

並列処理方式による三次元色彩図形表示用プロセッサ

A Parallel Processor Architecture for 3 Dimensional Graphics

新実 治男[†], 今井 慈郎[‡], 村上 昌義[‡], 富田 真治[†], 萩原 宏[†]

Haruo NIIMI, Yoshino IMAI, Masayoshi MURAKAMI, Shinji TOMITA, Hiroshi HAGIWARA

[†]京都大学・工学部,[‡]詫間電波高専,[‡]日本電子科学(株)

Faculty of Engineering,

TAKUMA Radio Technical College, Nippon Denshi Kagaku Co.Ltd.

KYOTO University

[1] はじめに

半導体技術とTV技術の発展により、従来のランダム・スキャン型に代わり、ラスター・スキャン型のディスプレイを使用する図形処理システムが主流となり、これに伴って、表示する図形も、ワイヤ・フレーム・モデルなどの線図形から、サーフェス・モデルなどの面図形が中心となりつつある。中でも、ソリッド・モデル^[1]に見られるように、色彩や明暗の変化によって実在感(reality)溢れる3次元図形の表示を行なうことは、計算機アニメーションやCADをはじめとする多くの応用分野と開拓する技術として、注目を集めてきている。

しかし、3次元図形の表示のためには、座標変換計算、隠面消去処理、陰影付けの処理など膨大な計算処理が必要であり、さらに、ディスプレイ・モニタの高解像度化、表示の対象とする図形の複雑化、および表示図形の実在感の高度化といった要求に伴い、その処理量は益々増大する方向にある。このような高速処理能力が要求される応用においては、現在商品化されている汎用ミニコンや各種図形処理端末では能力が不足しており、何らかの並列処理あるいはパイプライン処理を導入した図形処理専用のプロセッサが必要不可欠である。このような研究としては、Fuchs らのマルチプロセッサ・システム^[2]、Roman らのVLSI指向パイプライン・アーキテクチャ^[3]などがある。この2例はいずれも3次元図形処理における隠面消去処理のアルゴリズム: depth-buffer 法を並列に実行するものである。国内で

は、吉良による疎結合マルチ・マイクロプロセッサによる並列図形処理の報告^[4]がある。

我々は、実在感溢れる3次元色彩図形表示を、実時間で実行することを設計目標とした、並列処理方式による図形処理専用プロセッサ:

EXPERTS (EXpandable Parallel processors Enhancing Real-Time Scan conversion) の開発を行なっている。

EXPERTS は、2層の階層構成と採るマルチプロセッサであり、3次元図形処理における隠面消去処理を並列に実行する。さらに、曲面の実在感を表出するための輝度補間処理、ラスター・スキャン型ディスプレイに固有に発生する階段状のギザギザ(jag)を抑制するための境界線平滑化処理、および、表示画面全体の現実感を増すための付影処理などを実行する。

本報告では、EXPERTS が採用している3次元図形処理アルゴリズムと、EXPERTS を含む全体のシステムの概要について述べた後、EXPERTS のアーキテクチャの特徴と並列処理方式について詳述する。

[2] 3次元図形処理システム

2.1 システム構成と処理の概要

3次元図形処理は大きく分けて次の2段階の処理から成る^[5]:

- i) 図形データに対するモデリング変換、視野変換、透視変換(または平行投影変換)などの《座標変換処理》;
- ii) 幾何学的(geometrical)、位相幾何学的(topological)に表現されている図形データを、画素

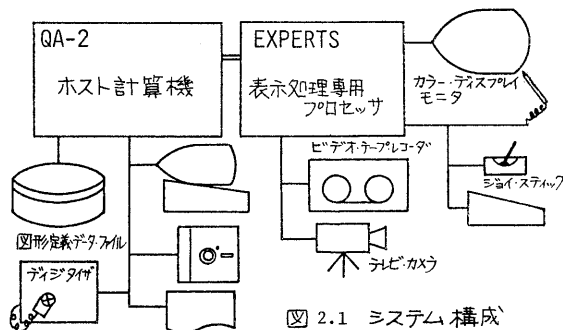


図 2.1 システム構成

(pixel) の 2 次元配列: フレームバッファ (Frame Buffer, 以下 FB と記す) 上のデータに変換する《表示処理》。

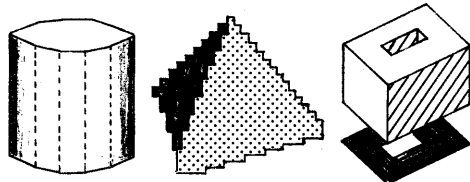
i) と ii) は互いに処理内容が大きく異なり、かつ明確に分離可能な関係にある。そこで我々は、i) と ii) の処理をそれぞれ個別の計算機に分担させ、双方が独自に処理の最適化を行なうことにより、3 次元図形処理全体の処理効率の向上を図っている。図 2.1 にシステム構成の概略を示す。

i) の座標変換処理は、座標値やバクトル値に対する行列演算が中心であり、図形データ量の増加に対する処理量の増加は線形であると考えられる。図 2.1 に示した QA-2 は、4 個の 32 ビット ALU と大容量レジスタ・ファイルとを装備し、ALU-レジスタレベルの並列処理を実行できるダイナミック・マイクロプログラム制御計算機^[6]があり、4 次の配列演算などを高速に実行することができる。

一オ ii) の表示処理は、3 次元図形処理の中心となる隠面消去処理 (Hidden Surface Elimination, 以下 HSE と記す) を含んでいる。HSE は一般に、ソーティングや多数の数値比較の処理を必要とするので、表示図形の複雑化に伴って急激に処理量が増加する。また表示処理は FB の総ての画素データ (色情報) の値を決定し書き込みを行なうという、大量データ処理でもある。特に、実在感のある 3 次元図形表示を行なうためには、HSE に加えて以下のような処理が必要である:

- a) 曲面の実在感を表出するために、連続する画素の輝度 (intensity) を滑らかに変化させる処理: Smooth Shading (以下 S.S. と記す);
- b) ラスタスキャン型ディスプレイに発生する階段状のギサギサ (jag) を和らげ、奥和感の少ない表示を行なうための、境界線平滑化処理: Anti-Jagging (以下 A.J. と記す);

の表示画面全体の実在感を増やすと同時に、各図形の形状をより詳しく表現するために、図形の影 (shadow) と表示する付影処理: Shadowing; d) その他、透明感 (transparency) や光の屈折 (refraction)、反射 (reflection) の表示、あるいは図形表面の質感 (texture) の表示など、高度の現実感を表わすための処理。(以上、図 2.2 参照)



a) Smooth Shading b) Jag c) Shadowing

図 2.2

これらの処理と行なった場合、1 画素の値を決定するために多くの計算時間を費やすことになる。従って、表示処理が 3 次元図形処理全体に占める比重はかなり大きく、この部分に並列処理を導入して処理効率を上げることは、全体の高速化を図る上での必須の条件である。従って、EXPERTS の設計にあたり、SSE に加えて、上述の S.S., A.J., および shadowing の処理と並列に効率よく実行できるよう配慮している。

2.2 EXPERTS における隠面消去処理 (HSE)

我々の図形処理システムでは、多面体 (polyhedron) モデルを処理対象としており、曲面を有する図形は多面体と近似することにより、表示することができ。従って、ホスト計算機 QA-2 と EXPERTS との間のインタフェースとなるデータは、各多面体を構成する多角形 (polygon) のリスト (Polygons List, 以下 PL) であり、この PL を基にして HSE を開始する。

HSE は、図形の相互の位置関係を調べ、視点から見える (visible) 図形と見えない (invisible) 図形とを区別する処理である。このためのアルゴリズム (Hidden Surface Algorithm, 以下 HSA と記す) には各種のものが提案されており、主要な HSA 相互の効率比較も Sutherland らによく行なわれている^[7]。

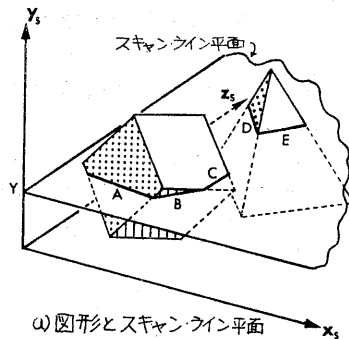
EXPERTS が採用している HSA は、スキャンラインアルゴリズム (Scan Line Algorithm, 以下 SLA) と呼ばれるものである^[8]。SLA は 1 スキャンライン (FB の横 1 行分の画素集合に対応するもの) に関係する図形だけを 1 回の処理対象とし、こ

のスキャン・ラインを順々に進めることにより、全体の表示処理を行なうものである。(以下、図2.3を参照のこと。)

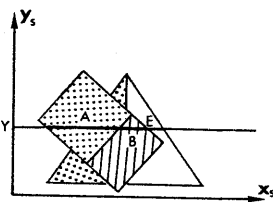
X_s-Y_s 平面は、図形を表示するスクリーンである。スキャン・ライン [$y_s=Y$, $Z_s=0$] に対応する平面 [$y_s=Y$] をスキャン・ライン平面と呼び、スキャン・ライン平面と各次元図形の表面との交線をセグメント(segment)と呼ぶ。図2.3(c)における黒丸(●)は、スキャン・ライン平面上に現われた、図形の稜線(edge)である。

SLAにおけるHSEとは、スキャン・ライン平面におい(可視)セグメントと決定する手続きであり、

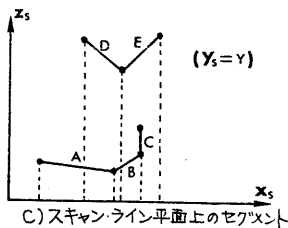
通常は X_s の正の方向(スクリーン上では右)に順に進められる。このためには、スキャン・ライン平面上のセグメントを順番に結びつけたアクティブ・セグメント・リスト(Active Segment List, 以下ASL)、または稜線を結びつけたアクティブ・エッジ・リスト(AEL)を保持している必要がある。スキャン・ラインを1つ次に進める際にはこのASL/AELを更新しなければならぬ。ところが、隣接する2つのスキャン・ライン平面の間での図形の変化分はごくわずかであり(この性質をscan line coherenceと呼ぶ)、かゝる一定の増分値を加えるだけのインクリメンタル(incremental)な計算だけで更新が行なえる。また、セグメントという「広がり」を持つ単位で可視/不可視を判定するので、一旦可視セグメントが決定できればその中では一律に計算で各画素の値を決める



(a) 図形とスキャン・ライン平面



(b) スクリーン平面



(c) スキャン・ライン平面上のセグメント

図 2.3

ことができる(この性質をspan coherenceと呼ぶ)。

以上のことから、SLAは各種のcoherenceを利用した効率の良いアルゴリズムであり、かつFBを有するラスタスキャン型ディスプレイには最も適合するアルゴリズムであると言える。

近年の3次元図形処理の並列処理の提案には、depth-buffer法(その座標軸の名称からZ-buffer法とも呼ばれる; 以下Z-B法と記す)を採用したものが多い。Z-B法は処理の各段階が単純であり、画素間での相互干渉もないので、並列処理向きアルゴリズムであると言える。しかし、1画素毎に1つの Z_s 値のみを保持しているので処理の履歴を残すことができず、前述のA.J.やshadowingの処理を組込むことが困難である。また、本質的に処理量が大きいので、実時間処理を達成するためには非常に多くのプロセッサ台数を必要とする。

一方SLAは、S.S.やA.J.の処理の組込みが比較的容易であり、shadowingも従来のアルゴリズムの少しの変更と実現することができる。また、前述した各種のcoherenceを利用することにより、処理量を軽減することもできる。これらの理由から、我々のEXPERTSではHSAとしてSLAを選択し、これにS.S., A.J., shadowingの各処理を組込んだ表示処理アルゴリズムを採用した。

2.3 EXPERTSにおけるSmooth Shading (S.S.)

EXPERTSでは、多面体近似によって曲面を表示する。これを元の曲面らしく表示するためにS.S.の処理を施す。

S.S.の方法としては、Gouraudの方法^[7]と類似した、輝度(intensity)の線形補間法を用いた。即ち、多角形(polygon)の各頂点(vertex)に予め適当な輝度を与えておき、これらの値を線形補間することによって、多角形を構成している各画素の輝度を求める。

EXPERTSでは、この近似多角形を三角形だけに限定している。この制限を設けると、輝度の変化量を一度計算しておけば、あとはインクリメンタルな計算だけで各画素の輝度値を求めることができる。従って、S.S.のための計算量を大幅に節減することができる。かつ、SLAにも簡単に組込むことができる。また、動画表示を考えた場合、図形とスキャン・ラインの方向との相対的な位置関係の変化による補間輝度値の変則

的なゆらぎ (shading anomalies) が起こらない、という利点がある (図2.4参照)。

図2.5に沿って具体的に説明する。三角形ABCの各頂点の輝度をそれぞれ i_a, i_b, i_c とすると、 $y_s=k$ における稜線AB上の点の輝度 $i_{AB}(k)$ は：

$$i_{AB}(k) = \{(k - y_a) \cdot i_b + (y_b - k) \cdot i_a\} / (y_b - y_a) \quad (1)$$

ここで、稜線ABの方向に沿って y_s の増える向きに対する輝度の変化分を di_{AB}/dy とすると、

$$\frac{di_{AB}}{dy} = (i_b - i_a) / (y_b - y_a) \quad (2)$$

(2)式を用いて(1)式を書き換えると次式を得る。

$$i_{AB}(k) = i_a + (k - y_a) \cdot \frac{di_{AB}}{dy} \quad (1')$$

また、 x_s 軸方向 (スキャンラインの方向) の輝度の変化分を di/dx とすると、

$$\frac{di}{dx} = \frac{(y_b - y_c)(i_a - i_b) - (y_a - y_b)(i_b - i_c)}{(y_b - y_c)(x_a - x_b) - (y_a - y_b)(x_b - x_c)} \quad (3)$$

従って、 $x_s - y_s$ 平面上の (j, k) における画素の輝度 $i(j, k)$ は次式で与えられる：

$$i(j, k) = i_{AB}(k) + (j - x_{AB}(k)) \cdot \frac{di}{dx} \quad (4)$$

(2)式、(3)式はいずれも三角形の各頂点の輝度が与えられた時点で計算できる定数値である。

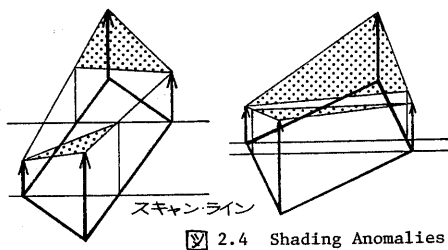


図2.4 Shading Anomalies

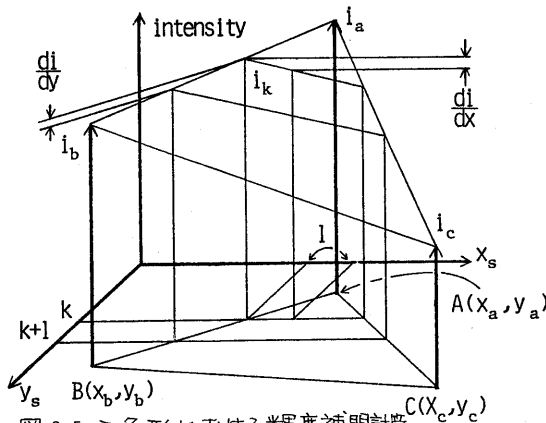


図2.5 三角形における光強度補間計算

従って、この値を図形データ中に格納しておけば、HSEによって可視セグメントが決定した後、直ちに(4)式によってS.S.を開始する事ができる。

2.4 EXPERTSにおける境界線平滑化処理 (A.J.)

ラスタースキャン型ディスプレイではスキャンラインに斜交する直線成分が階段状に表示され (jag) 現実感のある表示を行なう上大きな障害となっている。このようなjagの発生を抑える方法が、いくつか提案されている。^{[9][10][11]}

EXPERTSで採用したA.J.の手法は、多角形の境界線のところに発生するjagの抑制と処理の目的とするもので、SLAにも相対み易いアルゴリズムとなっている。

本A.J.の手法の基本は、画素データの重み付き混合計算である。即ち、直線成分の通過する画素に対し、その直線の左右の画素データ候補値 (これはHSEおよびS.S.により計算される) を次式により混合するものである：

$$C = \alpha \cdot C_L + (1 - \alpha) \cdot C_R \quad (5)$$

ここでCは求める画素データ、 C_L, C_R はそれぞれ左、右の画素データ候補値、 α ($0 \leq \alpha \leq 1$) は混合比である。 α の選定方法としては (図2.6参照)。

- 1画素領域の面積を1としたときの、直線の左側の部分の面積「A」；
- 画素領域の中心と直線との距離「D」；
- 画素領域の中心からスキャンライン方向、またはスキャンラインに垂直な方向 (これは、直線の勾配によって選択) に測った距離「d」；

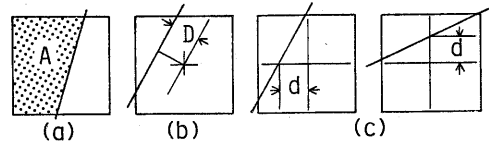
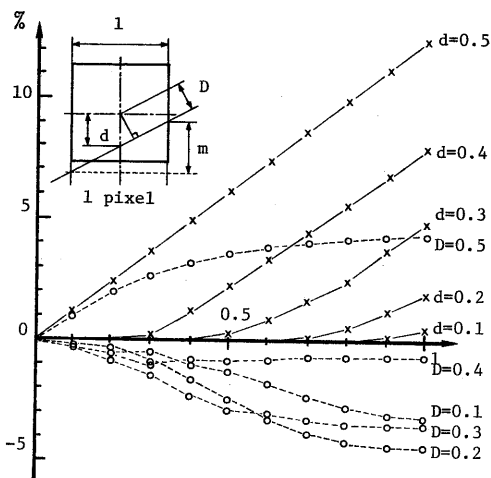


図2.6 混合比 α の計算

などが考えられる。各方法による α の計算値と理想値との誤差を図2.7に示した。

各方法の特徴は次のようにまとめられる。

- 混合比としては理想的である。しかし、面積計算のため処理の負荷が大きい。
- 線分の表示に適している。ただし、距離計算がやや複雑である。
- 距離dの計算が簡単であり (HSEによる計算している値を使用できる)、インクリメンタルな処理が可能である。しかし、線分の勾



(b)の方法 -----

(c)の方法 -x-

図2.7 理想的な混合比に対する誤差

配が 45° を越えるか否かにより、計算の方法を変えなければならない。

SLAに組込んだ形でA.J.を行なう場合は(c)の方法が適しており、EXPERTSではこの方法によるA.J.の処理を実行する。

以下に、上記の(b)および(c)の方法について、処理量の増加を測定するために、に実験の結果を示す。表2.1は、4種類の図形に対し(c)の方法によるA.J.を施した場合の、処理量の増加分を測定したものである。また、表2.2は、(b)および(c)の方法により、線分に対するA.J.を施した場合の測定結果である。(b)と(c)の結果の差異は、混合比を求めるための計算量の差によるものである。(b)および(c)の方法による視覚的な効果はほぼ同じであることから、(c)の方法が最も適当であると判断した。

表 2.1

	A.J.を施さない場合の処理量に対する増分%	線分の数
三角錐	4.4	5
四角柱(写真1)	5.0	9
八角錐	6.4	15
カウパーン(写真2)	12.4	20

表 2.2

線分の数	(b)の方法での処理量	(c)の方法での処理量
8	1.15 (写真3)	3.35 (写真4)

A.J.を施さない場合の処理量を1とする

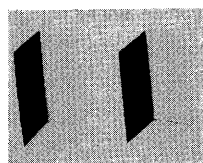


写真1

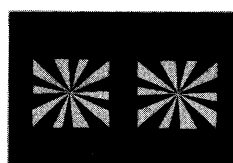


写真2

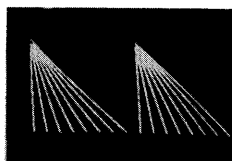


写真3

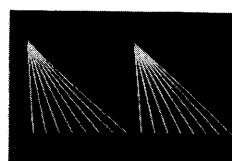


写真4

[3] 表示処理専用プロセッサ: EXPERTS

3.1 EXPERTS のアーキテクチャの概要

前章で述べたように、EXPERTSは、スキャン・ライン・アルゴリズム(SLA)による隠面消去処理(HSE)と、輝度補間処理(S.S.)、境界線平滑化処理(A.J.)、付影処理の各処理を組み込んだ形で高速に実行し、実在感のある3次元図形表示を実時間で行なうための専用プロセッサである。したがって、EXPERTSのアーキテクチャは、SLAの並列処理ということとを目的として、最適化されている。

SLAは1スキャン・ライン単位の表示処理(これを「スキャン・ライン処理」と呼ぶ)を次々に進めていくアルゴリズムである。そこで、我々は次の2点に注目して並列処理による高速化を図った。即ち、

- i) スキャン・ライン上の画素点集合に対する並列処理;
- ii) 複数のスキャン・ライン処理を同時に実行する並列処理。

また、SLAの処理の流れも大きく分けて次の2段階から成る。即ち、

- a) ホスト計算機から送られてきた多角形のリスト(PL)からアクティブ・セグメント・リスト(ASL)を作成する処理、および、スキャン・ラインを次に1ノ進める度にASLを更新する処理;
- b) ASLに基づいてHSEを行ない、続いてS.S., A.J.の処理を施して、フレーム・バッファ(FB)

に画素データを書き込む処理。

この2つの処理段階は、ASLをインタフェースのデータとして明確に分離することが出来る。

これらのことを考慮した結果、EXPERTSのアーキテクチャとして、SLP (Scan Line Processor) 及び PXP (Pixel Processor) と呼ぶ上位・下位の2レベルの階層構造を有する並列プロセッサ方式を採用した (図3.1 参照)。即ち、SLPには上記i)の処理、PXPには上記ii)の処理をそれぞれ分担させ、かつ、複数のPXPによる並列処理により上記i)の高速化を、複数のSLP系 (SLPとそれに属する複数のPXPを統括した単位) による並列処理によって上記ii)の高速化をそれぞれ達成することを意図している。

また、これまでの記述からも明らかなように、ホスト計算機とSLPの関係、およびSLPとPXPとの関係は、それぞれ前段の処理結果の出力データが後段の処理の入力データとなっており (具体的には、それぞれPL, ASL)、一方向的なデータの授受を行なうだけで、処理の分担が可能である。そこで、ホスト計算機、SLP、PXPの各処理段階を非同期的に動作させ、いわゆるパイプライン処理を実行することによって、表示処理

の効率向上を図ることが出来る。

また、EXPERTSの他の特徴として、その拡張性といふことが挙げられる。表示図形の複雑度、あるいは図形表示処理速度に対する要求は、応用分野によってそれぞれ異なることが予想される。その場合、必要な性能を発揮するために、容易に機能拡張 (あるいは縮小) が行なえることが望ましい。この要請を満たすために、EXPERTSは、SLPの台数、及び1 SLP系におけるPXPの台数を変更出来るよう、拡張性を備えた構成を採用している。

3.2 ハードウェア構成と並列処理方式

前述したように、EXPERTSの並列処理には、i) SLPレベルでの並列処理と、ii) PXPレベルでの並列処理がある。

《SLPレベルでの並列処理》: SLAの効率の良さは、2章に述べて scan line coherence を利用して計算量を低く抑えている (インクリメンタルな計算で済む) ことに一因がある。従って、ある程度まとまった本数のメキャンラインを1プロセッサの処理対象とする必要がある。そこで、画面を垂直方向に分割し、それぞれの複数スキャ

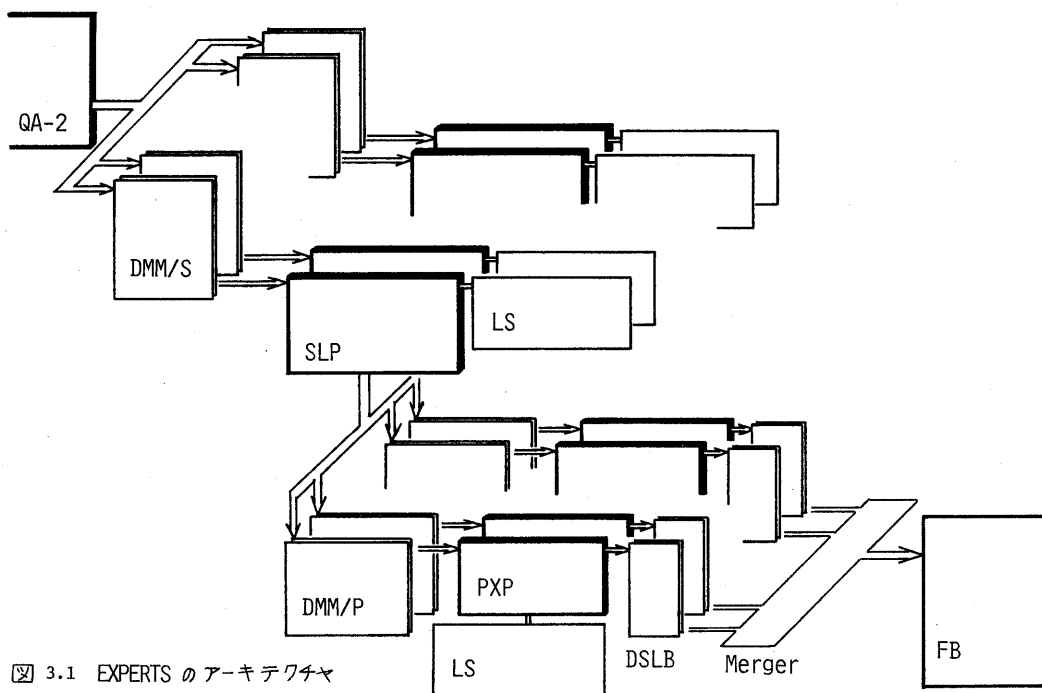


図 3.1 EXPERTS のアーキテクチャ

ン・ライン領域を1つのSLP系で処理することとした。

各SLPは、ホスト計算機から送られてPLを基に、自らの担当画面領域における最初のスキャンラインについてのASLを生成する。これは、一種のクリッピング(clipping)処理をSLPが分担したことになり、ホスト計算機はすべてのSLPに同一のPLを送出するだけでよい。一旦ASLを作成すれば、あとは通常のSLAと同様の手続きで次にASLの更新処理を実行する。この間、SLP間での処理の干渉は全くなく、各SLPは相互のデータ通信なしに独立に動作できる。

《FXPレベルでの並列処理》：SLPがASLの作成と更新という、スキャンラインについての大域的な処理を行なうのに対して、ASLを基にしてHSEを行ない、同時にS.S.やA.J.を施して各画素データの値を決定するという処理は局所的な処理の積重ねである。ここでも、SLAの特長である“span coherence”の利用を行ない、計算量を削減することが望ましい。従って、SLP1台に対して複数のFXPを従属させ、スキャンラインを分割して、それぞれを各FXPに処理させることとした。

各FXPは、SLPから送られてきたASLを基に、自らが担当する部分スキャンライン上の最初の画素(左端の画素)における可視セグメントをまず決定する。このためには、SLPの初期処理と同様、一種のクリッピング処理を行なうことになる。即ち、担当範囲よりも左側の部分に存在しているセグメントをASLから除去すると共に、担当範囲の開始点における各セグメントの

Zs値(奥行き座標値)、輝度値など、以降のHSE等のための初期データを計算する。これらをFXPで実行することにより、上述のホスト計算機とSLPの関係と同様、SLPはすべてのFXPに対して同一のASLを送出すればよい。ただし、FXPでの初期処理によるオーバーヘッドを軽減するために、図3.2に示すBlock Boundary Bucket List, およびBlock Leading Segment PointerとASLの付加的な情報としてASLと共にFXPへ送付する。これらは、SLPにおけるASL作成・更新処理の際に容易に生成することができる。

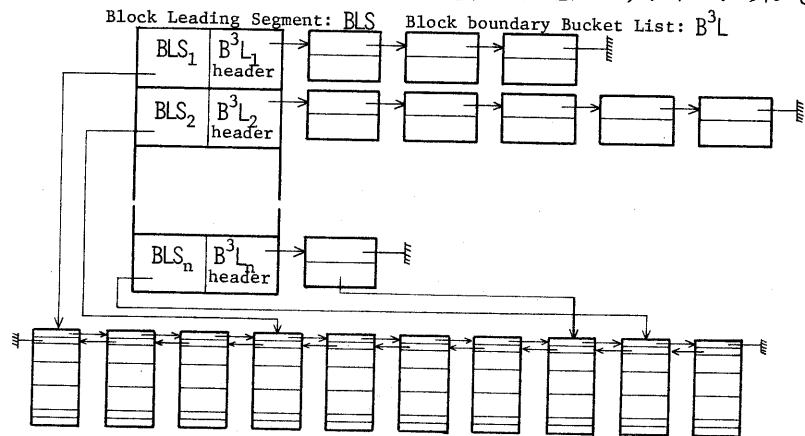
初期処理が終了した後の処理は、2章で述べたSLAと同一であり、自らの担当範囲の終点となる画素まで、他のFXPとの通信なしに独立に動作できる。

以上のことから、SLPレベル、FXPレベルの両方とも、ブロードキャスト型並列処理方式を採用することができ、通信オーバーヘッドを極小に抑えることができる。

《ホスト計算機、SLP、FXPの処理のオーバーラップ》：ホスト計算機、SLP、およびFXPごとの各処理をオーバーラップさせて、パイプライン効果によるスループットの向上を図るために、DMM(Double Message Memory)と呼ぶ2重系の共有メモリを用いた。図3.1に示すDMM/Sはホスト計算機とSLP間でのPLの授受、DMM/PはSLPと各FXP間でのASL等の授受に用い、上位プロセッサによる書込みと下位プロセッサによる読出しとを並行して同時に行なう。DMMは、物理的には下位のプロセッサ1台に1組ずつ装備しており、上位からは1枚の共通メモリ、下位からはそれ

next segment pointer	
last segment pointer	
R/G/B	Tags
$\frac{dz}{dx}$	$\frac{dt}{dx}$
(x_L, z_L, i_L)	
$\frac{dx}{dy} _L$	$\frac{dy}{dx} _L$
(x_R, z_R, i_R)	
$\frac{dx}{dy} _R$	$\frac{dy}{dx} _R$
work 1	
work 2	

Active Segment Block



Active Segment List: ASL

図3.2 Block Boundary Bucket List, Block Leading Segment

Active Segment List

ぞれ専用の局所メモリとし、見えるよう構成している。

《FBへの書き込み》：図3.1に示すDSL B(Double Scan Line Buffer)は、各PXPとFBとの間にあって、PXPの画素データ書き込みの高速化と、PXP間でのFBアクセス競合の解消とを図る目的で導入したものである。FBは大容量を必要とするために、アクセス時間の短縮をそれほど望めない。DSL Bは、1スキャンライン分の画素データを格納する高速メモリであるが、FBに対するキャッシュ(cache)メモリとしての機能を有する。これと共に、PXP毎にDSL Bを1組ずつ装備してPXPのメモリアクセスを局所化することにより、複数のPXPによるFBのアクセス競合問題を解消し、PXPレベルでの並列処理効果を最大限に発揮させることができる。DSL BもDMM同様、2重系の構成とし、PXPによる書き込みと、FBへの転送のための読出しとを並行して同時に行なうことができる。

《FBおよびCRTディスプレイへの出力》：FB全体に対する更新処理が終了していない段階でCRTディスプレイへの出力を開始すると、表示処理過程の途中段階が見えてしまうことになり、非常に違和感を与えてしまう。EXPERTSでは、FBをDSL BやDMMと同様に2重化し、更新処理が完了した時点でCRTディスプレイへの出力を開始することとする。

DSL BからFBへ画素データを転送する際の問題は、SLP系の間でのFBアクセス競合である。しかし、FBの読出し(即ち、CRTディスプレイへの出力)は、ラスタスキャン型ディスプレイである以上、一度に高々1スキャンライン分の画素データに対してしか必要ない。そこで、物理的なFBは、各SLP毎に分散して装備しておき、D/A変換(Digital to Analog変換)を含めたビデオ出力処理を担当するハードウェア:VOC(Video Output Controller)が画面領域分割の設定に基づいて、各SLP系内のFBから画素データを読出すという構成を採用した。これにより、FBは各SLP系にとって局所的なメモリとなり、アクセス競合問題は解消した。

論理的な意味でのFBは、512×512個の画素で構成する。各画素は16ビットのデータとして表現する。16ビットの内訳は、色相9ビット、輝度7ビットである。色相はさらに赤、緑、青

それぞれ3ビットずつから成る。色相と輝度とを分離したのは、S.S.のための計算量を軽減するためにめである。

EXPERTSの内部データ幅、即ち、SLP、PXPの基本演算データ幅は16ビットであり、DMMをはじめとして各プロセッサの局所メモリ、内部レジスタ・ファイル等を1語16ビットで構成する。

SLPにおけるASL作成の処理は、以後のインクリメンタルな計算のための各種の増分値の計算を含んでいるため、乗除算の高速化を行なう必要があった。そこでSLPには、加減算およびシフト演算を行なうALU(ビット・シフタのマイクロプロセッサで構成)の他に、乗算器、および逆数表(ROMによる)を装備した。PXPにおいても、担当の画面領域での初期処理、及び、A.J.における画素データ値の混合計算など、乗算が多数出現するため、SLP同様、乗算器を装備している。

3.3 動的負荷分散

前節では、EXPERTSの並列処理方式について述べ、それが表示画面上での分割に従って実行されていることを示した。しかし、表示図形の複雑さは常に全画面で均一に分布している訳ではなく、端と中央では画素データの決定に要する処理時間は大きく異なることがあり得る。このような性質を持つ表示図形を対象として並列処理による高速の表示処理を行なう場合、次の点に配慮が必要である。即ち、表示図形の複雑さの分布とは無関係に画面の領域を等分割し、各SLP及び各PXPの処理範囲を決定すると、プロセッサ間で著しい処理量の差が生じ、全体の処理速度を大幅に減少させる可能性がある。このため処理量の均等分散を考慮する必要がある。これは並列処理における問題点の1つである¹⁰⁾。EXPERTSでは、この負荷分散を表示処理の実行中に行なう動的負荷分散方式を採用している。

負荷分散の対象はSLPレベルとPXPレベルであるが、同一の負荷分散方式は採用しなかった。これは、前者がリストのソート・マージが中心の処理量とその扱うデータ量とが相関関係を持ち、ホスト計算機がPLをブロードキャストする時点で領域割当が容易に決定できるのに対し、後者はHSE、S.S.及びA.J.等を含む処理を行なうため、処理量と扱うデータ量とが相関関係を持

つとは限らず、SLPがASLをブロードキャストする時点での領域割当の決定は困難であることによる。

PXPレベルの動的負荷分散は以下のような方式で行なう。まず、ある負荷分散による領域割当のもとで各PXPにスキャンライン処理を実行させ、それぞれの終了時間を記憶し、これと基に次のスキャンライン処理における領域割当を決定する。この方式の問題点はスキャンライン処理の開始時に、そのスキャンライン処理における最適負荷分散を行なえない点である。しかし、表示図形の複雑さの分布は隣接するスキャンライン間ではほとんど変化しないので、このような追従的方式でも有効であると判断できる。

各PXPが処理の終了とその所要時間をSLPに知らせるため、SLPとPXPとの間にはカウンタとデコーダから成るハードウェアが用意されている。これを基にSLPがPXPの動的負荷分散を決定する。そのアルゴリズムとして、次の2つが考えられる。

i) 各PXPの処理時間に逆比例した領域分割を行なう。

ii) 処理時間の分布の定量的解析と正確に行なうのではなく、所要時間が一定の閾値を越えたPXPに対して、その分担領域を縮小して、逆に早く処理が終了したPXPから順に縮小分に相等する領域を割当てて。

i)は一意的に決定できるが、かなりの処理時間を必要とし、かつ総てのPXPが処理を終えるのを待たなければならぬ。ii)は比較的簡単に決定できるが、精度がi)に比べて劣る。両者は共にそれぞれ得失を持ち、負荷分散の決定には特に高速性が要求されることから、我々はii)の方法を採用した。

以上、SLPレベルとPXPレベルでは、動的負荷分散方式を採用して、並列処理効率の向上を図っている。

[4] EXPERTSの性能評価

EXPERTSにおける並列処理、パイプライン処理が表示処理をどのように効率よく実行するかを示し、アーキテクチャの性能を次の2点で概略評価する。

a) SLP及びPXPの階層構造を持ち、SLP台数1

に対しPXP台数を複数に設定し、負荷バランスをとっている点；

b) SLP系と拡張可能にし、並列処理効果を目差している点。

スキャンラインアルゴリズム(SLA)による表示処理は、機能的に次の5つの段階に分かれ、Fr:「明らかに不可視の面を除く表示面数(relevant Faces)」をパラメータとして各段階の処理量の概算値を示す²⁾。但し、Dcは1画素あたりの面の重なり(の平均値(Depth complexity)), α, β はそれぞれFrの横方向、縦方向の画素数である。

i) 表示面に含まれるすべての稜線を、 y_{min} (頂点の y_0 座標で小さい方の値)をキーとしてバケットソートする《y-sort》: $4Fr$

ii) 1スキャンライン内に現れるセグメントを、その左稜線の x_0 座標値をキーとして、マージソートする《x-merge》: $4Fr \cdot \sqrt{Dc \cdot Fr} \cdot \frac{\alpha}{\beta} \cdot \frac{1}{2}$

iii) スキャンラインの更新に伴い進入、退出するセグメントとii)で作成したセグメント・リストとをバブルソートする《x-sort》:

$$y \cdot \left\{ \sqrt{Dc \cdot Fr} \cdot \frac{\alpha}{\beta} + \frac{10 \cdot (Dc-1) \cdot 2Fr/4}{y \sqrt{Dc \cdot Fr} \cdot \frac{\alpha}{\beta}} \right\}$$

iv) スキャンラインをスパンに分割し、そのスパン内に現れる総てのセグメントを抽出する《span cull》: $y \cdot \sqrt{Dc \cdot Fr} \cdot \frac{\alpha}{\beta}$

v) iv)で抽出したセグメントについてデプスサーチを行ない、可視セグメントを決定する《depth search》: $30 \cdot y \cdot Dc \cdot \min(\alpha, 2\sqrt{Dc \cdot Fr} \cdot \frac{\alpha}{\beta} / Dc)$

並列処理を行なわない場合、表示処理の処理量はi)~v)の総和(Q_{seq})となる。ここで、 $Dc=3$, $\alpha=\beta=512$ とすると、

$$Q_{seq} = 4Fr + 3.46Fr\sqrt{Fr} + 32500\sqrt{Fr}$$

一オ、EXPERTSではi)~iii)をSLPに、iv)~v)をPXPに分担させている。従って、SLPが S_n 個、PXPが1SLP系につき p_n 個(従って総計 $p_n \cdot S_n$ 個)の並列処理を行なうので、SLP、PXPの処理量をそれぞれ Q_s, Q_p とすると、

$$Q_s = \frac{1}{S_n} \cdot (4Fr + 3.46Fr\sqrt{Fr} + 893\sqrt{Fr}),$$

$$Q_p = \frac{1}{(S_n \cdot p_n)} \cdot (31607 \cdot \sqrt{Fr})$$

パイプライン方式でSLP、PXPの処理を時間的に重ね合わせた場合の、総合的な処理量 Q_{par} は、

$$Q_{par}(S_n, p_n) = \max(Q_s, Q_p)$$

となる。

EXPERTSにおける並列処理効果を調べるために、 Q_{seq} と $Q_{par}(S_n, p_n)$ との比をとってグラフ

に描く。(図4.1参照) グラフは表示面数 F_r が1000面、及び2000面の場合についてのものである。これによると、SLP 1台に対するPXPの個数は4~8台が適していると考えられる。即ち、PXPレベルの並列処理は処理全体の高速化に必要不可欠である。以上、EXPERTSが採用したアーキテクチャが表示処理を効率よく実行するために適した構成であることが示された。

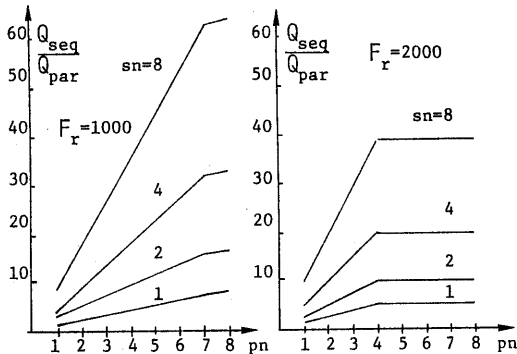


図 4.1 EXPERTS における並列処理、パイプライン処理効果

[5] おわりに

3次元色彩図形表示専用のマルチプロセッサ: EXPERTS について、その設計目標、アーキテクチャ、および並列処理方式を述べた。

EXPERTSは、表示処理のアルゴリズムとしてスキャンラインアルゴリズムを採用し、その効率的な並列処理を行なうために、SLP-PXP 2レベルの階層構造を持ったマルチプロセッサとして設計した。即ち、SLPによって表示画面の領域分割を行ない、各PXPによって並列に表示処理を実行すると共に、SLPにおける各スキャンライン毎のASL更新処理とPXPにおける画素データ値計算の処理とを時間にオーバーラップさせることにより、表示処理のスループットを向上させている。また、分割した画面の各領域における処理量の不均衡を是正し並列処理の効果を最大限に発揮させることを狙って、動的に領域分割の設定を変更できるような構成を採った。さらに、将来の処理性能向上の要求に対応するため、SLP系、PXP共に、その台数を容易に増設することができるよう考慮してある。SLP系8台、各SLP系にPXP8台(総計64台)の構成で約1000面から構成される3次元図形の実時間表示処理が可能であるとの評価結果を得た。

現在、さらに詳細に性能の見直しを行なうと同時に、ハードウェアの設計作業を進めている。

[謝 辞]

最後に、EXPERTSのスキャンラインアルゴリズムについて検討し貴重な意見を提示してくれた坂本靖子君、動的負荷分散方式について考察してくれた小幡高靖君の両氏に心より感謝する。

また、日頃貴重な御意見を頂き熱心に討論に加わってく下さる萩原研究室の各位に感謝する。

[文 献]

- [1] Requicha, A.A.G. and Voelcker, H.B.: "Solid Modeling: A Historical Summary and Contemporary Assessment," IEEE CG&A, Vol. 2, No. 2, Mar. 1982, pp. 9-24.
- [2] Fuchs, H. and Johnson, B.W.: "An Expandable Multiprocessor Architecture for Video Graphics (preliminary report)," Proc. of IEEE 6th Conf. on Computer Architecture, 1979, pp. 58-67.
- [3] Roman, G.C. and Kimura, T.: "VLSI Perspective of Real-Time Hidden-Surface Elimination," Computer-Aided Design, Vol. 13, No. 2, Mar. 1981, pp. 99-107.
- [4] 吉良: "疎結合並列処理方式の動画生成への適用," テレビジョン学会技術報告, IPD65-2, Feb. 1982, pp. 7-12.
- [5] Newman, W.M. and Sproull, R.F.: "Principles of Interactive Computer Graphics," 2nd Ed., McGraw-Hill Co., New York, 1979.
- [6] 柴山他: "低レベル並列処理機能を備えたグラフィックマイクロプログラム計算機QA-2のアーキテクチャ," 電子通信学会技術研究報告, EC79-63, Jan. 1980, pp. 31-40.
- [7] Sutherland, I.E. et al.: "A Characterization of 10 Hidden-Surface Algorithms," ACM Computing Surveys, Vol. 6, No. 1, Mar. 1974, pp. 1-55.
- [8] Watkins, G.S.: "A Real-Time Visible Surface Algorithm," Univ. Utah Computer Science Dept., UTEC-CSC-70-101, Jun. 1970.
- [9] Gouraud, H.: "Continuous Shading of Curved Surfaces," IEEE Trans. on Computer, Vol. C-20, No. 6, Jun. 1971, pp. 623-629.
- [10] Catmull, E.: "A Hidden-Surface Algorithm with Anti-Aliasing," ACM SIGGRAPH, Vol. 12, No. 3, Aug. 1978, pp. 6-11.
- [11] Pitteway, M.L.V. and Watkinson, D.J.: "Bresenham's Algorithm with Gray Scale," CACM, Vol. 23, No. 11, Nov. 1980, pp. 625-626.
- [12] 西田, 中前: "カラー・ディスプレイを用いた線画表示におけるJAGの除去," 情報処理学会コンピュータビジョン研究資料, CV10-1, Jan. 1981, pp. 1-8.
- [13] 高橋: "並列処理のためのプロセッサ結合方式," 情報処理, 23巻, 3号, Mar. 1982, pp. 201-207.
- [14] Ni, L.M. and Hwang, K.: "Optimal Load Balancing Strategies for a Multiple Processor System," Proc. of 10th Int'l Conf. Parallel Processing, Aug. 1981, pp. 352-357.