

2J-8

マルチプロセッサワークステーションでの部分更新光線追跡法

片平 昌幸 堀口 進 木村 正行 中田 武男

(東北大学 工学部) (日本IBM)

1. はじめに

コンピュータグラフィックスの分野において、現在最も高品質な画像を生成できるアルゴリズムの一つに光線追跡法(Ray Tracing)がある[1]。光線追跡法は光の反射、屈折などを統一的に扱えるため非常にリアルな画像を生成できる反面、処理速度が非常に遅いという欠点がある。

本稿では、光線追跡法による動画生成の際の高速化手法である部分更新光線追跡法を並列化し、共有メモリ型密結合マルチプロセッサシステム上にインプリメントして実験を行った結果について述べる。

2. 部分更新光線追跡法による動画生成

光線追跡法は、視点からスクリーン上の各画素を通る光線を光学現象に従って追跡して、各画素の色を決定し画像を生成するものである。この光線追跡の状況は木の形に表すことができ、光線追跡木と呼ばれる。

光線追跡法を用いて動画を生成する場合、各フレームごとにすべての画素を生成する方法では、各種の高速化手法を用いても膨大な時間を要するが、動画の連続するフレーム間では画像の変化量が少ないことに着目して前フレームから変化した部分のみを再計算するという部分更新光線追跡法[2][3]が提案され、高速な動画生成ができることが報告されている。このためには、各画素における光線追跡の情報を記憶しておく必要があるが、全ての画素に関する光線追跡の情報を光線追跡木の形で保存するためには膨大な記憶容量を必要とする。

そこで我々は、視点を固定した動画生成を行う場合において、光線追跡木を保存する画素は動画の1シーンの間の各フレームで変化することが予想されるものだけに限定して要求される記憶容量を少なく抑えるための手法を提案した[4]。各画素の光線追跡木を保存すべきか否かは、移動するオブジェクトの軌跡から判定することができるが、これを高速に行うために、光線追跡法の高速化手法の一つであるオブジェクト空間の三次元等分割法[5]を応用した。この手法を用いる事によって、メモリ使用量を全画素の全ての光線追跡木を保存する場合に比較して五分の一程度まで減少させることができた。

3. 共有メモリ型並列計算機による部分更新光線追跡法の性能評価

部分更新光線追跡法を画素分割型の並列処理方式に基づいて共有メモリ型並列計算機上にインプリメントして実行する実験を行った。

実験に使用した並列計算機は共有メモリ型密結合マルチプロセッサシステムである[6]。システム構成はローカル・キャッシュ付きの複数のプロセッサ(80386)間で主記憶を共有し、各プロセッサカード間を高速バスで結んだものとなっている。

前章で述べた部分更新光線追跡法を画素分割型の手法を用い並列化して並列計算機上にインプリメントした。実験に使用した並列計算機は共有メモリ型であり、オブジェクト等の各プロセスに共通なデータはプロセス間で共有する事ができる。一方、光線追跡木の情報は各プロセス固有のメモリ空間上に配置されるため、各プロセスに処理する画素を動的に割り当てて負荷分散を図ることはできない。従って、各プロセスで処理する画素の割当方法を工夫することによって、プロセス間の静的負荷分散を図ることになる。そのための画素割当法として、図1に示すブロック、シーケンシャル、ドットの3種類を用いた。

実験は、プロセッサ10台のシステム上で6種のサンプル画像を用いて行った。画素数は160×160、オブジェクト空間の分割数は1辺当たり16とし、それぞれプロセス数2、4、8の場合について実行した。一部の画像では、16プロセスについても実行してみた。

図2、図3に、サンプル画像MVTSTについてプロセス数を変えて実行した結果を示す。各プロセスに対する画

1 2 3 4 1 2 3 4	1 2 3 4 5 6 7 8	1 1 2 2 3 3 4 4
5 6 7 8 5 6 7 8	9 10 11 12 13 14 15 16	1 1 2 2 3 3 4 4
9 10 11 12 9 10 11 12	1 2 3 4 5 6 7 8	5 5 6 6 7 7 8 8
13 14 15 16 13 14 15 16	9 10 11 12 13 14 15 16	5 5 6 6 7 7 8 8
1 2 3 4 1 2 3 4	1 2 3 4 5 6 7 8	9 9 10 10 11 11 12 12
5 6 7 8 5 6 7 8	9 10 11 12 13 14 15 16	9 9 10 10 11 11 12 12
9 10 11 12 9 10 11 12	1 2 3 4 5 6 7 8	13 13 14 14 15 15 16 16
13 14 15 16 13 14 15 16	9 10 11 12 13 14 15 16	13 13 14 14 15 15 16 16

dot type sequential type block type
図1 画素のプロセッサへの割当方法

(16台のプロセッサを8×8の画素に割り当てた例)

Incremental Ray Tracing on Multiprocessor Workstation
Masayuki Katahira*1, Susumu Horiguchi*1, Masayuki Kimura*1, Takeo Nakada*2
*1 Tohoku University, *2 IBM Research, Tokyo Research Laboratory

素割当法はシーケンシャル型を用いた。それぞれ、1フレーム目、2フレーム目以降の平均の実行時間である。

理想的には、プロセス1つの場合の実行時間と比較して並列動作させるプロセス数分の一になることが望ましいが、実際には負荷の偏りなどのためにそこまで達することはないのが普通である。1フレーム目の実行結果では、2、4プロセスではかなり理想値に近い結果が得られているが、8プロセスではやや低下している。また、16プロセスでは、システムが持つプロセッサ数を越えているため、かえって遅くなっている。一方、2フレーム目以降の平均では、プロセス数以上に大きなスピードアップが2、4プロセスの時に得られている。これは、膨大なメモリを使用する部分更新光線追跡法が並列化されることにより、1プロセス当りの使用メモリ量が減少し、キャッシュメモリの効果が増したことによるものと考えられる。

次に、3種類の画素割当法による負荷分散の効果について示す。サンプル画像ORBIT3についての実験結果を図4、図5に示す。1フレーム目では全ての画素が計算されるため、2フレーム目以降の平均と比較して3種の割当法の間は差は小さくなっている。しかし、2フレーム目以降の計算においては、再計算の行われる画素が画面上に偏って存在するようになるため、負荷分散の効果の高いシーケンシャル形、ドット形とブロック形の差が大きく開くようになる。実験の結果、一般にシーケンシャル形の方がわずかにドット形よりも速度向上度が上回っていることが判明した。

4. まとめ

本稿では、光線追跡法による動画生成を高速化するための手法である部分更新光線追跡法を並列化して共有メモリ型並列計算機上で実行させた実験結果を示し、検討を行った。今後の課題としては、

- ・ 移動物体の数、大きさ、種類、移動軌道などを変化させた、各種の動画をを用いた性能評価
- ・ 光線追跡木の情報をプロセス間で共有し、2フレーム以降の再計算時に画素再割当を行うことによる動的負荷分散などが挙げられる。

[参考文献]

[1] T. Whitted, Comm. ACM, Vol. 23, No. 6, June 1980
 [2] 村上 他: "レイトレーシングにおけるフレーム間コヒーレンス方式", 情報処理学会第33回全大, 4Q-6
 [3] 広田 他: "部分更新レイトレーシング", 情報処理学会研究報告, 87-CAD-27-5, Vol. 87, No. 46, Jul., 1987
 [4] 片平 他: "動画生成のための高速並列光線追跡法", 第4回NICOGRAPH論文コンテスト論文集, 1988
 [5] A. Fujimoto et. al, IEEE CG&A, April 1986
 [6] 鈴木 他: "高速並列処理ワークステーション(TOP-1)", 情報処理学会第37回全大, 7N-1~7N-7

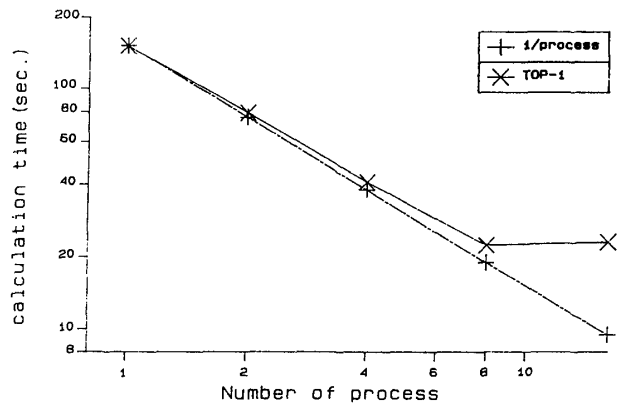


図2 プロセス数と処理時間の関係(MVTST:7V-A1)

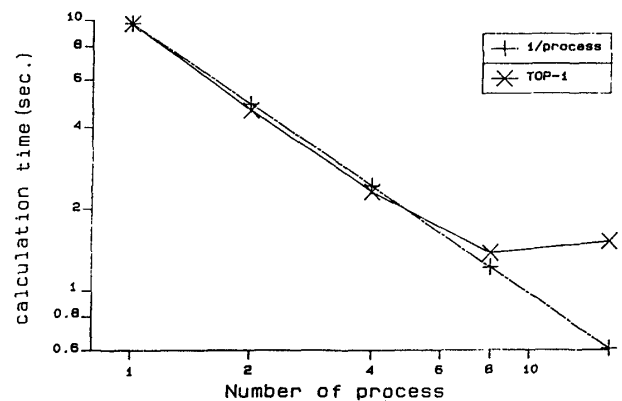


図3 プロセス数と処理時間の関係(mvtst:27V-A以降)

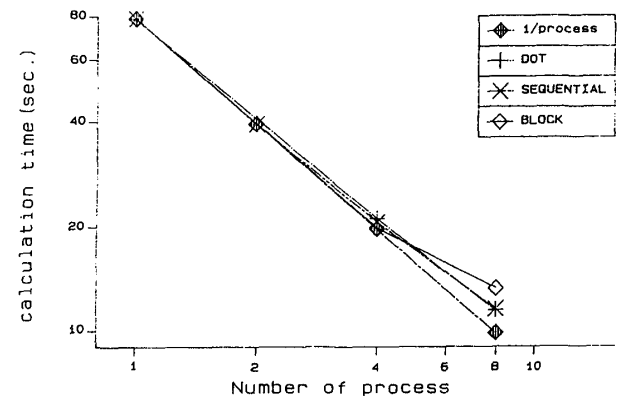


図4 画素割当法と処理時間の関係(ORBIT3:7V-A1)

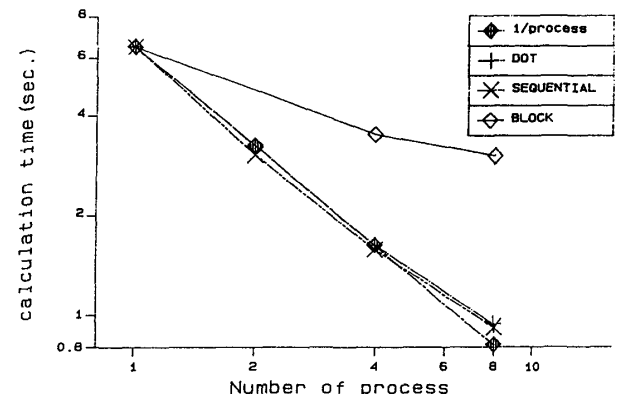


図5 画素割当法と処理時間の関係(ORBIT3:27V-A以降)