

High-Speed Computation of the EM Algorithm for PET Image Reconstruction

K. Rajan, L. M. Patnaik, and J. Ramakrishna

Abstract—The PET image reconstruction based on the EM algorithm has several attractive advantages over the conventional convolution backprojection algorithms. However, two major drawbacks have impeded the routine use of the EM algorithm, namely, the long computational time due to slow convergence and the large memory required for the storage of the image, projection data and the probability matrix. In this study, we attempt to solve these two problems by parallelizing the EM algorithm on a multiprocessor system. We have implemented an extended hypercube (EH) architecture for the high-speed computation of the EM algorithm using the commercially available fast floating point digital signal processor (DSP) chips as the processing elements (PEs). We discuss and compare the performance of the EM algorithm on a 386 / 387 machine, CD 4360 mainframe, and on the EH system. The results show that the computational speed performance of an EH(3, 1) using DSP chips as PEs executing the EM image reconstruction algorithm is about 130 times better than that of the CD 4360 mainframe. The EH topology is expandable with more number of PEs.

I. INTRODUCTION

POSITRON emission tomography (PET) is an imaging technique, which is potentially useful in the study of human physiology and organ functions. PET aims at obtaining a quantitative map of spatial and temporal distribution of the radio-nuclides inside the human body by measuring the event counts of positron-electron annihilation. PET images can be reconstructed by either analytic methods such as convolution back projection (CBP) algorithm [13] or by iterative algorithms such as EM algorithms. An analytic algorithm usually consists of two main computations. One is filtering and the other is back projection. An iterative algorithm, on the other hand, starts with an initial guess of the solution and iteratively updates (corrects) the object according to the computed pseudo-projection and the measured projection data. The major computations in an iterative method are forward (pseudo-projection) and backward (correction) projections. The EM algorithm requires longer computation time than the CBP method. But, the EM algorithm has some advantages over the CBP algorithm. The image reconstructed using the EM algorithm has somewhat lower

noise in regions of high activity than the CBP image and is less noisy than the CBP in the regions of low activity. In addition, the EM algorithm does not require the projection data to be equally spaced.

Algorithmic improvement, dedicated hardware and parallel processing are the three general methods used to speed up the image reconstruction. Various efforts have been made to improve the algorithmic efficiency [17], [19], [7], and [13]. Most of the attempts in this category have been to reduce the number of iterations to make convergence faster. Dedicated hardware approach has been tried in [8], [9], and [20]. Jones *et al.* have suggested a VLSI based architecture to support forward and backward projection in [8]. However, the potential parallelism involved in the EM algorithm was not thoroughly used in this architecture. The approach to parallelize the image reconstruction algorithm has been attempted in [3], [4], [5], and [6]. A parallel implementation of CBP algorithm on a hypercube multiprocessor is reported in [3]. Chen *et al.* have studied the parallelization of the EM on a message passing system (Intel iPSC/2) and on a shared memory system (BBN Butterfly GP 1000) in [4]. A data and task partitioning scheme called partition-by-box is proposed in this study. Though Chen *et al.* have used the iPSC/2 hypercube system, the pseudo-binary tree embedded in the hypercube has been used for the EM algorithm. The partition-by-box scheme proposed by Chen uses the broadcast and the partial result integration algorithms. The binary tree architecture is more efficient to perform the broadcast and integration algorithms. Since a system with a complete binary tree interconnection topology was not available, they used the embedded tree on the hypercube.

The implementation of the EM algorithm on a supercomputer (CRAY-1) has been reported in [10]. The high cost/performance ratio of a supercomputer is not preferable for the EM implementations. Instead of using an expensive supercomputer, a parallel architecture consisting of multiple low cost PEs, is preferable. The high computation time of the PET image reconstruction algorithm can be brought down using a parallel implementation. The high storage space requirement can be overcome by partitioning the projection data, the image and the probability matrix. In addition, a close look at the EM reconstruction algorithm shows that most of the computation time is spent in executing *multiply* and *multiply-accumulate* type of instructions. Digital signal processors

Manuscript received January 18, 1994; revised May 9, 1994.

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, Indian Institute of Science, Bangalore 560 012, India.

J. Ramakrishna Department of Physics and Instrumentation and Services unit, Indian Institute of Science, Bangalore 560 012, India.

IEEE Log Number 9402950.

(DSPs) are optimized processors to execute fast *multiply* and *multiply-accumulate* instructions.

Along with performance, price is an important factor of a computer system. DSPs are high performance processors that can be used to build multiprocessor systems with a high-performance-to-cost ratio [16]. In this study, we have implemented a parallel architecture based on the EH topology for the PET image reconstruction based on the EM algorithm. The DSPs are used as the processing elements.

II. THE EM ALGORITHM

The EM algorithm [18], [11] 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 (complete data).

In a given voxel, the number of radioactive decays, and the resulting emission of photons can be modelled as a Poisson point process. As radio nucleids decay spontaneously, photon emissions occurring inside a given voxel are independent of emissions in other boxes. So the emission process in the complete set of boxes can be modelled as a spatially independent Poisson process with the joint probability density function given by,

$$P(X = x) = \prod_{i=1}^N e^{-\lambda[i]} \frac{(\lambda[i])^{x[i]}}{x[i]!}, \quad (2.0)$$

where $x = (x[i]; i = 1, \dots, N)$ is the emission vector. Since the mean of emission process is assumed to be proportional to the concentration of radio nucleids, an estimate of the intensity parameter vector $\lambda = (\lambda[i]; i = 1, \dots, N)$ is also an estimate of the radio nucleid concentration in the whole object, where the object is made up of N boxes. A pair of photons fly off from the point of annihilation at the speed of light in opposite directions along a line which is uniformly distributed in angle. There is an array of discrete detector elements surrounding the object and the two photons are detected in time coincidence by a pair of detector elements defining a detector unit or detector tube t . Detector unit t may be thought of as a tube defined by two detectors D1 and D15 at opposite ends of the strip $t = t(D1, D15)$ in Fig 1.

Let $y = (y[j]; j = 1, \dots, M)$ be the observed measurement data, where $y[j]$ is the number of coincidence photon observed in tube j . The data vector y is a sample observed from the measurement process. The probability for a photon emitted at a point inside a box to be detected by the measurement system depends on (1) the geometry of the system, (2) the decomposition of the object space, (3) the physical properties of the medium, and (4) the response of the detector system. Let $p(i, j)$ be the average probability that a photon emitted in box i is detected in the detector tube j , the average being taken over all points inside the box.

We study only two-dimensional, or single-ring PET and assume that there are $N = N_b \times N_b$ boxes surrounding

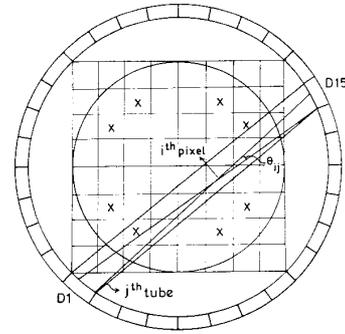


Fig. 1. The schematic diagram of the PET system.

the unit circle (the patient circle). We assume N_d discrete detectors equally spaced around the circle of radius $\sqrt{2}$ circumscribing the object boxes. We assume that the object is bounded by the unit circle. We have to reconstruct the boxes within the unit circle. The measurement system is shown in Fig. 1. A positron emission and a subsequent annihilation with an electron produces two gamma ray photons and are detected in time coincidence by a pair of detectors defining a detector tube as shown in Fig. 1. The likelihood of the observed data, i.e., the probability for observing the measured data given that the emission parameter is λ , is expressed as a function of λ given by,

$$L(\lambda) = \prod_{j=1}^{N_t} \frac{e^{-\sum_{i=1}^{N_t} \lambda[i] p(i, j)} (\sum_{i=1}^{N_t} \lambda[i] p(i, j))^{y[j]}}{y[j]!}, \quad (2a)$$

where $N_t = (N_d/2) \times ((N_d/2) + 1)$, is the number of detector tubes, $p(i, j)$ is the probability of detection of a pair of annihilation photons created in the box i , by the detector tube j , $\lambda[i]$ is the number of photon pairs emitted from box $[i]$, and $y[j]$ is the number of coincidence photons observed in tube j , and the data vector $y(y[j], j = 1, \dots, N_t)$ is a sample observed from the measurement process.

The EM algorithm is derived by maximizing the likelihood function $L(\lambda)$. A simple iterative solution for λ_i can be written as,

$$\lambda[i]^{\text{new}} = \frac{\lambda[i]^{\text{old}}}{\sum_{j=1}^{N_t} p(i, j)} \sum_{j=1}^{N_t} \frac{y[j] p(i, j)}{\sum_{i'=1}^{N_t} \lambda[i']^{\text{old}} p(i', j)}, \quad i = 1, \dots, N, \quad (2b)$$

where λ^{new} and λ^{old} are the new and old estimate of λ , respectively.

The standard EM iteration step given by (2b) can be rewritten in an additive form [12] as

$$\lambda[i]^{\text{new}} = \lambda[i]^{\text{old}} + \frac{\lambda[i]^{\text{old}}}{\sum_{j=1}^{N_t} p(i, j)} \sum_{j=1}^{N_t} \frac{y[j] - \sum_{i'=1}^{N_t} \lambda[i']^{\text{old}} p(i', j)}{\sum_{i'=1}^{N_t} \lambda[i']^{\text{old}} p(i', j)}, \quad i = 1, \dots, N. \quad (2c)$$

The equation (2c) has been implemented on the EH architecture. The EM-based MLE algorithm converges

towards a possible unique minimum, and the image obtained after convergence should be independent of the initial estimate $\lambda[i]^0$. However, if the procedure is stopped before the maximum likelihood is reached, the initial estimate can strongly influence the result. All the reconstructions carried out in this study were started with identical pixel values.

III. COMPUTATIONAL COMPLEXITY

The complexity of the EM algorithm is given in [18]. For a 128×128 square object, there are 16384 object boxes, out of which 12892 boxes are inside the object circle. The probability $p(i, j)$ that an emission in box i is detected in a tube j depends on a number of physical factors such as the geometry of the measurement system, the decomposition of the object space, the physical properties of the medium and the response of the detector system. In this study, we assume that the probability of an emission in box i and its detection in tube j depend only on the geometry of the measurement system. In such a case an annihilation event in box i is detected in a tube j with the probability $p(i, j)$ proportional to the angle of view from the center of the box i into the detector tube j . Shepp et al. have shown [18] that the choice of $p(i, j)$ based only on the geometry of the measurement system is reasonable, and that the results of the reconstruction do not depend critically on the choice of $p(i, j)$. Since there are N_t number of detector tubes and N object boxes, the dimension of probability matrix $P(i, j)$ is $N_t \times N$. For a system with 128 detectors and an object space of 128×128 , the dimension of the probability matrix $P(i, j)$ is 4160×16384 . Since only a very small percentage of the total detector tubes passes through a box, the probability matrix $P(i, j)$ is highly sparse. To minimize the storage requirement, only the nonzero $p(i, j)$ s associated with those boxes in the partition region is stored. The partition region is the region enclosed by $0 \leq x \leq N_b, 0 \leq y \leq x$, where, $N_b \times N_b$ is the size of the object. The partition region consists of about 1/8th of the object image. For circular ring geometry with the number of detectors N_d being a multiple of 4, there is an 8 fold symmetry available in the system. As shown in Fig. 2, if the angle of view of the box b marked 'x' is known, then the angle of view of all the other seven boxes marked 'x' can be obtained by the operation of a simple rotation and/or flipping. The group composed of all the eight box-tube pairs is defined as the bt-group [3] associated with the box b and the tube t passing through box b . The boxes that fall on the diagonal have only 4 box-tube pairs in a bt-group.

We precompute four one-dimensional arrays, ij, pij, kl

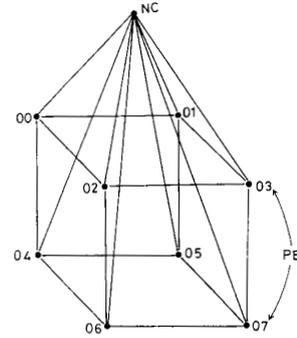


Fig. 2. The implementation of the EH (3, 1).

and pkl . The arrays ij and pij store the indices of the detector tubes passing through each of the boxes in the partition region and the corresponding box-detector tube probability values respectively. Similarly, arrays kl and pkl store the indices of the boxes that fall in each of the detector tubes and the corresponding box-tube probability values respectively.

The image reconstruction algorithm based on the EM is iterative in nature. The major computations in an iterative scheme are forward (pseudo-projection) and backward (correction) operations. From an estimate of the object, the forward algorithm finds the pseudo-projection on each detector tube t . The ratio of the difference between the projection data (actual measurement data) and the pseudo-projection data (computed value) to the pseudo-projection data, i.e., $\Delta P(t)$ gives the extent of resemblance between projection of the object and the projection of the reconstructed object. Based on ΔP , corrections are made to the initial guess of the object. In the correction operation, ΔP in the projection domain is back-projected onto the object domain.

There are three possible data and task partitioning schemes [4] to solve the image reconstruction problem based on the EM algorithm. The first approach, the *partition-by-box scheme*, is based on computing the pseudo-projection using the emissions from the boxes in the partition region. This scheme uses the two 1-D arrays ij and pij to compute the pseudo-projection and ΔP . Based on ΔP , the correction to the emission intensities of the boxes in the partition region are computed. Since the box-detector tube probabilities of all the boxes in a bt-group are known, the emission intensities of all the boxes in each of the bt-groups are updated. The algorithm based on *partition-by-box scheme* is given below.

Image reconstruction algorithm based on partition-by-box scheme

```
{
  for each box i in the partition region
    get the indices of the tubes j that pass through box i from array ij
    get the corresponding box-detector probability p(i, j) from array pij
    partial pseudo-projection on tube j, Δpseudo-projection[j] = λ[i] × p(i, j)
```

```

for each box  $b$  in the same bt-group
  get the indices of the tubes  $j$  that pass through box  $b$  from array  $ij$ 
  get the corresponding box-detector probability  $p(i, j)$  from array
   $p_{ij} \Delta \text{pseudo-projection}[j] = \sum \lambda[b] \times p(i, j)$ 
}
for each tube  $j$  /* $\Delta P$  computation*/
   $\Delta P[j] = \frac{(\text{projectionontube}j - \text{pseudo-projection}j)}{\text{pseudo} - \text{projection}j}$ 
for each box  $i$  in the partition region /*correction phase*/
{
  partial correction to  $\lambda[i], \Delta \lambda[i] = \sum_j (\Delta P[j] \times p(i, j))$ 
  for each box  $m$  in the same bt-group as box  $i$ 
     $\Delta \lambda[m] = \sum_j (\Delta P[j] \times p(m, j))$ 
}
for all the boxes  $i$  in the object /*image update*/
   $\lambda[i] = \lambda[i] \times \lambda[i] \times \Delta \lambda[i]$ 
}

```

The second approach, *partition-by-tube scheme* uses the detector tube as a subpartition. This scheme computes the pseudo-projection and correction to the emission intensities of the boxes using the two 1-D arrays kl and pkl . The algorithm is given below.

Image reconstruction algorithm based on partition-by-tube

```

{
  for each detector tube  $t$ 
    get the indices of the boxes  $i$  that are within the tube  $t$  from array  $kl$ 
    get the corresponding probability from array  $pkl$ 
    pseudo-projection on tube  $t$ ,  $\sum_i (\lambda[i] \times p(i, t))$ 
    for each detector tube  $t$  /* $\Delta P$  computation phase*/
       $\Delta P[t] = \frac{\text{projection}[t] - \text{pseudo-projection}[t]}{\text{pseudo-projection}[t]}$ 
    for each detector tube  $t$  /*correction phase*/
      get the indices of the boxes  $i$  that are within the tube  $t$ 
      get the corresponding probability values  $p(i, t)$ 
       $\Delta \lambda[i] = \Delta P[t] \times p(i, t)$ 
    for each box  $i$  /*image update*/
       $\lambda[i] = \lambda[i] + \Delta \lambda[i] \times \lambda[i]$ 
}

```

The third approach *partition-by-tube and box scheme* uses the *partition-by-tube scheme* to compute the pseudo-projection and *partition-by-box scheme* to compute the back-projection. Hence this scheme uses all the four one dimensional arrays, ij , p_{ij} , kl , and pkl .

The pseudo-projection and the back-projection algorithms consist of a series of *multiply* and *multiply/accumulate* instructions. Since DSP chips are optimized processors for executing *multiply* and *multiply/accumulate* instructions, DSP devices are ideal processing elements for the image reconstruction tasks.

A stopping rule for iterative EM image reconstruction algorithms

The objective of the stopping rule is to pick the image just before the estimates begin to turn noisy. For the EM iterations, this turning point is dependent upon both the number of photon counts in the measurement data and

the nature of the object distribution itself. Hence, it is inappropriate to predetermine the number of iterations required for obtaining the desired image.

The stopping rule proposed in [21] is based on a statistical approach, where after each iteration of the MLE algorithm, the estimate is accepted or rejected as the final image based on the result of a statistical hypothesis test. The measurement data in emission tomography are con-

sidered to be samples from independent Poisson random variables whose means are given by $\psi_j = \sum_{i=1}^N \lambda[i] p(i, j)$, $j = 1, \dots, N_t$, where $\{\lambda[i], i = 1, \dots, N\}$ are emission densities in the object. This assumption is the basis for the hypothesis test for stopping the iterations. After each iteration of the MLE algorithm, we propose to test that the observed measurement data $y(j)$, $j = 1, \dots, N_t$, are samples from Poisson random variables with means $\psi_j^k = \sum_{i=1}^N \lambda[i]^k p(i, j)$, $j = 1, \dots, N_t$, where $\{\lambda[i]^k, i = 1, \dots, N\}$ is the estimate after k iterations.

We divide the possible values for y into a finite number (S) of sets of equal probabilities. Since $y(j)$ is a sample from Poisson distribution function, $P(y, \psi_j)$, $y(j)$ has equal probabilities of being in any of the S sets. In the test procedure, we first generate a histogram, $\mathbf{h} = \{h_i, i = 1, \dots, S\}$ corresponding to the S sets. This histogram is

obtained by classifying each of the N_i measurement data into one of the S sets. Since each measurement data has a uniform probability for being in one of the S sets, we can test the hypothesis that the measurement data are samples from Poisson distribution by testing the uniformity of the histogram \mathbf{h} . To test if the histogram \mathbf{h} is uniformly distributed, we use the Pearson procedure. If \mathbf{h} is uniformly distributed, then

$$H = \sum_{i=1}^S \frac{h_i - N_i/S}{N_i/S}$$

is approximately X^2 distributed with $(S - 1)$ degrees of freedom. Therefore, the hypothesis is rejected if the value of H exceeds a critical value. In our experiment, we used 20 different sets (i.e., $S = 20$) and classified each measurement data to belong to one of these 20 sets.

IV. EXTENDED HYPERCUBE

The extended hypercube (EH) [14] architecture is suited for hierarchical expansion of multiprocessor systems. The EH is built using basic modules consisting of a k-cube of processor elements (PE's) and an additional node for handling communication – the Network Controller (NC). The NC is used as a communication processor to handle intermodule communication; 2^k such basic modules are interconnected via 2^k NC's, forming a k-cube among the NCs. The EH is essentially a truly expansive, recursive structure with a constant predefined building block. The number of I/O ports for each PE/NC is fixed and is independent of the size of the network. The EH structure is shown to be most suitable for implementing a class of highly parallel algorithms, *viz.*, the DESCEND and ASCEND. The EH can emulate the binary hypercube in implementing a large class of algorithms with insignificant degradation in performance. The EH architecture retains the positive features of 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.

A. Message routing in the EH

Using the NC, the interprocessor message traffic of a module gets redistributed into two categories, *viz.*, local communication and global communication. The communication among the PEs belonging to the same k-cube is called local communication and is handled by employing the hypercube message passing algorithms. The NCs are involved in global communication. The algorithm is given in [14].

B. Single node broadcast in the EH

In single node broadcasting, the message packet from a single node is transmitted to all the other nodes. The utilization factor of the PEs in the EH for the single node broadcast algorithm is shown [15] to be 1.75 times higher than that of the hypercube.

C. Multinode broadcast in the EH

The multinode broadcast problem involves message transfer from each of the nodes to all other nodes. The utilization factor of the PEs in the EH for the multinode broadcast algorithm is 1.75 times higher than that of the hypercube.

The EH architecture also supports fast integration of partial results. The algorithm for the integration of partial results is given in Section VI. The fast execution of single node broadcast and multinode broadcast makes the EH architecture ideal for execution of the image reconstruction algorithms.

V. PARALLELIZATION OF THE EM ALGORITHM

The image reconstruction algorithm based on EM is iterative in nature. It has a forward step followed by a backward correction step in each iteration. After each iteration, the convergence of the solution has to be estimated. The forward step and the back-projection step can be efficiently parallelized in data space. That is, in an iteration step, each processor element computes a part of the output data, without communicating with the other processors. After the forward step, data distribution takes place. The EM algorithms are given in Section III.

A. Description of the Parallel Algorithm

Partition-by-box scheme: The EM algorithm based on *partition-by-box scheme* is given in Section III. The NC at the topmost level partitions the data corresponding to the boxes of the partition region into 2^k number of blocks and sends one block each to the nodes at the $(l - 1)$ st level. The nodes at the $(l - 1)$ st level may be NCs or PEs. If they are NCs, each NC at level $(l - 1)$, subpartitions its block into 2^k number of smaller blocks. This kind of subpartition is continued by NCs at all levels j ($j > 0$). Finally, each PE at level 0 receives a small block of the boxes of the partition region. The indices of the tubes j that pass through the box i and the corresponding box-tube probability $p(i, j)$ are the data received by the PE at level 0. The host broadcasts the emission intensities of the boxes λ to all PEs.

The PEs at level 0 compute Δ pseudo-projection (the partial contribution to the pseudo-projection from boxes in the subpartition) and execute the integration of partial results using the algorithm given in section 6 and send the results to the NC at level 1. The NCs at level 1 integrate the Δ pseudo-projections received from the PEs at level 0 and send the integrated Δ pseudo-projections to the NCs at level 2, and finally the NC at the topmost level integrates the Δ pseudo-projections. The topmost level NC computes the ΔP and broadcasts ΔP to all PEs. The PEs at level 0 compute the corrections to the emission intensities of the boxes ($\Delta\lambda[i]$) in the subpartition and update the emission intensities of the boxes $\lambda[i]$. The updated emission intensities of the boxes are sent to the NC at the topmost level. While PEs at level 0 execute the correction algorithm, the NC at the topmost level computes the \mathbf{H}

value and decides whether to go ahead with the next iteration.

Partition-by-tube scheme: The algorithm is given in Section III. The NC at the topmost level partitions the data corresponding to N_i tubes 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 subblocks, and sends one subblock each to PEs at $(l - 2)$ nd level. Finally, each PE at level 0 receives data corresponding to a small block of detector tubes. The indices of the boxes i within a tube j and the corresponding box-tube probability $p(i, j)$ are the data received by a PE at level 0. The host broadcasts the emission intensities of the boxes λ and the $\Delta\lambda$ to all PEs.

Each PE at level 0 computes the pseudo-projection and ΔP for the tubes in the subpartition. Using ΔP of the tubes in the subpartition, each PE at level 0 computes $\Delta\lambda$ (the partial correction to the emission intensities of the boxes in the tube). The PEs at level 0 integrates the partial $\Delta\lambda$ using the algorithm given in section 6 and send $\Delta\lambda$ to the NCs at level 1. The NCs at level 1 integrate the $\Delta\lambda$ it receive from PEs at level 0 and send the partial result to NCs at level 2. This process is continued till the NC at the topmost level has the $\Delta\lambda$. The NC at the topmost level updates the emission intensities of the boxes using $\Delta\lambda$ and also computes the H value to decide whether to go ahead with the next iteration.

Partition-by-tube-and-box: The forward step is based on *partition-by-tube scheme*. The algorithm computes ΔP . The multinode broadcast property of the EH [15] is used to send ΔP from each of the nodes to all other nodes. The back projection step is based on *partition-by-box scheme* and updates the emission intensities of the image boxes.

VI. IMPLEMENTATION OF THE EH(3, 1)

An EH(3, 1) is has been implemented using ADSP 21020 [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 PEs through direct links.

A typical DSP such as ADSP 21020 has three independent computation units: an arithmetic and logic unit (ALU), a multiplier with fixed-point accumulator (MAC), and a shifter. The three units are connected in parallel and the output of any one unit can be the input to any unit on the next clock cycle. A 10-port register file is used

for transferring data among computation units and data buses, and for storing immediate results.

ADSP 21020 has two independent memories – one for *data* and the other for *program instructions and data*. A program sequencer with a 32-word instruction cache allows the ADSP 21020 to access data from both *data memory* and *program memory* and fetch an instruction – all three operations in the same clock cycle.

In the pseudo-projection (forward) algorithm, the λ of the boxes and the corresponding $p(i, j)$ values are stored in the *program memory* (PM) and *data memory* (DM), respectively. The pseudo-projection algorithm does the following four operations in one clock cycle: 1. read the λ value of a box from the program memory, 2. read the corresponding box-tube probability value from the data memory, 3. multiply the λ value with the probability value, and 4. update the address generators to point to the next box and the probability values.

The back-projection algorithm does the following four operations in one clock period: 1. read ΔP value of a box from *program memory*, 2. read the corresponding $p(i, j)$ value from *data memory*, 3. Multiply ΔP with $p(i, j)$ and 4. update the address generators to point to the next ΔP and $p(i, j)$ values. The multiple functional units inside the DSP chip allow the above operations to be done in a single clock cycle.

The PE to PE/NC link is created through a message channel generated out of dual-port RAM (DPR) [2]. The implementation of the EH(3, 1) is shown in Fig. 2. The size of the message channel is 1 KWord. The DPR is configured in the memory map of the PEs. The DPR allows accessing of every word in the memory array from two independent sources. The message transfer is basically a memory-to-memory transfer. The message is formatted with the source address, destination address, and the size of the message. The message transfer is implemented as a blocked synchronous transfer.

The NC is connected to a PC/386 system to use the resources such as memory, keyboard, display and the disk storage. The ADSP 21020 operates at 33 MHz with a cycle time of 33 ns. The fast DSP memories are made up of 20 ns static memories.

The EH with memory channel linking the PEs/NCs efficiently executes the integration of partial results. An algorithm to integrate the partial results is given below.

Integration of partial results

```
{
  for i = 0 to (l - 1) step 1
  {
    for j = 0 to (k - 1) step 1
    {
      for all PEs/NCs at level i in parallel
      if  $bit_j(m) = 1$ 
      send partial result to the channel linking PE[m]/NC[m] and PE[m - 1]/NC[m - 1]
```

TABLE I
EXECUTION TIME FOR ONE ITERATION OF
THE EM ALGORITHM (PARTITION-BY-TUBE)

Image size	Single node of an EH	EH EH(3, 1)	386/387	CD 4360
16*16	1508 μ s	244 μ s	113.36 ms	31.61 ms
32*32	11.41 ms	1.76 ms	867 ms	242 ms
64*64	93.05 ms	13.79 ms	7.30 s	1.95 s
128*128	712.42 ms	104.39 ms	62.85 s	13.10 s

TABLE II
EXECUTION TIME FOR ONE ITERATION OF THE
EM ALGORITHM (PARTITION-BY-BOX)

Image size	Single node of an EH	EH (3, 1)	386/387	CD 4360
16*16	1695 μ s	269 μ s	120 ms	23.71 ms
32*32	12.61 ms	1.91 ms	894 ms	176 ms
64*64	97.81 ms	14.40 ms	7.35 s	1.40 s
128*128	771.76 ms	111.93 ms	62.38 s	11.14 s

TABLE III
EXECUTION TIME FOR ONE ITERATION OF THE
EM ALGORITHM (PARTITION-BY-TUBE-AND-BOX)

Image size	Single node of an EH	EH (3, 1)	386/387	CD 4360
16*16	1547 μ s	261 μ s	85.09 ms	20.38 ms
32*32	11.60 ms	1.94 ms	655.42 ms	157.91 ms
64*64	90.62 ms	14.92 ms	5.61 s	1.28 s
128*128	715.80 ms	118.20 ms	48.12 s	10.59 s

for all PEs/NCs at level i in parallel
if $bit_i(m) + 0$

$$T(m) = T(m) + T(m + 1) \quad /* T(m) \text{ is the partial result in PE}[m]/\text{NC}[m]*/$$

}
OUT($NC_{i+1}\{PE/NC\}_i$) /* THE NC/PE at level i to NC at level $i + 1$ */
}

The computation is first done in all EH($k, 0$)s in parallel, and the results are transferred to next higher level EH($k, 1$) for further computation. This process is continued until the NC at the top level gets the result. The algorithm is executed in $O(k^*1)$ steps.

The Tables I, II, and III give the execution times for one iteration of the EM algorithm on a single node of the EH, on an EH(3, 1) system, on a 386/387 running at 40 MHz under Unix OS and on a CD4360 computer system for the *partition-by-tube*, *partition-by-box* and *partition-by-tube-and-box* schemes. The execution times for the EH have been calculated, assuming that the DSPs run at 33 MHz with zero 'wait' states. It is assumed that the DPRs also operate at 33 MHz. The execution times given for the EH include the communication overhead. The communication overhead and the storage requirement for the *partition-by-tube* and the *partition-by-box* schemes are roughly the same. The communication overhead for the *partition-by-tube-and-box* scheme is twice that of the other two schemes. The *partition-by-tube* scheme gives the lowest execution time.

The results show that the computation speed of an EH(3, 1) system for the EM image reconstruction algo-

rithm is 130 times faster than that of a CD 4360 computer system. Compared to a 386/387 Unix based system running at 40 MHz, the EH system gives 600 times better speed performance.

VII. CONCLUSIONS

In this paper, we have described the parallelization of the EM algorithm on an extended hypercube topology. Some of the features of the EH topology, such as single node broadcast, multinode broadcast and integration of partial results, speed up the execution of the EM algorithm on an EH. The EM algorithm breaks down to a sequence of *multiply* and *multiply/accumulate* type of instructions. Since DSP chips are optimized processors for executing *multiply* and *multiply/accumulate* instructions, they give high performance. Another factor that speeds up the EM algorithm execution is the DPR. The NCs and PEs are linked using high-speed DPRs. The computation and communication tasks overlap in the EH system because of the DPR. It has been found that the computational speed performance of an EH(3, 1) executing the EM image reconstruction algorithm is 130 times better than that of CD 4360 computer system.

REFERENCES

- [1] Analog Devices's Users Manual, ADSP 21020, 1992.
- [2] *Advanced Micro Devices Statis Memories Handbook*, 1990.
- [3] C. M. Chen, and S. Y. Lee, "A parallel implementation of 3-D CT image reconstruction on Hypercube multi processor" *IEEE Trans. Nucl. Sci.*, vol. 37, no. 3, June 1990, pp. 1333-1346.
- [4] —, 'Parallelization of the EM Algorithm for 3-D PET image reconstruction' *IEEE Trans. Med. Imag.*, vol. 10, no. 4, pp. 513-522, Dec. 1991.
- [5] Eric Shieh *et al.* 'High-speed computation of the radon transform and back projection using an expandable multi processor architecture' *IEEE Trans. Circ. Syst. Video Technol.*, vol. 2, no. 4, pp. 347-359, Dec. 1992.
- [6] R. Hartzz, D. Bristow, and N. Mullani, 'A real-time TOFPET slice back projection engine employing dual AM 29116 microprocessors' *IEEE Trans. Nucl. Sci.*, vol. 32, no. 1, pp. 839-842, Feb. 1985.
- [7] T. Hebert and R. Leahy, "Fast methods for incorporating attenuation in the EM algorithm," *IEEE Trans. on Nucl. Sci.*, vol. 37, no. 2, pp. 754-758, 1990.
- [8] W. F. Jones, L. G. Byars, and M. E. Casey, "Design of a super fast three-dimensional projection system for positron emission tomography" *IEEE Trans. Nucl. Sci.*, vol. 35, no. 2, pp. 800-804, Apr. 1990.
- [9] W. F. Jones, L. G. Byars, and M. E. Casey, "Positron emission tomographic and expectation maximization: A VLSI architecture for multiple iterations per second" *IEEE Trans Nucl. Sci.*, vol. 35, no. 1, pp. 620-624, Feb. 1988.
- [10] L. Kaufman, "Implementing and accelerating the EM algorithm for PET" *IEEE Trans. on Med. Imaging.*, vol. 6, pp. 37-50, Mar. 1987.
- [11] K. Lange, and R. Carson, "EM reconstruction algorithm for emission and transmission tomography" *J. Comp. Assisted Tomog.*, vol. 8, pp. 306-316, 1984.
- [12] K. Lange, M. Bahn, and R. Little, "A theoretical study of some maximum likelihood algorithm for emission and transmission tomography" *IEEE Trans. Med. Imag.*, vol. MI-6, no. 2, pp. 106-114, 1987.

- [13] R. M. Lewitt, and G. Muehlechner, "Accelerated iterative reconstruction for positron emission tomography based on EM algorithm for maximum likelihood estimation" *IEEE Trans. Med. Imag.*, vol. MI-5, no. 1, pp. 16-22, 1986.
- [14] J. Mohan Kumar, and L. M. Patnaik, "Extended hypercube: a hierarchical interconnection network of hypercubes" *IEEE Trans. Parallel Distrib. Syst.* vol. 3, no. 1, pp. 45-57, Jan. 1992.
- [15] J. Mohan Kumar, and L. M. Patnaik, and D. Prasad, 'A transputer-based extended hypercube' *Microprocessing and Microprogramming.*, vol. 29, pp. 225-236, 1990.
- [16] U. A. Muller, *et al.* "Achieving Super computer performance for neural net simulation with an array of digital signal processors" *IEEE Micro.*, pp. 55-64, Oct. 1992.
- [17] N. Rajeev, K. Rajagopal, and G. Krishna, "Vector-extrapolated fast maximum likelihood estimation algorithms for emission tomography" *IEEE Trans. on Med. Imag.*, vol. 11, no. 1, pp. 9-20, Mar. 1992.
- [18] L. A. Shepp, and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE Trans. Med. Imag.*, vol. MI-1, no. 2, pp. 113-121, Oct. 1992.
- [19] M. Samuel, and B. Ivan, "A high speed reconstruction from projections using direct fourier method with optimized parameters-A'n experimental analysis," *IEEE Trans. on Med. Imag.*, vol. 9, no. 4, pp. 421-429, Dec. 1990.
- [20] C. J. Thompson, and T. M. Peters, "A fractional address accumulator for fast back projection," *IEEE Trans. Nucl. Sci.*, vol. 28, no. 4, pp. 3648-3650, Aug. 1981.
- [21] E. Veklerov and J. Llacer, "Stopping rule for the MLE algorithm based on statistical hypothesis testing" *IEEE Trans. Med. Imag.*, vol. 6, pp. 313-319, 1987.