ヘミキューブ分割による並列ラジオシティ法

上嶋 明　　山崎 勝弘　　渡部 透　　得丸 英勝
立命館大学理工学部

ラジオシティ法による画像生成では，処理の大部分を占めるフォームファクタ計算の高速化を図ることが重要である．本研究では，フォームファクタの計算に用いるヘミキューブを各プロセッサに分割して並列化し，かつ負荷均衡を図るための静的 / 動的負荷分散法を考案した．4 種類のベンチマークシーンの計算を対象として，トーラス結合された 64 台の並列計算機トランスピュータ T805 上で実行した結果，プロセッサ数の増加とともに実行時間がほぼリニアに減少すること，及び動的負荷分散法がやや有利となることがわかった．さらに，並列化の効率はパッチ数に依存し，パッチ数が少ないほど良くなることが分かった．速度向上の点からは，プロセッサ 64 台の場合，1 台に対して平均で 28.3 倍の速度が得られた．また，現実的な部屋のシーンを処理した結果，同様に 30.3 倍の速度が得られた．

1　はじめに

ラジオシティ法は，物体表面を構成するポリゴンを細かいパッチに分割し，各々について光の出入りを計算して輝度を求める画像生成方法である．特徴は，物体間の光の相互反射を取り扱えるため，非常に現実感の高い画像を生成できることである [1]〜[3]．

ラジオシティ法が考案された初期の頃は光の平衡状態から得られる連立方程式を直接解いていたが，漸進法 [4] が発表されてからは，これが一般的に用いられるようになった．漸進法では徐々に解に近づけていくため，計算の途中でもそれなりの結果が得られる点，およびメモリを大幅に節約できる点が非常に有利である．ラジオシティ法ではフォームファクタと呼ばれる，パッチ間で光が到達する割合の計算が重要となる．これを積分計算で求めるのは非常に困難であるが，近似値を求める方法としてヘミキューブ法 [5] がある．一般に，ラジオシティ法ではフォームファクタの計算時間が全計算時間の 90%以上を占める．

本研究では，ラジオシティ法の処理を，トーラス結合されたマルチトランスピュータシステム（T805×64 台）上で並列化する方法と，その効果について述べる．並列化方法は，フォームファクタ計算の際にヘミキューブを分割して各プロセッサに処理を割り当て，ポリゴンの描画を行わせる方法を用いる．各プロセッサ間の負荷均衡を実現するために，全処理を単純に等分割し，インタリーブ方式による割り当てを行う静的負荷分散と，実行時に割り当てを行う動的負荷分散を用いる．

マルチトランスピュータシステム上での並列化の例として，ラジオシティの計算とレンダリングそれぞれを 8 台のプロセッサで並列処理した研究 [6] があるが，この例ではシーンのデータを完全に各プロセッサに分散させているため，ヘミキューブへの描画の際，必要なデータをプロセッサ間で転送する必要がある．それに対し，本研究でのシーンのデータの分散方法では，ポリゴンデータのコピーを全プロセッサに配置し，パッチデータを各プロセッサに分散させている．この方法では，座標のデータを各プロセッサ間で転送する必要が無く，かつ描画に無関係なパッチデータは各プロセッサへ分散できる．

本稿では，4 種類のベンチマークシーンの計算を対象として処理時間の測定を行い，ヘミキューブの解像度による静的負荷分散法と動的負荷分散法の比較，およびパッチ数を変化させた場合の比較を示し，考察を加える．さらに，現実的な部屋のシーンの処理時間を測定し，プロセッサ 64 台での速度向上を示す．

2　マルチトランスピュータシステム

本研究では動作クロック周波数 25MHz の T805 を 64 台接続したシステムを用いる．結合網として，$2\times2, 4\times4, 4\times8, 8\times8$ の 4 種類のトーラス接続，および 2 台ずつの組合せによる 2^5のハイパーキューブ接続を切り替えて使用できる．ローカルメモリは，1 台での逐次処理の場合にも使用する

プロセッサに 4M バイト，他の 63 台のプロセッサに 1M バイトを備えている．なお，計算を担当するプロセッサ群とは別に，外部と入出力を行うためのルート ・プロセッサを 1 台接続している．

本研究では言語として Parallel C を用いる．Parallel C は一般の C 言語に対して，プロセッサ間の通信やマルチスレッド処理を行うための関数などが拡張されている．スレッドとは，トランスピュータで実行される最小のプロセスで，複数のスレッドがデータを共有し，並行に実行される．マルチスレッド処理を行うことで，計算と通信を並列に行うことができる．

3 ラジオシティ法

3.1 計算手順

ラジオシティ法による画像生成の手順は，以下のようになる [1][3]．

1. ポリゴンで物体を定義する．ただし，その表面は理想的な拡散反射面とする．
2. 各ポリゴンを適当な大きさのパッチに分割する．このパッチがラジオシティの計算単位となる．
3. 光のエネルギーが平衡状態に達しているとして，すべてのパッチについて光の入射と反射に注目すると，次の式が成立する．

$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij} \qquad (1)$$

ここで，

B_i : パッチ i のラジオシティ・エネルギー
B_j : パッチ jのラジオシティ・エネルギー
E_i : パッチ i の自己放射エネルギー
ρ_i : パッチ i の反射率
F_{ij} : パッチ i からパッチ jへの フォームファクタ
n : パッチ数

4. 式 (1) より次の連立方程式が立てられ，これを解けば全てのパッチのラジオシティが求められる．

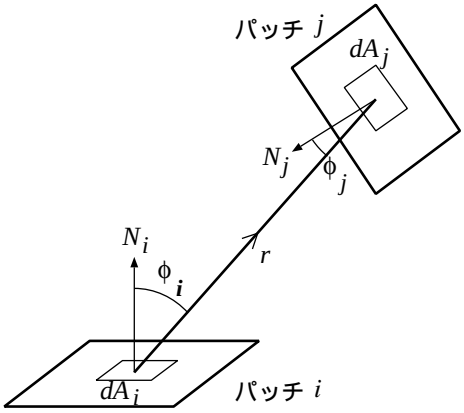

図 1 フォームファクタ

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_2 F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1-\rho_n F_{nn} \end{bmatrix} \cdot \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix} \qquad (2)$$

5. 求めた各パッチの中心のラジオシティから頂点のラジオシティに変換し，パッチ内部を線形補間してレンダリングする．

3.2 フォームファクタ

二つのパッチ i, jが図 1 のような位置関係にあるとき，フォームファクタ F_{ij}は，次のように表される．

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\phi_i \cos\phi_j}{\pi r^2} dA_j dA_i \qquad (3)$$

二つのパッチの間に障害物が存在するような複雑な位置関係のときに，この式によりフォームファクタを正確に計算するのは非常に困難である．そこで，高速に近似値を求める方法として，ヘミキューブ法が研究された．

3.3 ヘミキューブ法によるフォームファクタの計算

パッチ i を中心として，図 2 に示すようなヘミキューブ (半立方体) を考え，表面を適当な大きさのメッシュに分割する．この図 2 の解像度は 8

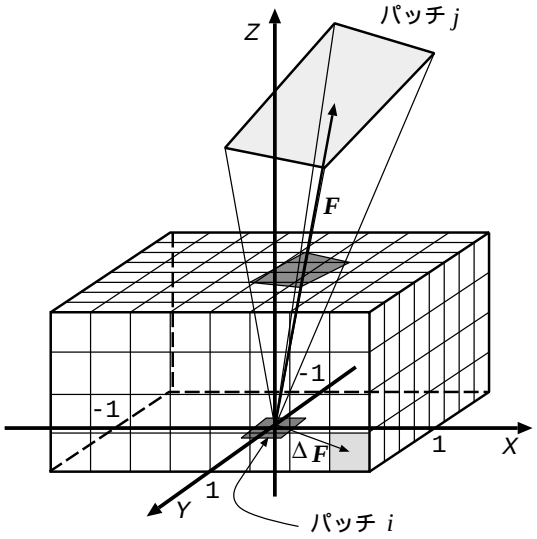

図 2 ヘミキューブ

である．次に，パッチ i から各メッシュへのファームファクタΔFを次の式を用いて計算する．

上面については，

$$\Delta F = \frac{1}{\pi(x^2+y^2+1)^2}\Delta A_i \qquad (4)$$

側面 $(y = 1, -1)$ については，

$$\Delta F = \frac{z}{\pi(x^2+z^2+1)^2}\Delta A_i \qquad (5)$$

この計算結果を後の計算のため，ルックアップ・バッファに入れておく．そして，Z バッファ法を用いて，パッチをヘミキューブに投影し，隠面消去によって距離が最も短いもののみを残す．ルックアップテーブルを用いて，パッチ jが投影されたメッシュのΔFの和を計算すると，F_{ij}が得られる [5]．

本研究ではパッチではなく，ポリゴンをヘミキューブに投影して隠面消去を行った後，各メッシュがポリゴン内の，どのパッチに対応するかを決定している．これは，ポリゴンの形状が長方形の場合，縦方向と横方向それぞれ等間隔に分割してパッチとしているため，ポリゴン内の任意の点の座標が分かれば，その点を含むパッチを決定できることを利用している．ポリゴンの描画とパッチの決定の処理が終わると，パッチを直接描画したのと同じ結果が得られ，続いてルックアップテーブルを用いたフォームファクタの計算を行う．この方法では，ヘミキューブへの投影数はポリゴン数に等しく，パッチ数とは無関係となる．

3.4 漸進法

式 (1) は各パッチの光の入射に注目しているが，逆に，各パッチの光の放射に注目し，他のパッチへの放射を繰り返すことで，全パッチのラジオシティの平衡状態に徐々に近づけていく方法である [4]．

放射に注目すると，以下の式ができる．

$$B_j = B_j + B_i(\rho_j F_{ji}) \quad (j = 1, 2, 3, \ldots, n) \qquad (6)$$

この式を，ラジオシティの大きいパッチ i から順に適用することで徐々に解に近づけていくことができ，これを収束するまで繰り返す．従来の連立方程式を解く方法では，計算の最中は式 (2) にある全フォームファクタを使用する必要がある．一方，この方法では一度に使用するフォームファクタは，行列の縦 1 列分のみでよいため，メモリを大幅に節約することが可能である．

4 ヘミキューブの分割によるラジオシティ法の並列化

データ配置は，ポリゴンデータのコピーを全プロセッサに持たせ，パッチデータを各プロセッサにインタリーブ方式に分散させている．例えば，プロセッサ 0~3 の 4 台で並列化する場合，プロセッサ 0 は図 3 の網掛部分のデータを保持する．全プロセッサがヘミキューブへの描画に必要な座標のデータを含むポリゴンデータを保持するため，座標などのデータをプロセッサ間で転送する必要は無い．さらに，描画の際に必要が無いパッチデータを各プロセッサに分散させることで，メモリの節約を行う．

並列化漸進法による計算は，図 4 のようになる．この中の部分フォームファクタの計算では，ヘミキューブの 5 面のメッシュを，図 5 の展開図の実線で示すように 1 ラインごとの領域に分割する．そして，各ラインを描画処理の単位として，各プロセッサへの割り当てを静的または動的に行う．各プロセッサでのヘミキューブへの描画後，部分フォームファクタのデータを，まずトーラスの横方向に，次に縦方向に加算しつつ転送を繰り返すことで全体の合計を求め，完全なフォームファクタを得る．全体としては，最大ラジオシティを持つパッチの検索，フォームファクタの計算，そして漸進法の

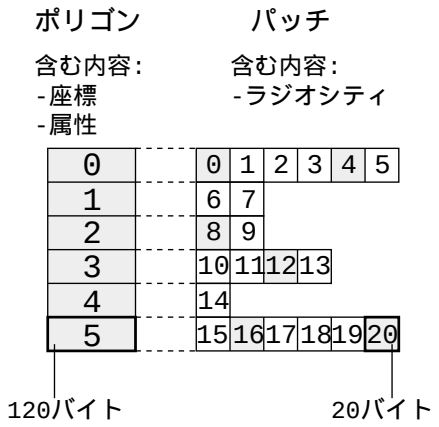

図 3 シーンデータの配置

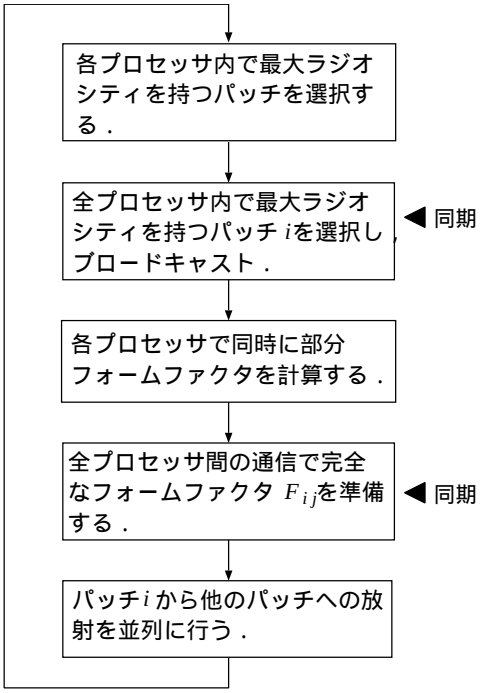

図 4 漸進法によるラジオシティ法の並列化

放射の過程を並列化している [7]．繰り返しの 1 回の中には，図 4 に示すように，全プロセッサが 2 箇所で同期をとることになる．

4.1 静的負荷分散

ヘミキューブを分割したものを，各プロセッサに均等に割り当てる．この際，物体の配置に偏りがある場合でも，なるべく各プロセッサに均等に負荷が分散できるようにインタリーブ方式で行う．まず，図 5 に示すように，分割したヘミキューブの領域にライン番号を振っておく．プロセッサを 0

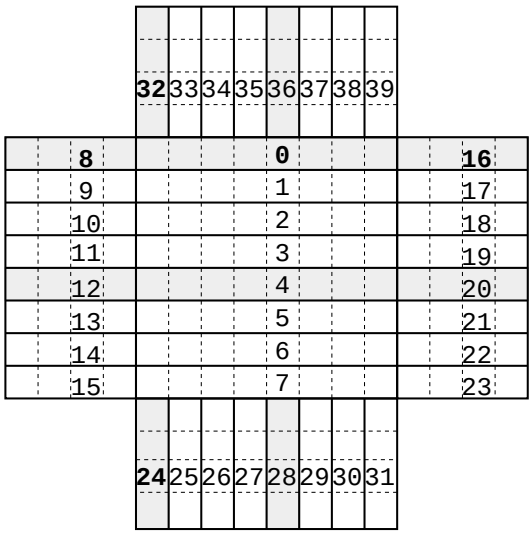

図 5 ヘミキューブの分割

番から $(Tn-1)$ 番までの Tn 台とすると，ヘミキューブのライン l は，$(l \bmod Tn)$ 番のプロセッサが計算を担当する．例えば，解像度 8 のヘミキューブをプロセッサ 4 台で並列化する場合，プロセッサ 0 はライン $0,4,8,\ldots,36$ の 10 箇所（図 5 の網掛部分）を担当することになる．

ヘミキューブの一つのメッシュに必要メモリな量は 8 バイトである．従って，解像度が 192 の場合，全体で 864k バイトのメモリが必要である．しかし，プロセッサ 64 台で並列処理の場合，必要なメモリ量はプロセッサ 1 台あたり $864/64 = 13.5$k バイトでよい．

4.2 動的負荷分散

各プロセッサでは，制御部，計算部の二つのスレッドを生成する．そして，各プロセッサの制御部の間でトークンを流す．トークンはライン番号と終了フラグを含んでいる．このライン番号により各プロセッサは計算するラインを指示される．終了フラグは，ヘミキューブのすべての部分の割り当てが終了したことを意味し，各プロセッサに部分フォームファクタの通信の開始を指示する．

動的負荷分散法においてプロセッサ数が多くなると，一つの処理が終わった後，次の処理が到着するまでに間隔が空くようになる．その対策として，グループ分割による動的負荷分散を用いる．すなわち，図 6 に示すように，トーラスの横 1 列のプロセッサを一つのグループとして，そのグループ

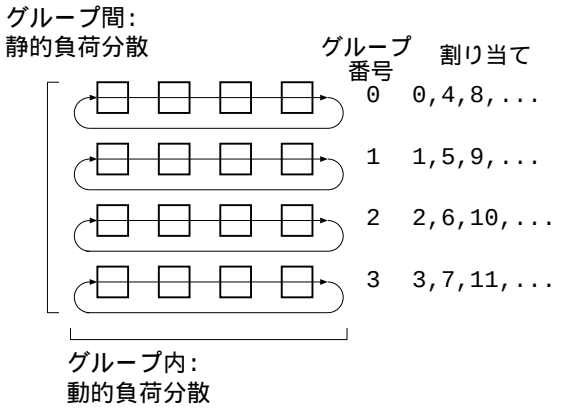

図 6 グループ分割による動的負荷分散

内で動的負荷分散を行う．各グループに対しては静的に処理を割り当てるので，グループ間静的負荷分散，かつグループ内動的負荷分散となる．

各グループにおいて，ライン番号は各々のグループ番号から開始する．そして，図 7 に示すように，計算中の場合はライン番号をそのまま次のプロセッサに送り出す．待機中の場合にはそのライン番号を計算部に伝えると同時に，グループ数を加えたものを次のプロセッサへ送り出す．このようにして，待機中のプロセッサに次々と処理を割り当てていく．

動的負荷分散法では各プロセッサの担当するライン数が不定のため，各プロセッサはヘミキューブ全体のためのメモリ量を確保している．

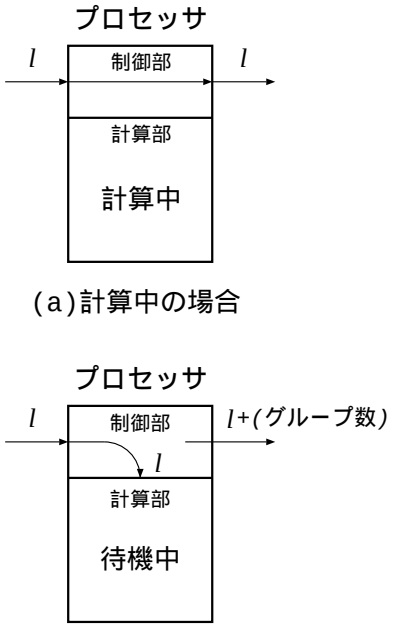

図 7 マルチスレッドによる動的負荷分散の制御

表 1 ベンチマークシーン

シーン	解像度	ポリゴン数	パッチ数
1	192	462	1074
2	160	462	1074
3	192	462	624
4	192	462	2874

5 並列化の実験と考察

5.1 実験

静的負荷分散，動的負荷分散の 2 通りの並列化方法で，ラジオシティ法の計算を実行する．1 台のプロセッサを用いた場合と，4，16，32，64 台のプロセッサを用いた場合，それぞれの処理時間を測定する．ここでの処理時間とは，データを全プロセッサに送り，ラジオシティの計算を行い，結果をルート・プロセッサに回収するまでの時間とする．

実行する内容は，表 1 に示すような 4 通りのベンチマーク用のシーンにおいて，漸進法の放射を 100 回行うものとする．全ベンチマークシーンに共通して，天井，床，4 枚の壁で囲まれた空間内に光源がある．天井と床そして 4 枚の壁のポリゴンは各々5×5 個のパッチに，光源の 6 個のポリゴンは，各々2×2 個のパッチに分割されている．そして，空間内に存在する 75 個の立方体を構成する 6 個のポリゴンのパッチへの分割数がシーンにより異なる．シーン 1 の計算結果をレンダリングしたものを図 8 に示す．なお，すべてのシーンはプロセッサ 1 台のローカルメモリ上で実行可能な規模としており，メモリ量が最大のシーン 4 は約 110k バイトである．

また，このベンチマークシーンとは別に，238 ポリゴン，661 パッチで構成される現実的な部屋のシーン (図 9) を，ヘミキューブの解像度 128，動的負荷分散法で処理する場合の速度向上の評価を行う．

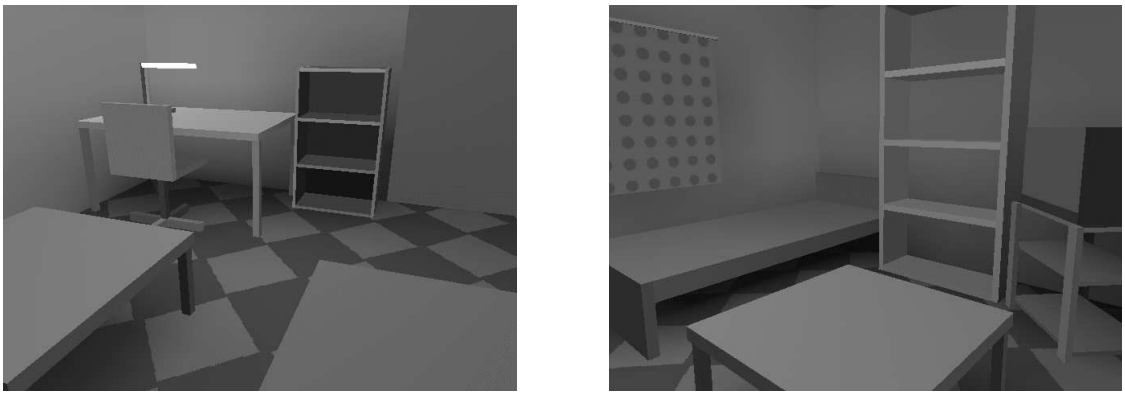

図 9　部屋 1

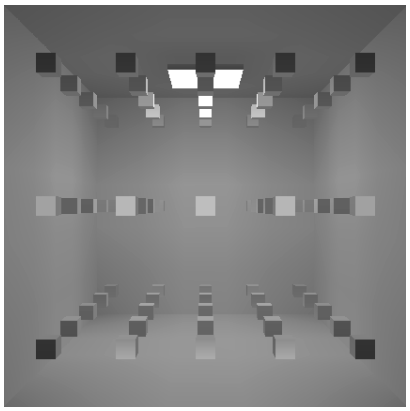

図 8　ベンチマークシーン 1

5.2　結果と考察

表 1 に示すベンチマークシーンの速度向上は，プロセッサ 64 台の場合，1 台に対して 22.8 ～ 33.6 倍，平均 28.3 倍の結果が得られた．また，図 9 に示す現実的なシーンを処理した結果，プロセッサ 64 台の場合で処理時間が 254.0 秒となり，プロセッサ 1 台に対して 30.3 倍の速度となった．

5.2.1　ヘミキューブの解像度による比較

シーン 1, 2 を用いたヘミキューブの解像度による比較を図 10 に示す．ここで，HcR はヘミキューブの解像度，LB は負荷分散法で，S が静的負荷分散，D が動的負荷分散を表す．図 10 より，プロセッサ数の増加とともに処理時間がほぼリニアに減少することが分かる．負荷分散法による比較では，解像度 160 でプロセッサ数 64 台の場合，差が約 12%と大きく表れている．これは，動的負荷分散では，160 がプロセッサ 64 台の場合のグループ数である 8 の倍数であるため，各グループの担当ライン数が等しくなるのに対し，静的負荷分散では 160 がプロセッサ数である 64 の倍数ではないため，各プロセッサの担当ライン数に差が生じるためと考えられる．なお，プロセッサ 32 台以下の場合は，静的負荷分散と動的負荷分散の差はほとんど表れていない．

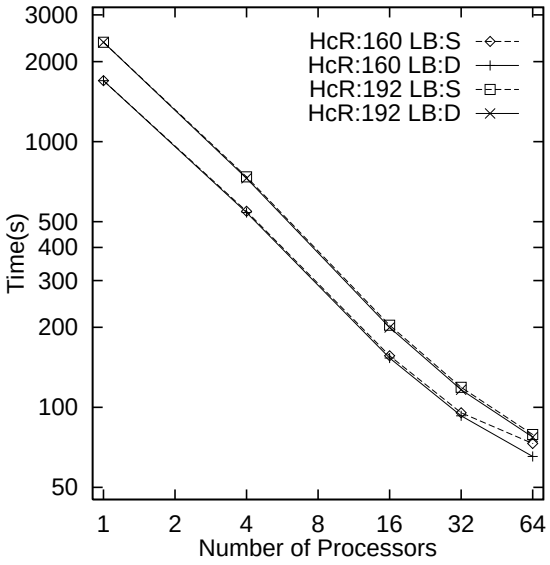

図 10　ヘミキューブの解像度による比較

5.2.2　パッチ数による比較

シーン 1, 3, 4 を用いたパッチ数による比較を図 11 に示す．ここで，Pt はパッチ数を表す．負荷分散法はすべて動的としている．図 11 より，プロ

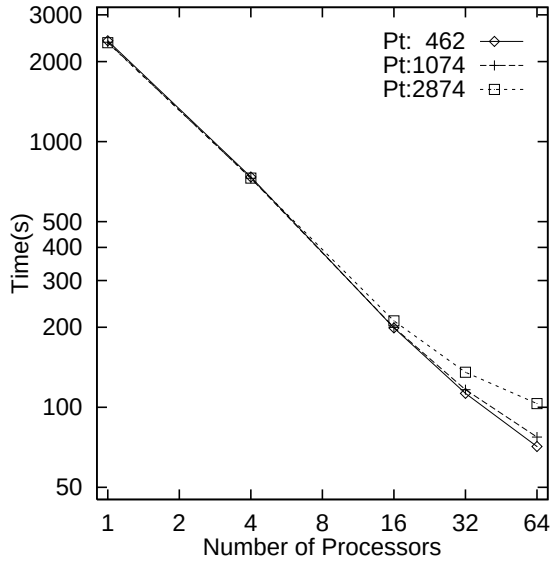

図 11 パッチ数による比較

セッサ 1 台の場合，処理時間はパッチ数とほとんど無関係であることが分かる．これは，3.3 節で述べたように，ヘミキューブにポリゴン単位で描画を行うからである．ここで，プロセッサ 1 台での処理時間が $T(1)$，n 台での処理時間が $T(n)$ の場合の並列化効率を $T(1)/nT(n)$ と定義すると，パッチ数が少ないほど並列化効率が高いことが分かる．これは，プロセッサ間の通信量がパッチ数に比例して増加するのに対し，計算量は一定であるため，パッチ数の増加とともに並列化効率が低下するものと考えられる．プロセッサ数が多くなるにつれて 1 台当りの計算時間の割合は小さく，また，通信時間の割合は大きくなるため，この現象はより顕著になる．

6 おわりに

マルチトランスピュータシステム上で，ヘミキューブを細分化してラジオシティ法の計算を並列化する方法と，その効果について述べた．

処理時間はプロセッサ数の増加とともにほぼリニアに減少すること，および，動的負荷分散法は，解像度がプロセッサ数の倍数でなくても，グループ数の倍数であるならば効率の低下が起こらず，静的負荷分散法よりも有効であることが分かった．また，並列化効率はパッチ数に大きく依存し，パッチ数が少ないほど良いことが分かった．

今後の課題として，パッチ数を増加させた場合の並列化効率の改善などがあげられる．

参考文献

[1] Goral C. M., Torrance K. E., Greenberg D. P. and Battaile B.: “Modeling the Interaction of Light Between Diffuse Surfaces”, *Computer Graphics(Proc. of SIGGRAPH ’84)*, Vol. 18, No. 3, pp. 213–222(1984).

[2] 西田友是, 中前栄八郎: “陰および拡散照度を考慮した 3 次元物体の陰影表示”, 情処学グラフィクスと CAD 研報, 14–2, pp. 1–8(1984).

[3] Cohen M. F. and Wallace J. R.: *“Radiosity and Realistic Image Synthesis”*, Academic Press(1993).

[4] Cohen M. F., Chen S. E., Wallance J. R. and Greenberg D. P.: “A Progressive Refinement Approach to Fast Radiosity Image Generation.”, *Computer Graphics(Proc. of SIGGRAPH ’88)*, Vol. 22, No. 4, pp. 75–84(1988).

[5] Cohen M. F. and Greenberg D. P.: “The Hemi-cube : A Radiosity Solution for Complex Environments”, *Computer Graphics(Proc. of SIGGRAPH ’85)*, Vol. 19, No. 3, pp. 31–40(1985).

[6] Feda M. and Purgathofer W.: “Progressive refinement radiosity on a transputer network, *Photorealistic Rendering in Computer Graphics(Proc. of 2nd Eurographics Workshop on Rendering)*, pp. 139–148(1994).

[7] 上嶋明, 山崎勝弘, 渡部透, 得丸英勝: “マルチトランスピュータシステム上でのラジオシティ法の並列化”, 並列処理シンポジウム JSPP’95, Vol. 95, No. 2, pp. 19–26 (1995).