

通信処理の競合を考慮したレイトレーシング分散処理法

本田 彰[†] 宮尾 淳一[‡]

[†] 広島大学大学院 工学研究科

[‡] 広島大学 総合科学部

あらまし 画面分割によりレイトレーシング法の並列処理をする場合の負荷分散手法には、静的負荷分散法と動的負荷分散法がある。これらの手法をネットワーク上で分散処理する場合、負荷分散とともに通信にかかる時間も考慮しなければならない。今回我々は、通信量を減らしつつ負荷を均一にするためのSR法を提案する。さらに、SR法と従来の動的負荷分散法をPVMにより実装した実験結果を示す。その結果、動的負荷分散法は画像生成時間を最小にする画面分割数の最適値が存在するが、その値はスレーブ数と画像内のピクセルの計算量により変動することがわかった。また、SR法は動的負荷分散法の最適画面分割数の場合よりも高速であることがわかった。

キーワード レイトレーシング 並列処理 画面分割 負荷分散 ネットワーク 分散処理

Distributed Parallel Ray-Tracing reducing Network Overhead

Akira Honda[†] Jun'ichi Miyao[‡]

[†] The Graduate School of Engineering, Hiroshima University

[‡] The Faculty of Integrated Arts and Sciences, Hiroshima University

Abstract In case of parallel Ray-Tracing with screen subdivision algorithms, there are two techniques for load balancing. One is dynamic load balancing method (D-method) and the other is static one (S-method). In case of network distributed systems, We should consider network overhead. In this paper, we propose SR method for reducing network overhead and realizing load balancing. Then we have implemented D-method and SR on PVM, and examined their characteristics and execution time. The results show that there exists an optimum number of subdivisions in D-method which depends on complexity of a screen and the number of slaves, and SR is more effective than D-method.

keywords Ray-Tracing parallel screen subdivision load balance Network Distributed system

1 はじめに

近年、LAN による分散処理環境の普及に伴い、多くの計算機資源を同時に利用可能となっている。そこで、並列化可能な仕事は並列に分散処理をすることで、処理時間の短縮が期待できる。

レイトレーシング法は、高品質で写実的な 3 次元陰影画像を容易に生成できる反面、計算時間が膨大にかかることで知られている。しかし、計算単位は画面上の 1 ピクセル単位であるため、並列処理における処理時間短縮が可能であり、それにおける研究が行なわれている[1]~[7]。

並列化の手法の一つとして、画面分割法がある。これは生成する画像をいくつかの小画面に分割し、各プロセッサに割り当てる手法である。この方法では、画像内の各ピクセルごとの計算量が異なるため、各プロセッサの負荷を均一にする必要がある。負荷分散手法の一つである動的負荷分散法は、小画面を微小化することにより計算量を均一にしようとするものである。この方法では、微小画面数を増やすと各プロセッサ間での計算時間の均一化ははかられるが[3]、その分通信量が増大し、全体の処理の効率が下がる。

本研究では、分散処理環境でのレイトレーシング動的負荷分散法に改良を加え、通信量削減を考慮したアルゴリズム SR を提案し、従来の動的負荷分散法との比較実験を行った。実験はこれらの手法を PVM (Parallel Virtual Machine) [8][9] 上に実装し、分割画面数と処理時間の比較を行った。その結果、動的負荷分散法には、画像生成時間を最速にする最適な分割画面数が存在するが、その最適値はスレーブ数と画面内のピクセル値の計算量により変動することがわかった。また、SR 法は動的負荷分散法の最適分割画面数の場合よりも高速であることがわかった。

2 画面分割による並列化手法

レイトレーシング法の並列化手法として、画面をいくつかの小画面に分割し、各プロセッサ毎に小画面を処理する、画面分割法[3]がある。この方法では、各プロセッサはすべての物体データを必要とするが、各小画面に対して従来のレイト

レーシング法を使用できる。ここでは画面の分割方法について述べ、ネットワーク環境における分散処理を行なった場合の各並列化手法の問題点を述べる。

2.1 静的負荷分散法

静的負荷分散法は、画面内の負荷を予測し、その結果に従って各プロセッサの負荷が均一になるよう画面を分割する。各プロセッサは、これらの小画面の 1 つを受けとり計算を行なう。この手法では、小画面割り当てのための通信は 1 度でよいが、高速かつ完全に負荷を均一に分割することは難しい[4]。

2.2 動的負荷分散法

動的負荷分散法は、画面をプロセッサ数より十分多くの小画面に分割し、それらを動的に各プロセッサに割り当てる。この手法では、小画面の大きさを小さくすることにより、各プロセッサにおける負荷を均一にすることができるが、それに反して小画面割り当てのための通信回数が増大する。このため、一台のスレーブに対するマスターの処理遅延から、小画面割り当てに対する待ち行列が発生し、全体の処理時間が増加する。

2.3 ネットワーク分散処理

各負荷分散法により並列化されたレイトレーシング法をネットワーク環境において分散処理させた場合、通信にかかる時間を考慮しなければならない。各分割法における通信時間の増加を引き起こす要因を示す(表 1)。

静的負荷分散法では、小画面のサイズが大きいために、各スレーブがマスターに返す画像のピクセルデータの量が膨大となる。このため、これらの画像データをマスターが受けとり、実際の画像を生成する間の通信時間は無視できない。

動的負荷分散法では、スレーブからの処理終了メッセージに対して、次の小画面を割り当てる間、スレーブは処理を中断している。このため、ネットワーク負荷が高くなると、スレーブの空き時間が長くなり、全体の処理時間が増加する[5]~[7]。

表 1: 各分割法のメリット・デメリット (Merit, Demerit of Screen subdivision algorithm)

	動的	静的
通信量	×	○
負荷分散	○	×
マスターの負荷	×	○
画像生成時間の均一性	○	×

3 SR 法

ネットワーク環境下において、レイトレーシング法を分散処理させる場合、動的負荷分散法は通信量の増加、マスターの負荷の増加、静的負荷分散法では負荷分散の不均一、画像データ送信時間の増加などが欠点となる。ここでは、これら従来法のもつ欠点、通信量と負荷分散、ならびに画像データ送信時間を改善することを考慮した SR (Suspend and Redistribution) 法を提案する。

この手法での基本的な概念は、スレーブの数を小画面数と等しくして、処理要求を出す。計算を終了したスレーブが出ると他のスレーブを強制的に中断させ、現在の処理した画面ピクセル数をマスターに返させる。その後、マスターは残った画面をスレーブ数で再度割り、それらを各スレーブに発送することを繰り返す。

以下にマスターとスレーブのそれぞれの手順と、本手法の利点を述べる。

図 1 では、本手法の初期データ送信後のマスターとスレーブの動作を図解している。この図中の数字番号は以下の処理説明における番号部分である。また、図中の矢印は、細い矢印が一台のスレーブから、または一台のスレーブへのメッセージを表し、太い矢印は全スレーブから、または全スレーブへのメッセージを表している。

3.1 マスターの処理

- (1) 物体データファイルからデータを取り出し、初期データを計算した後、それを全スレーブに送信する。
- (2) 生成画像をスレーブ数で小画面に分割し、各スレーブに小画面範囲を送信する。送信のさい、各スレーブの識別 ID 番号と小画面範囲

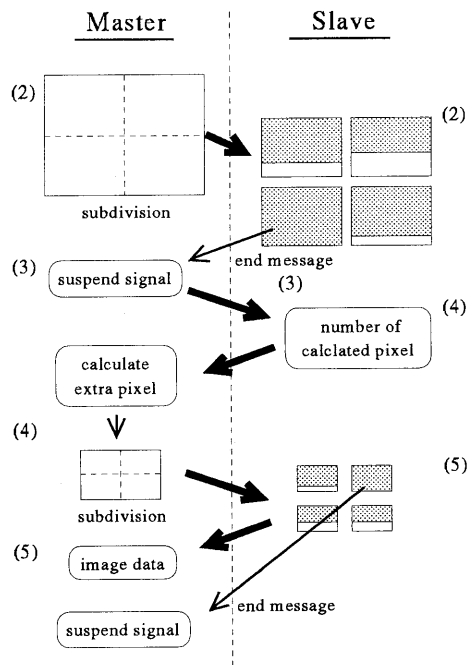


図 1: SR 法の動作 (movement of SR algorithm)

をデータベース化し、保存する。その後、いずれかのスレーブが処理終了メッセージを返すのを待つ。

- (3) 処理終了メッセージが届いたら、直ちに全スレーブに処理中断シグナルを送信する。
- (4) 全スレーブから返ってきたピクセル数により、データベースから各スレーブの残りのピクセルを集計する。集計された残りピクセルをスレーブ数分に分割し、全スレーブに再発送する。
- (5) 処理範囲を発送した後、スレーブからの小画面ピクセルデータ配列を受けとり、生成画像配列に格納する。
- (6) 分割される小画面の大きさが処理終了値に達すると、最後のピクセルデータを受けとり、全スレーブに処理終了メッセージを送信する。もし、処理終了値に達する前に全スレーブが要求範囲を終了させたなら、ピクセル値送信要求を出し、処理終了メッセージを送信する。

3.2 スレーブの処理

- (1) 初期データを受けとった後、マスターからの計算要求を待つ。
- (2) 始めの計算要求が来たら、直ちに計算開始。
- (3) 計算が終了すると、マスターに処理終了メッセージを送る。
- (4) マスターから処理中断シグナルが来ると即座に計算したピクセル値を返し、次の計算要求を待つ。
- (5) 2 回目以降の計算要求を受けとると、計算を始めるまえに前回計算したピクセル値配列をマスターに発送する。
- (6) 計算するピクセル数が規定値に達したら終了メッセージを送らず、ピクセル値配列を発送する。
- (7) 処理終了メッセージを受け取り終了。

3.3 本手法の利点

本手法は、従来手法における利点となる部分を取り込んでいる。SR 法の利点を以下に述べる。

- スレーブ一台に対する通信回数が減るため、全体の通信量が減らせる。
- スレーブからの計算データを、全体の処理時間に影響のない時間帯にマスターが受け取り、処理を行なえる。
- スレーブの処理を中断させることにより、各スレーブの負荷分散がほぼ均一になる。

4 実験結果

実験に使用した環境は、PC/AT 互換機 (OS:Linux, CPU:pentium133, memory:32MB) をスイッチングハブによりイーサネットで繋いだ 19 台のマシン上で行なった。

実験に用いたシーンは、画素数 1024×768, 物体数 80 個, 論理演算あり, 反射処理なしと言う条件で行なった (図 2)。

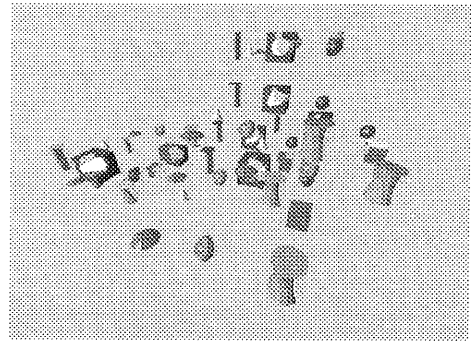


図 2: 実験に使用した画像 (Image of using research)

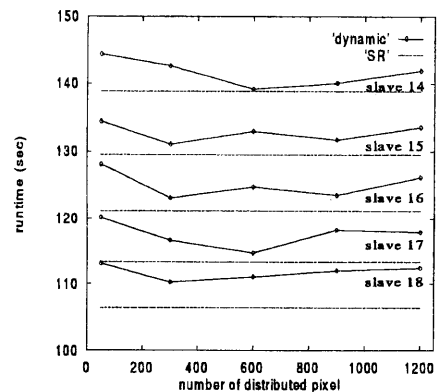


図 3: 小画面の大きさによる速度差 (1) (Speed for number of sending pixel (1))

表 2 は、動的負荷分散法の分配される各小画面の大きさ毎の画像生成時間と、SR 法の画像生成時間である。表 3 は、1 回目のスレーブの計算終了後の、処理中断から再分配までにかかった時間と、スレーブから 1 回目の処理結果を受けとり終るまでの時間、2 回目のスレーブの処理時間を示す。

図 3, 4 では、同一物体データで、ターゲット座標のみを変えた場合における各分散数の画像生成時間を示す。

図 5 では、表 2 における実験において、平均的に最も速かった分散ピクセル数 300 の動的負荷分散法と SR 法の画像生成時間を示す。

表 2: 動的負荷分散法と SR 法の画像生成時間 (Total time of Dynamic method and SR method)

	50 pixel	300 pixel	600 pixel	900 pixel	1200 pixel	1500 pixel	SR 法
Slave 2	972.01	950.83	950.84	944.18	943.87*	945.87	940.29
Slave 4	487.21	475.71	476.19	474.56	475.01*	476.79	471.41
Slave 6	325.24	320.03	318.02	319.98	320.96	317.09*	316.16
Slave 8	245.72	240.84	239.26*	243.81	240.53	244.22	237.86
Slave 10	197.12	193.14	192.34*	194.60	197.82	198.33	190.76
Slave 12	169.23	160.74*	162.62	164.51	165.20	162.06	160.57
Slave 14	144.36	142.60	139.15*	140.04	141.89	142.90	138.65
Slave 16	128.07	123.07*	124.70	123.48	126.14	127.49	121.08
Slave 18	114.11	111.23*	112.05	112.98	114.42	116.41	107.02

表 3: SR 法の各処理速度 (A partial speed of SR)

	再分配	格納	処理
slave 2	0.0104	13.8013	18.9433
slave 4	0.0114	11.3823	27.6001
slave 6	0.0135	10.7454	33.7729
slave 8	0.0200	10.9279	27.3950
slave 10	0.0235	11.3480	24.5684
slave 12	0.0291	11.3310	22.5668
slave 14	0.0368	11.1265	21.3410
slave 16	0.0423	11.2014	15.8467
slave 18	0.0480	11.3953	16.4708

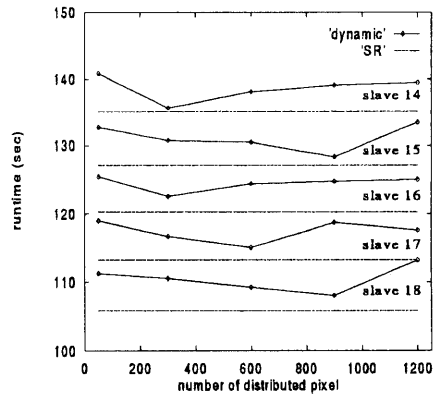


図 4: 画面の大きさによる速度差 (2) (Speed for number of sending pixel (2))

5 考察

図 3, 4 において, 動的負荷分散法では分散する数によって, 画像生成時間を最小にする小画面の大きさが変動する. さらに, 同一物体データでも, 視点の位置等を変えることにより, 同一分散数にもかかわらず最小値の変動が見られる. これは, 小画面の大きさにより通信量とスレーブの処理時間との釣り合いがとれ, 画像生成時間を最小とする最適値は存在するが, その最適値は分散数と画像内の各ピクセルの計算量の違いで変動することが考えられる. したがって, 最適値を完全に予測することは静的負荷分散法の場合と同様な理由で難しいと思われる.

図 5 において, 分散数 13 以降から動的負荷分散法と SR 法の間に速度差が生じている. これは, 動的負荷分散法が, 分散数の増加にとま

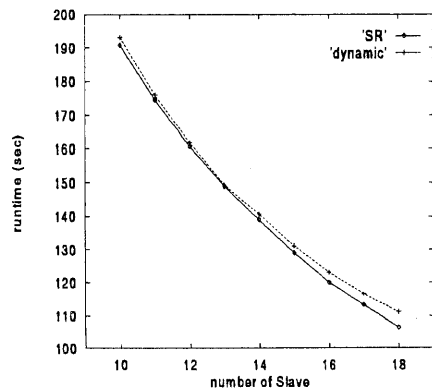


図 5: 画像生成時間 (Total computation time)

て、通信量とマスターにかかる負荷が増加したためと考えられる。

表 3 において、マスターがスレーブに対して処理中断シグナルを発生してから再分配までにかかる時間は、分配数の増加に伴い増えている。しかし、増加量は 0.01 秒にも満たない。これは、データ量が微小なため、通信負荷がほとんどかからなかったことと、マシン性能によりマスターの計算が高速であったためと考えられる。

また、画像データの返信にかかる時間は大きいですが、分散台数にあまり依存していない。それに対して、2 回目のスレーブの処理時間は分散数に比例して減少している。これは、スレーブ数が増えても一回の計算量の差は変わらなかったか微小なため、通信時間に影響を与えなかったことが考えられる。

6 おわりに

動的再分配法を用いた場合、分散数と画像内の計算量の違いにより、画像生成時間の最適値が存在する。しかし、最適値は分散数と画像内の各ピクセルの計算量に依存している。最適値に近い分配数の場合では、1 画像においては大きな時間差とはなっていない。しかし、レイトレーシングにより高品質なアニメーションを作成するような場合、画像ごとに各ピクセルの計算量が増え、かつ大量に画像を生成しなければならないため、大きな速度差につながる。この点において、本研究による SR 法は常に最速の画像生成時間が得られることから有効であるということがわかった。

また、SR 法の各処理時間において、マスターは一括した処理中断、収集を行なっているにも関わらず、それにかかる時間はデータ量が微少でマシンとネットワーク性能が高いため、非常に小さいことがわかった。

また、スレーブの台数が増えると 2 回目のスレーブの計算時間がさらに減少し、マスターがピクセル値データを格納し終る前に処理が終了することが予想される。

今後の課題として、より多くの台数を使っての実行時間を測り、どの程度でピクセル値データの格納とスレーブ処理時間の逆転が生じるのかを

調べたい。また、ネットワーク負荷が高い場合や、異なるマシン性能化での性能を調べたい。

参考文献

- [1] H.Kobayashi, H.Kubota, et al: "Effective parallel processing for synthesizing continuous images" Proc. of CG International'89, pp.343-352, 1989.
- [2] M.D.J.McNcil, et al: "Performance of Space Subdivision Techniques in Ray Tracing", *COMPUTER GRAPHICS forum*, pp.213 - 220, 1992.
- [3] 栗野, 河合 他: "画像生成システム MAGG における並列視線探索法とその評価", 信学論 (D-II), J76-D-II, pp.1763-1771, 1993.
- [4] 栗野, 河合 他: "画像生成装置 MAGG における並列視線探索法の負荷分散", 情処学グラフィックスと CAD 研報, pp.97-104, 1994.
- [5] 近内, 飯倉, 吉岡: "クライアント/サーバ型並列レイトレーシングモデルの解析", 平成 6 年度前期情処大会 5U-9, 1994.
- [6] 原田, 山崎, 渡部, 得丸: "画面分割と空間分割によるレイトレーシング法の高速度化", 平成 7 年度前期情処大会 1C-6, 1996.
- [7] 吉岡, 飯倉, 渡部: "クライアント・サーバ型情報処理環境における並列レイトレーシングモデルの解析", 平成 4 年度前期情処大会 6B-8, 1992.
- [8] A.Beguelin, et al: "A User's Guide to PVM (Parallel Virtual Machine)", Tech.Report ORNL/TM-11826, Oak Ridge Nat'l Laboratory, Oak Ridge, Tenn., 1991.
- [9] Sheue-Ling Chang, David Hung-Chang Du, et al: "Enhanced PVM Communication over a High-Speed LAN", IEEE Parallel & Distributed Technology Fall, pp.20-32, 1995