

分散メモリ型マルチプロセッサシステムを用いた 三次元 X 線 CT 像の再構成

佐々木 徹[†] 福田 安志[†]

我々は、撮影オブジェクトにコーンビーム状の X 線を照射することによって得られた二次元プロジェクション像から、フェルドカンブの方法により三次元ボクセルボリュームを高速に再構成するシステムを開発した。三次元 CT の場合には従来の二次元 CT に比べて次元数が 1 つ大きいため、データ量や演算量が膨大となり、現在のワークステーションで再構成すると 1~2 時間費やしてしまう。そこで、今回我々は再構成プログラムが並列化しやすい構造を持っていることに着目し、分散メモリ型マルチプロセッサシステムを使用することにより処理時間の短縮を図った。アプリケーションプログラムを移植し、処理時間を実測してみたところ、32 PE 構成の場合が最もバランスがとれており、この構成で 256^3 ボクセルの三次元 CT 像が約 6 分半で再構成できることが確認できた。ハードウェアは要素プロセッサに市販の DSP を用い、システムバスに標準 VME バスをを用いるなど、極力コストを抑えた設計となっているが、このような短期間に作製できる安価なマルチプロセッサシステムでもプログラムの並列化手法等、並列処理の技術を適切に応用することによりワークステーションの 10~20 倍の性能を達成することができた。

Reconstruction of 3-D Xray Computerized Tomography Images Using a Distributed Memory Multiprocessor System

TOHRU SASAKI[†] and YASUSHI FUKUDA[†]

We developed the system that can make fast reconstruction of 3-D voxel volume from 2-D cone beam Xray projection images using Feldkamp's method. 3-D CT machine requires much higher performance than conventional 2-D CT machines because 3-D CT machine must process much larger data, and current workstations takes 1-2 hours for 3-D reconstruction. Therefore we developed the distributed memory multiprocessor system to reduce processing time because CT application program has the good structure for parallel processing. The result from our measured reconstructing time show it takes about 6.5 minutes to reconstruct 256^3 3-D voxel volume on a 32-processor element machine. The hardware of our system is low-cost designed using DSPs as processor elements and standard VME bus as system bus. But we applied some techniques for parallel processing to our multiprocessor system and we could achieve 10-20 times as high performance as some current workstations.

1. はじめに

従来、X 線 CT は二次元断層像を生成するものであったが、近年、三次元立体像を生成するものが主流になってきている。現在三次元 CT の方式としては、ヘリカルスキャン（スパイラル）方式が一般的であるが、今回、我々は図 1 に示すコーンビーム方式を採用した。

ヘリカルスキャン CT が二次元プロジェクション像から一定ピッチの二次元 CT 像を多数再構成し、それらを補間して三次元像を生成する方式であるのに対し、

コーンビーム CT は二次元プロジェクション像から直接三次元ボクセルボリュームを再構成するという方式である。

コーンビーム CT は複雑な補間を行うことなく三次元 CT 像が得られるという長所を持つ反面、二次元 CT やヘリカルスキャン CT と比較して計算のオーダーが 1 つ大きくなってしまったため、再構成を行う際の処理すべきデータ量や計算量が膨大になってしまうという問題点もある。

実際に装置を運用する際に被験者を X 線撮像装置に搬入/搬出するのに 10 分程度かかるので、その間に再構成できることが望ましい。また、装置の分解能として最低 256^3 程度のボクセル数が必要なので、 256^3 ボクセルを 10 分以内で再構成する必要がある。とこ

[†] 株式会社ソニー木原研究所
Sony Kihara Research Center, Inc

ろが, Feldkamp¹⁾が提案した方法に基づいて現在のワークステーションを使用して 256³ ボクセルを再構成すると 1~2 時間費やしてしまい実用には耐えられない。

そこで, 我々は再構成プログラムが並列化しやすい構造を持っていることに着目し, 汎用 DSP をプロセッサとする VME バスベースの分散メモリ型マルチプロセッサシステムを採用することにより処理時間の短縮を図った。図 2 に三次元 CT 装置と, 図 3 に再構成装置の概要をそれぞれ示す。

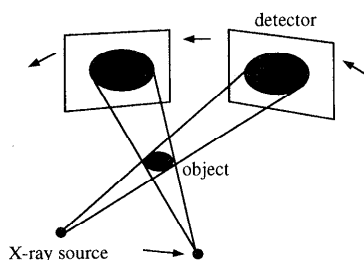


図 1 コーンビーム CT の概念
Fig.1 Conebeam CT.

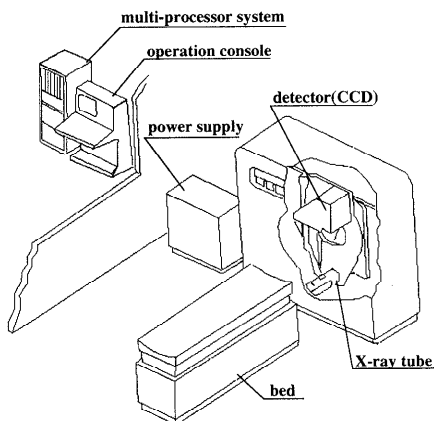


図 2 三次元 CT 装置の概要
Fig.2 3D CT system.

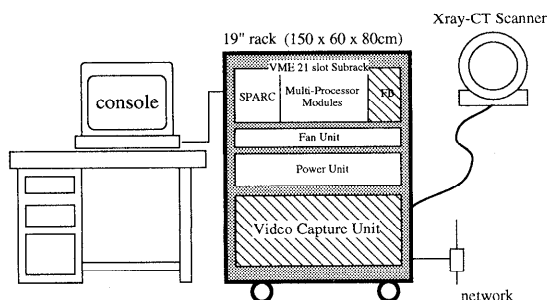


図 3 再構成装置の概要
Fig.3 Reconstruction system.

以下本稿では主にマルチプロセッサシステムによる三次元 CT 像の再構成について述べる。CT 装置全体に関しては文献 2)~4) 等を参照されたい。

2. 再構成装置のアーキテクチャ

2.1 ハードウェアの構成

図 4 に再構成装置の外観, 図 5 にその構成図を示す。

再構成装置のハードウェアは

- (1) ホストワークステーション
- (2) マルチプロセッサシステム
- (3) 画像入力部

から構成され, 各モジュールは VME バスによって接続されている。また, ハードウェアの設計に際して汎用 DSP や標準 VME バスといった部品を使用して極力コストを抑えるよう努力した。

2.2 ホストワークステーション

ホストワークステーションとして, SPARC ボードコンピュータにハードディスクやディスプレイやネットワーク I/O を接続したものを使用している。GUI の開発や IO 等は SPARC ステーションと同一の環境が利用できる。

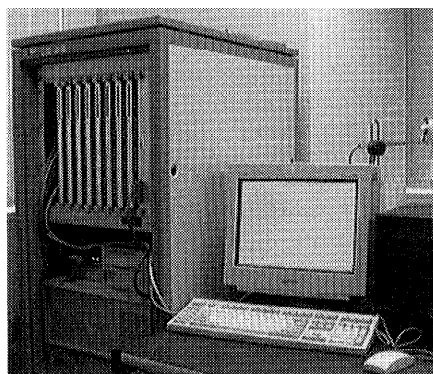


図 4 再構成装置の外観
Fig.4 Photograph of the reconstruction system.

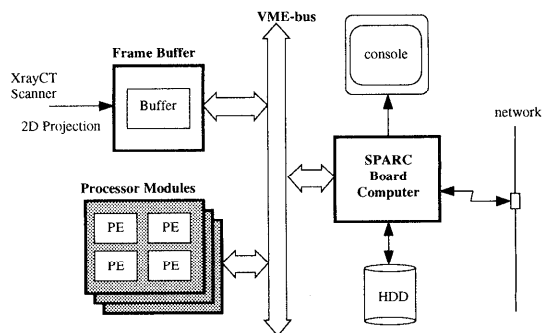


図 5 再構成装置のシステム構成
Fig.5 System configuration of the reconstruction system.

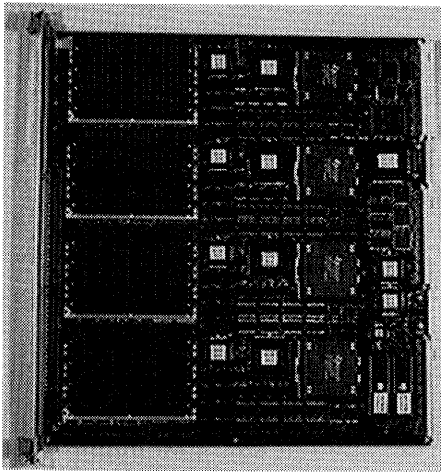


図6 マルチプロセッサボード

Fig. 6 Photograph of the multiprocessor board.

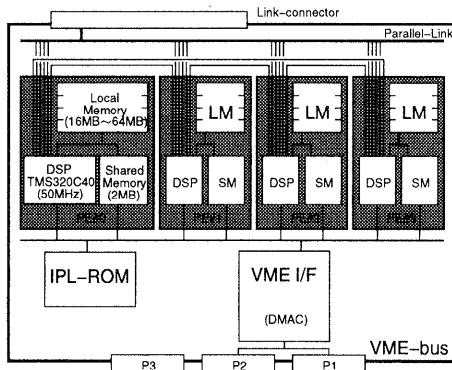


図7 マルチプロセッサボードのブロック図

Fig. 7 Blockdiagram of the multiprocessor board.

2.3 マルチプロセッサシステム

テキサスインスツルメント社製の DSP TMS320C40 をプロセッサとして使用したマルチプロセッサの演算アクセラレータである。TMS320C40 は最大 64 個まで実装することができ、この場合のピーク性能は 3.2 GFLOPS である。

2.3.1 マルチプロセッサボードの構成

マルチプロセッサボードはユーロ規格 9U×340 mm のボードサイズの基板上に 4 つの PE を搭載している。図 6 にマルチプロセッサボード、図 7 にそのブロック図を示す。

PE の諸元は表 1 に示す。扱うデータサイズが大きいことを考慮して、ローカルメモリや共有メモリのサイズを大きくとっている。

VME インタフェースには Cypress 社製の VIC068 を使用しており、VME バス上の他のデバイスと共有メモリとの間のバースト転送が可能である。また、

表1 プロセッサエレメントの諸元

Table 1 Organization of processor element.

プロセッサ	TMS320C40 (@50 MHz)
演算性能	50 MFLOPS at peak
ローカルメモリ	16~64 MB
共有メモリ	2 MB (1 MB × 2 Bank)

VIC068 に内蔵されている DMA コントローラは DSP 上のプログラムから利用することができる。

アーキテクチャ上、工夫した点は次の 2 点である。

- 共有メモリに対するブロードキャスト転送

標準 VME バスをシステムバスとしているため、システムバス上のブロードキャスト転送が行えないので、同一ボード上の 4 つの共有メモリに対するブロードキャスト転送によってそれを補っている。

- 共有メモリのダブルバッファ

2 ポートの共有メモリを 2 バンク構成にし、ダブルバッファリングを可能にした。

これらの効果については後述する。

なお、本マルチプロセッサボードは Verilog シミュレーションにより十分設計検証を行ったので、設計/製作は約 6 カ月という比較的短期間で終了することができた。

2.3.2 プロセッサ間リンク

TMS320C40 はプロセッサあたり 8 ビットのプロセッサ間リンクを 6 本具備している。本マルチプロセッサシステムでは同一ボード上の 4 つのプロセッサを正方形に接続し、それ以外のリンクは専用ケーブルを使ってフロントパネルで接続することができるので、6 次元以下のハイパキューブ構成や 3 次元トラス等の接続も可能であるが、本アプリケーションではコストを考慮して同一ボード上のリンクのみ使用し、ボード間のプロセッサ間リンクは使用していない。

2.4 画像入力部

X 線撮像装置は 6~12 秒で 360° 分のデータを撮影してしまうため、X 線撮像装置から送られてくるプロジェクションデータをいったんこの部分でバッファしている。バッファのメモリ容量は 512 Mbyte まで実装できるので、ワーク領域として中間結果や生成した 3 次元ボクセル格納にも使用している。

2.5 VME Bus

再構成装置のシステムバスは標準 VME32 の仕様に準拠している。

2.5.1 アドレス/データ転送

本システムは主に VME Bus 標準の A32 D32 を使ってデータ転送を行っており、マルチプロセッサボード

のソフトウェアリセットにのみ、A16 D8 を使用している。

2.5.2 アービトレーション

アービトレーションのプライオリティはホストが DSP ボードより高く、各 DSP ボードは公平にアービトレーションされる。

2.5.3 インタラプト

各 PE からホストへのインタラプトのみ VME インタラプトを使用している。逆にホストから各 PE へのインタラプトには VME バスのインタラプト機能は使用せず、各 PE の CSR へのアクセスによって実現している。

2.6 VME バスのデータ転送能力

演算能力とデータ転送能力のバランスを評価するためのデータとして、本システムの VME バスを介したデータの転送能力を各デバイスごとに実測してみた。結果を表 2 に示す。なお、計測はマスタボード上のプロセッサが行った。

表中以下の点に注意。

- FBM はフレームバッファメモリ、L.M はローカルメモリ、Sh.M は共有メモリをそれぞれ表している。
- フレームバッファメモリはバースト転送をサポートしていないので、フレームバッファメモリがスレーブの場合は、シングルデータ転送である。
- ホストがマスタの場合、SPARC ボード上の DMAC が実際の転送を行う。バースト長は

32 byte である。

- DMAC がマスタであるというのは DSP ボード上の VME I/F チップの DMAC がスレーブと DSP ボードの共有メモリとの間の転送を行うということを指す。バースト長は 65280 byte (64 Kbyte - 256 byte) である。
- DSP がマスタであるというのは DSP が VME バスに直接アクセスするということを指す。
- PE がスレーブであるというのは PE の共有メモリに対するアクセスを指す。
- WB は write 動作時の VME 上のマスタ/スレーブの Write Buffering である。

ボードの種類によっては、Write Buffering やブロック転送をサポートしていないものもあるため、細かい定量的な解析は難しいが、実測値からだいたい次のような点が確認できた。

- 21 スロットの VME バスを使用しているため、あまり大きな転送能力は期待できないが、バースト転送や Write Buffering 等を組み合わせると 7~9 MB/sec 程度の転送能力は得られる。
- VME I/F Chip の Write Buffering の効果のため Read サイクルよりは Write サイクルのほうが高速に実行される。

比較のため PE 内部の LBus を介した DSP とローカルメモリとの転送能力も測定したところ、VME バス上の転送との速度比は 5~10 倍であった。また、使用した DSP の出力する制御信号が write サイクルより read サイクルを優先する設計になっているため、回路設計上 write サイクルのほうが read サイクルよりも 1 wait 多くなってしまい、DSP がローカルメモリをアクセスする場合には、read サイクルのほうが write サイクルよりも高速である。

2.7 ハードウェアの実装

21 スロットの VME 19 inch 9 U ラックに SPARC ボードと DSP マルチプロセッサボード、画像入力部のフレームバッファメモリを実装している。画像入力部の X 線撮像装置とのインタフェースボードは別のラックに実装されており、ケーブルを介してデータをフレームバッファメモリに送っている。

2.8 ソフトウェアの開発環境

ソフトウェア構成を図 8 に示す。従来 DSP のマルチプロセッサシステムは、ソフトウェアの開発環境が問題になることが多いが、今回我々は、各プロセッサ上に SPECTRON Micro Systems 社のマルチタスクリアルタイム OS SPOX を移植してソフトウェアの開発環境を改善した。それにより、DSP 上のアプリケ-

表 2 データ転送能力 (実測値)

Table 2 Measured data transfer rate.

VME バスの転送能力					
Master	Slave	R/W	B/S	WB	Rate
Host	FBM	R	S	-/-	4.12 MB/s
Host	FBM	W	S	M/-	4.46 MB/s
Host	PE	R	B	-/-	5.59 MB/s
Host	PE	W	B	M/S	7.35 MB/s
DMAC	FBM	R	S	-/-	4.74 MB/s
DMAC	FBM	W	S	M/-	4.05 MB/s
DMAC	PE	R	B	-/-	9.17 MB/s
DMAC	PE	W	B	M/S	9.35 MB/s
DSP	FBM	R	S	-/-	4.69 MB/s
DSP	FBM	W	S	M/-	7.55 MB/s
DSP	PE	R	S	-/-	3.86 MB/s
DSP	PE	W	S	M/S	9.54 MB/s

LBUS の転送能力				
Master	Slave	R/W	B/S	Rate
DSP	L.M	R	S	47.7 MB/s
DSP	L.M	W	S	31.8 MB/s
DSP	Sh.M	R	S	47.7 MB/s
DSP	Sh.M	W	S	31.8 MB/s

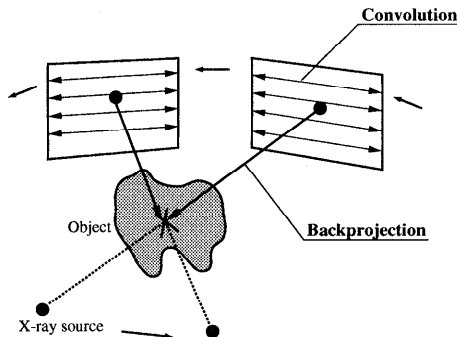


図 11 再構成の概念

Fig. 11 Concept of voxel reconstruction.

- N_x はプロジェクションデータの x_d 方向のピクセル数
- Δx_d は 1 ピクセルの x_d 方向のサイズをそれぞれ表している。

以上まとめると、図 11 に示すように再構成の処理は式 (2) に対応する二次元プロジェクション像に対する一次元コンボリューションと式 (1) に対応する各ボクセルへのバックプロジェクションから構成されるということが出来る。

3.1 コンボリューション

コンボリューションは個々の二次元プロジェクション像に対して回転軸と直交する x_d 方向に一次元のコンボリューションを行うという処理である。我々の装置仕様では撮影フレームのピクセルがもともと 10 bit の精度しか持たないため、全領域にわたってコンボリューションしなくても、Shepp-Logan 関数が、

$$|g(0, a)/g(n, a)| < 1/10000 \dots (|n| > 50)$$

という特性を持つことを考えて、近接 ± 50 pixel 程度の範囲でコンボリューションを行えば実用上十分な画質が得られる。

また、実際に計測されるデータは X 線強度であるが、ボクセル値は透過率ではなくて吸収係数であるから、プロジェクション像のピクセル値として必要なのは、式 (3) に示す吸収係数の経路積分値である。そこで、コンボリューションの前処理として X 線強度から吸収係数の経路積分値に変換しておく。

$$-\ln(I/I_0) = \int_S \mu(x, y, z) dl \quad (3)$$

- I : 測定された X 線強度
- I_0 : 物体を置かない場合に測定された X 線強度
- $\mu(x, y, z)$: 各点の X 線吸収係数

また、吸収係数の経路積分値への変換以外にもコンボリューションの前処理として、測定系に依存したノイズの除去等も行うので、以下本稿ではコンボリュ

ーションという場合にはプロジェクション像の 1 フレームごとに行う次のような一連の処理を指すものとする。

- (1) **Shading 補正** : ディテクタの特性の補正。
- (2) **Shrink** : 隣接 4 画素から 1 画素を生成。
- (3) **Log 変換** : I/I_0 から吸収係数の経路積分値への変換。
- (4) **Trimming** : 信頼性に乏しい周辺画素の削除。
- (5) **Convolution** : Shepp-Logan 関数によるコンボリューション

3.2 バックプロジェクション

バックプロジェクションは、コンボリューションを施したプロジェクション像を三次元オブジェクトに逆投影する処理である。

同一ボクセルであってもディテクタの回転角度によってディテクタ上に投影される点の座標値 (x_{dk}, y_{dk}) が異なるため、角度ごとに投影点を算出し直す必要がある。また、実際にはボクセルが投影される点の座標値がコンボリューションを施したプロジェクション像上のピクセルの座標値とぴったり重なり合うわけではないので、投影点の近接 4 近傍ピクセルを線形補間した値をピクセル値としてボクセルに逆投影し、ボクセル値に累積加算する。

したがって、バックプロジェクションは、

- ボクセル投影点の座標計算
- 4 近傍ピクセルの線形補間
- ボクセルへの補間値の累積加算

ということが出来る。

4. 再構成プログラムの実装

4.1 再構成プログラムの構造

再構成プログラムを並列化する前にシングルプロセッサの場合の再構成プログラムの構造について簡単に説明する。

式 (1), (2) は 1 ボクセルに対する定式化であるが、実際に算出したいのはボクセルの三次元配列であるから、図 12 (1) シングルプロセッサのプログラムに示すように、まず、撮影された 1 フレームの全ピクセルについてコンボリューションを行い、そのデータを使用して全ボクセルにバックプロジェクションする。これを全フレームに対して繰り返し行えばよい。

4.1.1 ループの深さ

y_d を固定して x_d 方向のコンボリューションを施すので、コンボリューションの部分のループは角度、 y_d, x_d, x'_d に関する四重ループとなる。一方、バックプロジェクションの部分では、各々の角度のプロジェクション像を各三次元ボクセルへ逆投影するので角度

(1) シングルプロセッサのプログラム

```

loop Angle
  convolution
  loop X
    loop Y
      loop Z
        backprojection
      endloop Z
    endloop Y
  endloop X
endloop Angle

```

(2) 並列化したプログラム

```

parallel loopAngle
  convolution
endloop Angle

loop Angle
  parallel loop X
    loop Y
      loop Z
        backprojection
      endloop Z
    endloop Y
  endloop X
endloop Angle

```



図 12 プログラムの並列化

Fig. 12 Parallelization of the reconstruction program.

とボクセルの座標値 (x, y, z) に関する四重ループとなる。

今回の評価で使用するデータサイズを使って最内ループの実行回数を計算してみると、コンボリューションの場合には、

$$2.8 \times 10^9 \cong 360 \text{ angle} \\ \times (320 \times 240 \text{ pixel}) \times 101$$

となる。近接 ± 50 ピクセルの範囲でコンボリューションしているので、 x'_d に関するイタレーションの回数は 101 回となっている。

一方、バックプロジェクションの場合には

$$6.0 \times 10^9 \cong 360 \text{ angle} \times 256^3 \text{ voxel}$$

となる。

4.1.2 最内ループの処理

全体処理の重さはほとんど最内ループの演算量に依存するので、次に最内ループの処理について考えてみる。

実際に X 線撮影する際には X 線源とディテクタが回転しているのであるが、ボクセル投影座標の計算を行う際には、X 線源とディテクタに座標軸を固定し被写体を回転させると回転した後の座標 (x_k, y_k, z_k) が、

$$x_k = x \cos(-\Phi_k) - y \sin(-\Phi_k)$$

$$y_k = x \sin(-\Phi_k) + y \cos(-\Phi_k)$$

$$z_k = z$$

として求められるので、三角関数の計算を一番外側のループに追い出すことができて有利である。

ボクセル投影座標 (x_{dk}, y_{dk}) は、

$$x_{dk} = \frac{(a+b)}{(a+y_k)} x_k, \quad y_{dk} = \frac{(a+b)}{(a+y_k)} z_k$$

として求められるが、 x_{dk} はボクセルの x, y 座標だけで算出できるので、最内ループから追い出すことができる。さらに、空間位置補正係数も

$$[a/(a+r \cos(\phi - \Phi_k))]^2 = [a/(a+x_k)]^2$$

と変形できるので、バックプロジェクションの最内ループにおける計算は y_{dk} の算出と 4 近傍ピクセルの線形

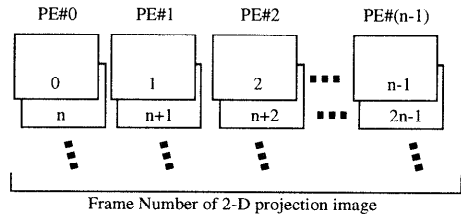


図 13 コンボリューションのデータ分割

Fig. 13 Data mapping in the convolution.

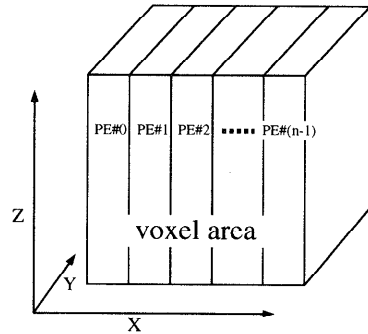


図 14 バックプロジェクションのデータ分割

Fig. 14 Data mapping in the backprojection.

補間および補間値のボクセルへの累積加算だけである。

一方、コンボリューション場合、Shepp-Logan 関数の値を前もってテーブルに展開することができるので、最内ループの処理は単なる積と計算である。前処理はすべて 1 ピクセル単位の処理なのでループの深さは一階層浅いが Log 変換等の比較的重い処理もある。

4.2 再構成プログラムの並列化

再構成プログラムをマルチプロセッサ上に並列化して実装する場合、コンボリューションとバックプロジェクションでは最適なデータの分割方法が異なるため、まず、角度に関するループをコンボリューションとバックプロジェクションに分離し、その後それぞれの処理に適した次のような並列化を施した。

- コンボリューションの場合はプロジェクションフレーム単位の処理になるので図 13 に示すように角度に関するイタレーションを各 PE にスケジュールする。
- バックプロジェクションの場合はプロジェクションフレームのイタレーションでは分割せずに、図 14 に示すようにボクセルを領域分割して異なる領域の再構成を各 PE にスケジュールする。

これをまとめると図 12 のようになる。このように再構成プログラムは粒度の大きい並列性を持っているため一般的に用いられている並列化のテクニックを適用して容易に並列化することが可能である。並列化の

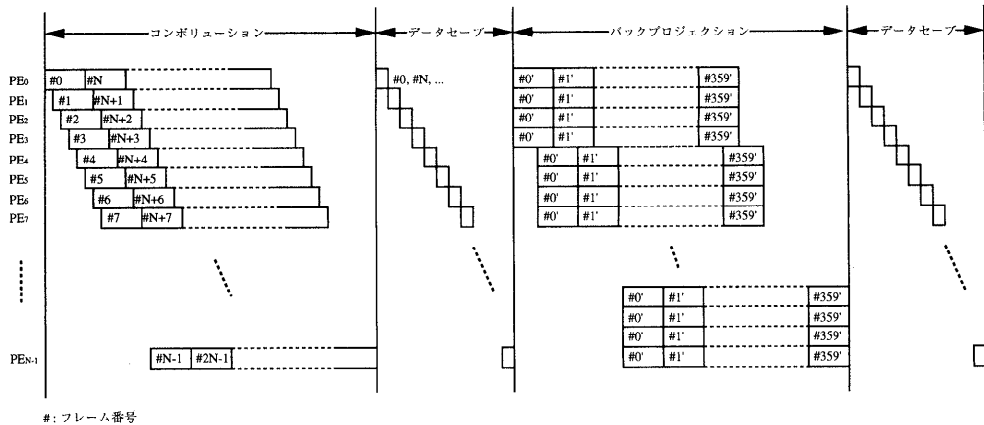


図 15 動作の概要
Fig. 15 Timing chart.

方針が決定できたので次節ではデータの受渡しや同期点置き方等を述べる。

4.3 処理手順とデータの流れ

X 線撮影装置からフレーム単位に送られてきたプロジェクションデータをフレームバッファメモリが自動的に取り込み、ホストワークステーションは取込み終了を検出すると各 PE を起動する。マルチプロセッサとして処理はここから開始される。

再構成の処理は以下 4 つのフェーズによって行われるが、各 PE はそれぞれのフェーズの処理を終了するとホストに終了を通知し、ホストは PE の終了が確認できた時点で全 PE に対して次のフェーズの開始を通知する。処理の全体的な様子を図 15 に示す。ただし、実際の動作では、必ずしも番号の小さい PE の処理がつねに先行しているとは限らない。

(1) コンボリユーション

各 PE は図 16 に示すように自ボード上の DMAC を使用して自ノードの共有メモリにプロジェクションデータをフレーム単位に読み出し、それと並行してこれにコンボリユーションを施す。その際、同一ボード上の 4 つの PE が 1 つの DMAC を共有するのでプロセッサ間リンクを用いて PE 間でセマフォを受け渡ししている。

(2) コンボリユーションデータの書き戻し

各 PE はコンボリユーションを施したプロジェクションデータをフレームバッファメモリにいったん書き戻す。

(3) バックプロジェクション

各 PE はコンボリユーションを施したプロジェクションデータの転送と並行してバックプロジェクションを行う。フレームデータの供給をホストの IO サー

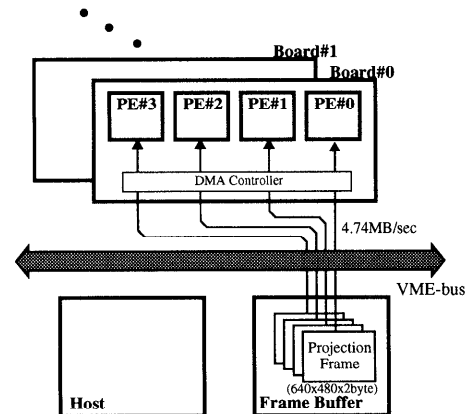


図 16 コンボリユーションの場合のフレームデータの供給
Fig. 16 Frame data transfer scheme in the convolution process.

ビス機能の 1 つとして実装しているため、ハードディスク上のファイルアクセス等と同様に各 PE はホストにデータ転送の要求を出し、ホストによってデータが供給される。その際、図 17 に示すように同一ボード上の 4 PE に対してブロードキャストして VME バス上のデータ転送量を減らしている。ブロードキャストする都合上、同一ボード上の 4 PE 間で同期をとる必要が出てくるが、プロセッサ間リンク上の同期メッセージによって実現している。

(4) ボクセルデータの書き戻し

各 PE は再構成したボクセルデータはフレームバッファメモリに書き戻す。書き戻されたボクセルデータはホスト上の Viewer 等の GUI ソフトウェア等によって使用される。

第二フェーズでコンボリユーションしたデータをフレームメモリに書き戻しているが、フレームメモリに

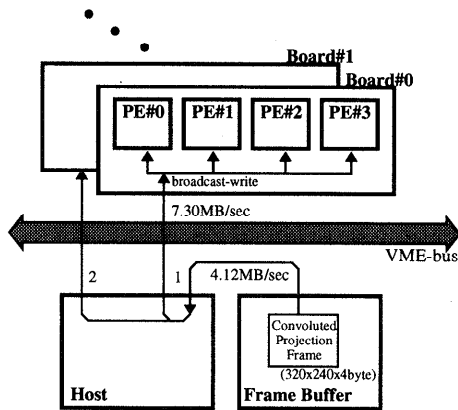


図 17 バックプロジェクションの場合のフレームデータの供給
Fig. 17 Frame data transfer scheme in the backprojection process.

書き戻さず、バックプロジェクションを行う際にプロセッサ間通信によって直接 PE 間でデータ転送する方法も考えられる。そのほうがデータを書き戻す時間が省略できて有利であるが、書き戻すのに要する時間を見積もってみると今回の評価に使用したデータのサイズの場合、コンボリューション後のフレームデータはトータルで 105 Mbyte であるから、書き戻すのに要する時間は VME バスの転送速度からみてただか 14 秒程度と見積もられ、これは処理時間全体に対しての数%程度の値である。また、データをフレームメモリに書き戻してしまうとバックプロジェクションの実行中はボード間でなんら同期をとる必要がなくなり、プログラムの実装がはるかに容易になるため、フレームメモリに書き戻す方法をとった。

第三フェーズのバックプロジェクションの処理において、フレームデータの読み出す際にホストに IO 要求を出してサービスさせているが、マルチプロセッサボード上の DMAC を使用して読み出す方法も考えられる。そこでホストと DMAC が 1 フレーム転送する時間を見積もってみる。ホストが転送する場合、ホストはフレームバッファメモリから 1 フレーム読み出し、それをバッファしてすべてのボードに送るためリード 1 回とライト N 回のデータ転送となるので、その所要時間は VME バスの転送速度の実測値から、

$$\text{FrameSize} \div 4.12 \text{ MB/s}$$

+ $\text{FrameSize} \div 7.30 \text{ MB/s} \times N \text{ Board}$
となる。一方、DMAC を使用した場合には、

$$\text{FrameSize} \div 4.74 \text{ MB/s} \times N \text{ Board}$$

となり、ボード数 N が 4 以上だとホストがデータを供給したほうが有利である。

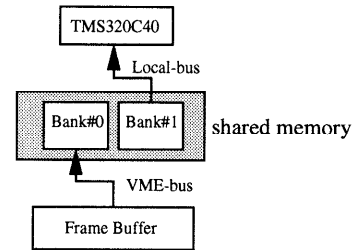


図 18 共有メモリのダブルバッファ
Fig. 18 Double buffering in the shared memory.

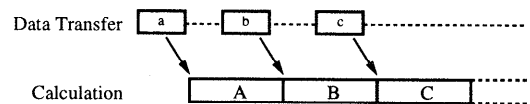


図 19 データ転送と演算のパイプライン化
Fig. 19 Pipelined execution of data transfer and calculation.

4.4 データ転送と演算のパイプライン化

2 バンク構成の共有メモリを積極的に利用して、2 つのバンクを図 18 のようにダブルバッファとして使用することができる。したがって、前もって次のプロジェクションフレームを要求しておけば、一方のバンクにすでにロードしてあるフレームを処理している間にもう一方のバンクに次のフレームがロードできるので、図 19 のようにデータ転送と演算のパイプライン化が可能になり、1 フレーム処理する間に次のフレームが到着していれば、データ転送時間を完全に隠すことができる。

5. 評価

5.1 再構成した画像

評価に使用したデータのサイズが表 3 であり、3 軸方向にそれぞれ 1 mm ピッチのボクセルデータを生成している。図 20 は実際に人体の胸部を撮影した二次元プロジェクション像である。これをもとに再構成した三次元 CT 像を図 21、図 22、図 23 に示す。

5.2 再構成時間の測定

4 章で定義した 4 つのフェーズの処理時間を測定してみた。

5.2.1 コンボリューションの処理時間の測定

プロセッサ数に対するコンボリューションに要する時間の変化を図 24 に示す。ホストが最初の PE にコンボリューション開始のトリガを出した時点から、全 PE がコンボリューションの終了をホストに通知した時点までの時間をホストが計測した。



図20 プロジェクション像 (0°, 45°, 90°)
Fig. 20 Projection images (0°, 45°, 90°).

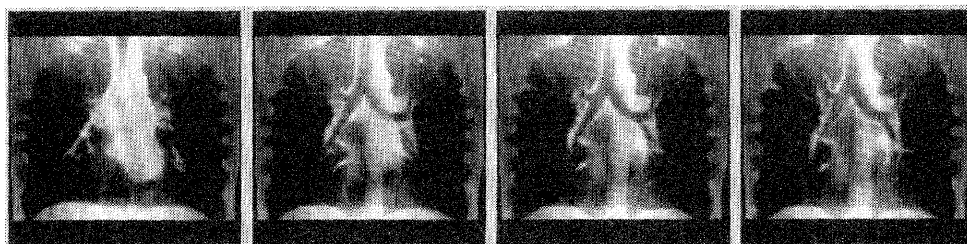


図21 CORONAL 像
Fig. 21 Coronal images.

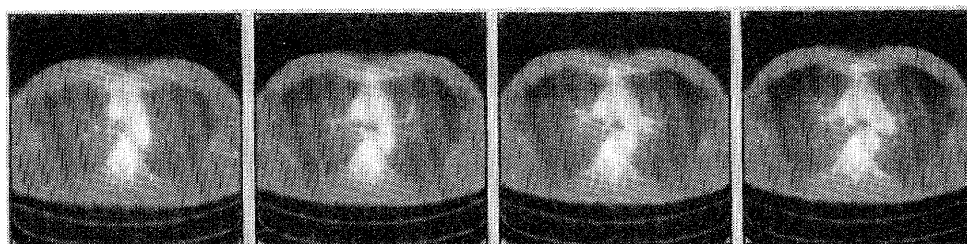


図22 TRANSVERS 像
Fig. 22 Transvers images.

表3 評価に使用したデータのサイズ
Table 3 Data size for evaluation.

2D-Projection Data	
フレームサイズ	640 × 480 (16b 固定小数点)
フレーム数	360 (1° ごとに 360° 分)
2D-Convolved Projection Data	
フレームサイズ	320 × 240 (単精度浮動小数点)
フレーム数	360 (1° ごとに 360° 分)
3D-Voxel Data	
ボクセル数	256 ³ (単精度浮動小数点)

5.2.2 コンボリューションデータの書き戻し時間

実測値はPE数によらず15secである。ホストが最初のPEにデータの書き戻し開始のトリガを出した時点から、全PEが書き戻しの終了をホストに通知した時点までの時間をホストが計測した。

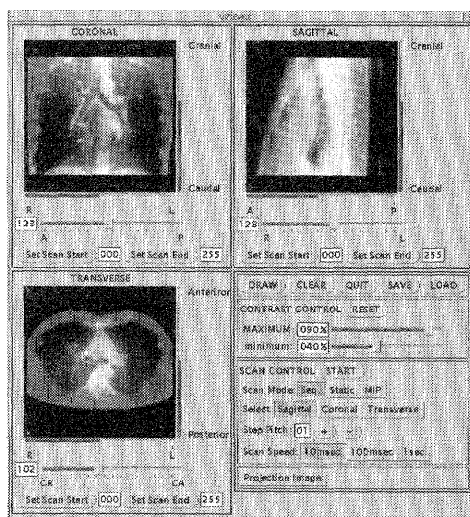


図23 3軸直交スライスのViewer
Fig. 23 Sample viewer.

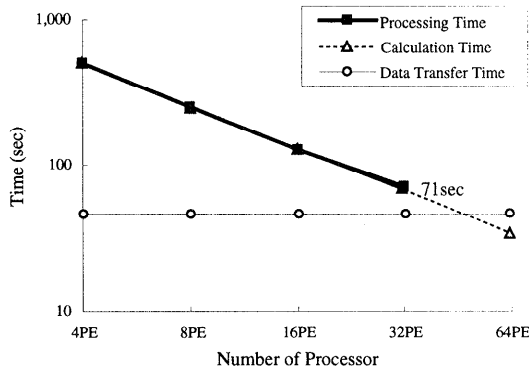


図 24 コンボリューションの処理時間 (実測値)

Fig. 24 Measured processing time of convolution.

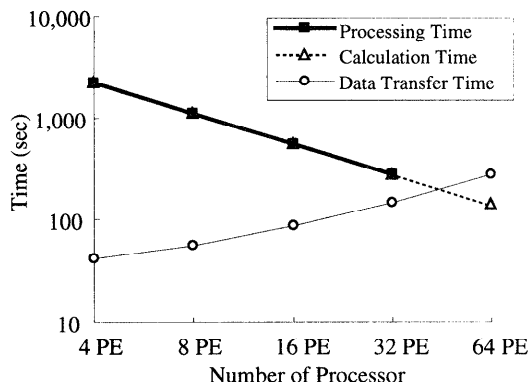


図 25 バックプロジェクションの処理時間 (実測値)

Fig. 25 Measured processing time of backprojection.

転送するデータのサイズ

$$105.5 \text{ Mbyte} \cong 360 \text{ frame}$$

$$\times (320 \times 240 \text{ pixel}) \times 4 \text{ byte}$$

と、VME バスを介した DSP からフレームメモリへのデータ転送速度の実測値 (7.55 Mbyte/s) から転送時間を見積もると 14.0 sec となり、ほぼ一致する。

5.2.3 バックプロジェクションの処理時間の測定

プロセッサ数に対するバックプロジェクションに要する時間の変化を図 25 に示す。ホストが最初の PE にバックプロジェクション開始のトリガを出した時点から、全 PE がバックプロジェクションの終了をホストに通知した時点までの時間をホストが計測した。

5.2.4 ボクセルデータの書き戻し時間測定の測定

実測値は PE 数によらず 9 sec であった。ホストが最初の PE にデータの書き戻し開始のトリガを出した時点から、全 PE が書き戻しの終了をホストに通知した時点までの時間をホストが計測した。

転送するデータのサイズ

$$64 \text{ Mbyte} = 256^3 \text{ voxel} \times 4 \text{ byte}$$

と、VME バスを介した DSP からフレームメモリへのデータ転送速度の実測値 (7.55 Mbyte/s) から見積もると 8.47 sec となり、ほぼ一致する。

5.3 1 フレームの処理時間の測定

演算時間とデータ転送時間と

コンボリューションの場合、1 つの PE が 1 フレーム処理するのに要する演算時間を DSP の内蔵タイマを使って測定すると 5.9 sec であった

バックプロジェクションの場合、1 つの PE では全ボクセル (64 Mbyte) をすべてローカルメモリに割り付けられないため、1 ボード (4 PE) 使用して 1 フレーム処理する時間を測定した。測定値は 6.16 sec であった。これも計測は DSP が行った。バックプロジェクションの場合、1 PE が担当するボクセル数が PE 数に反比例するため N ボード使用した時の 1 フレームの処理時間は 6.16/N sec と見積もることができる。

5.3.1 データ転送時間の見積り

コンボリューションの場合、VME バス上のデータ転送量は、プロセッサ数によらず一定であるから、全データ量は

$$(640 \times 480 \times 2) \text{ B/frame} \times 360 \text{ frames} \\ \cong 221 \text{ MB}$$

となる。表 2 からマルチプロセッサボード上の DMAC が VME バスを介してフレームメモリからデータを読み込む場合の速度は 4.74 MB/s である。したがって、データ転送時間は

$$211 \text{ MB} \div 4.74 \text{ MB/s} \cong 44.5 \text{ sec}$$

と見積られる。

一方、バックプロジェクションの場合、個々の PE に対してコンボリューションしたフレームデータをすべて転送する必要があるが、VME バスはブロードキャストができないのでホストはデータをフレームメモリから読み出し、いったんバッファしてから全ボードに分配している。そのためボード数を N とすると 1 フレームあたりホストからフレームメモリのフレームデータの読み出し (4.12 MB/s) が 1 回、同一ボード上の 4 PE には同時に転送できるのでホストから PE へのフレームデータの書き込み (7.30 MB/s) が N 回発生する。

1 フレームのデータサイズは

$$(320 \times 240 \text{ pixel}) \times 4 \text{ byte} \cong 300 \text{ Kbyte}$$

であるから、1 フレームの転送時間は

$$300 \text{ Kbyte} \div 4.12 \text{ MB/s} \\ + 300 \text{ Kbyte} \div 7.30 \text{ MB/s} \times N \\ \cong 71.1 \text{ msec} + 40.1 \text{ msec} \times N$$

である。バックプロジェクションの場合のデータ転送

時間はこれにフレーム数 360 を乗じて

$$360 \text{ frame} \times (71.1 \text{ msec} + 40.1 \text{ msec} \times N) \\ \cong 25.6 \text{ sec} + 11.4 \text{ sec} \times N$$

と見積もることができる。

5.4 32 PE 構成の場合の演算能力とデータ転送能力のバランス

処理時間のほとんどを占めるコンボリューションとバックプロジェクションのフェーズでは演算とデータ転送を並列に行っている。図 26 を見ると 32 PE の構成が最も演算時間とデータ転送時間のバランスが良いので、32 PE の構成の場合の演算時間とデータ転送時間のバランスをこの 2 つのフェーズに対して評価してみる。

コンボリューションの場合、1 PE が処理する平均フレーム数は

$$360 \div 32 = 11.25$$

であるから、8 PE が 12 フレーム処理し、24 PE が 11 フレーム処理する。1 フレームの処理時間が 5.9 sec/frame なので演算時間は

$$12 \text{ frame} \times 5.9 \text{ sec/frame} \cong 70.8 \text{ sec}$$

と見積もられる。一方、データ転送時間は 44.5 sec と見積もられるので演算能力が処理時間をリミットしていると予想できる。

実測値は 71 sec なので演算時間の見積り値とほぼ一致しており、確かに演算能力が処理時間を決定していることが分かる。また、VME バスの稼働率は

$$44.5 \text{ sec} \div 71 \text{ sec} \cong 63 \%$$

と見積もることができる。

バックプロジェクションの場合、演算時間は

$$6.16/8 \text{ sec/frame} \times 360 \cong 277.2 \text{ sec}$$

と見積もられる。一方、データ転送時間は

$$25.6 \text{ sec} + 14.4 \text{ sec} \times 8 \cong 140.8 \text{ sec}$$

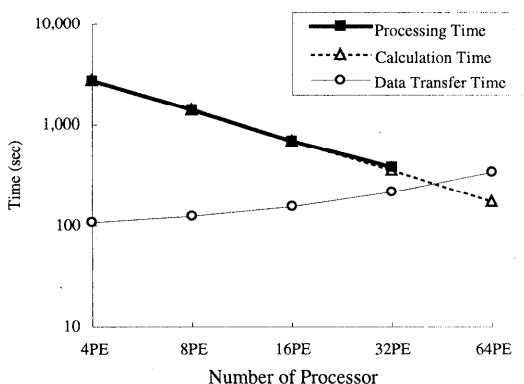


図 26 再構成に要する時間 (実測値)

Fig. 26 Measured processing time of reconstruction.

と見積もられるので、コンボリューションの場合と同様に、演算能力が処理時間をリミットしていると予想される。

実測値は 277 sec なので演算時間の見積り値とほぼ一致しており、確かに演算能力が処理時間を決定していることが分かる。

また、ボード上の 4 つの PE に対するブロードキャストができないとすると、ホストからの書き込み回数がボード数ではなく、PE 数になるためデータ転送時間の見積り値は

$$25.6 \text{ sec} + 14.4 \text{ sec} \times 32 \cong 486 \text{ sec}$$

となり、データ転送がボトルネックになり、かなり性能が低下することが予想できるので、同一ボード内の PE に対するブロードキャストはかなり効果があるということが分かる。

バックプロジェクションの場合の VME バスの稼働率は

$$140.8 \text{ sec} \div 277 \text{ sec} \cong 51 \%$$

と見積もられる。

5.5 再構成の全処理時間

再構成に要する全処理時間を図 26 に示す。ホストが最初の PE にコンボリューション開始のトリガを出した時点から最後の PE がボクセルデータの書き込みの終了をホストに通知した時点までの時間を計測したものである。

32 PE (8 ボード) 構成の場合の全処理時間は 6 分 12 秒であり、処理時間 10 分という目標はクリアしている。

CPU に R4400 を使用したソニー製のワークステーション NEWS-5000 で 200^3 ボクセル再構成した場合、約 1 時間強かかる。この場合、 256^3 と比べてボクセル数は約半分なので、本マルチプロセッサシステムは NEWS-5000 よりも 20 倍程度高速であるということが出来る。

6. ま と め

3 次元 X 線 CT 用の分散メモリ型マルチプロセッサシステムを開発し、実測による評価により次の点が確認できた。

- 32 プロセッサ構成の場合、360 フレームから 256^3 ボクセルを約 6 分半で再構成できるので、32 PE 構成の場合でも性能としては十分臨床試験に使用できるレベルにある。
- データ転送能力と演算能力のバランスという面では 32 プロセッサ構成が適当である。プロセッサ数 32 までは処理時間が演算時間によりリミットさ

れ, この場合, VME バスの稼働率は 50~65% と見積もることができる。

- 共有メモリのダブルバッファリングによるデータ転送時間の隠蔽や同一ボードの共有メモリに対するブロードキャスト転送等, アーキテクチャ上工夫した点も性能に対してかなり効果がある。
- また, 今回の試作/評価により,
- プログラムの並列化等, 並列処理の技術を適切に用いれば, VME バスベースの安価なシステムにおいても既存のワークステーションと比較して 10~20 倍程度の性能を得ることが可能である。
 - ワークステーションをホストとし, マルチプロセッサシステムをアクセラレータとするようなシステムの場合, マルチプロセッサを構成する個々の PE にはリアルタイム OS 程度の軽い OS を実装し, DSP 上のアプリケーションプログラムからホスト上の UNIX の IO 等が利用できるようになっていけば, 開発環境としては十分である。

ということも確認できた。

今後の課題としては, 三次元ボクセルの表示方法の充実を図るとともに, 本マルチプロセッサシステム上に三次元 CT 以外のアプリケーションを移植し, マルチプロセッサシステムとしてのコストパフォーマンス等の調査も行っていきたい。

参 考 文 献

- 1) Feldkamp, L.A., Davis, L.C. and Kress, J.W.: Practical Cone-beam Algorithm, *J. Optical Society of America*, Vol.1, pp.612-619 (1984).
- 2) Endo, M., Satoh, K., Okazaki, T., Hattori, Y., Kobayashi, S. and Jimbo, M.: Development of a 3D CT Scanner with a Cone Beam and Videofluoroscopic System, *Suppl. Radiology*, RSNA, Vol.197(P), p.292 (1995).
- 3) 遠藤, 吉田, 松本, 館野, 日下部: コーンビーム CT による正常人胸部の撮影, 第 55 回日本医学放射線学会学術発表会抄録, S119, 演題 426 (1996).
- 4) Okazaki, T., Satoh, K., Kusakabe, M., Endo, M., Fukuda, Y., Sasaki, T. and Iida, R.: A Par-

allel Processing System for High-speed Volume Reconstruction of a 3D X-ray CT, *Proc. SPIE Medical Imaging*, Vol.2431, pp.456-459 (1995).

- 5) Webb, J., Sutcliffe, J., Burkinshaw, L. and Horsman, A.: Tomographic Reconstruction from Experimentally Obtained Cone-beam Projection, *IEEE Trans.*, Vol.MI-6, pp.67-73 (1987).
- 6) 佐々木, 福田: 疎結合型並列処理システムを用いた三次元 X 線 CT 像の再構成, 情報処理学会, HOKKE96, HPC-3, 60-10, pp.55-60 (1996).
- 7) 佐々木, 福田, 岡崎, 遠藤: 分散メモリ型マルチプロセッサシステムを用いた三次元 X 線 CT 像の再構成, 情報処理学会, JSPP96, pp.251-258 (1996).
- 8) 佐々木, 福田, 石橋: 三次元 X 線 CT 用マルチプロセッサシステムのハードウェア評価, 情報処理学会, SWoPP 秋田 96, ARC-5, 119-18, pp.101-106 (1996).

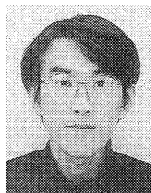
(平成 8 年 9 月 13 日受付)

(平成 9 年 2 月 5 日採録)



佐々木 徹 (正会員)

1958 年生。1983 年東京大学教養学部基礎科学科卒業。1985 年同大学大学院工学系研究科金属材料学専攻修士課程修了。同年 (株) 東芝入社。半導体技術研究所において CPU, キャッシュメモリの研究開発に従事。1990 年から (株) ソニー木原研究所において, 並列処理システムの研究開発に従事。



福田 安志

1962 年生。1985 年日本大学理工学部電気工学科卒業。1987 年同大学大学院博士課程前期修了。同年 NEC 入社。PC のハードウェア開発に従事。1991 年から (株) ソニー木原研究所において, 並列処理システムの研究開発に従事。