

J-16

# サイクリックにデータを配置した 並列ボリュームレンダリング処理における ERT の効果

生雲 公啓<sup>1</sup>, 高山 征大<sup>1</sup>, 丸山 悠樹<sup>1</sup>, 中田 智史<sup>2</sup>  
津邑 公暁<sup>3</sup>, 五島 正裕<sup>1</sup>, 森 眞一郎<sup>1</sup>, 中島 康彦<sup>3</sup>, 富田 眞治<sup>1</sup>

- 1) 京都大学大学院 情報学研究科
- 2) 京都大学工学部
- 3) 京都大学大学院 経済学研究科

## 1 はじめに

近年, 様々な分野において大規模な数値シミュレーションが行われるようになってきたが, そこから得られる膨大な数値データを処理するためには, 多大な計算資源と時間がかかってしまう。例えば, 大規模なボリュームデータの可視化を並列処理で行う場合, 処理対象となるボリューム空間の全てのボリュームデータを各プロセッサのローカルメモリに置くことを考えると, 処理速度は高速になるが, 膨大なメモリ資源が必要となる。

本研究ではそれとは反対に, ボリュームデータを分割して各プロセッサに分散させて並列ボリュームレンダリングを行うことを考える。

## 2 対象とする並列ボリュームレンダリングアルゴリズムと ERT

ボクセル (3 次元立方格子) で構成されたボリューム空間でのボリュームレンダリングは, 図 1-(a) のようにスクリーンの 1 ピクセル毎に生成された視線にあるボクセルの色をそのボクセルの透明度に従って, 視点のほうから積算していきそのピクセル値 (色&透明度) とする。このピクセル値計算の際に, 透明度がある基準値よりも不透明になったら, それ以降のボクセルからの影響はごく小さいため, そこで計算を打ちきってしまうという手法が, Early Ray Termination (ERT) [3] である。

本稿では,  $N$  枚の  $N^2$  サイズのスライスからなる  $N^3$  のボリュームデータを  $P$  台のプロセッサに  $\frac{1}{P}$  ずつ分解する。そして各プロセッサは, 担当する  $M (= \frac{N}{P})$  枚のスライスに対して, Ray Casting アルゴリズム (Front-to-Back) に基づき, ピクセル値計算のための  $P$  段のパイプラインを搭載し並列処理を行うことを考える。ピクセル  $P(i,j)$  に対するピクセル値計算の流れを図 1-(b) に示す。

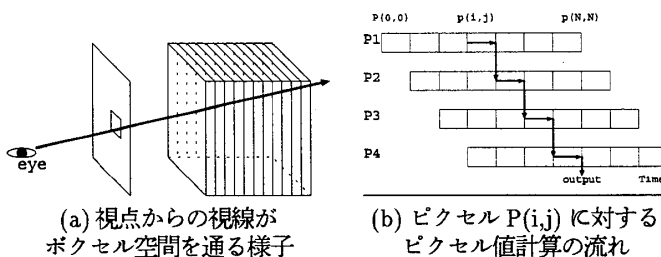


図 1: 評価の対象とする並列ボリュームレンダリング処理

<sup>1</sup>実際は視線方向の依存性をなくすため 3 重化しているため  $\frac{3}{P}$  なる [1]

## 3 サイクリック分割における ERT

データに依存して計算量が変化する場合は静的負荷分散手法の 1 つに, データをサイクリックに分割してプロセッサに割り当て, 担当領域に対して処理を行う手法がある。しかし, ボリュームレンダリング処理に対してこの手法を適用すると, プロセッサ間通信回数が増加するため逆に速度低下を招く恐れがある。

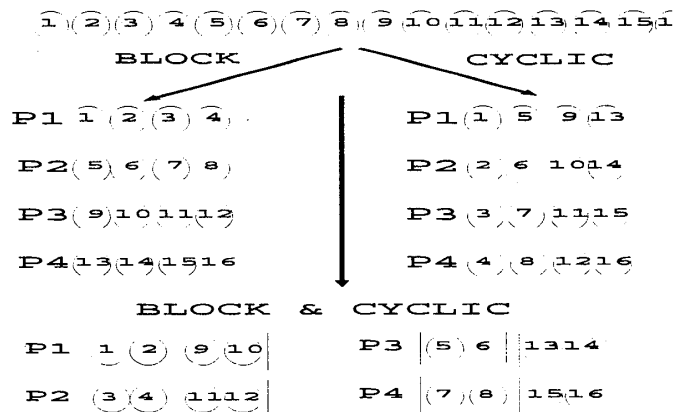


図 2: データのブロック分割とサイクリック分割

今,  $P = N$  の場合に 1 フレーム分の処理に要する時間を通信時間  $C$  と計算時間  $T$  の和として表現すると, ある  $P (< N)$  に対して幅  $\frac{M}{m}$  のブロックサイクリック分割時 (以下,  $m$  回サイクリックと呼ぶ) に描画に要する時間  $T_{M,m}$  は,

$$T_{M,m} = (C + \frac{M}{m} \times T) \times m \quad (1)$$

となり, ブロック分解 ( $m=1$  に相当) に比べて  $C \times (m-1)$  だけ処理時間が増加する。

この問題に対して, サイクリック分割であることを利用して, Early Ray Termination (ERT) の手法を適用することが考えられる。  $E_i$  % の視線が  $\frac{i}{m}$  までのサンプリングで Terminate すると, 描画時間  $T_{M,m,E_i}$  は,  $E_0 = 0$  とすると,

$$T_{M,m,E_i} = (C + \frac{M}{m} \times T) \sum_{i=1}^m i \times \frac{E_i - E_{i-1}}{100} \quad (2)$$

となる。

ここでもし  $E$  の値と  $\frac{T}{C}$  の値がある程度大きくなれば, ブロック分割の場合に比べて高速化が図れると考えられる。

## 4 評価

本章では前章で述べた、サイクリック分割における ERT による高速化を、実際にいくつかのポリウムデータ (256<sup>3</sup>) に適用して、その効果について検証を行った。

各ポリウムデータに対して、透明度が 0.05 以下になったらピクセル値計算を打ち切るという方針で、それぞれ E<sub>i</sub> を計算した。サイクル数 m=8 の時の結果を表 1 に示す。

使用したデータは次の 4 種類である。

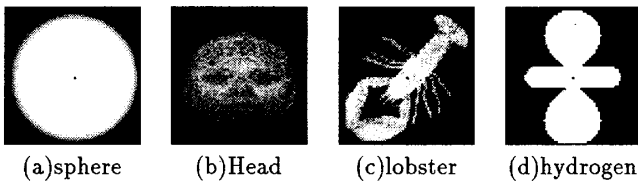


図 3: 使用したポリウムデータの概観

- sphere  
中心が (128,128,128) で半径が 128 の球。透明度は中心が 0.8 で最も不透明で徐々に透明になっていき球の表面では 1.0 になっている<sup>2</sup>。
- Head  
人の頭部のポリウムデータで、透明度は全て 0.5 である。
- lobster  
物体を構成している部分の透明度は 0.8 から 1.0 の間で変化する。
- hydrogen  
物体を構成している部分の透明度は 0.9 から 1.0 の間で変化する。

全ての画像とも物体の外側の部分 (黒い空間) の透明度は 1.0 になっている。

表 1: m = 8 の時の E<sub>i</sub> の値 (%) E<sub>8</sub> = 100

	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>
sphere	0.00	0.96	24.99	41.34	50.22	53.40	53.50
Head	0.00	1.90	31.13	41.53	45.45	46.53	46.53
lobster	0.00	0.00	0.00	1.00	5.17	5.17	5.17
hydrogen	3.15	9.45	26.15	40.31	40.84	41.02	41.02

表 2: m=1 の時に対する実行時間比 (%) (C:T=164:207)

m	P = 8 (M = 32)				P = 16 (M = 16)			
	1	2	4	8	1	2	4	8
sphere	100	81	81	84	100	83	86	95
Haed	100	81	83	85	100	82	88	97
lobster	100	101	105	114	100	104	112	130
hydrogen	100	81	82	87	100	83	88	99
ERT 未使用	100	102	107	116	100	104	114	133

次に我々が開発しているポリウムレンダリング専用並列計算機 ReVolver/C40demo での実測データ C=164msec, T=207msec[2] を適用して、プロセッサ数 P を変化させ<sup>3</sup> 実行時間を計算すると、

<sup>2</sup>透明度の値は 0.0(不透明) から 1.0(透明) までの値をとる

<sup>3</sup>画像の大きさは全て 256<sup>3</sup> のため M =  $\frac{256^3}{P}$  である

表 2 のような結果になった。lobster の画像は lobster 自体がうすっぺらく不透明になるピクセルが非常に少なかったため、効果が少なかったと思われるが、他のデータでは ERT を使うことにより軒並実行時間が短縮している。

M の値をある程度大きくし、ERT を用いることで、データをサイクリックに分割することで生じる通信時間の増加による描画速度の低下を解消できることが分かった。またレンダリングの対象となるポリウム空間に、ある程度広範囲にポリウムデータが存在すれば、サイクリックに分割しない場合よりも数%から 20% 近く速度向上が見られる。また、ERT を利用して生成された画像と、利用せずに生成した画像を比較しても、人間の目ではほとんど違いが分からず、ERT は十分有効であるといえる。

## 5 考察・まとめ

シミュレーションと連携させた可視化システムにおいては、その膨大なデータの転送時間が問題となるため、うまく負荷分散して並列に処理しなくてはならない。負荷分散の一形態であるサイクリック分割では、通信時間が増加して描画速度が低下することが懸念されるが、ERT を用いることで、画像をほとんど劣化させることなくより高速な描画が可能であることが確認できた。

ERT を用いて、サイクル数 m の値を大きくすることで通信量の増加を抑制することはできたが、今回使用したポリウムデータのよう、半数近くの視線が Terminate することなく最後まで計算されるため、サイクル数を多くしても劇的に高速になるということはない。今回のポリウムデータでは、サイクル数は 2 程度が妥当であると思える。

なお、今回は ReVolver/C40demo での実測値をベースに議論を行ったが、PC クラスタ等での並列ポリウムレンダリングで同様のサイクリック分割を行うと、一般に C >> T であるため、M, E を相当大きくとって初めて有効性が確認できる。また、そのような環境においては、視線を複数同時に処理し、通信回数を減らすことが考えられる。その場合、どのように視線を組み合わせて処理をするかが重要な問題になってくる。多くの視線を同時に処理すると、通信回数は大きく減少するが、その分 E の値が低下してしまう。E の値が低下してしまうと今回の lobster の様に、ERT の効果が望めなくなってしまう。現在はこれらの問題について研究を進めているところである。

## 謝辞

日頃より御討論いただく京都大学大学院情報学研究所富田研究室の諸氏に感謝します。なお、本研究の一部は文部省科学研究費補助金 (基盤研究 (B)(2) 課題番号 13480083 ならびに特定領域研究 (C)(2) 「情報学」 課題番号 13224050 ) による。

## 参考文献

- [1] 對馬 雄次 他, “ポリウム・レンダリング専用並列計算機 ReVolver のアーキテクチャ”, 情報処理学会論文誌, 第 36 巻, 第 7 号, pp.1709-1718, 1995.
- [2] 吉谷直樹 他, “ポリウムレンダリング専用並列計算機 ReVolver/C40 の性能評価”, 情処研報告, 99-ARC-132, pp.79-84, 1999.
- [3] Philippe Lacroute and Marc Levoy, “Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation,” Proc. of SIGGRAPH'94, pp.451-458, July 1994.
- [4] 原瀬 史靖 他, “数値シミュレーション過程の実時間可視化を支援するハードウェア”, 可視化情報シンポジウム, 2002.7.