

並列画像生成方式とアーキテクチャ

大宅伊久雄, 吉田隆義, 守谷信行
 沖電気工業(株) 研究開発本部 総合システム研究所

画像生成技術は、マンマシンインタフェイス向上によって増々重要な技術となってきた。従来のパイプラインアーキテクチャでは、おぼろげバスの高度化にともなう生成処理の高速化や、3次元色彩画像などの高機能化への要求に対するが困難となっている。本論文では、並列画像生成方式にもとづくアーキテクチャを提案する。画像生成の基本演算処理を、幾何計算処理とピクセル処理に分け、それぞれを並列処理で実行し高速化をはかっている。特に、幾何計算処理では、ハイパーキューブネットワークを用いているが、大量のデータ通信に適した高速転送制御方式を採用した。また、ピクセル処理部では、高速なピクセル演算とメモリ更新を実現するためのSIMD方式による高機能画像メモリ方式を採用した。

Parallel processing and its architecture for image generation

Ikuro Oyake, Takayoshi Yoshida, and Nobuyuki Moriya
 Systems Laboratory, R&D Group, Oki Electric Industry Co., Ltd.
 11-22, Shibaura 4-Chome, Minato-ku, Tokyo 108, Japan

Image generation becomes very important to man-machine interface in many applications. The conventional pipeline architecture for image generation has some critical problems to achieve high speed processing and 3-dimensional functions such as hidden surface removal and smooth shading. A parallel approach to image generation and its architecture is presented in this paper. The architecture consists of two major components, a geometric processing part and a pixel processing part. Some important aspects of the architecture are addressed.

1. はじめに

画像生成技術は、マンマシンインタフェイスの向上によって増々重要な技術となってきた。画像生成の応用分野は、ビジネスグラフィックス、CAD/CAMからビジュアルシミュレーションやコンピュータアートにまで広がっており、その中で、より精度のよいモデルを扱ったり、よりリアルタイムかつ美しい画像を生成することが要求されている。また、CRTやプリンタなど画像生成に関連したデバイスの高速度化により、解像度はさらに精細になり、画像生成に要求される処理能力も現実の重要な問題となってきた。

本論文で用いる画像生成あるいは画像表示の簡単な定義を述べると、アプリケーションプログラムで扱う対象モデルのデータ構造をエンドユーザに表示するため、ラスタメモリ上に画像情報として展開することという。画像生成に関連した技術として、エンドユーザと計算機との対話をいかに制御するかというユーザインタフェイス管理の問題もあるが、本論文ではそれにはふれない。また、画像生成処理においては、応用により静止画像か、対話画像か、あるいは動画像かに分けられるが、本論文では、処理能力が要求される対話画像か動画像の分野を対象とする。

画像生成における処理の高速度は本論文の課題である。まず従来の画像生成方式とアーキテクチャの特徴をふりかえり高速度化の諸問題を明らかにする。そして、画像生成への新規要求機能を考慮した並列処理方式に基づくアーキテクチャを提案する。このアーキテクチャは、SIMDとMIMD方式を組み合わせた多層並列処理方式を採用している。特に、画像データのメモリ更新を高速度に行なう高性能画像メモリと

高速プロセッサ間通信のデータ伝送方式、および並列プロセッサのモニタ機能について詳しく述べる。

2. 従来方式

計算機による画像生成方式としては大別して光線追跡法(レイストレーシング法)と物体座標変換法(オブジェクトトランスフォーメーション法)がある。光線追跡法はアートなどの映像制作分野で、物体座標変換法はビジュアルシミュレーション、CAD、ビジネスグラフィックスなど広範囲な分野で用いられてきた。光線追跡法は、アルゴリズムが比較的シンプルで膨大な計算量を必要とすることから並列処理への関心が強く、従来から並列処理マシンの応用プログラムとして多く発表されている。

一方、物体座標変換法は1960年代の前半にさかのぼりアルゴリズムの蓄積やソフトウェアインタフェイスの標準化が進んでいる反面、並列処理方式についてはあまり提案されていない。物体座標変換法を用いた従来の画像生成アーキテクチャは文献(1)に詳述されている。ホスト計算機と表示処理プロセッサを密結合にし、表示処理機能を専用パラライズンアーキテクチャにより高速度に実現している。専用パラライズンアーキテクチャは画像表示処理をすべてホスト計算機で行なう場合に比較してホストの負荷を軽減し、表示処理の高速度が図れる有効な方式とされてきた。

このようなパラライズンアーキテクチャにおける処理形態を図1に示すとともに、以下に各段階の処理を説明する。

- (1) ホスト計算機ではアプリケーションプログラムで表示モデルを生成し、表示処理プログラムライブラリを用いて、表示情報をシステムバスを介して表示処理プロセッサに送り、以下のパイプライン処理を施す。
- (2) 表示情報はホスト計算機から構造リストとして出力される。構造リストは表示要素の階層構造として表わされている。
- (3) 表示要素は一般に局所座標系で表現

- されているのでそれを世界座標系に統一するため必要な座標変換を行う。
- (4) 画面は有限領域であるので、その枠の外に出る表示要素の部分をクリッピングする。
- (5) クリッピングされた表示要素をピクセル展開し、各ピクセルごとに色情報を画像メモリに書き込む。

以上のようなパイプラインアーキテクチャには、高速化、高機能化に

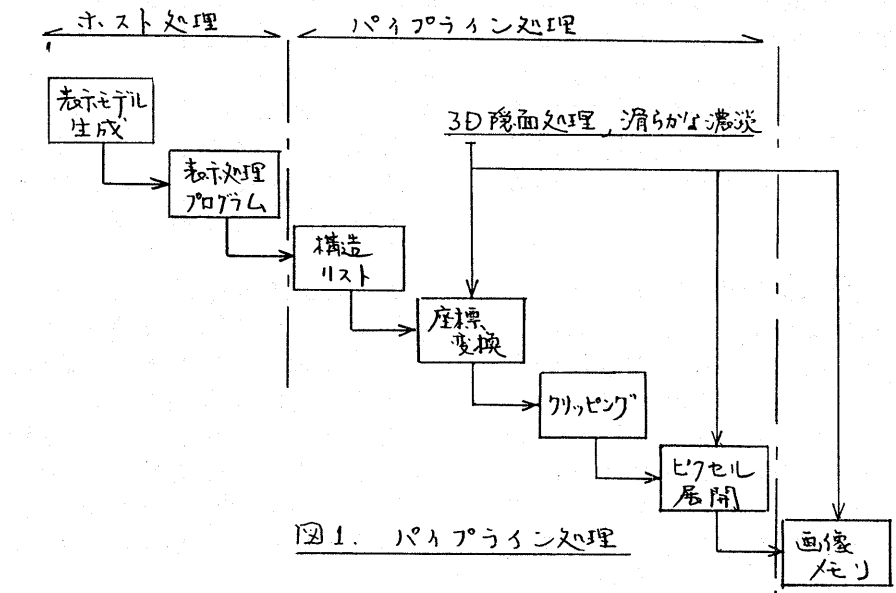


図1. パイプライン処理

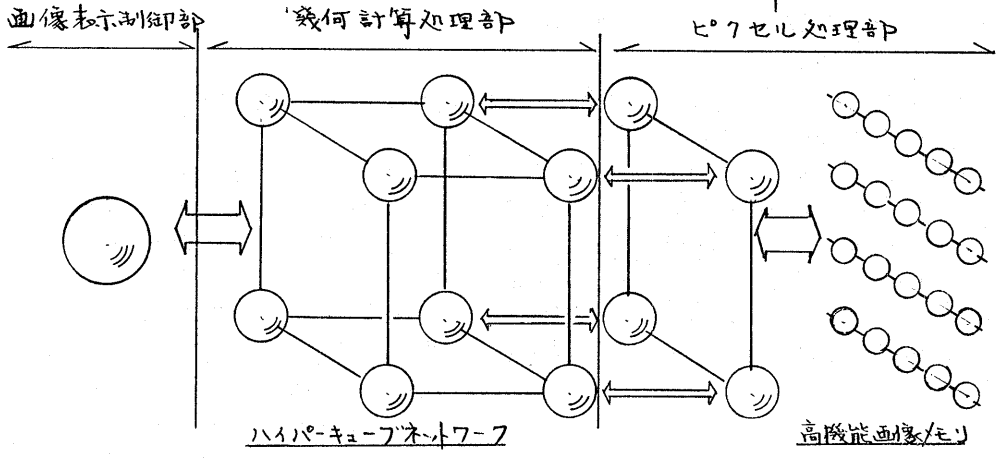


図2. 並列アーキテクチャ

り、以下の問題点がある。

(1) N段のパイプライン処理は各ステップの処理時間が等しくなった場合最大の処理性能が得る。しかし画像表示処理では、各ステップの処理時間は表示要素の形状、サイズに大きく依存する。そのため、多種多様な表示要素を大量に流した場合、パイプラインに乱れが生じ性能劣化の現象があらわになる。

(2) 色情報の多ビット表現、表示画面の高精細化に対応するためには、ピクセル展開部で行っているピクセル(画素)更新の処理能力に負担がかかる。

また、アプリケーションが扱うモデルが2次元から3次元に拡大するにつれ、3次元モデル特有の表示処理である陰面消去や滑らかな濃淡づけの機能をアーキテクチャ上に組み入れる必要がある。

これらの諸問題を解決するために、新たな並列画像生成方式を以下に提案する。これは従来の物体座標変換法を実現するパイプライン処理において、基本となる演算でかつ負荷の増大が指摘される処理を、幾何計算処理とピクセル処理に大別して、それぞれ処理を並列化し高速化を達成しようとするものである。また、幾何計算処理とピクセル処理は全体としてパイプライン的に動作させる。

物体座標変換法と異なった光線追跡法は、画面の各画素ごとに光線と発生し表示対象モデルとの交差をしらべながら、色情報を計算する方法であるがその演算処理は主に幾何計算処理に属するものといえる。ここで提案する並列処理方式は、光線追跡法にも適用可能であるが、以下では物体座標変換法を例に説明する。

3. 並列画像生成方式

3.1 3次元機能の拡張

ここで3次元機能とは、3次元の対象モデルを2次元の表示画面に透視写像(または平行写像)して画像生成することを意味する。カメラをとおして3次元被写体とみることに相当しているが、写像された対象をより正確に表示するためには、3D陰面処理と滑らかな濃淡づけ機能は必須といえる。

陰面処理方式としては、対象モデルの表現レベルで深さ方向にソートする方法と、画面の走査線に対応したスクリーンラインレベルでソートする方法、および、画面の画素レベルで深さの値(Z値)で比較する方法がある。ここでは、ソート処理がなく並列性に適したZ値比較、すなわちZ-バーン法を採用する。

さらに滑らかな濃淡づけでは、対象モデルの法線方向と光線方向で決定される輝度値を、ピクセル展開時に線形内挿する方式を採用する。

図1に示したように、基本的には3次元機能の処理を従来のパイプライン処理の幾何計算処理とピクセル処理に分散させることを意図した方式である。

3.2 並列化方式

画像生成処理の並列化を考えた場合、その手法はいろいろあるが、大きく分けると次の3つが考えられる。

(1) オブジェクト分割

対象モデルを複数のプロセスに分散し並列処理する方法

(2) 画面分割

表示画面を空間的に分割して各プロセスは一つの分割画像をうけもつ方法

(3) フレーム分割(時分割)

動きをもつ表示を行なう場合に各プロセッサは連続したフレームを別々にうけもち、表示はフレームを順次切り替えながら行なう方法

このらふの方式のどつが良いかは、アプリケーションが扱う対象モデルの値と量、表示処理の内容、表示画面の解像度などによつてかわってくる。そこでアーキテクチャとしての柔軟性を保持するため、並列化の方式は固定することなく、アプリケーションプログラムの記述により適応できる方式が一番望ましい。

3.3 並列アーキテクチャ

並列アーキテクチャの構成を図2に示す。画像生成は、処理の基本演算タイプにより幾何計算処理部とピクセル処理部に分けられる。幾何計算処理部は、浮動小数点を強化したプロセッサであり、座標変換処理、クリッピング処理やモデル生成などのアプリケーションが実行される。幾何計算処理部におけるプロセッサ間の結合ネットワークは、プロセッサ間通信の様々なデータフローの可能性を想定してハイパーキューブ接続方式を採用した。このハイパーキューブネットワーク結合をもつマルチプロセッサは、文献(2)にあるように当初数値計算用に提案された。画像生成に用いる場合、より多くのデータ転送量が必要となる。そのため、第4節では特にハイパーキューブネットワークにおいて工夫したデータ転送方式について詳述する。

ピクセル処理部は、ピクセルプロセッサと高機能画像メモリにより構成される。ピクセルプロセッサは、通常グラフィックプロセッサと呼ばれるもの

で、表示要素のピクセル展開を行なう。表示要素としては、面要素、線要素、画像要素が考えられる。従来このようなピクセル展開を実行するプロセッサは、画像メモリを集中制御するものが多かった。しかし、画面の高精細化にともなう画像メモリの増大、色彩画像の表現によるビットプレーン数の増大、さらには、Zバッファ法、滑らかな濃淡づけの補間機能などにより、性能を達成するには集中メモリより分散メモリの方法が好ましくなっている。分散メモリとは、画像メモリを分割し、それぞれに画素演算機構を外付けし、各演算機構をRAMD方式でピクセルプロセッサを制御する方法である。これについては第5節高機能画像メモリで詳述する。また、並列プロセッサのモジュールについては第6節で説明する。

4 ネットワークのデータ転送方式

ハイパーキューブネットワーク結合のマルチプロセッサは $N (= 2^n, n$ は正の整数) 個のノードからなり、各ノードは、主として処理を実行するプロセッサとノード間通信を行なうデータ通信部よりなる。各ノードには n 本の転送路がある。ノード a のアドレスは2進数 $(a_n a_{n-1} \dots a_2 a_1)$ で表わされる。ノードアドレス a は n 以外のハミング距離が1だけ異なるノードに直接結合している。ノード $a (a_n \dots a_1)$ からノード $b (b_n \dots b_1)$ へのデータ転送は、最下位ビットから調べて、最初に $a_i \oplus b_i = 1$ とするノード $(a_n \dots b_i b_{i-1} \dots b_1)$ にデータを送る。これを順次繰り返すとノード a とノード b の通信ができる。このように、任意の2つのノード間でデータ転送を行なう場合、ノード数が多くあるほど

経由する転送路の平均数は増加するといえる。

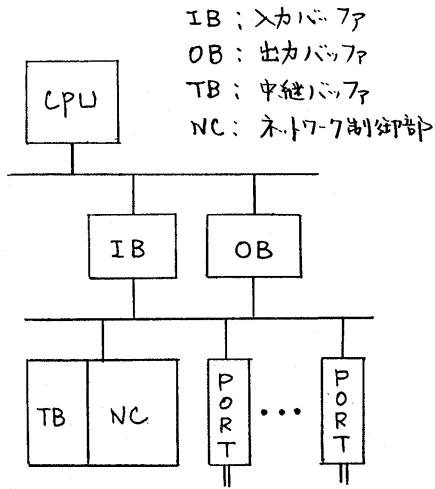


図3 ノード構成

ノードの構成を図3に示す。高速なデータ転送を実現するため、データ通信部はCPUと切り替えし、ネットワーク制御部が複数の入出力PORTを利用する。中継処理もこのネットワーク制御部で実行される。

ここで伝送制御を簡便にするため、半二重通信を行なう。すなわちデータの伝送を送信と受信に分け同時に実行しない方式である。しかし、この方式では隣接する2つのノードが同時に送信要求をだした場合にデッドロックとなる。さらに複数の中継ノードを経由してデータを伝送する場合、各中継ノードにおいて中継データが集中することがあり、それを保持するためのバッファが多くなる問題点がある。

これらの問題点を解決するため、ノード間(A, B)におけるデータ転送の優先順位を以上のように決定する。

- (1) 互いにデータ送信を行なう2つのノードの中継用バッファが空の場合、優先度はノードの番号により判定する。
- (2) 互いにデータ送信を行なう2つのノードの中継用バッファの何れか一方が

ふさがっているとき、中継バッファがふさがっているノードの優先度が高いものとする。

(3) 互いにデータ送信を行なう2つのノードの中継用バッファがいずれもふさがっているとき、ノード番号に従って優先度を決定する。

(4) 各ノードにおいて、データの送信は中継データを優先する。

これらの判定はすべてネットワーク制御部において実行される。

5. 高機能画像メモリ

次に、隠水面処理を行なうピクセル展開を例にして、高機能画像メモリの並列方式を説明する。

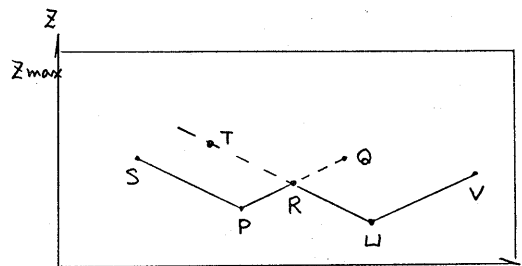
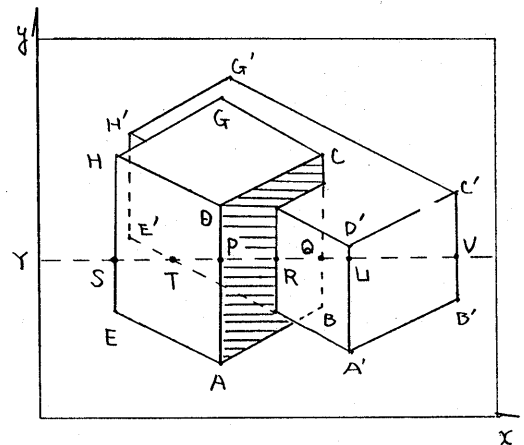


図4 隠水面処理例

図4のように空間的に交差する2つの直方体からなる立体の表示において、直方体A, B, C, D, E, F, G, Hのみがすでに描画されている状態で、次に面A, B, C, Dを表示するには、図でハッチングを施した領域の画素のみ塗りつぶす必要がある。画面上でy座標が一定値Yをとる走直平面で対象立体を切断したとき、各稜線との交点をS, P, R, U, Vとし、この断面を画面の上方から見た図を図4の下図に示す。このとき、線分PQのうち線分PRは塗りつぶし、線分RQは塗りつぶさないようにする必要がある。これを稜消面消去機能と呼ぶが、これを可能にするため、Zバッファメモリを利用する。

以下に処理手順を示す。

- ① 立体を構成する各面につき、以下の処理を反復する
- ② 面を構成する各線分につき、以下の処理を反復する

- ③ 左端点Pにおける深度を Z_i とし、増分を ΔZ とする
- ④ 左端点Pから右端点Qに至る各画素につき、以下の処理を反復する
- ⑤ Zバッファメモリの値 Z_M を読み出す
- ⑥ Z_i と Z_M の値を比較する
- ⑦ $Z_i < Z_M$ ならば
 $(R_i, G_i, B_i, Z_i) \rightarrow (R_M, G_M, B_M, Z_M)$
- ⑧ Z_i を累加する
 $Z_i + \Delta Z \rightarrow Z_i$

このような処理手順に従って、1画素ごとにメモリ読み出し、Z比較、条件付き書き込み、累加算という一連の処理を実行すると、画素数の膨大さを考えて、モ1つのプロセッサが集中メモリ制御で実現することは処理能力上大きな問題点が生じる。

そこでこの問題を解決するため、画素のX座標を $N (= 2^n)$ で除したときの剰余により分割したN個のメモリ素子群を考える。そしてこのN個のイン

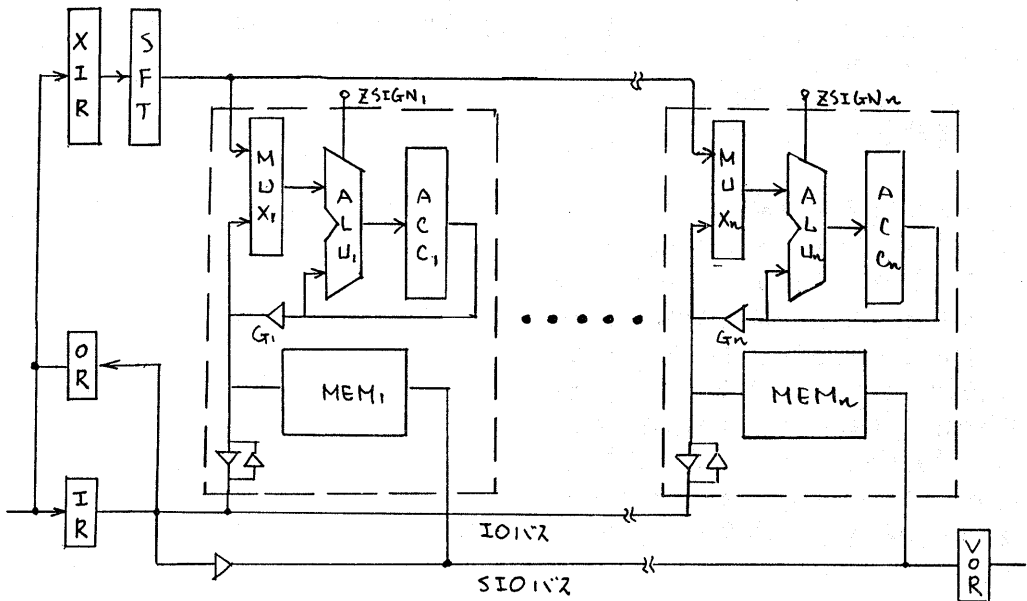


図5 高機能メモリの構成

タリーブメモリ各々に、各メモリのビット幅(w_1)より大きいビット幅(w_2)をもち、深度値の比較累加を行なう比較累加回路を設け、さらに深度値の初期値および増分値を設定する増分回路を設け、これを放送バスにより接続することにより、同時にN画素のメモリ更新を可能とする構成を採用する。

図5は高機能バッファメモリの構成を示す。メモリ素子群の各々に、マルチプロセッサ、算術論理演算ユニット、累加算器、ゲートからなる比較累加回路が付け加えられている。

さらに、読出し、比較機能を使用せずに、条件付き読みと累加機能を使用すれば、色情報と滑らかに変化する画像を生成することが可能となる。

6. 並列プロセッサのモニタ機能

幾何計算処理部は、MIMD方式のマルチプロセッサである。各プロセッサ上のプロセスは、メッセージ通信により他プロセッサ上のプロセスと通信する。このような通信機能は、各プロセッサに搭載されるモニタにより実行される。

モニタが実行する機能は、ユーザプロセス間通信、各プロセッサ内の資源管理、ネットワーク内エラー処理、およびアプリケーション実行制御である。ユーザプロセス間通信におけるメッセージの種類としては、リアルタイム画像生成制御を考慮して、即時処理メッセージと順処理メッセージの2種類をもうけた。即時処理メッセージは、プロセス間で同期をとる場合などに用いられる。

以上、アーキテクチャの概要と、プロセッサ間のデータ転送方式、高機能画像メモリ、およびモニタ機能について述べた。その他の重要な課題として並列プログラミングの開発支援環境の問題がある。本論文ではこれにふれず今後の課題としたい。

本研究は通産省工業技術院大型プロジェクト「科学技術用高速計算システム」の研究開発の一貫としてなされている。関係各位の日頃の助言に感謝したい。

(1) W. Myers, 「Staking Out Graphics Display Pipeline」 IEEE Computer Graphics and Applications, p60-65, July, 1984.

(2) J.C. Peterson, 「The MARK III Hypercube-Ensemble Concurrent Computer」 1985 IEEE

(3) 和守慶, 吉田, 大宅 「3次元色彩図形表示処理専用プロセッサ HPRG」 計算機アーキテクチャ 58-4