

## ボリュームレンダリング専用並列計算機 *ReVolver/C40* の性能評価

吉谷直樹<sup>†</sup> 重田大助<sup>††</sup> 五島正裕<sup>††</sup>  
森 眞一郎<sup>††</sup> 富田眞治<sup>††</sup>

*ReVolver/C40* は、3つのステージによるマクロパイプライン構成、各ステージにおける様々な並列処理、バンクコンフリクトのないメモリ構成、サンプリング方法の単純化、レイキャスティング法の採用などの特徴をもったボリュームレンダリング専用並列計算機である。本論文では現在動作している *ReVolver/C40* のサブセットを用いた性能評価を行った結果を示し、それに対する考察を述べる。*ReVolver/C40* は、1フレームを  $128^2$  pixel、ボリューム空間を  $128^3$  voxel、システムクロック周波数を 25MHz とした場合の画像生成速度の評価を行った結果、12.51 フレーム/秒の性能が得られることがわかった。

### Performance Evaluation of Parallel Volume Rendering Machine *ReVolver/C40*

NAOKI YOSHITANI,<sup>†</sup> DAISUKE SHIGETA,<sup>††</sup> MASAHITO GOSHIMA,<sup>††</sup>  
SHIN-ICHIRO MORI<sup>††</sup> and SHINJI TOMITA<sup>††</sup>

*ReVolver/C40* is a parallel machine designed specially for volume rendering. The features of this machine are three stages macro-pipeline structure, various kind of parallelism in each stage, conflict free volume memory, the simplification of the way to sampling and adoption of the ray-casting algorithm. This paper describes performance evaluation of *ReVolver/C40*. As a result of this performance evaluation, we find that *ReVolver/C40* can generate images of  $128^2$  pixels from  $128^3$  voxels volume in 12.51 frames/s under system clock frequency of 25 MHz.

#### 1. はじめに

ボリュームレンダリングは、3次元空間を単位立方体(ボクセルと呼ぶ)で構成されていると考え、この空間を2次元のスクリーンに投影する技法である。この技法は、医療画像分野における人体内部の可視化や、スーパーコンピュータなどの高速な計算機を利用した科学技術計算の結果の可視化による解析手法として重要視されている。そのため我々は医療画像生成だけでなく、科学技術計算の結果の可視化をも行えるような新しい専用並列計算機 *ReVolver/C40* を開発するために設計、実装を行ってきた。その結果、現在 *ReVolver/C40* はそのサブセットが動作している。今回そのサブセットを用いて *ReVolver/C40* の性能評価

を行った。

本稿では第2章において *ReVolver/C40* の概要について述べ、第3章では *ReVolver/C40* での並列処理ならびにアーキテクチャの特徴について述べる。第4章では *ReVolver/C40* のサブセットを用いた性能評価を行い、第5章でシステムの小型化に関する考察を行う。そして第6章でまとめる。

#### 2. *ReVolver/C40* の概要

##### 2.1 *ReVolver/C40* の特徴

*ReVolver/C40* によるボリュームレンダリングを達成するためには、半透明ボリュームの表示、遠近法による画像生成、高速描画という3つの要件を満足する必要があり、それぞれに対して以下に示すアプローチをとった。

**半透明ボリュームの表示** 科学技術計算の可視化に利用するために、*ReVolver/C40* はボリュームを半透明な物体として扱うことで、ボリューム全体の様子を把握することを可能にした。

<sup>†</sup> 京都大学大学院工学研究科情報工学専攻  
Division of Information Science, Graduate School of  
Engineering, Kyoto University  
<sup>††</sup> 京都大学大学院情報学研究科通信情報システム専攻  
Division of Communications and Computer Engineer-  
ing, Graduate School of Infomatics, Kyoto University

遠近法による画像生成 視点の位置や視線の方向に制限がなく、遠近法による奥行きのある画像表示を可能にした。

高速描画 ポリウムレンダリングは膨大な記憶容量と計算量が必要である。この際、最も計算時間を要するピクセル値計算を最大限並列化することを念頭においた。そのために、1) レイキャスティングアルゴリズムを採用し、2) 主軸等間隔サンプリングアルゴリズム<sup>1)</sup>に基づいた、3) バンクコンフリクトのない3次元メモリを開発した。

## 2.2 ReVolver/C40の構成

ポリウムレンダリング処理は、大きくわけて視線生成、ピクセル値計算、シェーディングの3つの処理にわけられる。ReVolver/C40ではこれらの3つの処理に特化した基板を設計、実装し、それらを接続することでポリウムレンダリング処理を高速に行うことを考えた。ReVolver/C40は図1に示すような構成をしている。

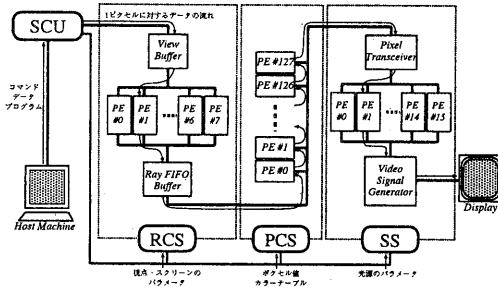


図1 ReVolver/C40の全体構成

以下に ReVolver/C40 の構成要素について述べる。  
システムコントロールユニット (SCU) システム全体の制御を行い、ホストマシンからの指示にしたがって、他の構成要素に対してコマンド/データを転送する。

視線生成ステージ (RCS) RCSはSCUから視点とスクリーンに関するパラメータ(ビューデータ)を受け取り、これらのデータと各ピクセルのスクリーン座標から視線ベクトルを求め、PCSにおけるボクセルサンプリング及びSSにおけるシェーディングに必要なデータ(視線データ)を出力する。RCSにおけるこれらの処理を総称して視線生成処理と呼ぶ。

ピクセル値計算ステージ (PCS) PCSはポリウムデータを保持しているステージであり、RCSからの視線データを受け取り、これらのデータに基づいてその視線線上にあるボクセル値を読み出す。ボクセル値は1byteのインデックス値なので、これを用いて属性テーブル<sup>1)</sup>を参照し、ボクセルの色や透明度を求めてピクセル値計算を行う。そし

てそのピクセル値をSSへ送る。

シェーディングステージ (SS) SSは、PCSから送られてくるピクセル値を、視線データに基づき適切な順序に並び替え、1フレーム分のデータが揃った段階でシェーディングを行い、その結果をCRTに出力する。

各基板にはProcessing Element(PE)として、TI社のDSPであるTMS320C40を使用している。SCUを除く各基板には、それぞれ8個のPEが搭載されている。ReVolver/C40を構成するためには、各ステージの基板はそれぞれ次のような枚数が必要である(表1参照)。これらの基板はSCUの一部を除いてVME

表1 ReVolver/C40の構成のために必要な各基板の枚数

基板名	基板枚数	
	最終構成	プロトタイプ
SCU	1枚	1枚
RCS	1枚 (PE8台)	1枚 (PE8台)
PCS	64枚 (PE512台)	16枚 (PE128台)
SS	2枚 (PE16台)	2枚 (PE16台)

のトリプルハイトサイズの筐体2つに収納される。プロトタイプシステムの筐体の様子を図2に示す。この筐体の中にはバックプレーンが取り付けられており、各ステージは一部を除いてバックプレーンにより接続される。他にもバックプレーンは、ReVolver/C40に電源やクロック信号、そしてリセット信号を供給する役割もはたしている。SCUの中心部分にはホストマシンのPCIスロットに挿入され、ReVolver/C40の各ステージとの通信にはフラットケーブルを用いる。

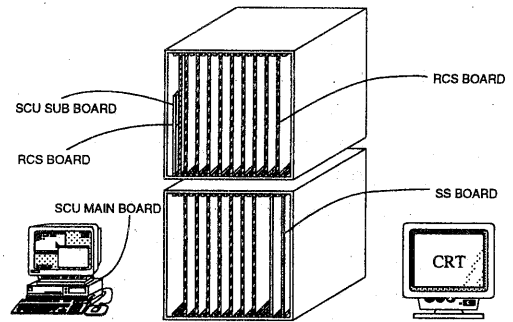


図2 ReVolver/C40の筐体(プロトタイプ)

## 3. ReVolver/C40での並列処理ならびにアーキテクチャの特徴

### 3.1 ReVolver/C40での並列処理

ポリウムレンダリング処理は多大な計算量が必要であるため、並列処理が不可欠である。このため Re-

Volver/C40 では様々な規模、形の並列処理を行っている。まず ReVolver/C40 は、フレーム単位で画像生成<sup>\*</sup>、シェーディング、画像出力を独立に処理することによってマクロパイプラインを形成し、高速な表示を可能としている。さらに、画像生成においてはピクセル単位のパイプライン処理が行われている(図3参照)。

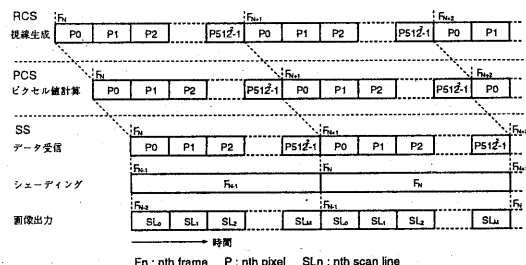


図3 ReVolver/C40 のパイプライン構成

また、各ステージでも以下に示すような様々な並列処理を行っている。

**RCSでの並列処理** PCSの速度に見合った速度で視線データを生成するため、8台のPEでフレームをスキャンライン単位のサイクルに分割し、静的負荷分散することによって、並列処理を行っている。隣接するスキャンライン内では負荷の急変が小さいため、PE間の負荷の偏りも少ない。また、RCSの各PEで並列に生成された視線データは、PCSに転送する際にシリアル化される。その際PCSでの処理の遅れにともなう通信遅延を吸収するために、FIFOメモリ(Ray FIFO Buffer)をPCSとRCSの各PEの間に搭載することにした。また各PEの調停によるオーバーヘッドを減少させるため、一度に複数(64words)の視線データを転送することにした。

**PCSでの並列処理** PCSはパイプライン構成になっており、それぞれのPEは視点に近いボクセルから順にピクセル値を計算し、自分の処理が終わると次のPEに計算結果を渡す。プロトタイプではPEを128台使用する。

**SSでの並列処理** SSでは、1)PCSから逐次的に送られてくるデータの受取りとそれを16台のPEで処理するための分配、2)シェーディング、3)シェーディングされたデータの収集と画像出力、という3段のパイプライン処理を行っている。1)と3)に対しては専用ハードウェア(それぞれPTおよびVSG)を設けており、パイプライン処理と並列処理それぞれのデータ授受方法の差異を吸収している。またスクリーンを16個の領域に分割し、

それぞれを一つのPEに割り当て、シェーディング処理を並列に行う。

上述の通りRCSとSSは、処理すべきデータに対して、空間的な並列処理を、またPCSは、時間的な並列処理をそれぞれ行っている。このようにRCSとPCS、PCSとSSの間では、並列処理の形態が異なるため、RCSならびにSSでは、並列処理効率を低下させずにこれらの差異を吸収するための措置が講じられている。

### 3.2 PCSのメモリアクセス機構の特徴

PCSは図4のようにピクセル値計算ユニット(PCU)を1次元に接続させた構成となっている。このPCUで、あるスライス<sup>\*\*</sup>のボクセル値を基にそのスライスまでのピクセル値を求める。ボクセル値を格納する3D Volume Memoryとして使用するDRAMは、アクセス時間が比較的大きく、ピクセル値計算時間と比較して無視できない。しかし、読み出すボクセル値のアドレスは視線データのみから求められることに着目し、サンプリングするボクセル値のアドレス計算と読み出しを行う専用ハードウェア(AG)を設けて、ボクセル値のプリロードを行う。このためにPCU間ではAGで扱うデータをAG bus、DSPで扱うデータをDSP busという2系統のデータパスにわけて通信を行っている。これにより、メモリアクセスとピクセル値計算の並列実行が可能となり、ボクセル値の読み出し時間の隠蔽を図っている。

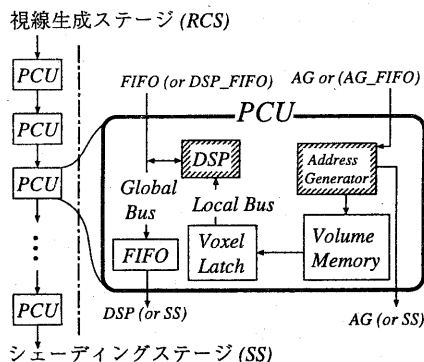


図4 PCUの構成図

## 4. ReVolver/C40の性能評価

ここでは主に ReVolver/C40 の画像生成速度についての評価を行なう。3.1節で述べたように、ReVolver/C40 は、フレーム単位でマクロパイプラインを形成している。よって、ReVolver/C40 の画面生成速度は1フレームにかかる処理時間が一番大きな部分

<sup>\*</sup> ReVolver/C40での画像生成は視線生成、ピクセル値計算、SSでのデータ受信の3つの処理をいう。

<sup>\*\*</sup> ボリューム空間のある軸に対して垂直に等分割した部分空間。

に律速される。

#### 4.1 ReVolter/C40 のプロトタイプの子バセツトを用いた性能測定結果

まず ReVolter/C40 のプロトタイプの子バセツトを用いての画面生成時間を測定する。この子バセツトは各基板の PE が RCS3 台、PCS8 台、SS8 台という構成となっている。よって処理可能な最大ボリューム空間は PCS の PCU が 4 平面処理した場合の  $32^3$  voxel<sup>\*</sup> である。

ReVolter/C40 の子バセツトを用いた画面生成速度を表 2 に示す。

表 2 ReVolter/C40 の画面生成速度 (実測)

1PCU 担当スライス数	画面サイズ (pixel)	1 画面処理時間 (ms)	画面生成速度 (frames/s)
1 スライス	$128^2$	79.94	12.51
2 スライス	$128^2$	93.58	10.68
4 スライス	$128^2$	135.82	7.36

次に子バセツトの各ステージの実測を示す。表 3 に RCS の実測、表 4 に画面サイズ  $128^2$  pixel での PCS の実測から求めた 1 ピクセルの処理に要する時間、表 5 に SS の実測を示す。

表 3 RCS の画面生成速度 (3 台での実測)

画面サイズ (pixel)	1 画面処理時間 (ms)	画面生成速度 (frames/s)
$128^2$	75.9	13.18
$256^2$	308.6	3.24
$512^2$	1206.2	0.83

表 4 PCS の画面生成速度と 1 ピクセルの処理に要する時間 (8 台での実測)

1PCU 担当スライス数	1 画面処理時間 (ms)	画面生成速度 (frames/s)	1 ピクセル処理時間 ( $\mu$ s)
1 スライス	79.94	12.51	4.88
2 スライス	93.58	10.68	5.71
4 スライス	135.82	7.36	8.29

表 5 SS の画面生成速度 (8 台での実測)

画面サイズ (pixel)	1 画面処理時間 (ms)	画面生成速度 (frames/s)
$128 \times 64$	20.41	41.00
$256 \times 128$	80.93	12.36
$512 \times 256$	323.26	3.09

<sup>\*</sup> ボリューム空間サイズ=(PCU 台数×担当平面数)<sup>3</sup>

ReVolter/C40 のユーザにとっては、画面生成速度も重要であるが、実際に使用するときの使いやすさも重要である。ReVolter/C40 でユーザインタフェースを担っているのは SCU である。以下ではユーザからの要求にこたえるための SCU の処理の実測を示す。

**ビューデータの転送 (変更)** ReVolter/C40 では視点の変更をリアルタイムに行なうために、ビューデータ (12word) を転送する。この転送時間を実測したところ、最速で  $4.32\mu$ s で 1 回の転送を終える。ビューデータの変更は 1 フレーム単位で行われるが、1 フレーム分の処理時間に対して十分短いため ReVolter/C40 の画面生成速度に影響を与えることはない。すなわち、ユーザからの視点変更指示にリアルタイムで追従できることがわかった。

**ボリュームデータの転送 (変更)** ReVolter/C40 では最初にボリュームデータを DRAM に格納する必要がある。また、途中でボリュームデータを変更したいという要求もでてくるであろう。この転送時間を実測したところ、 $32^3$  voxel 分のデータ (96KByte) の転送に約 82ms 要した。これより  $512^3$  voxel の場合でも約 7 分で転送が終了することがわかる。多くの場合ユーザは一度ボリュームデータを送ると、しばらくはそのデータで視点を変えながら観察を行うので、頻繁にボリュームデータを変更することはないと思われる。その点では十分ユーザの要求にこたえられる。

#### 4.2 ReVolter/C40 のプロトタイプのパフォーマンス予測

ここでは 4.1 節の結果を基にした ReVolter/C40 のプロトタイプ (表 1 参照) の画面生成速度のパフォーマンス予測を行う。予測の結果、RCS の画面生成速度は、視線生成処理と視線データ転送のどちらかに律速されるが、すべての画面サイズで視線生成処理が RCS の画面生成速度を決定することがわかった。また SS では PT でのピクセル値受信処理が SS の画面生成速度を決定する大きな要因であることがわかった。表 6 に ReVolter/C40 のプロトタイプのパフォーマンス予測を示す。

表 6 ReVolter/C40 のプロトタイプのパフォーマンス予測

画面サイズ (pixel)	画面生成速度 (frames/s)		
	RCS	PCS	SS
$128^2$	35.14	12.51 (1 スライス)	24.50
		10.68 (2 スライス)	
		7.36 (4 スライス)	
$256^2$	8.64	2.67 (2 スライス)	6.18
		1.84 (4 スライス)	
$512^2$	2.21	0.46 (4 スライス)	1.55

PCS 欄の括弧内は 1PCU 担当スライス数。

この結果、ReVolter/C40 のプロトタイプでは子バセツトの場合と同じように、PCS の画面生成速度が

ReVolver/C40の速度を決定することがわかった。以下ではReVolver/C40を律速するPCSの処理に関する評価を述べる。

### 4.3 PCSの評価

PCSの画面生成速度は、図5に示すように、各PCUでのピクセル値計算時間とAGがボクセル値を読み出す時間のどちらかに依存する。そこで、それぞれの1視線分の処理に要する時間を独立に測定した時間を表7に示す。\*

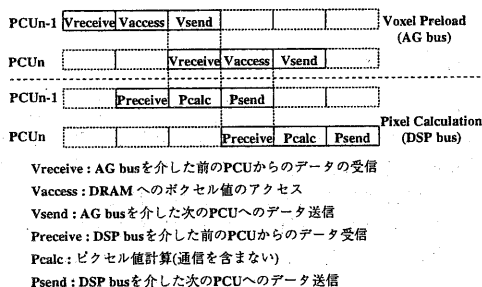


図5 ピクセル値計算とプリロードのパイプラインの様子

表7 1ピクセルの処理に要する時間

1PCU 担当 スライス数	ピクセル値計算 時間 (μs)	プリロード 時間 (μs)
1 スライス	3.28	2.52
2 スライス	4.80	2.80
4 スライス	7.24	3.36

#### 4.3.1 AGの効果に関する評価

図5と表7から考慮すると、ピクセル値計算時間がPCSの速度を決定し、ボクセル値のプリロードの時間は隠蔽できると推測できる。しかし表4の値は、表7のピクセル値計算時間の値よりも大きい。特に1PCUの担当スライス数が少ないほど性能低下の比率が大きい。この原因を調べた結果、基板間(RCS-PCS間、PCS基板間、PCS-SS間)の通信、特にその中でもAG busの通信で送受信の時間が長くなる現象が観測された。調査の結果、図5での前段のPCUのVaccessが終了せず、その次のPCUのVreceiveの前に待ちが生じ、遅延が後段に伝搬していることがわかった。これは基板内の通信距離にくらべて、基板間の通信距離が長く通信時間が大きいことに起因する。これに対処する方法として、プリロードをパイプライン化し通信処理時間を隠蔽することが考えられる。

現在は上述したように、AG間の基板間通信で遅延が発生しているが、もしAGを実装しなかった場合、ピクセル値計算時間は大きくなると考えられる。そこ

\* プリロード時間に関してはボード内での時間である。

でAGの効果を知るために、AGを使用しない場合のピクセル値計算時間を計測した。表8にその結果を示す。

表8 ピクセル値計算処理時間の比較

1PCU 担当スライス数	ピクセル値計算時間 (μs)	
	AGあり	AGなし
1 スライス	4.88	8.31
2 スライス	5.71	11.96
4 スライス	8.29	19.11

これより、AGを設けることで約1.7倍~2.3倍の処理時間向上が得られることがわかった。

#### 4.3.2 ピクセル値計算に関する評価

ボクセル値のプリロードが完全に隠蔽できた場合、PCSの画面生成速度はピクセル値計算処理によって決定する。そこでピクセル値計算処理についてさらに詳しく評価を行うことにする。この処理は細かく分けると、前段のPEからのデータ受信、ボクセル値を用いたピクセル値の計算(通信を除く)、後段のPEへのデータ送信という3つの処理にわけられる。ここでデータ送受信の時間を計測した結果、前段のPEからのデータ受信には1.20μs、後段のPEへのデータ送信には1.00μsであることがわかった。つまりピクセル値計算の中で、2.20μsは通信に利用されているということになる。これがピクセル値計算処理の中でどの程度の割合を占めるかを表9に示す。

表9 ピクセル値計算処理時間の中の通信時間の割合

1PE 担当スライス数	通信時間の割合
1 スライス	67.1%
2 スライス	45.8%
4 スライス	30.4%

この結果、データ送受信の時間がピクセル値計算時間の中でかなりの割合を占めていることがわかり、特に1PEで処理するスライス数が少なくなればなるほどその割合が大きくなることがわかった。つまりこの部分を改良すればReVolver/C40の画面生成速度は向上することになる。現在PE間通信にはFIFOメモリを使用し、ハンドシェイクを行うことでFIFOメモリへの送受信を行っている。これを1ピクセル分のデータを連続して受け取る改良を行うことによって、PE間送受信は最短で0.80μsで行えるようになる。この場合ピクセル値計算時間は約1.75倍の速度向上となる。

## 5. システムの小型化に関する考察

ReVolver/C40で使用しているDSP、Memory、FPGA等の要素デバイスは、近年ではより高性能化かつ小型化が図られている。これら最近の要素デバイ

スを用いた *ReVolver/C40* の性能を維持しつつ小型化する方法を考察する。

**プロセッサ** 現在は TMS320C40 とくらべて 10 倍以上の性能を持ったプロセッサが登場しており、これらを採用すれば、PCS でのピクセル値計算時間を十分小さくすることができる。しかしながら、1PE あたりのメモリバンド幅 (ランダムアクセス時) の不足のため、高速な汎用プロセッサを用いた小型化は困難である。ただし、視角を制限する等によりメモリアクセスの局所性を上げ、大容量キャッシュを併用することで高速化/小型化が図れる可能性が残っている。

**メモリ** 現在では容量的には、 $512^3$  voxel 分のボクセルデータを格納できる Volume Memory が DIMM1 枚で構成可能であるが、前述のプロセッサと同様にランダムアクセスに関しては、*ReVolver/C40* の開発着手当時とくらべても数倍程度の速度向上であり、*ReVolver/C40* として必要とされるメモリアクセスバンド幅を 1 枚の DIMM で実現することは不可能であり、従来通りのバンク構成が必要である。ただし、メモリアクセスのパイプライン化が可能であるので、ボクセル値のプリロードに必要なアドレス計算を含めたパイプライン化による速度向上は期待できる。

**FPGA** 最新の FPGA は、最大ゲート数 500000 ゲート、最大ユーザ使用可能ピン数 448 本、最速クロック周波数 200MHz の性能を持っており、特にゲート数に関しては飛躍的な向上である。

*ReVolver/C40* で使用している FPGA (XC4010D) は、そのゲート数とユーザ使用可能ピン数に制限されるため、現在はボクセル値のプリロードに専念している。しかし PCS でのピクセル値計算は、基本的には加算器と乗算器があれば実現可能であるので、FPGA で実装できる。よって FPGA がピクセル値計算とボクセル値のプリロードの両方の処理のパイプライン化に必要なゲート数を持っていれば、PE と AG の処理を FPGA 単独で実装することが可能と思われる。加えてこの場合ゲート数に余裕があれば、前述のメモリアクセスのパイプライン化を FPGA で行うことができる。よって FPGA を用いることにより、ボクセル値のプリロードを含めたピクセル値計算のパイプライン化による速度向上が可能であると考えられる。

## 6. ま と め

本論文ではボリュームレンダリング専用並列計算機 *ReVolver/C40* の性能評価をそのサブセットを用いて行い、それに対する考察を行った。

*ReVolver/C40* では PCS が全体の速度を決定している。そこで比較対照として Work Station を用いて

ピクセル値計算処理を行い、処理時間を比較する。マシンはキャッシュを 2MByte、メモリを 1GByte 搭載した Sun ULTRA2 Model 2300 を使用し、2CPU のうちの 1 つを使用して平行投影によるピクセル値計算処理を行った。その結果、画面サイズ  $128^2$  pixel、ボリュームサイズ  $128^3$  voxel で、画面生成速度が平均約 0.83frames/s という結果が得られた。同様の条件で *ReVolver/C40* は 12.51frames/s の性能がある。よって *ReVolver/C40* はコストパフォーマンスも考慮にいられて、汎用の計算機とくらべても有用なものであるといえる。

## 謝 辞

メンター・グラフィックス・ジャパン株式会社には Higher Education Program の一環として製品とサービスをご提供頂き、また、日本テキサス・インスツルメント株式会社の加藤賢二氏には、DSP ユニバーシティプログラムの一環として DSP 開発環境をご提供頂いたことに感謝します。また、プリント基板の作成にご協力頂いた、三精システム株式会社の藤代氏、森岡氏に感謝します。

なお本研究の一部は、文部省科学研究費補助金 (一般研究 (B) 課題番号 10480059 および 10558045、基盤研究 (C) 課題番号 09680334 ならびに 奨励研究 (A) 課題番号 09780268) による。

## 参 考 文 献

- 1) 對馬 雄次: ボリュームレンダリング専用並列計算機 *ReVolver/C40*, 並列処理シンポジウム JSP '95 発表論文 (1995)
- 2) 對馬 雄次: ボリュームレンダリング専用並列計算機 *ReVolver/C40* のアーキテクチャ, 並列処理シンポジウム JSP '94 発表論文 (1994)
- 3) 金 喜都: 視覚制限ピクセル並列処理によるボリューム・レンダリング向きの超高速専用並列計算機のアーキテクチャ, 並列処理シンポジウム JSP '96 発表論文 (1996)
- 4) 吉谷 直樹: *ReVolver/C40* の設計及び実装, 並列/分散/協調処理に関する「火の国」サマー・ワークショップ SWoPP 阿蘇 '97 発表論文 (1997)
- 5) 藤原 雅宏: 階層格子ボリュームデータの実時間可視化, 「ハイパフォーマンス・コンピューティングとアーキテクチャの評価」に関する北海道ワークショップ (HOKKE-98) 発表論文 (1998)
- 6) 鷺島 敬之, 西澤 貞次, 浅原 重夫 共著: 並列図形処理
- 7) 中嶋 正之, 川合 慧 共著: グラフィックスとマンマシンシステム
- 8) Texas Instruments: TMS320C4x User's Guide