

128-02

分散メモリ型並列計算機を用いた 3次元物体の衝突面検出の性能評価*

二宮 茂樹 渋谷 進†
茨城大学工学部情報工学科‡

1 はじめに

3次元物体の衝突問題は、ロボティクス、コンピュータグラフィックス等多くの分野で中心的な問題の一つである。正確に物体間の衝突を検出しようと考えるとき、作業空間中の物体の全ての面と稜線の組合せの交差を調べると、物体の数や物体の面数が多くなるにつれ、衝突面検出の計算量は大きく増加してしまう。

3次元物体の衝突面を検出する方法としては、離散時間の衝突検査で物体間の衝突面を実際に衝突が起きる直前に特定する衝突面検出アルゴリズムが開発されてきた[1]。また文献[1]では、共有メモリを用いて、並列アルゴリズムを実現している。

本研究では、3次元物体の衝突面検出を並列化し、分散メモリ MIMD 型並列計算機を用いて、衝突面検出実験を行った[2]。

2 衝突面検出のアルゴリズム

このアルゴリズムでは、3次元空間中の物体の衝突についてその衝突面を検出する。各物体は一離散時刻毎に移動し、その移動量はあらかじめ分かっているとす。各stepで、検出すべき領域や面が見つからないときは、次の時刻へ進む。

step1: 外接直方体の重なり領域の検出

各物体に対して外接直方体を生成し、各対の物体の重なり領域を検出する。

step2: 重なり領域と交差する候補面の抽出

重なり領域を持つ対の物体に対し、各物体を構成している全ての面と重なり領域との交差を調べる。また、このとき交差する面を候補面とする。

step3: 8分木を用いた面ペア検出

候補面が作業空間中に存在する位置を、深さ d の8分木で表す。8分木のルートノードをstep1で検出した重なり領域とし、順次空間を8分割した木で表現する。また、最下層のノードの大きさは、面が一離散時間内に移動する量よりも十分大きいとする。この8分木を用いて、候補面がどの空間

に存在するかを調べる。最下層のノードにおいて、異なる物体からの面の組合せを面ペアとする。

step4: 面ペアの衝突検査

各面ペアについて、現在の時刻を t_i とすると、時間 $[t_i, t_{i+1}]$ に各面によって構成される物体が交差していれば、この面ペアは衝突すると判断する。

3 AP1000 を用いた実験

本研究で用いた AP1000 は、セル数が16であり、各セルは25MHzのSPARCプロセッサで16Mbyteのメモリを持っている。

逐次実験からstep4が衝突面検出処理の全実行時間中で大きな比重を占めていることがわかり、本研究ではstep4の並列化を行う。衝突面検出のstep4は、step3までで得られた面ペア一つ一つに同様の処理を行っている。そこで、面ペアを各セルに均等に分配する、次の方法で並列化を行った。

方法1 面ペア数から、1セルが処理する面ペア数を計算し、その数ずつホスト-セル通信で送る。

方法2 ホストからセル0へ全ての面ペアを送り、セル-セル通信で全てのセルが均等な面ペアを持つように、図1のように送る。

方法3 方法2のセル0の役目をホストにさせて、面ペアを送る。このとき図1のセル0からの通信はホスト-セル通信になる。ホストは最後にセル0に残りの面ペアを送る。

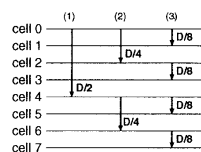


図1: 8セル用いたときの通信の方法 (D: 面ペア数)

次に、逐次時間と並列時間の比率と、衝突検査を行う面ペア数の関係を図2に示す。この実験では、衝突検査を行う面ペアのうちの半分が衝突するデータを扱った。また、測定した時間は5回測定した平均値をとった。

図2から、全ての方法について面ペアの数が増えるほど比率は良くなっていることが分かる。つまり、面ペア数が増えるほど、並列化の効果が上がっているといえ

*The Evaluation for the Detection of Colliding Pair-Faces among 3D Objects using Distributed-Memory Computers

†Shigeki Nimiya, Susumu Shibusawa

‡Department of Computer and Information Sciences, Faculty of Engineering, Ibaraki University

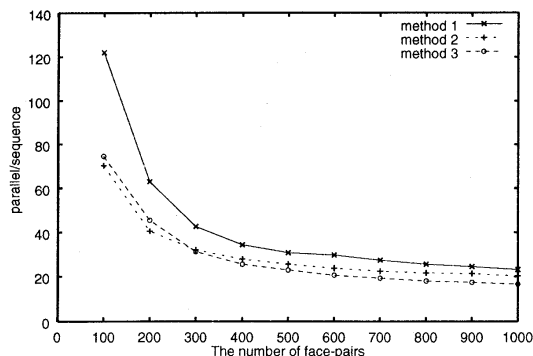


図 2: 比率と面ペア数の関係 (16 セル使用)

る。ただし、面ペアが増える毎に比率の改善が少なくなっているため、限界があると考えられる。また、100 から 300 辺りまでの面ペア数について、一番効率の良いのが方法 2 であり、それ以上の面ペア数では、方法 3 が一番効率が良いという結果を示した。面ペア数が小さいときに、方法 2 が一番効率が良い理由については、ホストセル通信がデータ量が小さいとき時間がかかるということが考えられる。

それぞれの測定値から次式ができる。ここで、 D はデータ量である。

$$\begin{aligned} T_{method1} &= 8.9 \times 10^{-5} \times D + 6.66 \times 10^{-2} \\ T_{method2} &= 10.1 \times 10^{-5} \times D + 3.66 \times 10^{-2} \\ T_{method3} &= 6.7 \times 10^{-5} \times D + 4.51 \times 10^{-2} \end{aligned}$$

4 SR2201 を用いた実験

SR2201 は、計算、入出力のノードから成っており、各ノードは 3 次元クロスバネットワークという高速のネットワークで接続されている。ノードは X 軸方向、Y 軸方向及び Z 軸方向に接続される。

AP1000 と同様に step4 の並列実験を行い、使用するプロセッサに均等な数の面ペアを割り当てるという方法で実験した。実験は、2,8 プロセッサを用いて行った。100 から 1000 個の面ペアを扱い、そのうちの半分の面ペアが衝突する。また、この章での実験で測定した結果は、10 回測定した値から最大値、最小値を除いた平均をとっている。

次に、8 プロセッサ実験の評価のために、総通信時間 (T_{comm}) を測定した。また、純粋な総計算時間を出すために、逐次実験の実行時間 (T_{p1}) を用いた。逐次実験はデータの通信を行わず、面ペアの計算のみの時間である。よって、この逐次実験の実行時間を 8 で割る ($T_{p1}/8$) ことで、8 プロセッサ実験で実際の計算時間とほぼ同等な値を得ることができると考える。

得られた測定値から、次式ができる。この式から、8 プロセッサにおいて、通信時間と計算時間の和と実行時間の間には差が見られるが、これはプロセッサの待ち時間と考えられる。

$$T_{p1} = 1.1 \times 10^{-4} \times D + 6.7 \times 10^{-4}$$

$$\begin{aligned} T_{p2} &= 5.7 \times 10^{-5} \times D + 8.4 \times 10^{-4} \\ T_{p8} &= 1.5 \times 10^{-5} \times D + 1.1 \times 10^{-3} \\ T_{p1}/8 &= 1.4 \times 10^{-5} \times D + 8.4 \times 10^{-5} \\ T_{comm} &= 0.1 \times 10^{-5} \times D + 5.9 \times 10^{-4} \end{aligned}$$

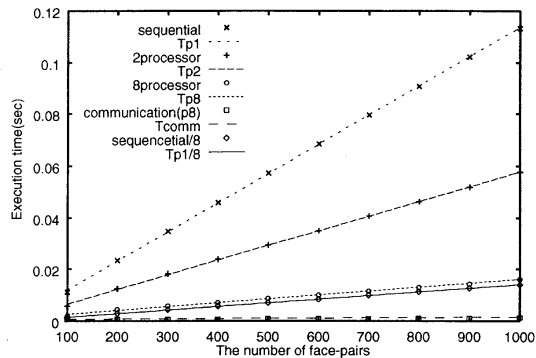


図 3: 実行時間と通信時間

これらの結果を示した図 3 から、各プロセッサ数での実験について、その実行時間は面ペア数に比例していることが分かる。また、8 プロセッサの実行について、その実行はほとんどが計算時間で占められていることが分かる。つまり、均等な面ペアを送って実行するという方法が効率が良いこと分かる。

5 二つの並列計算機の比較と考察

AP1000 では、方法 1 は他の方法と比べると時間がかかる処理であったが、SR2201 では、ほとんど時間の差が無いという結果が得られた。これは、SR2201 の通信性能が良いためと考えられる。実際、8 プロセッサを用いた実験では、データがある程度大きい場合、通信時間は実行時間の 7% 程度である。逆に、通信性能が低い場合にはその通信操作を効率的にすることで実行時間の改善を図ることができるといえる。

6 おわりに

本研究では、分散メモリ型並列計算機として AP1000 と SR2201 を用いて、3 次元物体の衝突面検出アルゴリズムの並列化を行った。そして、測定値から時間の式を作り、評価した。今後の課題としては、より一般的な物体を用いたときの時間式の適用などが挙げられる。

謝辞 ご討論頂いた渋谷研究室の皆様へ感謝致します。

参考文献

- [1] 北村喜文, Andrew Smith, 竹村雄雄, 岸野文郎: “並列計算機による 3 次元物体の実時間衝突面検出”, 信学論 (D-I), Vol.J78-D-I, No.8, pp788-797, 1995.
- [2] 二宮, 渋谷: “分散メモリ MIMD 型並列計算機を用いた 3 次元物体の衝突面検出”, 情報学アルゴリズム研報, AL63-7, July, 1998.