

『熱視線』：視線探索法を高速処理する専用並列レンダリング・マシン
—メモリ構成およびその評価—

村田 誠治 †権 五鳳 村上 和彰 ‡富田眞治

九州大学 大学院総合理工学研究科
†九州大学 工学部 ‡京都大学 工学部

視線探索法を高速処理する専用並列レンダリング・マシンとして『熱視線』を現在開発している。視線探索法の光線処理過程においては、イメージ・コヒーレンスおよびレイ・コヒーレンスと呼ばれる処理の局所性が存在する。また、光線は進行方向にあるプリミティブに対してのみ交差判定を行う。そこで、このようなデータ参照の局所性に着目し、グローバル・メモリとプロセッサ間のアクセス時間を削減するために、次のような2階層3種類のメモリ階層構成を採った。

第1層：レジダント・メモリおよびキャッシュ・メモリ

第2層：グローバル・メモリ

本稿では、『熱視線』で採用したメモリ・アーキテクチャについて述べるとともに、その評価を行う。評価の結果、レジダント・メモリを採用することにより、実効的なメモリ・アクセス時間を低減できることが判明した。

A Paralell Rendering Machine for High Speed
Ray-Tracing
—Memory Structure and Evaluations—

Seiji MURATA, †Oubong GWUN, Kazuaki MURAKAMI,
and ‡Shinji TOMITA

Department of Information Systems
Interdisciplinary Graduate School of Engineering Sciences
Kyushu University
6-1 Kasuga-koen, Kasuga-shi, Fukuoka 816 JAPAN

†Department of Computer Science and Communication Engineering, Kyushu University
‡Department of Information Science, Kyoto University
E-mail: {murata,gwun,murakami}@is.kyushu-u.ac.jp

A parallel rendering machine for high speed ray tracing has been being developed at Kyushu University.

In the process of ray tracing, there is the locality of process, called image coherence and ray coherence, and the intersection of ray and primitives are tried to the only primitives in the direction of ray-traverse.

Utilizing the localities of reference, we try to reduce the effective access time of global memory, and then we adopt the hierarchical memory structure as follows.

- (1) Local memory, including cache memory and resident memory.
- (2) Global memory.

This paper presents the memory architecture, and evaluates it through simulation.

1 はじめに

視線探索法を高速に処理するレンダリング・マシンを現在開発中である。本システムは以下の3レベルで並列処理を取り入れている。

- ① マルチプロセッサ・レベル並列処理：光線に関する処理が画素単位に独立して行われることに着目し、各画素を要素プロセッサに分配して並列に処理する。
- ② プロセッサ・レベル並列処理 [1]：光線と物体との交差判定回数を削減するため、空間等分割法を採用する。これにより、一本の光線処理のタスクは、オーバラップ処理可能な3つのサブタスク（物体探索、交点計算、輝度計算）に分割できる。この点に着目して、光線に関するタスクをマクロパイプライン処理する。
- ③ 命令レベル並列処理 [3]：物体探索においては、ボクセル決定のためのインデックス計算が、3次元の各方向とも独立に計算可能である。交点計算においては、光線とポリゴンまたは2次曲面との交点を求めるための方程式の各係数が独立に計算可能である。しかも、対称性を持つ3次元のベクトル演算が主体である。このように、物体探索アルゴリズムおよび交点計算アルゴリズムの特徴に着目して命令レベルで並列処理する。

上記の各項目に関しては、先にソフトウェア・シミュレーションにより評価を行い、その有効性について報告した [2]。本稿では、参照の局所性を利用したメモリ構成について述べ、その評価を行う。

まず、2章で『熱視線』の概要について述べる。次に3、4章で、グラフィックス・データの特徴と、それを利用したメモリ・アーキテクチャについて述べる。そして5章で、ソフトウェア・シミュレーションによる評価結果を示し考察を行う。

2 『熱視線』の概要

図1に『熱視線』の全体構成を示す。フロントエンドにホスト・プロセッサを、また、バックエンドにフレーム・バッファを備える、メモリ共有型マルチプロセッサである。構成要素は、以下の通りである。

- ① 負荷分配器 (DIS)：ホスト・プロセッサで生成した1次光線を PE に分配する。また、ホストによる動的負荷分散・均衡化を可能とするために、各要素プロセッサの負荷を監視し、それをホストに伝播する機能を有する。
- ② ピクセル輝度値コレクタ (COL)：PE の光線処理結果 (ピクセル輝度値)を集めて、随時フレーム・バッファに書き込む。

- ③ グローバル・メモリ (GM)：光線処理過程で用いる4種類のデータベース (ボクセル・データ、プリミティブ・リスト、プリミティブ形状データ、および、プリミティブ輝度データ)のうち、交点計算ステージおよび輝度計算ステージに占有される3種類のデータベースを、それぞれステージ固有のグローバル・メモリに格納する。

- 交点計算用グローバル・メモリ (ICGM)：プリミティブ・リストおよびプリミティブ形状データを格納する。全 PE の交点計算器 (IC) により共有される。
- 輝度計算用グローバル・メモリ (SCGM)：プリミティブ輝度データを格納する。全 PE の輝度計算器 (SC) により共有される。

- ④ プロセッシング・エレメント (PE)：3ステージのマクロパイプライン構成を採る。個々のステージは、専用のプログラムおよびプログラム・カウンタを有する独立したプロセッサである。これらのプロセッサは、FIFO バッファを内蔵する通信ユニット (CMU) を介してリング状に接続される。

- 物体探索器 (OS)：整数演算ユニット 2 本および浮動小数点演算ユニットを 2 本を有する VLIW プロセッサである。レジスタファイルには、読出しポート 5、書込みポート 2 のマルチポート・レジスタファイルを使用する。
- 交点計算器 (IC)：整数演算ユニット 1 本および浮動小数点演算ユニットを 3 本を有する VLIW プロセッサである。レジスタファイルには、読出しポート 6、書込みポート 3 のマルチポート・レジスタファイルを使用する。
- 輝度計算器 (SC)：汎用機能ユニット 1 本を有する逐次プロセッサである。

- ⑤ 通信ユニット (CMU)：デッドロック・フリー、高スループット、低レイテンシーといった要件を満たすために、文献 [4] で提案されている“ワームホール・ルーティング+構造化バッファ・プール”方式を採用している。送信 FIFO バッファ 3 個、受信 FIFO バッファ 4 個 (内 1 個はホストからの通信用)、小容量のフリット・バッファ 9 個、ステージ間単方向リンク 3 本から構成される。

3 コヒーレンス

レイトレーシング・アルゴリズムに内在するコヒーレンスについて述べる。以下のコヒーレンスが存在する [5]。

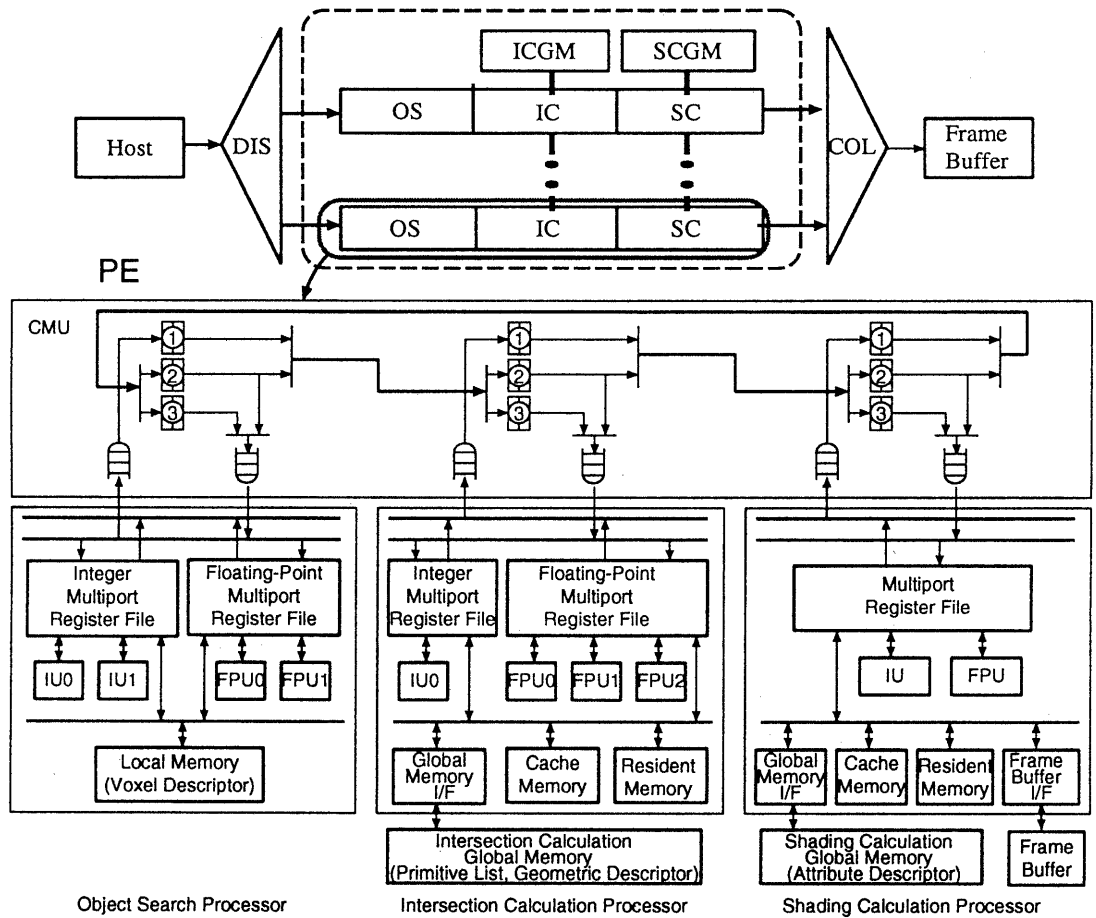


図 1: 「熱視線」

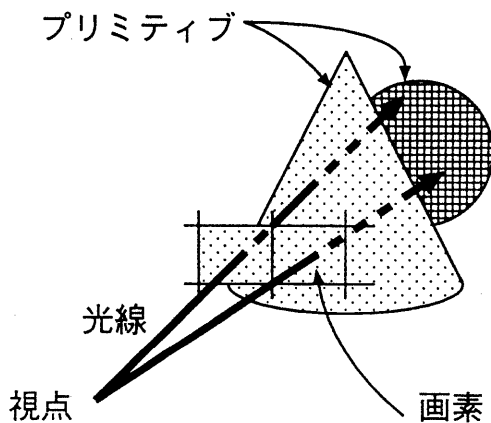


図 2: コヒーレンス

- ① イメージ・コヒーレンス (Image Coherence) : 画素に着目する。ディスプレイに最終的な画像を表示するためには、3次元空間上の物体を2次元平面に投射する必要がある。このとき、ある画素の近傍にある他の画素の輝度は同一である可能性が高い。
- ② レイ・コヒーレンス (Ray Coherence) : 光線に着目する。隣接した画素を通る光線は、方向ベクトルがほぼ等しいとみなせる。したがって、ある光線がプリミティブと交差したとき、隣接した画素を通る光線も同じプリミティブと交差する可能性が高い。

交差判定回数を削減するために、バウンディング・ボリューム、オクツリー分割法、空間等分割法といった方法が用いられる。これらの方法を用いると、光線とプリミティブとの交差判定が、ボクセルと呼ばれる部分空間に含まれるプリミティブに対してのみ行われる。つまり、任意の光線と

交差判定を行うために必要なプリミティブは、光線が通過する方向に存在するプリミティブのみである。このことは1次光線のみならず、交差した場合に生成される反射光線、透過光線および屈折光線といった分岐光線についてもあてはまる。さらに、イメージ・コヒーレンスやレイ・コヒーレンスにより、ある光線が交差判定処理を行う過程で参照されたデータは、隣接画素を通る光線にも参照される可能性がある。

このことは、データ参照に局所性 (locality of reference) が存在することを示している。よって、各PEとグローバル・メモリ間にキャッシュ・メモリを導入することにより、実効メモリ・アクセス時間を低減することが期待できる。

上記に加えて、隠面消去法には次の特徴がある。

- 映像としてスクリーン上に表示されるのは、光線と交差するプリミティブのうち、視点からもっとも近い位置にあるプリミティブである。
- 大きなプリミティブほど多くの光線と交差判定が行われる。

よって、ディスプレイ上に表示されるプリミティブは全プリミティブの一部分のみとなり、その前後左右にある多くのプリミティブは、光線と交差したプリミティブが反射、透過あるいは屈折面を持つときのみ参照される。また、大きなプリミティブほど光線との交差判定が頻繁に生じるため参照頻度が高くなる。

以上のことから、参照頻度の高いデータあるいは視点から可視なプリミティブのデータは、キャッシングすると同時にリプレース・アウトしないようにする必要がある。つまり、キャッシュ・メモリ内に常駐させたほうがよい。

4 『熱視線』のデータ・メモリ

4.1 データベース

光線処理には、以下の4種類のデータベースを必要とする。

- ① ボクセル・データ：物体探索ステージで用いる。ボクセル対応に設けられ、次の2つのフィールドから成る。
 - (a) フラグ：プリミティブの有無を示す。
 - (b) ボクセル識別子：プリミティブが存在する場合、対応するプリミティブ・リストの先頭へのポインタ。
- ② プリミティブ・リスト：交点計算ステージで用いる。プリミティブが存在するボクセル対応に設けられる単方向リストである。リストの各セルは、プリミティブ対応に設けられ、そのデータ部はプリミティブ形状データへのポインタとなっている。

- ③ プリミティブ形状データ：交点計算ステージで用いる。プリミティブ対応に設けられ、次の2つのフィールドから成る。

- (a) 形状値：交点計算に必要なデータ。
- (b) プリミティブ識別子：プリミティブ輝度データへのポインタ。

- ④ プリミティブ輝度データ：輝度計算ステージで用いる。プリミティブ対応に設けられ、輝度計算に必要な色、反射率、透過率、屈折係数、等のデータを保持する。

これら4種類のデータベースのメモリ配置にあたっては、以下の方針を採った。

- ① ボクセル・データ：
物体探索器 (OS) にて読み出されるボクセル・データ量の大小は、プリミティブの数よりは物体空間の分割数に依存する。さらに、この分割数は任意に定められる。したがって、ボクセル・データに関しては、グローバル・メモリを設けずに、各OSのローカル・メモリに完全なコピーを格納する。このローカルメモリの容量を上限として、その範囲内で空間分割数を調整する。
- ② プリミティブ関連データベース：
プリミティブ・リスト、プリミティブ形状データ、および、プリミティブ輝度データのデータ量は、モデリング・データに比例して増加する。したがって、ボクセル・データのように完全な同一コピーを各PEのローカル・メモリに格納するのは困難である。そこで、プリミティブ・リストおよびプリミティブ形状データについては交点計算用グローバル・メモリ (ICGM)、また、プリミティブ輝度データについては輝度計算用グローバル・メモリ (SCGM) と呼ぶ大容量のグローバル・メモリに各データベースを格納する。すなわち、各データベースはグローバル・メモリ中にしか完全なコピーは持たず、それが全PEに共有される。

4.2 構成

3章で述べた特徴から、交点計算器 (IC) および輝度計算器 (SC) それぞれに対して、次のような2階層3種類のメモリから成る構成を採る。

- ① 第1層：ローカル・メモリ
 - レジデント・メモリ：参照頻度の高いデータの完全なコピーを各PEに格納する。これらのデータのコピーは、グローバル・メモリに格納する必要はない。レジデント・メモリは、以下の条件を満足する必要がある。

- レジデント・メモリへの書込み権はホストのみに与える。各PEは読出し権のみ有する。
- レジデント・メモリへ書き込まれたデータは、処理が終了するまでリリース・アウトしない。
- キャッシュ・メモリ：グローバル・メモリから必要なデータのコピーを適宜キャッシングする。キャッシュ・メモリの諸元を以下に示す。
 - アクセス・アドレス：仮想アドレス
 - 連想度：ダイレクト・マッピング
 - ブロック・サイズ：64B

光線処理過程においては、プリミティブ・データに対する書込みは行われな
ない。つまり、プリミティブに関する全データベースが read-only であるので、
キャッシュの一貫性については保証する必要がない。

- ② 第2層：グローバル・メモリ
レジデント・メモリに格納すべきデータ以外をすべて格納する。

4.3 レジデント・メモリに格納するデータの選択

レジデント・メモリに格納すべきプリミティブ関連データベースとして、参照頻度の高いものを選択しなければならない。その1方法として、前処理の段階で、粗めの解像度で交差判定を行い、参照頻度を予め求めることにする。処理の流れを以下に示す。

- 前処理による結果として、各プリミティブの参照回数を出力する。
- この結果を元に、参照回数の大きなものから順に、プリミティブをソートする。
- ソートしたのち、各プリミティブの優先順位を決定する。優先順位は、もっとも参照回数が多いものを1とし、以後2, 3, ... とする。同時に、プリミティブ・リストに優先順位を追加する。
- 優先順位に基づいて、上位から順にグローバル・メモリからレジデント・メモリへプリミティブ関連データベースのコピーを格納する。
- レジデント・メモリの容量に達したなら処理は終了する。

このとき、前処理に要する処理時間が問題になる。たとえば、1000×1000の解像度で画像を生成する場合を考える。前処理段階では100×100の解像度でレイトレーシング処理を行うとすると、前処理に要する時間は、本処理に要する時間の $\frac{1}{100} = 1\%$ 程度となり、このオーバーヘッドは無視できると判断した。

表 1: ベンチマーク・シーン

ベンチマーク・シーン	tetra	balls	mount	rings
プリミティブ数	4096	7382	8196	8401
ポリゴン	4096	1	8192	1
球	0	7381	4	4200
円錐	0	0	0	0
円柱	0	0	0	4200
光源数	1	3	1	3
反射面	無	有	有	有
透過面	無	無	有	無

4.4 ローカル・メモリ・アクセス

キャッシュ・メモリあるいはレジデント・メモリのどちらへアクセスするかは、プリミティブ優先順位とレジデント・メモリに格納できるプリミティブ数を比較することにより決定する。いま、レジデント・メモリに格納できるプリミティブ数を M 、プリミティブ i の優先順位を P_i とする。

- プリミティブ i の優先順位をレジデント・メモリから読み出す。
- プリミティブ i の優先順位 P_i とレジデント・メモリに格納できるプリミティブ数 M を比較する。
 - $P_i \leq M$: レジデント・メモリをアクセスする。
 - $P_i > M$: キャッシュ・メモリをアクセスする。

5 評価および考察

「熱視線」で採用したメモリ構成について、交点計算器に限ってソフトウェア・シミュレータを用いて評価を行った。

5.1 評価方法

本シミュレータは、「熱視線」の動作を忠実にモデル化している。連想度は1(ダイレクト・マッピング方式)とし、キャッシュ・メモリ容量およびレジデント・メモリ容量は任意に変更可能とした。空間分割法として、文献[6]で提案されている2階層空間等分割法を用いた。

評価項目は、次の通りである。

- プリミティブに対する参照の局所性：
 - 各プリミティブの参照回数
 - 1次光線と交差するプリミティブ数
- キャッシュ・ヒット率：
キャッシュ・メモリの容量を各ベンチマーク・

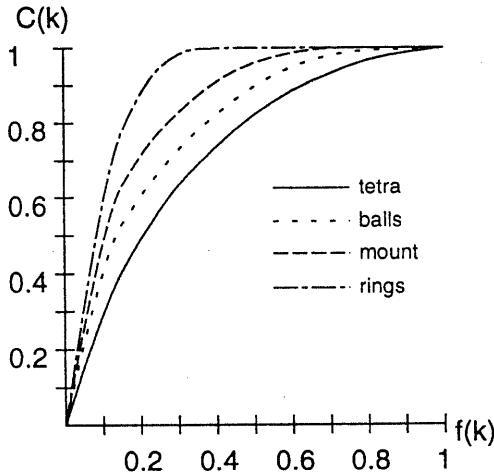


図 3: 参照の局所性

表 2: 視点から可視なプリミティブ

ベンチマーク・シーン	tetra	balls	mount	rings
プリミティブ数	4096	7382	8196	8401
可視プリミティブ数	2121	3072	1903	2571
可視プリミティブの割合	51%	42%	23%	29%
可視プリミティブの参照率	56%	73%	57%	77%

シーンにおけるプリミティブ関連データベース容量の 10%, 20%, ..., 100%と設定したときのキャッシュ・ヒット率を測定する。

- レジデント・メモリの効果：
ローカル・メモリの容量を各ベンチマーク・シーンにおけるプリミティブ関連データベース容量の 10%, 30%, 50%と設定する。このとき、ローカル・メモリ内でレジデント・メモリとキャッシュ・メモリの容量比を変化させたときのレジデント・メモリの参照率とキャッシュ・ミス率を測定する。

解像度は 512 × 512 とした。ベンチマーク・シーンには、Hains の SPD モデル [7] のうち、tetra, balls, mount, rings の 4 つのシーンを用いた (表 1 参照)。

5.2 評価結果

5.2.1 参照の局所性

図 3 に、プリミティブに対する参照の局所性に関する評価を示す。図 3 の縦軸の $C(k)$ 、および、横軸の $f(k)$ は、以下の式から求めた。

$$C(k) = \frac{\sum_{i=1}^k REF(i)}{\sum_{j=1}^N REF(j)}$$

$$f(k) = \frac{k}{N}$$

- N : プリミティブの総数
- k : 参照回数の多い方から数えた際の順位 ($1 \leq k \leq N$)
- $REF(i)$: プリミティブ i の参照回数 ($REF(1) \geq REF(2) \geq \dots \geq REF(N)$)
- $f(k)$: 上位 k 個のプリミティブ数 (k) がプリミティブ総数 (N) に占める割合
- $C(k)$: 上位 k 個のプリミティブの参照回数の合計が全体の参照回数に占める割合

図 3 より、全ベンチマーク・シーンとも参照がある特定のプリミティブに偏っていることが判る。特に、mount と rings では全体のわずか 20% のプリミティブが全参照回数の 70% ~ 80% を占めている。さらに、 $C(k) = 1$ となる $f(k)$ の値より、まったく参照されないプリミティブ (交差判定を行わないプリミティブ) が、mount では 20%、rings では 67% も存在する。

また、表 2 から、tetra および balls は全プリミティブの 40% ~ 50% が、mount および rings は 20% ~ 30% が視点から可視なプリミティブであることが判る。つまり、最終的に表示されるプリミティブが全体に占める割合はそれほど大きくない。視点から可視なプリミティブの参照率に着目してみると、balls および rings では全体の 70% を占めているが、tetra および mount では 50% となり、ばらつきがあることが判る。しかし、どのベンチマーク・シーンとも視点から可視なプリミティブの参照率は割と高く、3 章で述べた仮定の正しさを裏付けている。

以上のことから、全プリミティブの 20% ~ 50% を占めるプリミティブとその周囲に存在するプリミティブにデータ参照が偏っていることが確認できた。よって、キャッシュ・メモリの導入による実効アクセス時間の低減が期待できる。

5.2.2 キャッシュ・ヒット率

図 4 に、各ベンチマーク・シーンでのキャッシュ・ミス率を示す。図 4 の縦軸はキャッシュ・ミス率を、そして、横軸はキャッシュ・メモリ容量 (各ベンチマーク・シーンのプリミティブ関連データベース容量の $n\%$) を表す。これより次のことが判る。

- どのベンチマーク・シーンとも、キャッシュ・ミス率は低い。つまり、キャッシュ・ヒット率が高い。
- キャッシュ・メモリ容量がプリミティブ関連データベース容量の 30% を越えた時点で、ヒット率は飽和する。

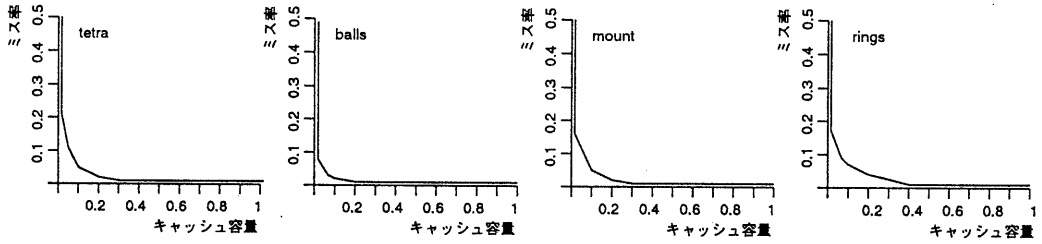


図 4: キャッシュ・ミス率

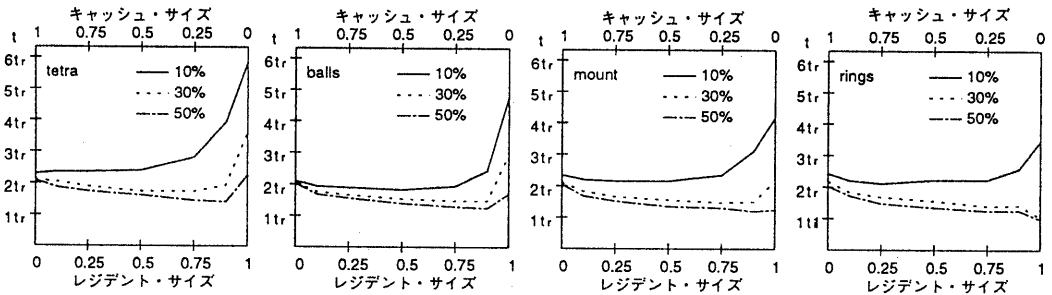


図 5: 実効メモリ・アクセス時間

以上から、キャッシュ・ヒット率はかなり高いことが判り、実効メモリ・アクセス時間の低減が可能である。

5.2.3 レジデント・メモリの効果

実効メモリ・アクセス時間 t を次の式で定義する。

$$t = \rho t_r + (1 - \rho) \{ \mu t_c + (1 - \mu) t_g \}$$

- ρ : レジデント・メモリの参照率
- μ : キャッシュ・ヒット率
- t_r : レジデント・メモリのアクセス時間
- t_c : キャッシュ・メモリのアクセス時間
($t_c = 2t_r$)
- t_g : グローバル・メモリのアクセス時間
($t_g = 8t_r$)

上式は以下のように変形できる。

$$t = t_r \{ \rho + (1 - \rho)(8 - 6\mu) \}$$

図 5 に実効メモリ・アクセス時間を示す。図 5 の縦軸は実効メモリ・アクセス時間を、横軸はローカル・メモリ内でのレジデント・メモリとキャッシュ・メモリの割合を示す。これより以下のことが判る。

- ローカル・メモリ内に占めるレジデント・メモリの割合を大きくするほど、実効メモリ・アクセス時間 t は小さくなる傾向にある。特に、レジデント・メモリの割合が 70% のあた

りで最小になる。ただし、ローカル・メモリ容量が、プリミティブ関連データベース容量の 10% の場合は、レジデント・メモリの割合が大きくなるほど、実効メモリ・アクセス時間は増加する。その理由として以下のことが考えられる。

- レジデント・メモリの割合を大きくすることにより、レジデント・メモリの参照率 ρ は大きくなるが、キャッシュ・メモリの割合は小さくなり、キャッシュ・ミス率 $(1 - \mu)$ は大きくなる。しかも、図 3 より、データ参照回数の多いプリミティブ 10% 程度では、参照率 ρ はそれほどよくない。したがって、レジデント・メモリの割合を大きくしても、キャッシュ・ミスの影響が無視できなくなり実効メモリ・アクセス時間は増加する。
- ローカル・メモリの容量を大きくするほど、実効メモリ・アクセス時間 t は減少する。その理由として以下のことが考えられる。
 - ローカル・メモリの容量を大きくすることは、当然レジデント・メモリおよびキャッシュ・メモリの容量を大きくすることにつながる。また、図 3 から判るように、データ参照のほとんどが全プリミティブの 30%~50% のプリミティブに集中している。したがって、ローカ

ル・メモリを大きくすることで、参照されるデータのほとんどがローカル・メモリに格納される。つまり、レジデント・メモリの参照率 ρ およびキャッシュ・ヒット率 μ が大きくなり、実効メモリ・アクセス時間 t が減少する。

- ローカル・メモリの容量がプリミティブ関連データベース容量の30%の場合と50%の場合では、実効メモリ・アクセス時間 t にそれほど差がない。これは、前述したように、ローカル・メモリの容量がプリミティブ関連データベース容量の30%を越えると、参照されるプリミティブのほとんどがローカル・メモリ内に格納されてしまうからである。よって、ローカル・メモリの容量は、コスト対性能比を考えると、プリミティブ関連データベースの30%程度が最適であると思われる。

6 おわりに

以上、レイトレーシング・アルゴリズムに内在するコヒーレンスを利用したメモリ構成について述べ、それについて評価を行った。その結果、次のことが判明した。

- 参照の局所性：レイトレーシング・アルゴリズムでは、プリミティブ参照が全体の30%~50%のプリミティブに偏っていることが判った。
- キャッシュ・ヒット率：キャッシュ・ヒット率が高いことが確認でき、実効メモリ・アクセス時間の低減が可能である。
- レジデント・メモリの効果：
 - － レジデント・メモリを導入したことにより、実効メモリ・アクセス時間はキャッシュのみの場合よりも減少した。特に容量が、レジデント・メモリ:キャッシュ・メモリ=3:1のときに最小となった。ただし、レジデント・メモリの参照率が低いときにはほとんど効果が現れなかった。
 - － ローカル・メモリ容量を大きくするほど実効メモリ・アクセス時間は減少した。しかし、ローカル・メモリ容量がプリミティブ関連データベース容量の30%の場合と50%の場合では実効メモリ・アクセス時間にそれほど差がなかった。したがって、ローカル・メモリの容量としては、グラフィックス・データの30%ほどが、コスト対性能比を考えると最適である。

今回用いたベンチマーク・シーン (SPD モデル) は、通常のグラフィックス・データと比べるとデータ・サイズが小さい。したがって、もっとサイズの大きいグラフィックス・データを用いて評価を行い、レジデント・メモリの有効性を確認する必要がある。

今後は、PE間の負荷分散・均衡化の方法などの問題を解決し、その結果を考慮しながら設計を行っていく予定である。

謝辞

日頃ご討論頂く九州大学 大学院総合理工学研究科 安浦寛人教授、および、安浦研究室の諸氏に感謝致します。

参考文献

- [1] 権五鳳, 村上和彰, 富田真治:「熱視線」: 視線探索法を高速処理する専用並列レンダリング・マシン—マクロパイプライン・アーキテクチャー, 情処研報, ARC-85-6, 1990年11月.
- [2] Gwun, O., Murata, S., Murakami, K., and Tomita, S.: A Parallel Rendering Machine For High-Speed Ray Tracing, *Proc. International Symposium on Supercomputing'91*, pp.173-182, Nov. 1991.
- [3] 村田誠治, 権五鳳, 村上和彰, 富田真治:「熱視線」: 視線探索法を高速処理する専用並列レンダリング・マシン—マクロパイプラインの各ステージにおける命令レベル並列処理—, 並列処理シンポジウム'91 論文集, pp.109-116, 1991年5月.
- [4] 堀江健志, 池波守夫, 石畑宏明: 並列計算機CAP-IIのルーティング・コントローラ, 情処研報, ARC-83-38, 1990年7月.
- [5] Green, S, A., Paddon, D, J.: Exploiting Coherence for Ray Tracing, *IEEE Computer Graphics & Applications*, vol.9, no.6, pp.12-26, Nov 1989.
- [6] 権五鳳, 村田誠治, 村上和彰: 3MPRT (3-stage Macropipelined Ray Tracing) マシン—物体探索法の高速化に関する評価—, 信学技報, CPSY-92-19, 1992年8月.
- [7] Hains, E.: A Proposal for Standard Graphics Environments, *IEEE Computer Graphics & Applications*, vol.7, no.11, pp.3-5, Nov. 1987.