

## マルチマイクロプロセッサシステムを用いた カラーグラフィックスシステム†

久保正敏<sup>†\*</sup> 戸島英一朗<sup>††</sup>  
星野寛<sup>††</sup> 大野豊<sup>††</sup>

マイクロプロセッサが安価、高機能化するにつれ、種々の応用向けにマルチマイクロプロセッサシステムが開発されている。われわれは、カラーグラフィックス向きに専用化されたマルチマイクロプロセッサシステムを開発し、これを用いて、グラフィックデザインシステムおよび3次元カラーグラフィックスシステムを開発した。これら二つのシステムは、マルチマイクロプロセッサシステムの特徴を生かし、機能分散と負荷分散に留意して開発されており、とくに3次元カラーグラフィックスシステムにおいては、演算量の多い隠面消去と付影の処理を一括して、しかも並列に処理する並列スキャニングアルゴリズムを開発している。本論文では、このマルチマイクロプロセッサシステムのハードウェアについては簡単な説明にとどめ、二つのカラーグラフィックスシステムの開発方針、アルゴリズムを中心に述べる。

### 1. まえがき

近年、カラーグラフィックスシステムのニーズが増大するとともに、ホスト計算機の負担軽減のため、高性能、高速処理能力をもつ画像生成端末の開発が盛んである。われわれはカラーグラフィックス向きに専用化されたマルチマイクロプロセッサを開発しこれを画像生成端末とするグラフィックデザインシステムおよび3次元カラーグラフィックスシステムを開発した。

マルチマイクロプロセッサシステムはG-PSYCOと名づけられ、並列プロセッサシステムを含んでいる。この並列プロセッサシステムは、画像生成処理に多く含まれるSIMD (Single Instruction stream Multiple Data stream) 型処理<sup>1)</sup>の高速化をねらい、独特のステップ同期方式と呼ぶ方式で制御され、PSYCO (Parallel processor system with SYnchronized COntrol) と名づけられている。

本稿で報告するグラフィックスシステムは、ホスト計算機も含めた機能分散に留意して開発されている。グラフィックデザインシステムは、そのベースとして用いたFORTH処理系の特徴を生かし、拡張性、操作性に優れている。3次元カラーグラフィックスシ

テムでは、shadow polygonの概念を利用し、隠面消去と付影を一括して並列処理する並列スキャニングアルゴリズムを開発した。

### 2. マルチマイクロプロセッサシステム G-PSYCO

G-PSYCOは、画像生成端末に必要な機能を分解し、それぞれをマイクロプロセッサで実現したシステムであり、図1に示すように、コントロールプロセッサを中心に、I/Oプロセッサ、ビデオコントローラ、および並列プロセッサシステムPSYCOが、メモリ共有で強結合された構成となっている。G-PSYCOのハードウェアについてはすでに他所で発表している<sup>2)</sup>ので、ここでは簡単な説明にとどめる。

PSYCOは、SIMD型処理に専用化された並列プロセッサシステムで、並列な各PE (Processing Element) はそれに固有のメモリLS (Local Storage) に格納されたMDに対し、共有メモリCS (Common Storage) に格納されたSIに従って演算を施す。CSには、データの値に依存しない制御フローをもつプログラムを格納しておけば、任意時点で各PEが実行する命令は同一となり、命令語フェッチのためのCSへのアクセスがPE間で揃い、アクセス競合が生じない。われわれはこのようなプログラムをDIF (Data Independent Flow) プログラムと呼び、これを用いた並列プロセッサシステムの制御方式を、命令のステップごとにPEが同期されるのでステップ同期制御方式と呼ぶ。この方式は、ハードウェア構成が簡単なためシステム

† Color Graphics System with Multi-Microprocessor System by MASATOSHI KUBO (Department of Information Science, Faculty of Engineering, Kyoto University), EIICHIRO TOSHIMA (Canon Inc.), HIROSHI HOSHINO and YUTAKA OHNO (Department of Information Science, Faculty of Engineering, Kyoto University).

†† 京都大学工学部情報工学科

††† キヤノン(株)

\* 現在 国立民族学博物館

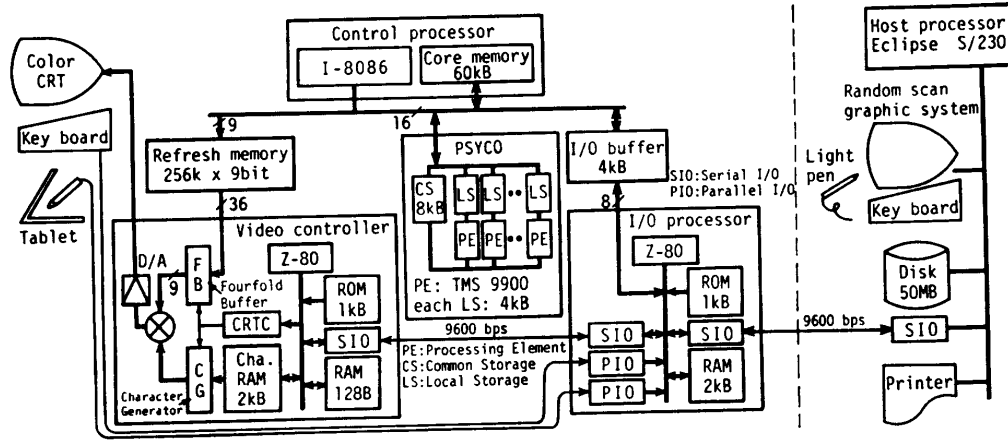


図 1 G-PSYCO のハードウェア構成  
Fig. 1 Hardware configuration of G-PSYCO.

の拡張性, 保守性に優れ, CS と各 PE を結ぶ共通バス上で競合が生じないため純粋な SIMD 型処理に対しては PE 台数に比例した速度向上が期待できる<sup>4)</sup>.

ビデオコントローラは, 256k x 9bit の画像メモリを順次読み出し, D/A 変換後カラー CRT へ送出する. 1画素に R, G, B 各 3bit を当て, 512 x 512 の解像度を得ている. グラフィック表示に, 64字 x 32行の文字を重畳表示する機能ももつ.

I/O プロセッサには, キーボード, タブレット, 文字出力機器としてのビデオコントローラが接続されていて, これら I/O 機器の制御を行うほかに, ホストプロセッサと G-PSYCO との通信制御も司る.

コントロールプロセッサは, G-PSYCO の核として, 各構成要素の制御を行うタスクスケジューラの機能や, PSYCO が生成した並列処理結果を編集するデータマネージャの機能を果たす.

### 3. グラフィックデザインシステム

#### 3.1 G-FORTH

G-PSYCO のように特殊なハードウェア環境におい

て大規模なソフトウェアシステムを開発するには, 支援ツールが欠かせない. われわれは, G-PSYCO のコントロールプロセッサ用ソフトウェア開発支援ツールとして FORTH 処理系を採用した. FORTH は, ワードと呼ばれる処理単位 (モジュール) を, すでに登録済みのワードを用いて定義し, システムのディクショナリに登録するという過程を, 対話的に動作確認しながら繰り返し, より大きなモジュールを次々と組み上げていく, 拡張性の優れた言語システムである. ディクショナリには, 高級言語風の制御構造記述ワードのほかに, アセンブリ言語レベルのワードも同じ形式で登録できるので, 特殊なハードウェア機能の対話的チェックと並行しながらソフトウェアを開発するのに便利である. われわれは, I-8086 用 FORTH を独自に開発し, G-FORTH と名づけている.

G-FORTH は, 図 2 に示すように FORTH 処理系に必要な諸要素を G-PSYCO 上に分散して実現したものである. すなわち, 入力テキストを解析するテキストインタプリタ, ワードをディクショナリに登録するコンパイラ, ワードの実行指令に対し, ディクショ

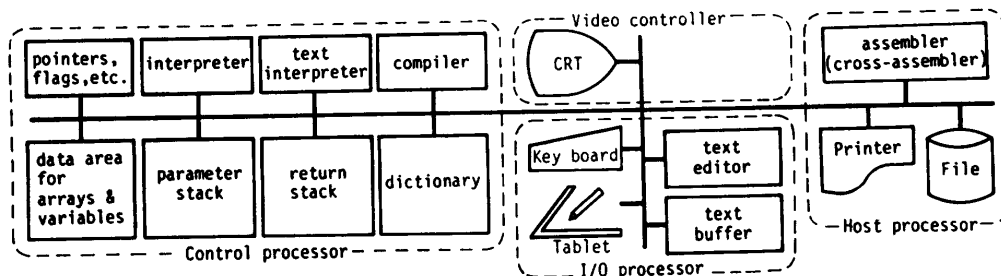


図 2 G-FORTH のシステム構成  
Fig. 2 System configuration of G-FORTH.

ナリを検索してその定義本体を取り出し実行するインタプリタ等の基本ルーチンはコントロールプロセッサに、コンソール制御、テキストエディタ等の入出力ルーチンはビデオコントローラおよび I/O プロセッサに、アセンブラ、ファイル等はホストプロセッサに、それぞれ置かれている。テキストインタプリタ、コンパイラは G-FORTH 自身で記述され、これに対応する基本ディクショナリはホストプロセッサ上でクロスアセンブラを用いてコンパイルできるので、G-FORTH 自身の開発は非常に容易であった。

### 3.2 グラフィックデザインシステム

G-FORTH をもとに、ポスターなどのグラフィックデザインを支援するシステムを G-PSYCO 上に開発した。グラフィックデザインにおいては、図面を次のような階層で捉えるのが自然であろう<sup>6)</sup>。

基本図形：直線、円、円弧、楕円、半楕円、文字。

部分図：基本図形を組み合わせた線図形。平行、回転移動の対象となる。

図面：部分図の集合、または図面の集合。

この階層に対応して、本システムでは、図 3 に示すように部分図作成モードおよび図面作成モードと、それぞれのモード用コマンドを用意している。

部分図作成モードでは、画面を見ながらカーソル、キーボード、タブレットを用い、希望する線図形の位置と組合せを決める。部分図は、内部データ形式の系列で表現され、G-FORTH の変数領域に置かれる。すなわち、部分図は名前が与えられ変数として扱われる。部分図の表示は、G-FORTH のアセンブリ言語レベルで記述された基本図形描図ワードが、この内部

データをディスプレイファイルとして解釈実行することにより行われる。

図面作成モードでは、部分図をさらに組み合わせ、線図の枠内を希望する色で塗りつぶして最終図面を作りあげる。枠内の塗りつぶしは、その枠の線図形をその中心に向かって順次縮小するという簡便な方法をとった。ただし、凹多角形に対しては枠からはみ出して塗色してしまうので、いったん凸多角形に分割しそれぞれを塗りつぶす。

図 4 に図面作成例を示す。図面作成モードには、この例に見られるように部分図を拡大、縮小しながら繰り返し描くコマンドも用意されている。これらコマンドはいずれも G-FORTH のワードであり、その定義は描図動作をそのまま手続きとして表現したものである。このように、本グラフィックデザインシステムでは、部分図作成モードにおいて線図形の組合せによる部分図を変数として定義し、図面作成モードにおいて変数を組み合わせる手続きとして図面を完成するという作成手順を採っており、これは、ポスタやイラスト図面等のデザイン活動に素直な対応をもつと考える。また、コマンドがすべて G-FORTH のワードとして定義されユーザに解放されており、これらを組み合わせさらに複雑な描図コマンドを定義できるので、FORTH の特長を生かした、柔軟性、拡張性に優れたグラフィックデザインシステムとなっている。

## 4. 3次元カラーグラフィックスシステム

### 4.1 3次元画像表示アルゴリズム

3次元画像表示では、隠面消去と付影処理の二つの問題を解決しなければならない。隠面消去の方法には、大別すると depth-buffer, scan-line, area-cohe-

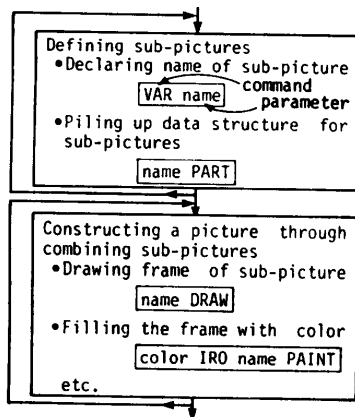


図 3 グラフィックデザインシステムの操作  
Fig. 3 Flow of user operation in graphic design system.

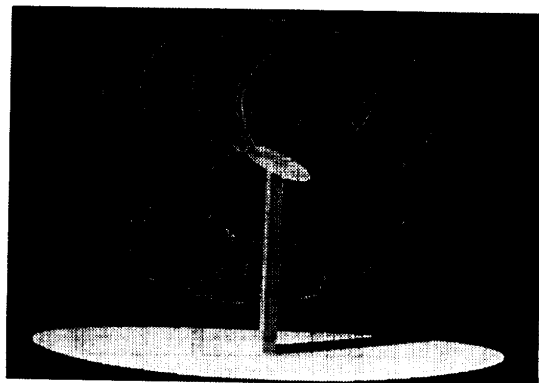


図 4 グラフィックデザインの例  
Fig. 4 An example of graphic design.

rence の3アルゴリズムがあるが<sup>5)</sup>, われわれは, 大量メモリを必要とせず処理も単純な古典的 scan-line アルゴリズム<sup>2)</sup> を基本とした. また, 付影処理には, shadow polygon<sup>3)</sup> の概念を利用した. この手法は, 凸多角形で構成されたいかなる形状の物体に対しても付影が容易であり, scan-line アルゴリズムに shadow polygon を含めることにより隠面消去と付影が一括して行えるという特長をもつ. また, われわれは実感的な画像を得るために, 同一平面上の点であっても光源との位置関係によって色彩を変化させている.

3次元画像表示アルゴリズムは, 図5に示すように三つのステージに分けられる. 3次元カラーグラフィックスシステムは, 処理データ量, 構造を考慮すれば, 立体モデル構造を仲介とする, 立体モデル作成段階とそれに対する画像の生成段階とに分けられるべきであり, 前者は工業製品設計, 都市計画等の応用システムに組み込まれ, 後者は画像生成端末に分担させるのが望ましいシステム形態である. われわれは, ホストプロセッサ上にステージ1を担う対話型立体モデル作成システムを開発し, G-PSYCO 上に, ステージ2,3を PSYCO を用いて並列処理する画像生成システムを開発した. コントロールプロセッサ上のソフトウェア開発には G-FORTH を用い, PSYCO 上の諸ルーチンは DIF プログラムの形態で開発した.

4.2 対話型立体モデル作成システム

ホストプロセッサ上でFORTRAN を用いて開発されたこのシステムは, ユーザによる立体モデル作成を支援し, G-PSYCO 上に開発された画像生成システムへの入力となるモデル構造を構築する. 立体は, 凸多

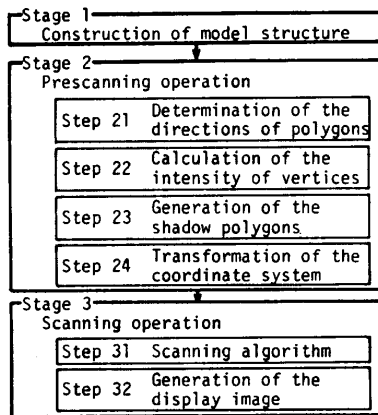
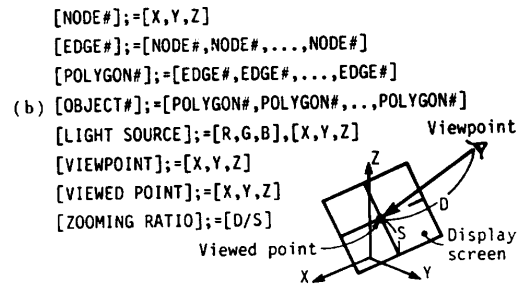
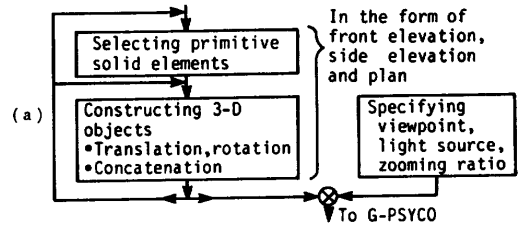


図5 3次元画像表示アルゴリズム  
Fig. 5 Algorithm of three-dimensional color graphics.



(a) 立体モデル作成手順  
(b) 立体モデルの構造

図6 立体モデル作成過程

Fig. 6 Stage of constructing 3-D model structure.

角形で構成される簡単なものに限定し, 曲面も平面で近似する手法を採っている.

立体モデル作成手順を図6(a)に示す. ユーザは, 基本要素として用意された, 直方体, 角錐, 角柱, 円錐, 円柱の中から, 寸法を指定しながら要素を選び, これらを移動, 接合しながら次々と組み合わせ, 立体モデルを完成する. 円錐, 円柱は40角錐, 40角柱で近似している.

二つの立体の接合は, 一方を固定, 他方を移動の対象として, 面, 稜, 点をそれぞれ介する接合を指定することにより行う. 立体各平面は, その立体を外から見て右まわりの順にリンクされた稜により記述されており, そのうちの2稜により平面方程式が計算される. この方程式を用いて, ユーザの指定する接合状態へ立体が移動される. 接合状態を保ったまま一方の立体を移動することも可能で, 位置合せのため指定された点の座標値も表示される.

立体要素の選択, 移動, 接合操作は, ホストプロセッサに結合されたベクトル型モノクログラフィックスシステムを用い, 線図で表示された立体の3面図を見ながら対話的に行われる. 高応答性を満たすため, 隠線消去は行っていない. 図7に, 面を介する立体接合途上の3面図による表示例を示す.

立体の組合せが完了すれば, 表示に関するパラメータとして, 立体各平面, 背景および光源の色, 視点お

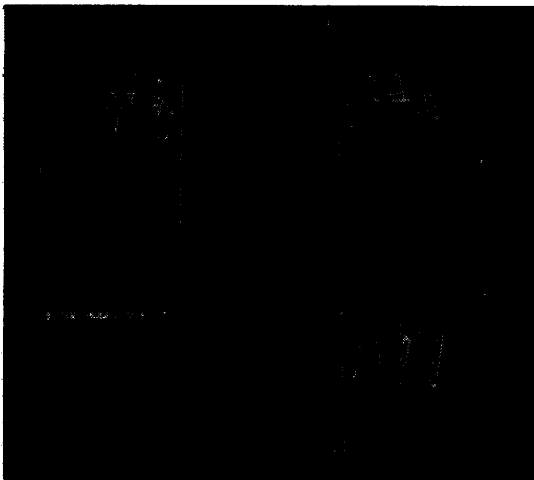


図 7 3面図表示による対話的立体モデル作成  
Fig. 7 Constructing 3-D model using orthogonal projection display.

および光源の位置，ズーム率等を入力する．これらを総合して，図 6 (b) に示す立体モデルが形成される．

#### 4.3 プリスキヤニング処理

G-PSYCO は，ホストプロセッサで形成された立体モデルを受け取り，まず図 5 で示したステージ 2 の処理を行う．このステージを構成する四つのステップはいずれも SIMD 型処理であり，PSYCO による並列処理に適している．コントロールプロセッサは，必要な処理を並列処理可能な子タスクに分解し，データを各 PE の LS に分配して PSYCO を起動する．並列処理が終れば，コントロールプロセッサは各 LS 上に生成された処理結果を収集し，次の並列処理サイクルに入る．以下に，各ステップを簡単に説明する．

##### ○ステップ 21 平面の向き判定

立体を構成する各平面の視点および光源に対する向きを判定する．各平面の平面方程式の係数は，その面の法線ベクトルに相当する．図 8 に示すように，このベクトルと，視点/光源へのベクトルとの内積を求め，

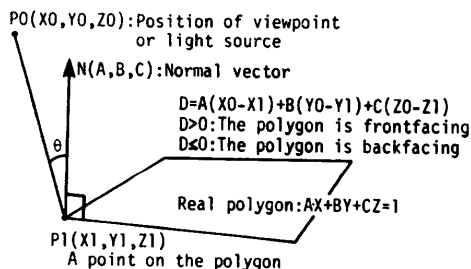


図 8 面の方向の判定  
Fig. 8 Determination of direction of polygon.

その正負で面の向きが判定できる．視点に裏向きの面は見えない面であり，スキヤニングアルゴリズムには不要である．光源に対する向きの判定は，後述するように shadow polygon の生成に必要である．

このステップでは，PSYCO の各 PE に平面に関するデータおよび視点，光源の座標値を与え，面単位の並列処理によって各面の向きを判定させる．

##### ○ステップ 22 代表点の輝度算出

簡単な式  $|\cos \theta|/R$  ( $\theta$ : 入射角,  $R$ : 光源との距離) により，平面頂点の輝度を算出する．平面上の他の点の輝度は内挿によりステージ 3 で求める．このステップでも，各 PE に平面のデータ，光源の座標値および色を与え，面単位の並列処理を行わせる．

##### ○ステップ 23 shadow polygon の生成

shadow polygon を生成するために，まず光源から見た立体の輪郭線，すなわち光源に表向きの面と裏向きの面との両者に含まれる稜を求める．shadow polygon とは，立体により影となる空間をとり囲む架空の平面であり，図 9 に示すように立体の輪郭線を光源に関して投射することにより得られる．これに対し，実際の立体を構成する平面を real polygon と呼んでいる．shadow polygon についても視点に対する向きを求める．視点に表向きの shadow polygon は，視点から見てその後方にある立体を影の内に入れ，裏向きのものは表向きのものの効果を打ち消す．表向きのものには +1，裏向きのものには -1 の，shadow count と呼ぶ値を与えておき，視点から real polygon 上のある点までの間に存在する shadow polygon の count 値の総計が正であれば，その点は影の内にある．この判定法により，付影処理を行う．

1本の輪郭線が一つの shadow polygon に対応するので，各 PE に輪郭線，光源および視点の座標値を与え，shadow polygon の生成と方向判定を行わせる．

##### ○ステップ 24 座標変換

スキヤニングアルゴリズムはスクリーン座標系上で適用されるので，これまでのステップで用いられた原

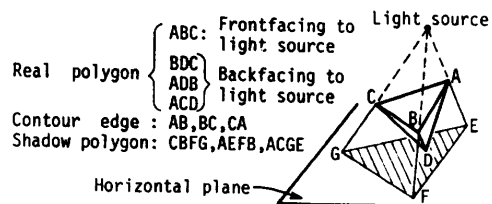
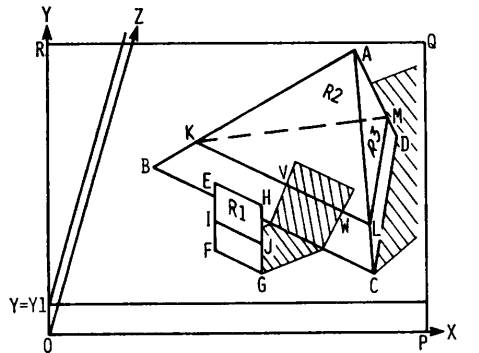


図 9 Shadow polygon の生成  
Fig. 9 Generation of shadow polygon.

座標系から、視点を原点とする視点座標系を経て、スクリーン座標系へ変換する。この過程は、立体各頂点座標値に同一の変換マトリクスを乗ずるといった典型的な SIMD 型処理であり、PSYCO の各 PE に頂点座標値を与え、点単位の並列処理を行わせる。

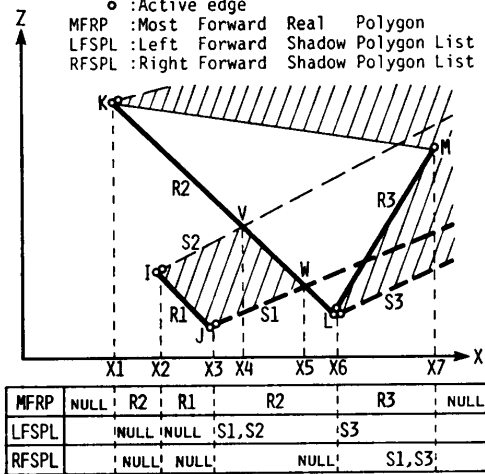
4.4 スキャンングアルゴリズム

スクリーン座標系上で隠面消去と付影を行うスキャンングアルゴリズムを、図 10(a), (b) を用いて説明する。(a)の平面 OPQR が表示画面に相当する。Y=Y1 のスキャン平面が物体 ABCD, EFGH を切断したときの切片同士の Z の値(深さ)の関係が(b)に示されている。左から右へ(b)をスキャンし、平面を構成する稜に出会う度に深さを比較し、隠面消去と付影



(a) スキャン平面

- : Intersection of frontfacing real polygon with scan plane
- - - : Intersection of backfacing real polygon with scan plane
- : Intersection of frontfacing shadow polygon with scan plane
- - - : Intersection of backfacing shadow polygon with scan plane
- o : Active edge
- MFRP : Most Forward Real Polygon
- LFSPL : Left Forward Shadow Polygon List
- RFSPL : Right Forward Shadow Polygon List



(b) 深さの関係

図 10 スキャンングアルゴリズム  
Fig. 10 Scanning algorithm.

を行う。このスキャン平面を上から下へスキャンすれば全画面の処理が終了する。

スキャンングアルゴリズム開始前に、複数の平面に含まれる稜を複製し、平面と稜との間に一対一対応をつけておく。現在のスキャン平面と交差する稜を、active edge と呼び、交点の X 座標値に従ってこれらをソートし、AEL (Active Edge List) と呼ぶリストを作成する。AEL から稜を次々と取り出し、その時点で視界に入る平面に “in”，視界から出て行く平面に “out” 状態のフラグを与える。現在 “in” 状態にある real polygon のなかで最も手前 (Z の値が最小) のものが見える平面であり、これを、MFRP (Most Forward Real Polygon) と呼ぶ。また、MFRP の左端の位置でその手前に存在する shadow polygon 群、同じく右端の位置でその手前に存在する shadow polygon 群をそれぞれ、LFSPL (Left Forward Shadow Polygon List), RFSPL (Right Forward Shadow Polygon List) と呼ぶリストにまとめる。このリストから、前述した shadow count 値により MFRP が影の内かどうかを判定する。

スキャン平面と稜との交点ごとに求められる MFRP, LFSPL, RFSPL が図 10(b) に示されている。X6 で求まる RFSPL は、X3 で求めた LFSPL と異なっている。これは、区間 X3, X6 で、R2 と shadow polygon S1, S2 とが交差することを表している。交点の X 座標値 X4, X5 を求め、この間で影の内に入ることを判定する。

これらの処理の結果、図 11 に示す key-square と呼ぶ情報が出力される。これは、スキャン平面に対応するラスタを左から右へスキャンしたときに色の変化が生じる X 座標値、新しい色および X の増加に対する色の差分を表すもので、図 10 では、X1~X7 の各点で生成される。この key-square に基づき、図 5 のステップ 32 でラスタ上の各画素の色が算出され、画像メモリに書き込まれる。

X	X-value where intensity is changed
R	Red intensity at the point
G	Green intensity at the point
B	Blue intensity at the point
DR	Difference of Red intensity
DG	Difference of Green intensity
DB	Difference of Blue intensity

図 11 Key-square の構造  
Fig. 11 Key-square.

4.5 並列スキャンニングアルゴリズム

スキャンニング処理を並列に行う手法として、われわれは図12に示す並列スキャンニングアルゴリズムを開発した。すなわち、スキャン平面内の active edge の個数を PSYCO の PE 台数で除して画面を分割し、それぞれの小画面を各 PE に割り当て、スキャンニング処理を行わせる。図は PE が4台の場合を示す。

図13にコントロールプロセッサおよび PSYCO の各 PE が分担する並列スキャンニングアルゴリズムのフローを示す。各 PE は、自分に割り当てられた小画面に対してステップ 31, 32 をまとめて実行する。コントロールプロセッサは分割した各小画面の左端におい

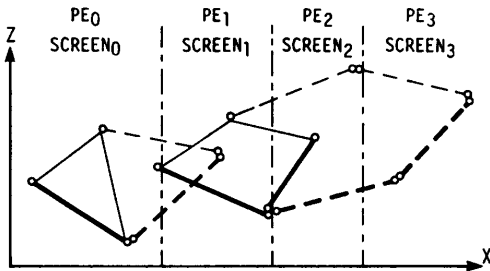


図12 並列スキャンニングアルゴリズムにおける稜の分割  
Fig. 12 Division of active edges for parallel scanning algorithm.

て“in”状態にある平面、および小画面に含まれる AEL サブリストを求めて PSYCO の各 LS に分配し PSYCO を起動する。各 PE は、与えられた AEL から一つずつ active edge を取り出し、その前後の active edge と X 座標値を比較しながら図に示す処理を行い、key-square を生成する。AEL の走査が終われば小画面内各画素の輝度を算出して LS に格納し、コントロールプロセッサに自分の処理終了を通知する。

コントロールプロセッサは、全 PE の処理が終了すれば各 LS から輝度を読み出し、画像メモリの適切な位置に転送する。PSYCO が1スキャン平面の処理を実行中に、コントロールプロセッサは次のスキャン平面に関する準備を行うという、2段階のパイプラインが実現されている。

4.6 画像表示例と並列処理効果

図14に画像表示例を示す。光源との位置関係により平面内の輝度を変化させ、実感的な画像を得ている。この例で、PSYCO の PE 台数を8に設定した G-PSYCO がホストプロセッサから立体モデルを受け取り画像表示を終了するまでに約1分を要した。

ここで、並列処理の効果を評価してみる。図15に PE 台数  $N$  と処理速度  $S$  との関係を示す。ただし、 $S$  は、 $N=1$  のときの速度が1となるように正規化して

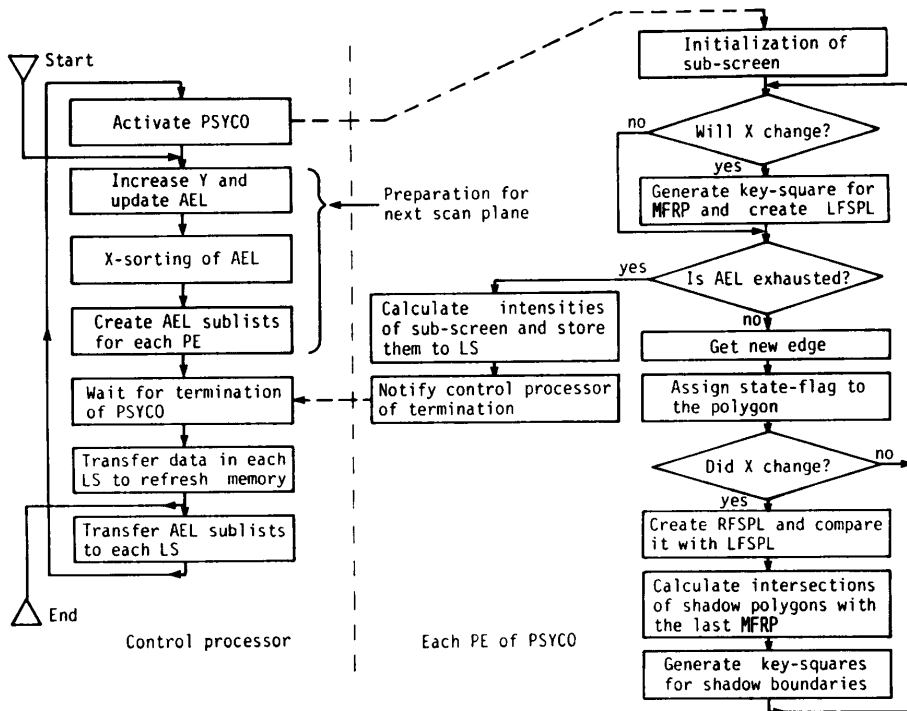


図13 並列スキャンニングアルゴリズムのフロー  
Fig. 13 General flow of parallel scanning algorithm.



図 14 3次元画像表示例

Fig. 14 An example of three-dimensional color image.

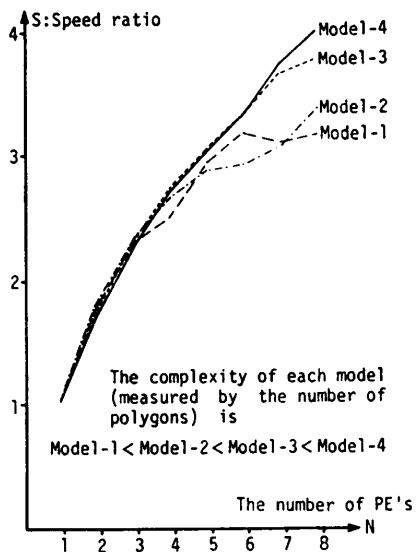


図 15 3次元画像表示における並列処理効果

Fig. 15 Effect of parallel processing for three-dimensional color graphics.

ある。Nの増加に伴い、Sは増加するが $dS/dN$ は減少している。この理由は次のように考えられる。処理時間の4/5を占めるスキャン処理のPSYCO側担当部は、図13に示したように二つのサブループを含み、平面の重なり合った部分においてはこのループの回数が増える。単純な構造の立体モデルでは、そのような部分は画面の一部に集中する。PSYCOはステップ同期方式で制御されているため、複雑な小画面を担当したPEと同期するために他のPEが待たされる状況が生じる。すなわち、特定のPEに負荷が集中し、Nの増加が負荷分散にあまり寄与しないので、 $dS/dN$ が減少すると考えられる。しかし、立体モデル構造が複雑となり、平面の錯綜する部分が画面に一

様に分散するにつれ並列処理により負荷が分散され、 $dS/dN$ の改善されることが、図からわかる。

## 5. むすび

本稿では、マルチマイクロプロセッサシステムを用いた画像生成端末の応用として開発した、グラフィックデザインシステムおよび3次元カラーグラフィックスシステムについて述べた。両システムは、グラフィックスシステムに必要な機能の分担に留意した機能分散システムとして実現し、とくに後者では並列プロセッサシステムによる負荷分散もあわせて実現したものである。今後、システム充実をはかるため、PSYCOのPEの高速化等のハードウェア拡充とともに、3次元画像表示アルゴリズムを再検討し、動画への応用も目ざした高速化についても考慮する予定である。

**謝辞** G-PSYCOのハードウェア開発には、京都大学工学部阿草清滋助教授の、3次元カラーグラフィックスシステム開発には、亀井朗（現東洋エンジニアリング）、氏原賢司（現四国電力）、田口泰志（現三菱電機）、森夏樹（現三洋電機）、渡辺正信（現シャープ）各氏の協力をいただいた。ここに記して謝意を表す。

## 参 考 文 献

- 1) Flynn, M. J.: Very High-Speed Computing Systems, *Proc. IEEE*, Vol. 54, No. 12, pp. 1901-1909 (1966).
- 2) Bouknight, W. J.: A Procedure for Generation of Three-Dimensional Half-Toned Computer Graphics Presentations, *CACM*, Vol. 13, No. 9, pp. 527-536 (1970).
- 3) Crow, F. C.: Shadow Algorithms for Computer Graphics, *SIGGRAPH-ACM*, Vol. 11, No. 2, pp. 242-248 (1977).
- 4) 久保, 阿草, 大野: カラーグラフィックス制御システム G-PSYCO, 信学技報, EC 78-37 (1978).
- 5) Newman, W. M. and Sproull, R. F.: *Principles of Interactive Computer Graphics*, 2nd Ed., McGraw-Hill, New York (1979).
- 6) 西原, 池田, 磯浜, 中山: 対話的図形定義における2分法の応用, 情報処理学会第21回全国大会, 1E-3 (1980).
- 7) Kubo, M., Taguchi, Y., Agusa, K. and Ohno, Y.: Multi-Microprocessor System for Three-Dimensional Color Graphics, *Proc. IFIP Congress*, pp. 145-150 (1980).

(昭和57年10月4日受付)

(昭和58年1月17日採録)