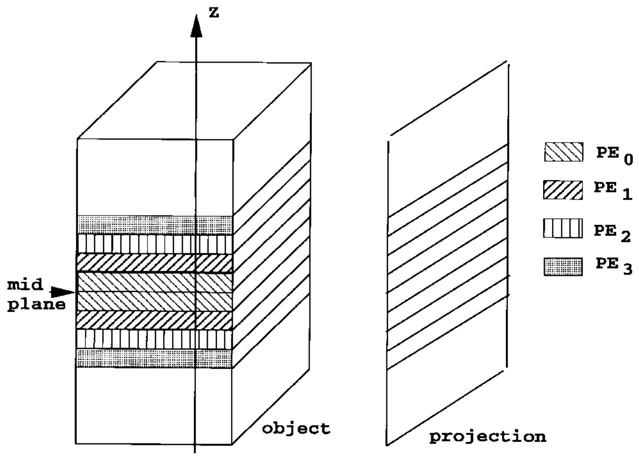


Fig. 4. Decomposition of the object.

procedure [3], [13] as given below:

$$f^{(q+1)}(x, y) = f^{(q)}(x, y) + \lambda^{(q)} W(x, y, k, n) \cdot \frac{P_k(n) - \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} W(x, y, k, n) f^{(q)}(x, y)}{\sum_{x=1}^{N_x} \sum_{y=1}^{N_y} W(x, y, k, n)^2}. \tag{2}$$

The scaling factor $\lambda^{(q)}$ is a relaxation factor; its value can be chosen in the range 0.0–2.0.

Improved reconstructed images can be obtained through a procedure of combining correction terms from all projections before the image function $f(x, y)$ is updated. This approach, the simultaneous iterative reconstruction technique (SIRT) [12], can produce in a few iterations a tomographic image of a quality comparable to that of CBP methods.

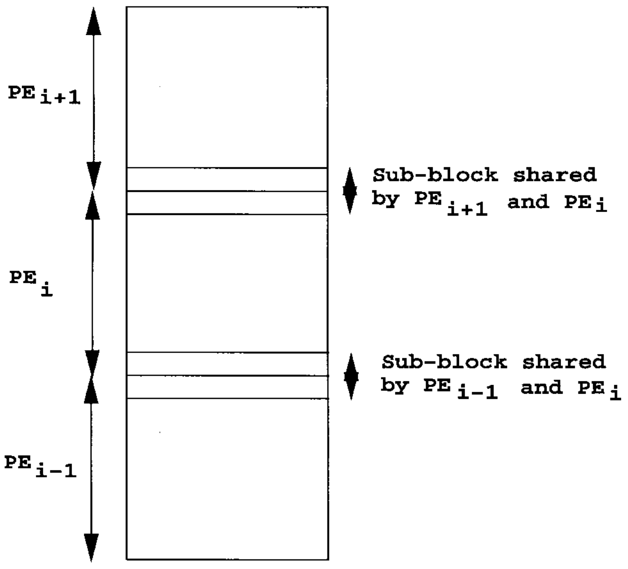

PE i+1
Sub-block shared by PE i+1 and PE i
PE i
Sub-block shared by PE i-1 and PE i
PE i-1

Fig. 5. Sub-block shared by PE_i and PE_{i+1}.

Fager *et al.* [10] have proposed a set of new algorithms called pixel-based reconstruction (PBR) algorithms for reconstruction of object function$f(x, y)$ from their fan-beam projections. Since the pixel update is done by combining corrections from all rays within a particular projection, PBR algorithms have the advantage of noise averaging and give better pictures than the SIRT algorithms.

B. PBR Algorithms

Two prominent PBR algorithms are *Weighted_Reverse_Projection* and *Modified Gilbert's Algorithm.*

1) Weighted_Reverse_Projection: The approach called *Weighted_Reverse_Projection* [10] gives the following expression for the pixel update:

$$\Delta f^{(q+1)}(x, y) = \frac{1}{N_r(x, y)} \cdot \sum_{k=1}^{N_r(x, y)} \frac{\Delta P_k^{(q)}(n)}{L_k(n)} \tag{3}$$

where

$N_r(x, y)$	number of projections passing through pixel (x, y);
$L_k(n)$	length of the ray $r_k(n)$ inside the object.
$\Delta P_k(n)$	difference between the actual projection measurement $P_k(n)$ and computer-calculated projection $R_k^{(q)}(n)$ for the ray $r_k(n)$.

2) Modified Gilbert's Method: The modified Gilbert's method proposed by Fager [10] has the following relation:

$$\Delta f^{(q+1)}(x, y) = \frac{\sum_{k=0}^{N_r(x, y)} \Delta P_k^{(q)}(n)}{\sum_{k=0}^{N_r(x, y)} L_k(n)}. \tag{4}$$

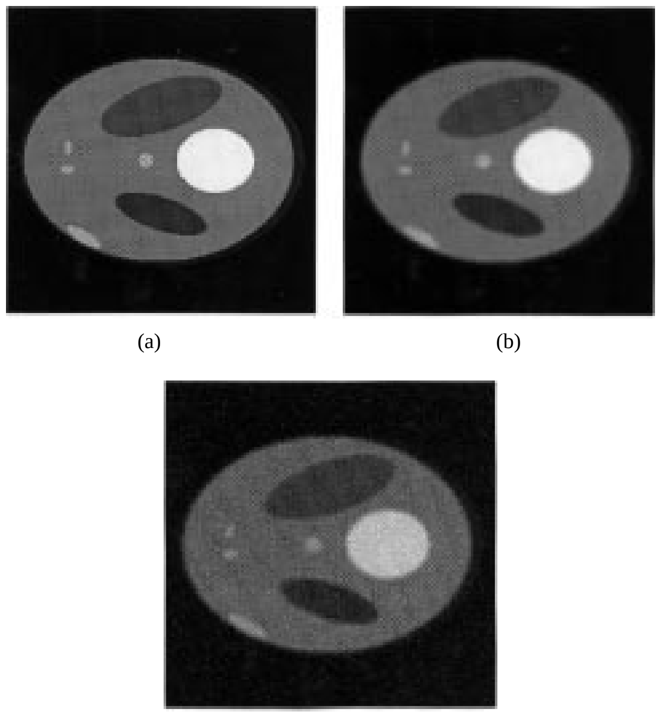


Fig. 6. (a) Original phantom, (b) blurred phantom, and (c) blurred with Gaussian noise.

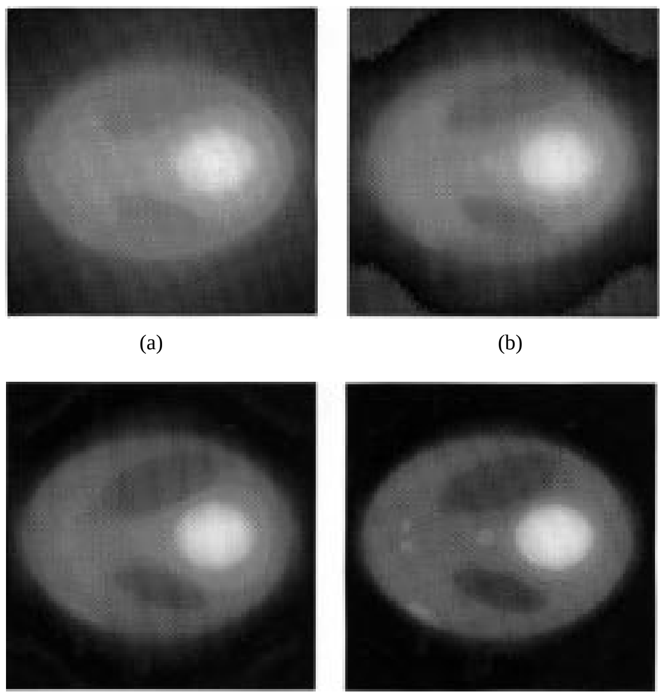


Fig. 7. Image reconstructed using the Reverse_Projection_1 algorithm (a) after one iteration, (b) after two iterations, (c) after five iterations, and (d) after ten iterations. The noise-free phantom is used as the source of the projection data.

We present three modified pixel-based algorithms which were found to give better results, compared to those obtained using Fager's algorithms.

C. Proposed Algorithms

1) Modified Weighted_Reverse_Projection_1: In Fager's *Weighted_Reverse_Projection* algorithm (3), the summation is over projection $k = 1$ to $N_r(x, y)$. For a fan-beam geometry with M projections, N_d rays originate from the source, penetrate the object, and are detected on the linear detector array in each projection. In such a case, multiple rays in a projection pass through a pixel (x, y). So, (3) should be modified as follows:

$$\Delta f^{(q+1)}(x, y) = \frac{1}{N(x, y)} \cdot \sum_{\Gamma(x, y)} \sum_{\Omega(x, y)} \frac{\Delta P_k^{(q)}(n)}{L_k(n)} \tag{5}$$

where we have the following.

$N(x, y)$	total number of rays passing through the pixel (x, y);
$\Gamma(x, y)$	all k that pass through (x, y);
$\Omega(x, y)$	all n that pass through (x, y).

2) Modified Weighted_Reverse_Projection_2: In Fager's algorithm (3), the correction factor is the summation of $\Delta P_k(n)/L_k(n)$ over all projections that pass through (x, y). The factor $\Delta P_k(n)/L_k(n)$ is the error per unit length. When a ray $r_k(n)$ passes through a pixel (x, y), with a finite length $\delta(x, y, k, n)$, the correction factor should be proportional to $\delta(x, y, k, n)$. The new equation for the pixel correction is given below:

$$\Delta f^{(q+1)}(x, y) = \sum_{\Gamma(x, y)} \sum_{\Omega(x, y)} \frac{\Delta P_k^{(q)}(n). \delta(x, y, k, n)}{L_k(n)}. \tag{6}$$

3) Improved Gilbert's Algorithm: The reasoning given for the Modified Weighted_Reverse_1 algorithm can be applied to Fager's modified Gilbert's algorithm (4). The modified reconstruction equation is

$$\Delta f^{(q+1)}(x, y) = \frac{\sum_{\Gamma(x, y)} \sum_{\Omega(x, y)} \Delta P_k^{(q)}(n)}{\sum_{\Gamma(x, y)} \sum_{\Omega(x, y)} L_k(n)}. \tag{7}$$

The modified PBR algorithms are computationally more complex than the conventional ART algorithm. In the *Modified Weighted_Reverse_Projection_1* algorithm, e.g., the number of rays that pass through a pixel is distributed on all processing elements (PE's). The result has to be integrated on a host node. Similarly, the partial corrections are also distributed on all PE's. In addition, the image has to be broadcast to all PE's. So, the parallel topology should be able to handle fast broadcast and integration of partial results. We have replaced the links of an extended hypercube (EH) with dual port RAM (DPR) to enhance the speed of broadcast and integration of partial results.

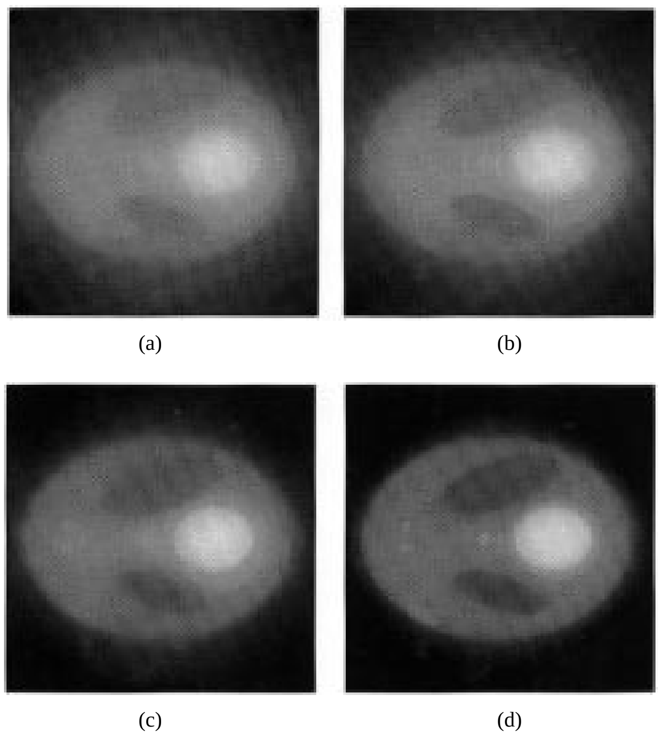


Fig. 8. Image reconstructed using the Reverse_Projection_2 algorithm (a) after one iteration, (b) after two iterations, (c) after five iterations, and (d) after ten iterations. The noise-free phantom is used as the source of the projection data.

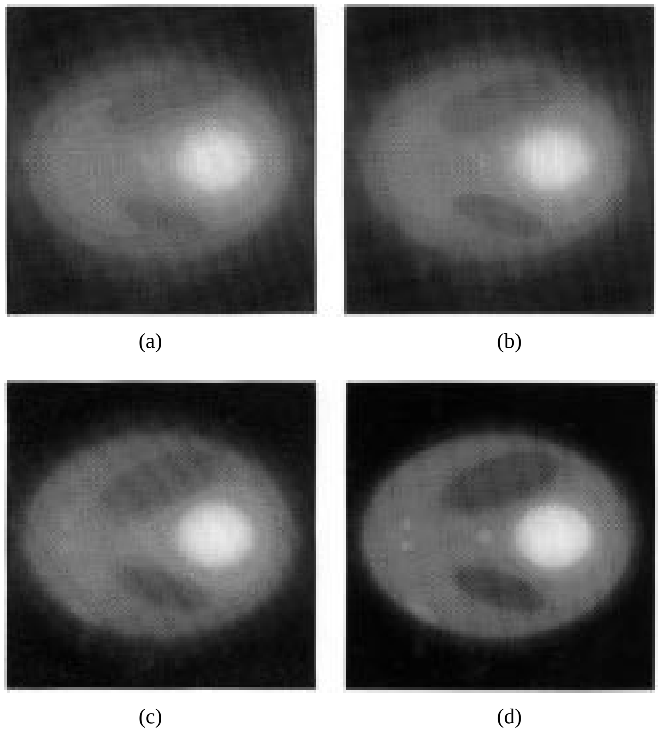


Fig. 9. Image reconstructed using improved Gilbert's algorithm (a) after one iteration, (b) after two iterations, (c) after five iterations, and (d) after ten iterations. The noise-free phantom is used as the source of the projection data.

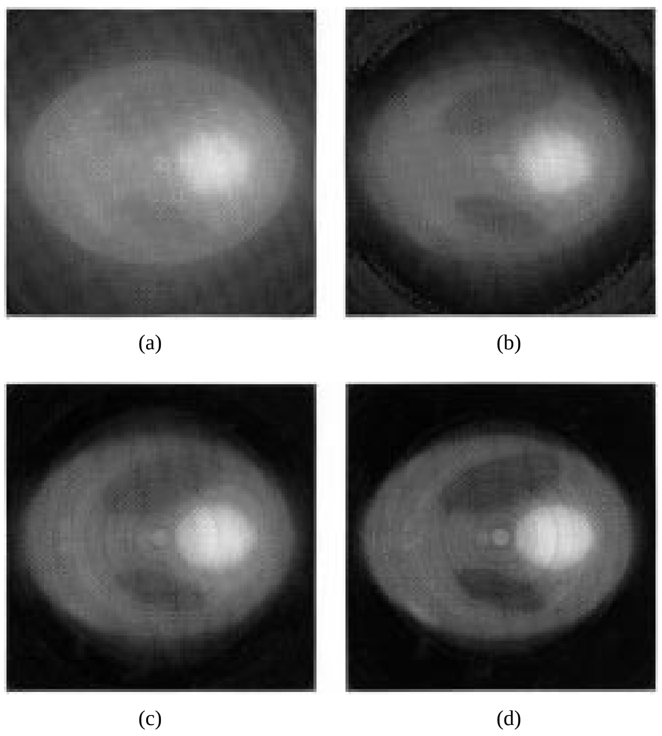


Fig. 10. Image reconstructed using the Reverse_Projection_1 algorithm using the noisy projection data (a) after one iteration, (b) after two iterations, (c) after five iterations, and (d) after ten iterations.

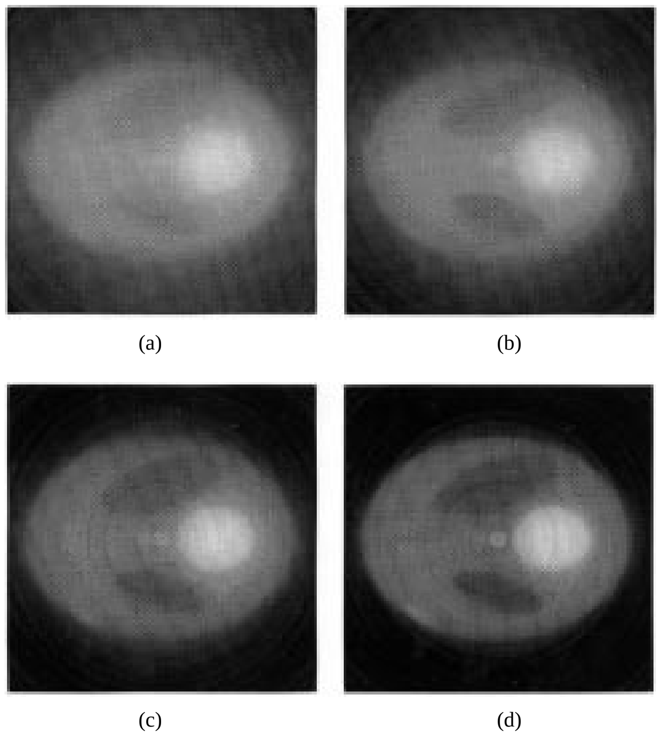


Fig. 11. Image reconstructed using the Reverse_Projection_2 algorithm using noisy projection data (a) after one iteration, (b) after two iterations, (c) after five iterations, and (d) after ten iterations.

III. Extended Hypercube Topology

The EH architecture [20] is suited for hierarchical expansion of multiprocessor systems. The basic module of the EH(k, l) (Fig. 1) consists of a k-cube and an additional node for handling communication—the network controller (NC). 2^k such basic modules are interconnected via 2^k NC's, forming a k-cube among the NC's. An EH(k, l) (l is the degree of the

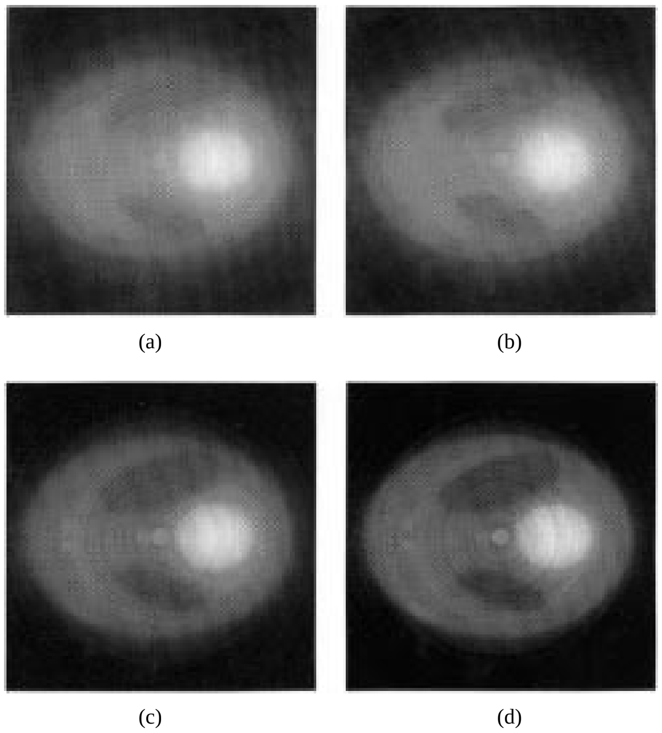


Fig. 12. Image reconstructed using the improved Gilbert's algorithm using noisy projection data (a) after one iteration, (b) after two iterations, (c) after five iterations, and (d) after ten iterations.

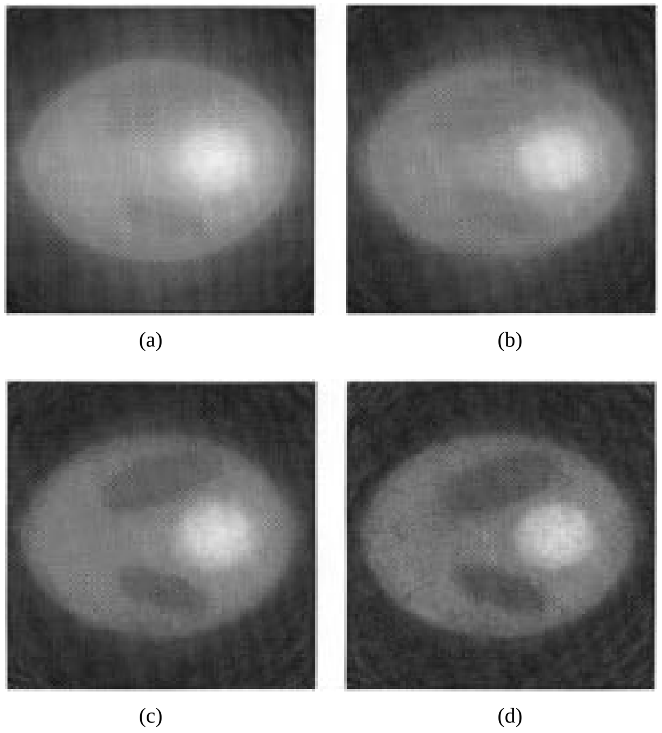


Fig. 13. Image reconstructed using the Reverse_Projection_1 algorithm (a) after one iteration, (b) after two iterations, (c) after five iterations, and (d) after ten iterations. The blurred image with additive Gaussian noise of variance 0.1 is used as the source of the projection data.

EH) consists of one NC at the lth level and a k-cube of 2^k NCs/PE's at the $(l-1)$st level. The NCs/PE's at the $(l-2)$nd level of hierarchy form 2^k distinct k-cubes. Thus, we have k-cubes at all levels j for $0 \leq j \leq (l-1)$. The EH architecture

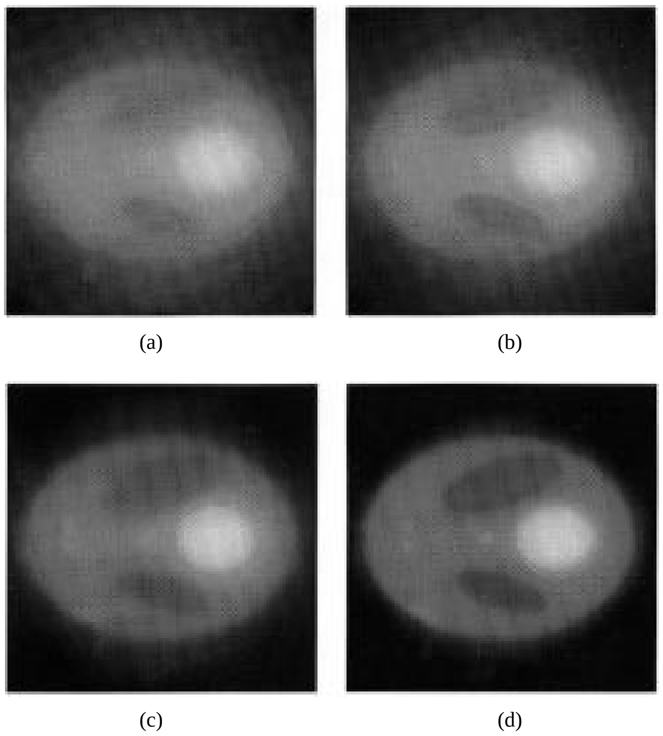


Fig. 14. Image reconstructed using the Reverse_Projection_2 algorithm (a) after one iteration, (b) after two iterations, (c) after five iterations, and (d) after ten iterations. The blurred image with additive Gaussian noise of variance 0.1 is used as the source of the projection data.

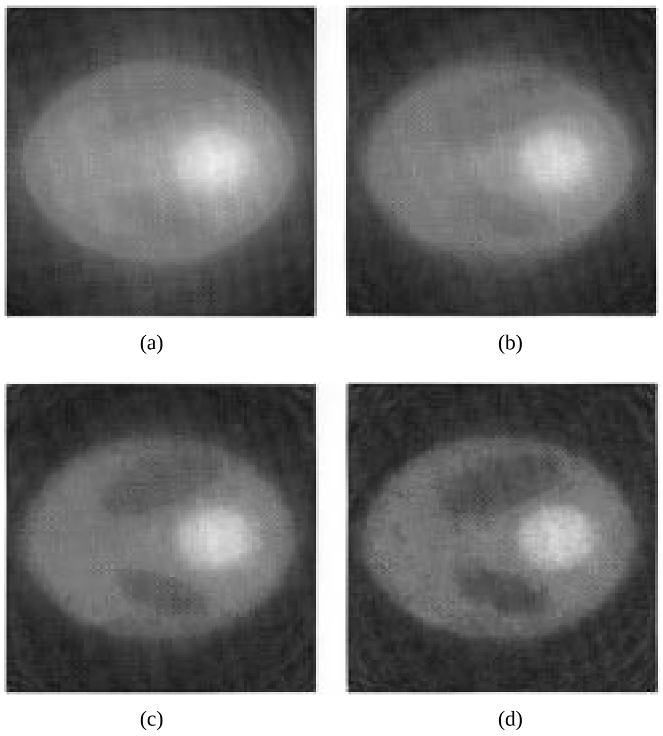


Fig. 15. Image reconstructed using the improved Gilbert's algorithm (a) after one iteration, (b) after two iterations, (c) after five iterations, and (d) after ten iterations. The blurred image with additive Gaussian noise of variance 0.1 is used as the source of the projection data.

retains the positive features of a k-cube at different levels of hierarchy and at the same time has some additional advantages like reduced diameter and constant degree of a node.

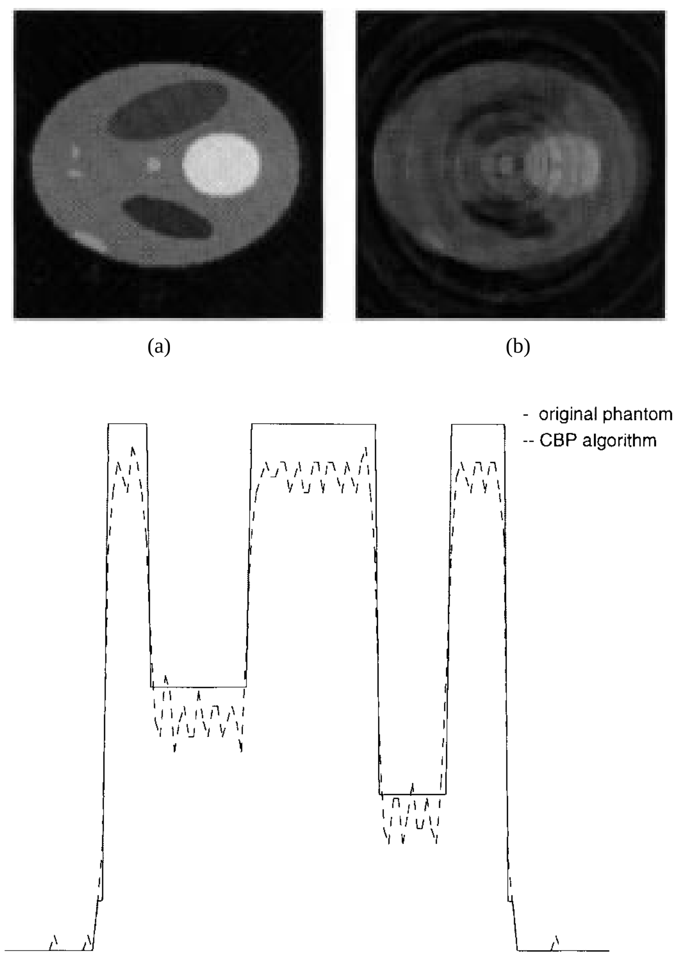


Fig. 16. Image reconstructed using the CBP algorithm. (a) The noise-free phantom, (b) image reconstructed from a noisy projection data, and (c) plots of cross section through 128th column of the image shown in Fig. 16(a).

A. Implementation of the EH(3, 1)

An EH(3, 1) has been implemented (Fig. 1) using ADSP 21 020 [1] DSP chips. Eight DSP chips form the 3-D cube, and one DSP chip is configured as an NC; the NC is connected to all the eight PE's through direct links. The links between PE's and PE's/NC's are created using DPR. A 32 KWord memory channel between the PE and PE/NC is configured as two 16-KWord unidirectional channels. This supports the high-speed data transfer across the link. It takes one processor clock cycle to transfer a word across the link, which gives a transfer speed of 120 MB/s.

The NC is connected to a PC/486 system to use the resources such as memory, keyboard, display, and disk storage. The ADSP 21 020 operates at 40 MHz with a cycle time of 25 ns. The fast channel memories are made up of 15 ns static memories. Message passing among the nodes within a k-cube is performed by adopting one of the hypercube message passing algorithms, and message passing among the nodes at different hierarchical levels is via one or more NC's.

The algorithm for implementing the integration of partial results follows.

1) Integration of Partial Results:

```
begin
  for i=0 to (l−2) step 1
    begin
      for all $PE_i[j]/NC_i[j]$, j=0 to (k−1) in parallel
        begin
            send partial result to the channel linking
            $PE_i[j]/NC_i[j]$ and $PE_{i+1}[j]/NC_{i+1}[j]$
        end $j$
          for all NC's at level i+1 in parallel
            begin
              for j=0 to k−1
              $T_{i+1}(m) = T_{i+1}(m) + T(j)$
               /* $T_{i+1}(m)$ is the partial result in $NC_{i+1}$
                    /* T(j) is the contribution of
                    next lower level PE[j]/NC[j]
        end j
    end i
end.
```

The assembly code for implementing the integration of partial results is given in Appendix C.

2) Message Broadcast from NC to PE's: The message packet from NC is transmitted to all PE's/NC's at the next lower level. The dual-port memory is the bi-directional link between two nodes. The DPR is in the address map of two PE's sharing the DPR. The address map of an NC is shown as follows:

Local Memory
NC-to-PE$_0$
NC-to-PE$_1$
....
....
NC-to-PE$_{2^k-1}$
Broadcast Address

An NC can write a word to all PE's at the next lower level in one clock cycle by writing to the *broadcast address* space of the NC.

The broadcast algorithm from NC at the highest level to all PE's is given below. The broadcasting from NC at the highest level to all PE's is performed in $O(l)$ time.

Broadcast from NC at level (l-1) to all PE's

```
begin
  for each i, i=l−1 to 1 step 1
   $NC_i$ writes to broadcast_address $NC_i/[NC/PE]_{i-1}$
end.
```

IV. 2-D Implementation

A set of rays originating from S, penetrating the image, and detected on the linear detector array is shown in Fig. 2.

For each of the rays $r_k(n)$ from the source S to the detector element n, we compute the x and y coordinates of the pixels in the path of each ray $r_k(n)$ and store them as a 2-D array [[pxl_inx_x] [pxl_inx_y]]. The *pxl_inx_x* and *pxl_inx_y* are the x and y coordinates of the pixel in the path of the ray $r_k(n)$. The lengths of the ray $r_k(n)$ in each of these pixels are computed and stored as one-dimensional (1-D) array *weights*. The number of pixels in the path of the ray $r_k(n)$ is computed and stored as *Nweights*. The array count [[pxl_inx_x] [pxl_inx_y]] is the number of rays passing through a pixel [[pxl_inx_x] [pxl_inx_y]]. To reduce the computation and the memory requirement for storing the array [[pxl_inx_x] [pxl_inx_y]], *weights* and *Nweights*, a grouping technique namely *grouping by symmetry* [19] is employed. As shown in Fig. 2, if the x and y coordinates of the pixels, in the path of the ray marked X, are known then the x and y coordinates of the pixels in the paths of all other rays marked X (four rays if $\phi \neq 0$, and eight rays if $\phi = 0$) can be obtained by the operation of a simple rotation and/or flipping.

The NC at the topmost level broadcasts the data corresponding to $N_x \times N_y$ pixels to the PE's at the $(l-1)$st level. Each PE/NC at level $(l-1)$ broadcasts the data to PE's at the $(l-2)$nd level. Finally, each PE at level 0 receives $N_x \times N_y$ pixels. Each PE has to compute the pseudo-projection and Δf for M/N_p number of source positions (N_p is the number of computing nodes in the EH system). The NC partitions the M projection measurements in M/N_p, the number of partitions, and sends partition[i] to PE[i], $i = 0$ to $N_p - 1$. Each PE at level 0 computes the x and y coordinates of the pixels, length of the ray in each pixel, and the number of pixels in the path of the ray $r_k(n)$.

The pseudo-projection $R_k^{(q)}(n)$ is computed for each of the N_d rays originating from the source S. From the difference between $P_k(n)$ and $R_k^{(q)}(n)$, the correction to the intensity function $\Delta f^{(q)}(x, y)$ is computed. This computation is repeated for all the M positions of the source location S. The PE's at level 0 send $\Delta f^{(q)}(x, y)$ of the image to NC's at level 1. The NC's at level 1 integrate the $\Delta f^{(q)}(x, y)$ received from PE's at level 0 and send the partial results to NC's at level 2.

This process is continued until the NC at the topmost level has the image $f(x, y)$ and the corrections to the image $\Delta f^{(q)}(x, y)$. The NC at the top level computes the final image $f^{(q+1)}(x, y)$. Normally, the mean squared error (MSE) between the projection $P_k(n)$ and the computed projection $R_k^{(q)}(n)$ is used as the criterion for the convergence. This is given by

$$\text{MSE between projections} = \frac{1}{M.N_d}.\sum_{n=1}^{N_d}\sum_{k=1}^{N}[R_k^{(q)}(n) - P_k(n)]^2. \quad (8)$$

The PE's at level 0 also compute the partial MSE. These partial MSE's are integrated at the NC at the top level. The number of iterations required for an application depends on the MSE that application can tolerate. Depending on the MSE, NC decides if the next iteration has to be initiated. The NC distributes the image $f^{(q)}(x, y)$ to PE's for next iteration. The NC at the highest level decides if a new iteration has to be initiated, based on the value of MSE. The algorithm is given in Appendix A.

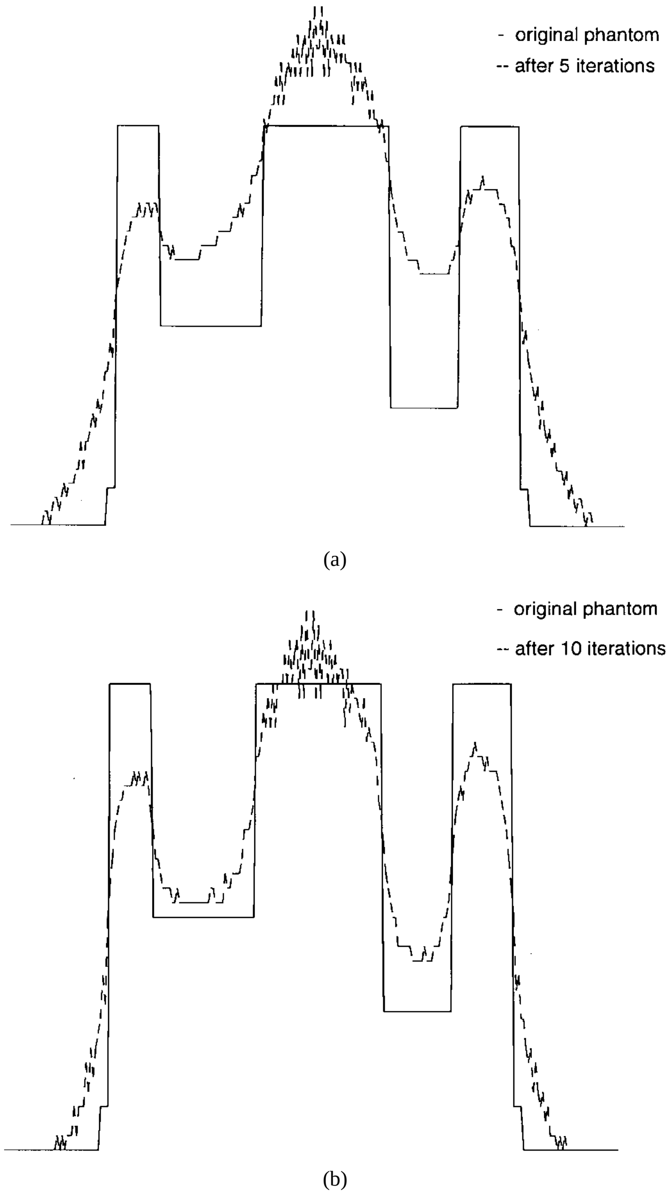
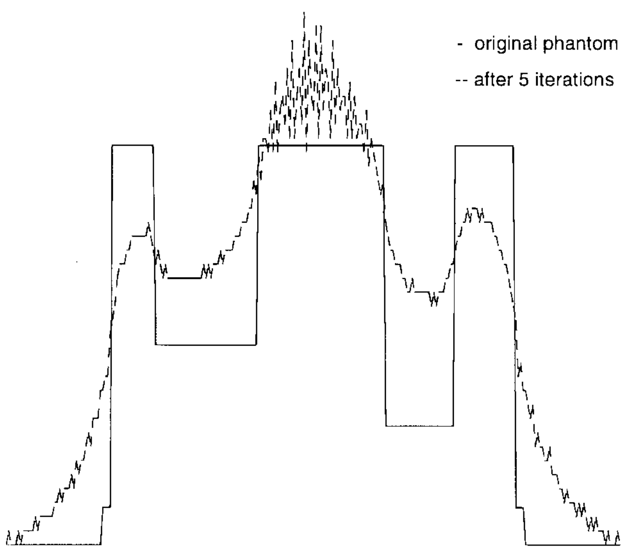
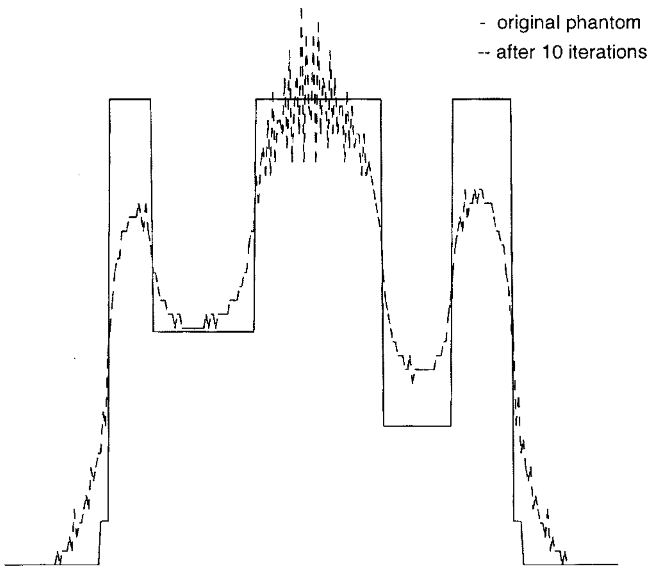


Fig. 17. Plots of the cross section through the 128th column of images (Fig. 7) reconstructed using the Reverse_Projection_1 algorithm: (a) five iterations and (b) ten iterations.

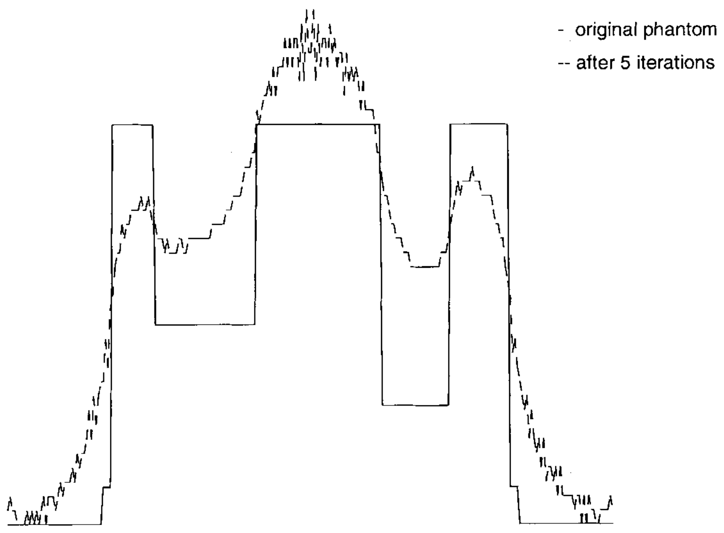
Table I gives the execution times for one iteration of the fan-beam reconstruction algorithm on a single node of an EH, an EH(3, 1) system, on an IBM RS 6000/340 workstation, and on a Silicon Graphics Indigo 2 workstation. The Indigo 2 workstation consisted of R4400 CPU, 4410 FPU running at 150 MHz, 16 kB instruction/16 kB data cache, 1 MB level 2 cache, and 64 MB memory. The Indigo 2 runs under IRIX 5.3 OS. The IBM RS 6000/340 consisted of RS 6000/340 CPU running at 33 MHz, 8 kB instruction/32 kB data cache, and 32 MB memory. The system runs under IBM AIX 3.5 OS. In both cases, the executables are created using the C compiler with O2 level optimization enabled. The EH consisted of ADSP



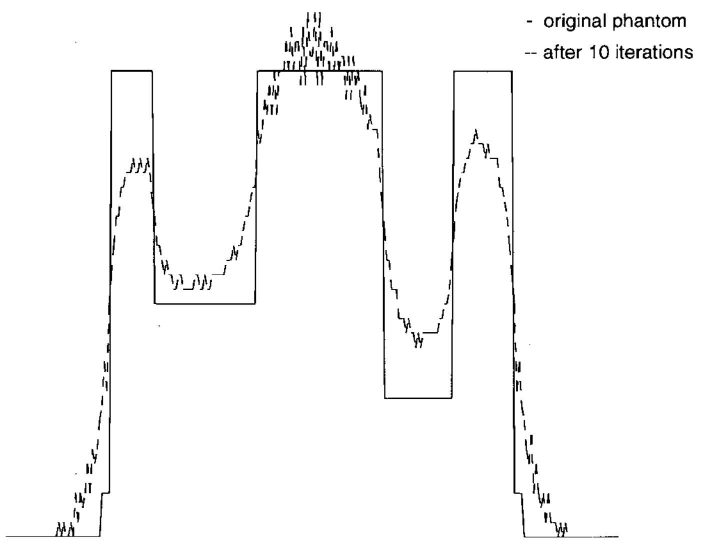
(a)



(b)



(c)



(d)

Fig. 18. Plots of cross section through 128th column of images (Fig. 8) reconstructed using Reverse_Projection_2 algorithm: (a) five iterations and (b) ten iterations, and (c) plots of cross section through 128th column of images (Fig. 9) reconstructed using the improved Gilbert's algorithm after five iterations, and (d) after ten iterations.

TABLE I
EXECUTION TIME FOR ONE ITERATION OF THE FAN-BEAM ALGORITHM BASED ON MODIFIED PBR

Image size $N_x \times N_y$	Detector Elements N_d	EH(3,1)	Single Node of an EH	IBM 6000 RISC	Indigo 2
64*64	128	91 ms	718 ms	7.8 s	2.15 s
128*128	256	338 ms	2.7 s	31 s	8.3 s
256*256	512	1.32 s	10.3 s	131 s	33 s

TABLE II
COMPARATIVE COMMUNICATION PERFORMANCE OF AN EIGHT-NODE IBM SP2 AND EH(3, 1)

Operation	Communication time SP2	Communication time EH(3,1)
point-to-point	186 μ s	25 μ s
broadcast	384 μ s	25 μ s
integration of partial results	83 ms	225 μ s

21 020 CPU's running at 40 MHz. The execution time on the EH includes the communication overhead. The speed-up of an EH(3, 1) with 8 PE's and one NC is 7.85.

The results show that an EH(3, 1) system executes the 2-D modified PBR about 25 times faster than a Silicon Graphics Indigo 2 workstation and about 90 times faster than an IBM RS 6000/340 system.

We have implemented the algorithms on an IBM SP2 system to see if one can achieve a shorter execution time using a faster hardware/parallel computing system. IBM SP2 is a massively parallel processor (MPP) with distributed memories. Interprocessor communication is achieved through explicit message passing. Each node is equipped with a 66-MHz POWER2 processor and 256 MB of local memory. Each POWER2 superscalar processor has a peak performance of 266 MFLOPs/s. The nodes are interconnected by multistage network called a high-performance switch (HPS). HPS is a packet switching with buffered wormhole routing [30]. Table II gives the comparative communication performance of an eight-node EH and an eight-node IBM SP2. The message size is 1024 words.

Table III gives the execution time of the modified PBR algorithm on an SP2 system consisting of one, two, and four nodes. Dedicated nodes were used to minimize interference from other tasks. The IBM message passing library (MPL) has been used for communication. The codes were compiled using the -O3 -qarch=pwr2 option switch, which generates most optimized POWER2 codes. The studies show that the EH(3,1) system gives better performance than the IBM SP2 system with four nodes in implementing the ART-type algorithms.

V. 3-D RECONSTRUCTION BY STACKING MULTIPLE PARALLEL PLANES

Most of the 3-D reconstruction techniques, however, are actually 2-D procedures for reconstructing an object from co-axial projections. The full 3-D reconstruction is achieved by reconstructing several parallel planes separately. We have

TABLE III
EXECUTION TIME OF A MODIFIED PBR ALGORITHM ON IBM SP2 FOR THE IMAGE OF 256 $\times$ 256 AND DETECTOR ELEMENTS OF 512

	1 node	2 nodes	4 nodes
execution time	4 s	3 s	1.5 s
Speedup	1	1.33	2.66

implemented the modified PBR algorithm by reconstructing several parallel planes separately. The equation for 3-D reconstruction is given by

$$\Delta f^{(q+1)}(x, y, z) = \frac{1}{N_r(x, y, z)} \cdot \sum_{\Gamma(x,y,z)} \sum_{\Omega(x,y,z)} \sum_{\Lambda(x,y,z)} \cdot \frac{\Delta P_k^{(q)}(i, j)}{L_k(i, j)} \tag{9}$$

where

$N_r(x, y, z)$	the number of rays passing through (x, y, z);
$\Gamma(x, y, z)$	all projections k that pass through (x, y, z);
$\Omega(x, y, z)$	all i that pass through (x, y, z);
$\Lambda(x, y, z)$	all j that pass through (x, y, z).

Smith [24] has given the necessary and sufficient conditions for cone-beam reconstructions. Geometries that satisfy this condition are referred to as being complete, and geometries that do not do so are referred to as incomplete. The twin cone-beam geometry [25] connected by a straight line is a complete geometry. The dual orthogonal circular geometry is also a complete geometry.

A. System Geometry

The schematic physical arrangement of the 3-D tomographic system based on cone-beam reconstruction is shown in Fig. 3.

TABLE IV
EXECUTION TIME FOR ONE ITERATION OF THE CONE-BEAM ALGORITHM BASED ON MODIFIED PBR FOR THE STACKED PLANES APPROACH AND "TRUE 3-D"

Object size $N_x \times N_y \times N_z$	Detector plane $N_{dx} \times N_{dy}$	EH(3,1)		Single Node of EH		IBM 6000 RISC	Indigo 2
		stacked	'true' 3-D	stacked	'true' 3-D		
16*16*8	16*8	9.4 ms	10.88 ms	73 ms	84 ms	1.27 s	.33 s
32*32*16	32*16	70.5 ms	84 ms	551 ms	655 ms	9.3 s	2.64 s
64*64*32	64*32	514 ms	651 ms	4 s	5 s	71.33 s	21 s
128*128*64	128*64	5.2 s	6.3 s	40.4 s	47.1 s	730 s	203 s
256*256*64	256*64	22.5 s	27.35 s	175 s	210 s	3259 s	1042 s

TABLE V
PARAMETERS OF THE PHANTOM USED FOR SIMULATION STUDIES

Ellipse No.	Origin		radius		Orientation	Density
	x0	y0	Semi-major	Semi-minor		
1	0	0	0.69	0.92	0	0.1
2	0	-0.018	0.66	0.87	0	0.9
3	0	0.35	0.21	0.25	0	1.0
4	0.35	0	0.11	0.31	-0.314	-0.7
5	-0.35	0	0.16	0.41	0.314	-0.5
6	0	-0.1	0.046	0.046	0	0.5
7	-0.08	-0.605	0.046	0.023	0	0.5
8	0.06	-0.065	0.023	0.046	0	0.5
9	0.5	-0.5	0.0375	0.125	-0.524	0.5

The object is placed such that its mid-plane passes through the source S. The planar 2-D detector array is placed opposite the source on the other side of the object. The point source S is perpendicular to the axis of rotation and is at the mid-plane. The detector plane is placed such that the mid-plane passes through the center of the detector plane. We take the object to be stationary, and the source and the detector array are rotated about the Z axis of the object. The intersection of the axis of rotation and the mid-plane is taken as the origin of the coordinate system. We have used dual orthogonal circular scanning [24], where the scanning loci of the first source S form a circle in the mid-plane. The second source scans on a perpendicular circle as shown in Fig. 3.

B. Parallelization of the PBR Algorithm

We investigate the possible task and data partitioning schemes by exploiting the potential parallelism in the PBR algorithm subject to minimizing the memory requirement. For the geometry selected, we have the data parallelism in the sense that each object plane can be processed independently. The object has to be reconstructed on a 3-D mesh of points, $N_x \times N_y \times N_z$.

The NC at the topmost level partitions the data corresponding to N_z slices into 2^k number of blocks and sends one block each to the nodes at the $(l-1)$st level. Each NC at level $(l-1)$ subpartitions its block into 2^k number of smaller blocks and sends one sub-block each to the PE's at the $(l-2)$nd level. Finally, each PE at level 0 receives N_z/N_p slices. Each PE at level 0 computes the arrays [[pxl_inx_x][pxl_inx_y]] and *weights*. The pseudo-projection $R_k^{(q)}(i, j)$ is computed for each of the N_{dx} rays originating from the source S. From the difference between $P_k(i, j)$ and $R_k(i, j)^{(q)}$, the correction to the intensity function $\Delta f^{(q)}(xyz)$ is computed. The PE's at level 0 integrate the partial results and compute $f^{(q)}(x, y, z)$ and send the partial object to NC's at level 1. The NC's at level 1 send the partial results to NC's at level 2. This process is continued until the NC at the topmost level has the complete object function $f(x, y, z)$.

The iteration is continued until the convergence condition specified by (8) is satisfied. The PE's compute the partial MSE's, and the partial MSE's are integrated at the NC. The NC decides if a new iteration has to be carried out, based on MSE.

Table IV gives the execution times for one iteration of the PBR reconstruction algorithm on a single node of an EH, on an EH(3, 1) system, on an IBM RS 6000 RISC workstation, and on a Silicon Graphics Indigo 2 workstation.

VI. TRUE 3-D RECONSTRUCTION

In this section, we look at the true 3-D reconstruction scheme using divergent X-ray beams.

We partition the object into slices, symmetrically about the mid-plane, i.e., each PE stores N_z/N_p slices of the object. The partitioning scheme is shown in Fig. 4. PE$_0$ gets the mid-slice, and there are $\{[(N_z + 1)/2.N_p] - 1\}$ slices on the $+Z$ direction and an equal number of slices in the $-Z$ direction, symmetrically about the mid-plane. All the other PE's get $(N_z/2.N_p)$ slices in the $+Z$ direction and an equal number

TABLE VI
MEAN-SQUARED ERROR BETWEEN THE IMAGE AND THE RECONSTRUCTED IMAGE USING MODIFIED PBR ALGORITHMS

Algorithm	Iterations			
	1	10	50	100
Proposed Algorithm				
Modified Reverse_Projection_1	3.6	0.38	0.19	0.15
Modified Reverse_Projection_2	6.0	1.2	0.32	0.22
New Gilbert's	4.5	0.5	0.2	0.15
Fager Algorithm				
Reverse_Projection	14	1.5	0.5	0.25
New Gilbert's	27	2.0	0.6	0.3
CBP	11			

of slices in the $-Z$ direction, symmetrically around the mid-plane. Similarly, the projection data is split into N_p number of partitions. When a cone beam diverges from the source and passes through a 3-D object, the ray traverses multiple slices in the object. So, a PE is required to share information with other PE's. We distribute the object and projection sub-blocks such that a PE shares the information with other PE's connected on a hypercube link. Each PE at level 0 has k number of PE's at a distance of one hop. Since the DPR is shared by two PE's, and the DPR can be accessed by both PE's simultaneously, the shared information can be stored in DPR. The duplication of the same data on two PE's is thus eliminated. Fig. 5 shows the data shared by two neighboring PE's. The dual-port memory connecting PE$_i$ with PE$_{i+1}$ holds the set of slices shared by PE$_i$ and PE$_{i+1}$. Similarly, the dual-port RAM linking PE$_{i-1}$ with PE$_i$ holds the set of slices shared by PE$_{i-1}$ and PE$_i$. The 3-D reconstruction algorithm is given in Appendix B.

The updated partial object $f(x, y, z)$ is sent to the NC, which integrates the result from all PE's and sends it to the host.

Table IV gives the execution times for one iteration of the PBR reconstruction algorithm on an EH(3, 1) system, on an IBM RISC RS 6000 workstation, and on a Silicon Graphics Indigo 2 workstation.

The results show that the computational speed of an EH(3, 1) system for the cone-beam reconstruction using stacked parallel planes is 40 times faster than that of a Silicon Graphics Indigo 2 workstation. Compared to the IBM 6000 RISC workstation, the EH system gives 130 times better speed performance. The "true 3-D" algorithm on an EH(3, 1) is 30 times faster than that of a Silicon Graphics Indigo 2.

VII. SIMULATION RESULTS

The phantom used for simulation studies is shown in Fig. 6(a). The parameters of the phantom are given in Table V.

Extensive simulation studies of the new algorithms have been carried out on: 1) noise-free projection data; 2) noisy projection data; and 3) on a blurred and noisy phantom. In order to simulate the algorithms using noisy projection data, we added a Gaussian noise of mean zero and variance 0.1 to the projection data. To study the effects of blurring on the reconstructed images, we used projection data from a blurred image with additive Gaussian noise.

Fig. 6(a) is the noise-free phantom. Fig. 6(b) is a phantom obtained by blurring the original phantom shown in Fig. 6(a) with a 5 × 5 window of constant coefficients (0.04). Fig. 6(c) is the blurred phantom with an additive Gaussian noise. The reconstruction region, a square of size 256 × 256, is placed such that its center coincides with the center of rotation of the source S. The linear array detector having 512 elements is placed, diametrically opposite the source, on the other side of the object.

The source and the detector array are rotated about the center of the object at equally spaced angles. The center of the object and the detector array lie on the central ray of each fan-beam. Simulation studies have been carried out with 120 projection measurements. We have assumed $f^{(0)}(x, y) = 0$ as the initial condition. The (MSE$_{\text{image}}$) between the object and the reconstructed object has been used as the criterion for comparing the algorithms.

$$\text{MSE}_{\text{image}} = \frac{1}{N_x . N_y} . \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} [f^{(q)}(x, y) - f(x, y)]^2. \quad (10)$$

Figs. 7–9 show the reconstructed images obtained using the *Modified Weighted_Reverse_Projection_1*, *Modified Weighted_Reverse_Projection_2*, and *Improved Gilbert's Algorithms*, respectively. The reconstructed images obtained after one, two, five, and ten iterations are shown in these figures.

Figs. 10–12 show the reconstructed images using the noisy projection data. Figs. 13–15 show the reconstructed images of the blurred phantom shown in Fig. 6(b).

Fig. 16(a) shows the reconstructed image based on CBP using the Shepp–Logan [26] reconstruction filter. The image reconstructed using CBP for a noisy projection data is given in Fig. 16(b). Rowland [22] has analyzed the effect of additive Gaussian noise in the projection data on the reconstruction. The results show that the reconstruction of noise with a constant variance in the projection measurements does not have a constant variance but is instead a function of the position (x, y). The standard deviation fluctuates as a function of position for a band limiting filter with linear interpolation. So, ring artifacts show up in the reconstructed image.

Figs. 16(c), 17, and 18 are the plots of the cross section through the 128th column of images reconstructed using CBP and the proposed algorithms, respectively. Table VI shows the MSE$_{\text{image}}$ using the proposed algorithms, Fager's algorithm, and the CBP algorithm. The MSE of the image after 10 iterations is 0.38. As the number of iterations is increased to 100, the MSE falls to 0.15.

VIII. CONCLUSIONS

In this paper, we have described the parallelization of a fan-beam reconstruction scheme based on modified PBR algorithm

```
fan-beam reconstruction based on PBR algorithm
begin
 The PEs at level 0 receive the image and projection partition from NC
  for each projection k
    begin
      for each n, n = 0 to ($N_d$ - 1) step 1
        begin
          compute the array [[pxl_inx_x] [pxl_inx_y]] /* x,y coordinates of pixel */
          compute 1-D array $weights$            /* length of the ray in each pixel */
          compute $Nweights$            /* number of voxels in the path of the ray */
            do for t = 0 to (Nweights-1)
              begin
                $R_k(n) = R_k(n)$ + f[pxl_inx_[t] pxl_inx_y[t]] * weights[t]
              endfor t
            $\Delta P_k(n) = R_k(n) - P_k(n)$
            do for t = 0 to (Nweights-1)
              begin
              $\Delta$ f[pxl_inx_x[t] pxl_inx_y[t]] =
                $\Delta$ f[pxl_inx_x[t] pxl_inx_y[t]] + $\Delta P_k(n)$ / $\sum_{l=0}^{Nweights-1}(weights[l])$
              increment count[[pxl_inx_x[t] pxl_inx_y[t]] /* number of rays thru pixel */
              endfor t
        endfor n
          repeat for all other 3/7 rays in the same ray-group.
          compute partial MSE   .
    endfor k
     send partial $\Delta f$, partial count and partial MSE to NC at the next higher level for integration
end begin
```

The NC at the highest level integrates (1) partial Δf, (2) partial count from all PEs/NCs of the next lower level, (3) updates the image f(x,y), and (4) integrates partial MSE from all PEs/NCs at the next lower level.

on an EH topology. The scheme has been extended for a 3-D object based on stacked multiple parallel 2-D planes. A true 3-D reconstruction scheme also has been developed. DSP chips have been used as PE's. Since DSP chips are optimized processors for executing *multiply* and *multiply/accumulate* instructions, they give high performance. Another factor that speeds up the execution of the PBR algorithm is the high-speed memory channel linking the PE's/NC's. The computation and communication tasks overlap in the EH system because of the memory channel. For faster speed, we have written the code for the DSP's entirely in assembly language.

Appendix A

See the algorithm shown at the top of the page. The NC at the highest level integrates 1) partial Δf; 2) partial count from all PE's/NC's of the next lower level; 3) updates the image $f(x, y)$; and 4) integrates partial MSE from all PE's/NC's at the next lower level.

Appendix B

See the algorithm at the top of the next page.

Appendix C

The time required to perform the integration of a partial result of one word is 2^k CPU clock cycles. There is no overhead for iterating N data items since the DSP device supports zero overhead loop facility. See the algorithm at the bottom of the next page.

```
True 3-D Cone-beam reconstruction based on modified PBR algorithm
begin
 receivethe object sub-block from NC
 send the object slice to be shared with $PE_{i+1}$ to the channel memory shared
                 by $PE_i$ and $PE_{i+1}$
 receive the object slice to be shared with $PE_{i-1}$ from the channel memory shared
                 by $PE_i$ and $PE_{i-1}$
  for each projection k
   begin
    for each (i, j) i = $0, \ldots, N_{dx}-1$; j = $0, \ldots, N_{dy}-1$
     begin
      compute array  [pxl_inx_x][pxl_inx_y][pxl_inx_z] /* coordinates of voxel */
      compute 1-D array weights              /* length of the ray in each voxel */
      compute Nweights              /* number of voxels in the path of the ray */
       do for t = 0 to (Nweights-1)
        begin
         $R_k(i, j)$ = f[pxl_inx_[t] pxl_inx_y[t] pxl_inx_z[t]] * weights[t]
        endfor t
       $\Delta P_k(i, j) = R_k(i, j) - P_k(i, j)$
       do for $t = 0$ to (Nweights-1)
        begin
         Δ f[pxl_inx_x[t] pxl_inx_y[t] pxl_inx_z[t]] =
         Δ f[pxl_inx_x[t] pxl_inx_y[t] pxl_inx_z[t]] + $\Delta P_k(i, j)/\sum_{l=0}^{Nweights-1}(weights[l])$
         increment count[[pxl_inx_x[t] pxl_inx_y[t] pxl_inx_z[t]]    /* number of rays
        endfor t                                   /* through a voxel ($x, y, z$) */
       repeat for all other 7/15 rays in the same ray-group
      endfor (i, j)
   endfor k
 integrate the partial count of the number of rays passing through the overlapped slices between
 $PE_{i-1}$ and $PE_i$, and $PE_i$ and $PE_{i+1}$,
 integrate the partial correction $\Delta f$ of the voxels it shares with that of $PE_{i+1}$, and $PE_{i-1}$
   integrate the partial MSE for the slices it shares with that of $PE_{i+1}$, and $PE_{i-1}$
   for each object voxel (x, y, z) in the slices allocated to $PE_i$
    Δ f[x, y, z] = Δ f [x, y, z] / count[x][y][z]
   if $f^{(q+1)}(x, y, z) \geq 0$   $f^{(q+1)}(x, y, z) = f^{(q)}(x, y, z) + \lambda.\Delta f^{(q)}(x, y, z)$
   else $f^{(q+1)}(x, y, z) = f^{(q)}(x, y, z)$
   send the object block to the NC
end begin
```

The assembly program to implement the integration of partial results is given below:

```
                R0=0.0                                   ;sum in register R0
counter = N, do repeat until counter underflows        ; N partial results to be integrated
                                    R1 = dm($PE_0$)     ;dm($PE_0$) is the $NC/PE_0$ link address
                R0 = R0 + R1,       R1 = dm($PE_1$)     ;dm($PE_1$) is the $NC/PE_1$ link address
                ...........,        ...........
                R0 = R0 + R1,       R1 = dm($PE_{(2^k-1)}$)
repeat:   R0 = R0 + R1
```

REFERENCES

[1] *Analog Device User's Manual*, ADSP 21020, 1992.

[2] M. D. Altschuler and G. T. Herman, "Fully 3-D image reconstruction using series expansion methods," *A Review of Information Processing in Medical Imaging,* A. B. Brill *et al.*, Eds. Oak Ridge, TN: Oak Ridge National Lab., 1977, p. 125.

[3] A. H. Andersen, "Algebraic reconstruction in CT from limited views," *IEEE Trans. Med. Imaging,* vol. 8, pp. 50–55, Mar. 1989.

[4] S. Barresi, D. Bollini, and A. D. Guerra, "Use of transputer system for fast 3-D image reconstruction in 3-D PET," *IEEE Trans. Nucl. Sci.*, vol. 27, pp. 812–816, Apr. 1991.

[5] C.-M. Chen and S.-Y. Lee, "On parallelizing the EM algorithm for PET image reconstruction," *IEEE Trans. Med. Imaging,* vol. 5, pp. 860–873, Aug. 1994.

[6] Z. H. Cho, C.-M. Chen, and S.-Y. Lee, "Incremental algorithm—A new fast backprojection scheme for parallel beam geometries," *IEEE Trans. Med. Imaging,* vol. 9, pp. 207–217, June 1990.

[7] C.-M. Chen, S.-Y. Lee, and Z. H. Cho, "A parallel implementation of 3-D CT image reconstruction on hypercube multiprocessor," *IEEE Trans. Nucl. Sci.,* vol. 37, pp. 1333–1346, June 1990.

[8] M. Defrise and R. Clack, "A cone-beam reconstruction algorithm using shift-variant filtering and cone-beam projections," *IEEE Trans. Med. Imaging,* vol. 3, pp. 186–195, Mar. 1994.

[9] E. Shieh, K. Wayne, P. J. Hurst, and I. Agi, "High-speed computation of radon transform and backprojection using an expandable multiprocessor architecture," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 2, pp. 347–359, Dec. 1992.

[10] R. S. Fager, V. P. Kumar, and G. N. Kumar, "Pixel-based reconstruction techniques for CT reconstructions," *IEEE Trans. Med. Imaging,* vol. 12, pp. 4–9, Mar. 1993.

[11] L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone-beam algorithm," *J. Opt. Soc. Amer.,* vol. 1, no. 6, pp. 612–619, June 1984.

[12] P. F. C. Gilbert, "Iterative methods for the three-dimensional reconstruction of an object from projections," *J. Theor. Biol.,* vol. 72, pp. 105–117, 1972.

[13] R. Gordon, R. Bender, and G. T. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography," *J. Theo. Biol.,* vol. 29, pp. 471–481, 1970.

[14] G. N. Hounsfield, "A method and apparatus for examination of a body by radiations such as X or gamma radiation," London, Patent 1283915, 1968.

[15] G. T. Herman, *Image Reconstruction From Projections.* New York: Academic, 1980.

[16] R. Gordon and G. T. Herman, "Algebraic reconstruction from projections: A review of the algorithms," *Int. Rev. Cytol.,* vol. 38, pp. 111–151, 1974.

[17] W. F. Jones, L. G. Byars, and M. E. Casey, "Design of a super fast 3-D projection system for PET," *IEEE Trans. Nucl. Sci.,* vol. 37, pp. 800–804, Apr. 1990.

[18] L. Kaufman, "Implementing and accelerating the EM algorithm for positron emission tomography," *IEEE Trans. Med. Imaging,* vol. 6, pp. 37–51, Mar. 1987.

[19] S.-C. B. Lo, "Strip and line path integrals with a square pixel matrix: A unified theory for computational CT projections," *IEEE Trans. Med. Imaging,* vol. 7, pp. 355–363, Dec. 1988.

[20] J. M. Kumar and L. M. Patnaik, "Extended hypercube: A hierarchical interconnection network of hypercubes," *IEEE Trans. Parallel Distributed Syst.,* vol. 3, pp. 45–57, Jan. 1992.

[21] K. Rajan, L. M. Patnaik, and J. Ramakrishna, "High-speed computation of the EM algorithm for PET image reconstruction," *IEEE Trans. Nucl. Sci.,* vol. 41, pp. 1721–1728, Oct. 1994.

[22] S. W. Rowland, *Image Reconstruction From Projections, Implementation and Applications,* G. T. Herman, Ed. Berlin: Spring-Verlag, 1979.

[23] B. D. Smith, "Cone-beam tomography: Recent advances and tutorial review," *Optical Engin.,* vol. 29, no. 5, pp. 524–534, May 1990.

[24] ——, "Image reconstruction from cone-beam projections: Necessary and sufficient conditions and reconstruction methods," *IEEE Trans. Med. Imaging,* vol. 4, pp. 14–28, 1985.

[25] M. Schlindwein, "Iterative 3-D reconstruction from twin-cone beam projections," *IEEE Trans. Nucl. Sci.,* vol. 25, pp. 1135–1143, Oct. 1978.

[26] L. A. Shepp and B. F. Logan, "The Fourier reconstruction of the head section," *IEEE Trans. Nucl. Sci.,* vol. 21, pp. 21–43, June 1974.

[27] C. J. Thompson and T. M. Peters, "A fractional address accumulator for fast backprojection," *IEEE Trans. Nucl. Sci.,* vol. 28, pp. 3648–3650, Aug. 1981.

[28] G. Wang, T.-H. Lin, P.-C. Cheng, and D. M. Shinozaki, "A general cone-beam reconstruction algorithm," *IEEE Trans. Med. Imaging,* vol. 12, pp. 486–496, Sept. 1993.

[29] E. L. Zapata *et al.,* "Image reconstruction on hypercube computers," *Proc. Frontiers '90—Third Symp. Frontiers Massively Parallel Computation,* Oct. 1990, pp. 127–134.

[30] X. Xu and K. Hwang, "Modeling communication overhead: MPI and MPL performance on the IBM SP2," *IEEE Parallel Distributed Technol.,* pp. 9–23, Spring 1996.