

コンピュータグラフィックスシステム LINKS-1の現状

山下伸⁺, 吉村浩⁺, 西田政人⁺, 出口弘⁺, 西村仁志⁺,
河田亨⁺, 白川功⁺, 大村皓一⁺ (†大阪大学工学部, †(株)東洋現像所)

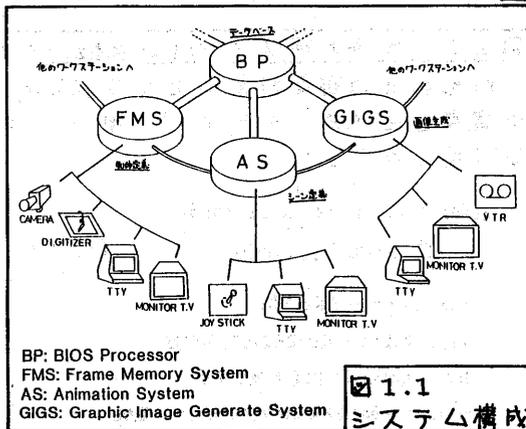
§1 はじめに

ここ数年でコンピュータグラフィックス(CG)は急速に普及し、テレビ映画等の映像分野でもCGを用いた作品が製作されるようになった。

CGを用いて映像(動画)を作成する手順は3段階に分けられる。

- (1) 物体定義: 個々の物体の形状を局所座標系で定義する。また色や透過率などの属性データを与える。
- (2) シーン定義: 物体と光源・視点の位置関係を定め、それらの時間的变化を定義する。
- (3) 画像生成: シーン定義に基づいて物体・光源・視点を配置したときの画像を発生する。

この一連の作業を総合的に処理するCGシステムLINKS-1を設計・試作した。本システムは、各作業を独立して行なうワークステーションとそれらを有機的に結合する共有データベースより構成されている^[1](図1.1)。



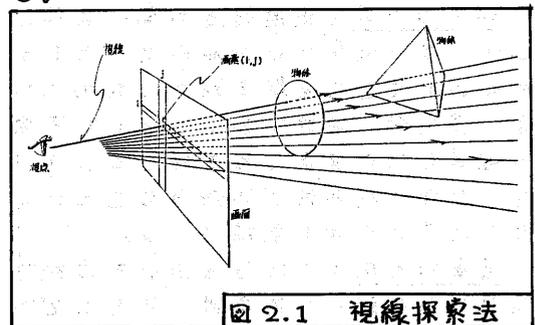
フレームメモリシステム(FMS)はディスプレイによる三面図入力などにより、物体定義を行なうシステムである^[2]。
アニメーションシステム(AS)は高速な動画表示を用いて、シーン定義を

行なうシステムである。FMSおよびASで定義されたデータを管理するデータベースは、BIOSプロセッサを用いて実現した。画像生成システム(GIGS)は定義されたデータに基づいて一連の画像を生成・収録するシステムで、疎結合マルチマイクロコンピュータによる並列処理を用いて、高速に実在感の高い画像を生成することを可能にした。

実用的な画像生成システムには、高速性のほかに、

- (1) ままざまな光学的現象(陰影・透過・屈折・反射など)の再現
- (2) 豊富な表現形式(主に自由曲面への対応)

が求められる。既にままざまな画像生成アルゴリズムが発表されているが、これらの条件を満たし、並列処理にも適したアルゴリズムとして、LINKS-1では視線探索法を採用した。本論文では、GIGSにみける視線探索法の実現手法ならびにその評価を報告する。



§2. 視線探索法

視線探索法は透視変換と隠面消去問題を陽に解決し得る画像生成手法であり、視点から光路を視線方向に探索し、視線との交点が最も近い物体を求めた

後、画面上の各画素の輝度を計算するものである(図2.1)。物体と視線との交点から仮想的な視線を出して再帰的計算を行えば、透過・反射処理も可能であり、また交点と光源を結ぶ直線と交差する物体を探索することにより影付けも行なえる。

2.1 表現形式

視線探索法に用いる物体の表現形式としては、視点との交差判定と交点における視線ベクトル[†]の計算が簡単に行なえるものであればよい。このためさまざまな表現形式が可能であり、現在のものは次の4種類の表現形式を用いている^{[7][8]}。

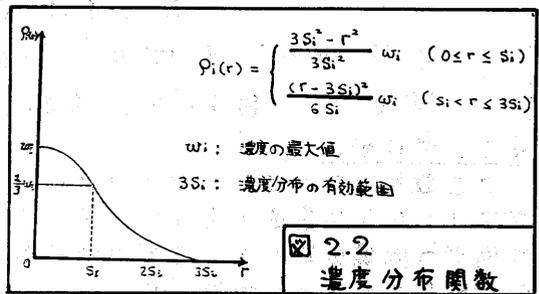
- (1) 三角板
- (2) 積円体
- (3) 濃度球表現
- (4) 直方体

三角板は、面が平坦な物体、すなわち多角形や多面体を表現するとその基本形状として広く用いられるほか、自由曲面の表現にも応用している。すなわち、曲面を充分小さな三角板に分割して近似表現するもので、交差判定は各々の三角板に対して行なう。ただしこのままでは曲面のなめらかさが出ないため、三角板の境界で視線が連続に変化するように補間する手法がとられる^{[10][11]}。

球・積円体は使用頻度の高い物体であり、かつ簡単な式で表現できるため、われわれはこれを基本形状に含めている。積円体にはオプションとして平面方程式を付加することを許しており、この平面で積円体を切断することにより円柱等の形状も表現できる。

濃度球表現は主に自由曲面を表現するための手段として考案したもので^[6]、空間に連続に分布する関数の等ポテンシャル面をもって物体の形状を定義するものである。濃度球は空間内で点対称に広がる濃度(ポテンシャル)分布であり(図2.2)、これを空間に多数配置することにより、さまざまな形状

が定義できる。



直方体は多面体であるから三角板でも表現できるが、より高速に処理するために基本形状として採用した。

このほかにも、必要に応じて新たな表現形式を比較的簡単に導入することができる。

2.2 輝度計算

視線探索法では光源に対する制約も少なく、光線方向ベクトルと強度がわかればよいための、LINKS-1では数種類の光源を用意している(図2.3)。

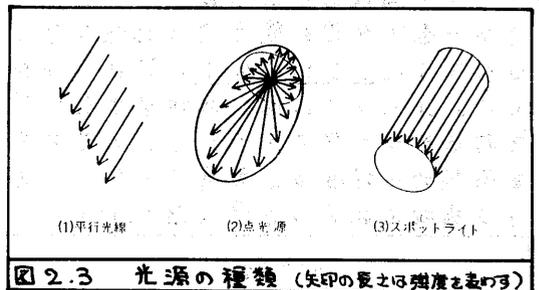


図 2.3 光源の種類 (矢印の長さとは強度を意味する)

われわれは輝度計算式として phong の提案した次式のモデル^[9]を用いている(図2.4)。

$$I_0(p) = drc + drc \sum_{i=1}^N Ls_i \cos \theta_i + \sum_{i=1}^N src(\theta_i) Ls_i \cos^n \theta_i \quad (2.1)$$

$I_0(p)$: 点Pにおける面の輝度

drc : 周囲から一様に照らされている強度

drc : 入射光に対する散乱強度係数

$src(\theta)$: 入射光の表面反射強度係数 θ の関数

n : 表面反射の鋭さを定める定数

Ls_i : i 番目の光源から来る光(視線)の強度

†: 交点における視線ベクトル。その点の輝度を求める際にも用いられるが、図中方向ベクトル、反射方向ベクトルと異なるものにも用いられる。

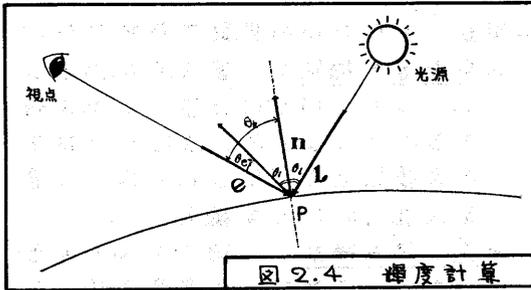


図 2.4 輝度計算

θ : 面の法線と光源の光線との間の角度

ϕ_e : 視線と見る側の光線の反射方向の間の角度

もし透過や反射が起こるような物体であれば、透過方向から来る光強度 I_t と反射方向から来る光強度 I_r を再帰的に求め、交点 P にあける輝度を次式で最終決定する。

$$I(p) = I_0(p) + \tau r_c(\theta) I_r + \tau t_c I_t \quad (2.2)$$

θ : 視線と法線の間の角度

τr_c : 物体の持つ透過率

また影処理は点 P と光源の間に存在する物体を探索し、見つけた物体の透過率を考慮して (2.1) 式の I_0 を決定することにより行なえる。

(2.1) 式の各係数は普通 1 つの面に対して 1 組与えるだけでよいが、表面の不均一性や印刷された模様を表現したい場合があり、このためにマッピングという手法が用いられる。マッピングとは、あらかじめ色や表面係数の 2 次元配列を用意し、これを物体表面に投影して平面の属性を変化させる手法である。

§3. 画像生成システム

視線探索法は各画素ごとに独立に計算が行なえる特徴を持つため、画素を単位とした並列処理を行なうならば、各並列処理モジュール間の通信は不要である。可能なシステムは星状の連結構成がよく、確実に高性能を發揮し、しかも安定に動作する。

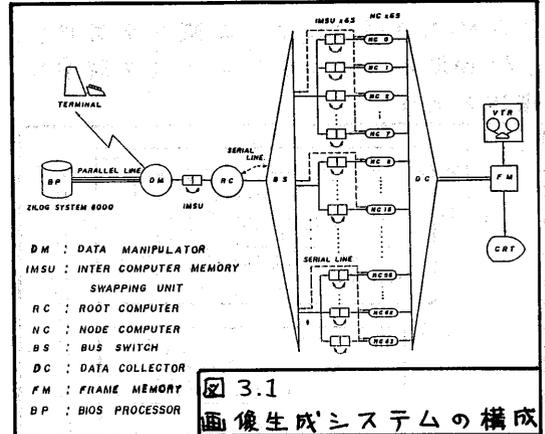


図 3.1

画像生成システムの構成

3.1 システム構成

LINKS-1 の画像生成システムは図 3.1 の構成をとっている。データ操作モジュール (DM) はデータベースとの通信をパラレル回線を通じて行ない、シーンデータに基づいた物体の配置・移動、および外接直方体を求めるなどの前処理を行なう。このように前処理を画像生成から独立したモジュールに行なわせることにより画像生成中に次のデータを準備することができ、連続運転時に効率のよいパイプライン処理が行なえる。画像生成は 64 台のノードコンピュータ (NC) が並列して行ない、ルートコンピュータ (RC) によってモニタリングされる。DM で前処理されたデータは RC に転送され、RC は各 NC に対し負荷を均等に分担させる働きをする。

DM・RC・NC はすべて同一のハードウェア構成を持ち、これらを単位コンピュータと呼んでいる (図 3.2)。単位コンピュータは Z8001 (クロック 4 MHz) をメイン CPU とするマイクログラファコンピュータで、浮動小数点演算を高速に行なうためのサブ CPU iAP X86/20 を備えたマルチ CPU 構成をとっている。またシステムバスによって

†: 詳細は 4 章参照のこと。

RAMを1MB単位で拡張できるほか、I/O機器とのインタフェースを容易に増設できる。

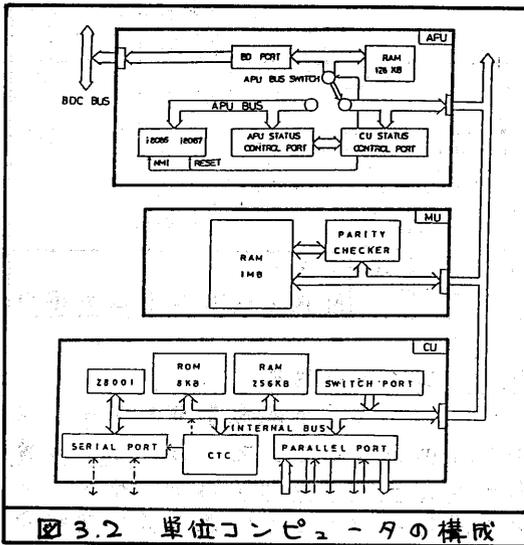


図3.2 単位コンピュータの構成

3.2 単位コンピュータ間の通信

単位コンピュータ間の通信は主としてメモリ交換ユニット (IMSU) を用いて行なっている。IMSUは図3.3のように128KBのメモリアドレスを2組持っている。これがそれぞれ単位コンピュータのシステムバスに接続され、CPUのアドレス空間の一部に割り当てられ通常のRAMとしてアクセスされる。そして簡単なハンドシェイクの後このメモリアドレスを交換することにより双方向通信を行なうので、大量のデータを高速に転送することができる。

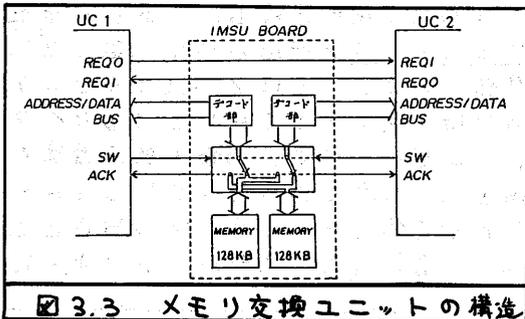


図3.3 メモリ交換ユニットの構造

IMSUには以下のような星状結合

に有効ないくつかの機能を持たせた。

- (a) 重ね書き機能: 図3.4 (a)のように1つのIMSUに書き込みを行なうことによって、並列に接続される複数のIMSUにも同時に書き込みが行なえる機能。
- (b) 差し読み機能: 図3.4 (b)のように1つのIMSUの内容を読み出すと同時に、他のIMSUに書き込みが行なえる機能。
- (c) アドレスシフト機能: 重ね書き・差し読みが広く行なえるように、CPUからのアドレスに任意のオフセットを加えてメモリをアクセスする機能で、図3.4 (b)は差し読みと併用した例である。

図3.1のようにRCのシステムバスには非常に多くのIMSUが接続されているため、ドライブ能力の増強とアドレス空間の拡張を行なうためのバススイッチを間に挿入している。

さらにLINK S-1画像生成システムでは、IMSUの他に、SDLC回線によってもRC-NC間の通信を行なうことができる。

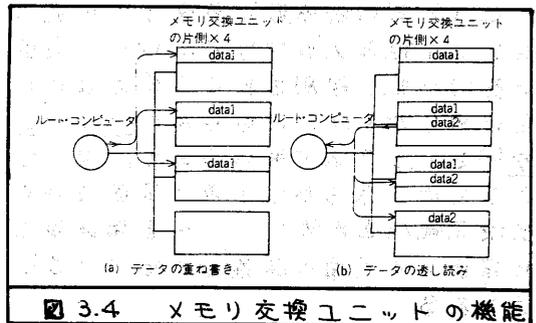


図3.4 メモリ交換ユニットの機能

3.3 フレームメモリ [3][12]

NCで計算された結果は、画面アドレスと輝度情報より成る54bitの画素データとしてFIFO型データコレクタ^[13]に集められ、フレームメモリに書き込まれる。フレームメモリは1024×1024画素、RGB各8bitの容量を持っており、NTSC方式あるいはNH

Kが提案する高精細度テレビ（HDTV）方式¹⁰のビデオ信号を出力し、画像をテレビモニタに表示することができるものである。

§4 画像生成プログラミング手法

視線探索法は光学系を正確にシミュレートすることにより奥在感の高い画像を生成できる秀れた手法であるが、他のアルゴリズムと比較して1画素あたりの計算量が多い。これを補うため、LINKS-1では、前章のハードウェア的手法（並列処理・パイプライン処理・教値演算プロセッサ）に加えて、いくつかのソフトウェア的手法をとり入れている。また各NCが効率よく並列動作するように、動的負荷分散法と呼ばれる手法を導入している。

4.1 クラスタ構造の階層化

シーンデータを定義する際、1つ1つの形状データについて動きを定めるよりは、いくつかの形状データを一括して扱えるようにするほうが便利である。このような形状データの集合をクラスタと呼んでいる。またいくつかの物体が集まって何かを形成している場合が多いため、クラスタを階層的な木構造にするとさらに都合がよい。たとえば人間の形状を図4.1のように階層化したクラスタで表現すれば、右腕を

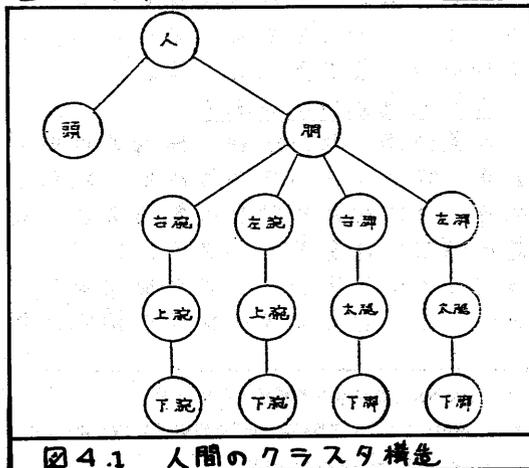


図4.1 人間のクラスタ構造

動かすことにより右腕の下のレベルにある上腕や下腕も同時に動かすことができ、より自然な記述が可能になる。

クラスタはこのようなデータ作成上の便宜だけでなく、画像生成時にも非常に有用であり、多くのアルゴリズムでクラスタ構造を利用して画像生成を高効率化している¹⁰。

4.2 外接直方体によるクリッピング

座標軸に垂直な面より構成される直方体を考え、ある形状データにこれが外接するものとする。このとき視線がこの外接直方体と交差しなければ、その形状データとも交差しはないことは明らかであり、これを利用して候補データの選抜ができる。外接直方体が(4.1)式で表わせるとき、点Pより出射

$$\begin{cases} x_{min} \leq x \leq x_{max} \\ y_{min} \leq y \leq y_{max} \\ z_{min} \leq z \leq z_{max} \end{cases} \quad (4.1)$$

した視線ベクトル e が交差する条件は、(4.2)式を満たす正の d が存在することである。

$$\begin{cases} x_{min} \leq P_x + d e_x \leq x_{max} \\ y_{min} \leq P_y + d e_y \leq y_{max} \\ z_{min} \leq P_z + d e_z \leq z_{max} \end{cases} \quad (4.2)$$

ただし、 $P_x, P_y, P_z, e_x, e_y, e_z$ はP, eの各座標成分

LINKS-1画像生成システムではこの手法をクラスタに対しても適用している。すなわち、各レベルのクラスタに対し、自分の子のクラスタあるいは形状データの外接直方体を前処理と

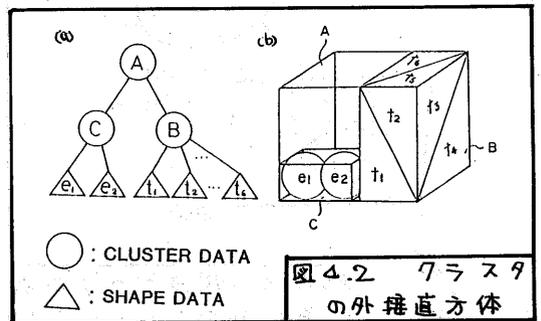


図4.2 クラスタの外接直方体

してあらかじめ求めておく(図4.2のクラスタ構造に対しては、図4.2bのように各クラスタに対応した外接直方体を与える)。そして視線との交点を求める際に、木構造のクラスタを根から順にたどる。つまり、視線がクラスタの持つ外接直方体と交差する場合のみ、さらに下のレベルを探索する。多くの場合局所的に存在する形状データをまとめて1つのクラスタとするため、視線から離れて存在する形状データ群を効率良くふるい落とすことができる。この手法は反射・透過・影処理に用いられている。

4.3 視点座標系への交換

反射処理・透過処理の際の仮想的な視線の出射位置は予測することはできないが、本来の視点は一定なので、ここを原点とすることによって処理は簡単になる。そこでわれわれはDMにおける前処理で、すべての形状データを視点座標系に変換している。視点座標系では、視点を原点とし、画面の右をX、上をY、そして画面に垂直な視線方向をZ⁺としている。この変換は画像生成と独立にパイプライン処理されるので、この変換時間自体は問題にならない。

4.4 画面上でのクリッピング

各々の形状データが画面上のどの部分に投影されるかを、図4.3のように外接直方体を画面に投影することによ

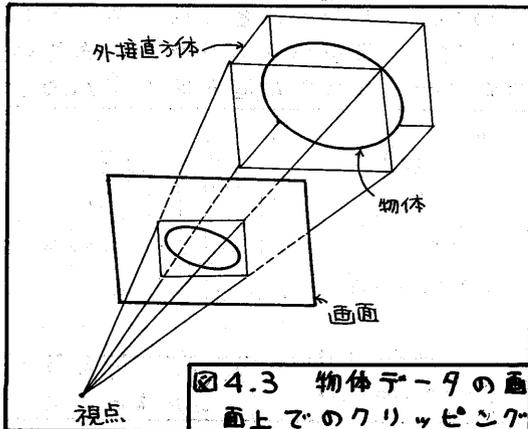


図4.3 物体データの画面上でのクリッピング

り求めることができる。この長方形領域を前処理として各形状データに与えれば、画面上の任意の画素を通る視線と交差するクラスタを抽出するのに非常に役立つ。

各NCが一度に担当する画面上の領域は走査線1本分とその一部分で、いずれにしても画面上で横方向に連続した領域を受け持つ。このためこの走査線と交わる形状データをはじめに抽出すれば、かなり高速な処理が期待できる。そこで各NCはRCから担当領域を受けとると、すべての形状データの中から画面上での長方形領域が与えられた担当領域と交わるものだけを抽出する(Y方向のクリッピング)。この作業は整数の比較演算だけで行なうことができ、またクラスタの階層構造も利用できるもので、かなり高速に終わることが出来る。

Y方向でのクリッピングを行なって得られた形状データの中から、ある画素を通る視線と交わる可能性があるものを抽出するために、X方向でのクリッピングを行なう。すなわち、その画素が各々の長方形領域の内部にあるかどうかを、画面上でのX座標を比較することによって決定する。Y方向でのクリッピングの結果得られた形状データ群を、画面上でのX座標値によってソーティングしておくことにより、この処理は一層高速化できる。

4.5 透過処理の高速化

透過処理・反射処理は再帰的に輝度計算を行なうものであるから、特殊な状況下では膨大な計算が要求され、最悪の場合には過度のネスティングによってシステムスタックが破壊されてしまう。そこでネスティングの最大回数

†: われわれは右手系の座標系を用いている。

‡: 三角板については、三角板をその向きに投影することにより、それによりクリッピングが行なえる。

を人為的に定めて、不必要な処理を行なわないようにしている。

しかし透過処理の場合、視線が物体を透過している途中で処理が打ち切られてしまうと、本来見えるはずの透過光が見えなくなり、その部分で不自然に輝度が下がってしまう。これを解決するため、非屈折透過処理を打ち切り時の透過計算に利用している。非屈折透過処理とは屈折を伴わない透過処理であり、視線と交差する形状データをすべてバッファに保存しておくことにより、比較的容易に実現できる。多くの場合、この非屈折透過処理だけで十分な実在感を出すことができるようである。

4.6 動的負荷分散法

画像生成速度を高めるためには、各NCの処理時間ができるだけ等しくなるように負荷を分散し、並列処理の効率を高める必要がある。ところが、画面を面積的に等分割して各NCに分配しても、多くの場合負荷の重い画素が画面上に偏在しているため、処理時間のばらつきがかかり大きくなる。また画面上での負荷の分布状態を正確に予測することも困難であるため¹⁰⁾、われわれは次に示すような動的負荷分散法を用いてこの効率を高めている。

画面をNCの数に比して十分に大きな数の小画面に分割し、NCにはその小画面を1つずつ分配して処理を開始する。NCは、それぞれの小画面での処理を終了したら、RCに対し次の処理領域を要求し、これに対してRCはNCに次の小画面を渡して処理を継続する。

この場合、NCの処理時間の分布の広がりやばらつきは、1つの小画面を処理するのに必要な時間より小さくなるはずであるから、小画面をさらに小さくすることによってこの効果は増大する。しかし一方で、RC-NC間の通信オーバ

ヘッドが増大し、またY方向でのクリッピングの効果が低下するため、全体の処理時間を長くしてしまう。これを解決するため、小画面の面積を一定にせず、はじめのうちは比較的大きな面積の小画面を分配し、処理が進むにつれて、より小さい小画面を分配する方法をわれわれは採用している。これにより、必要以上のオーバヘッドを抑え、かつ並列処理の効率の高い処理が行なえる。

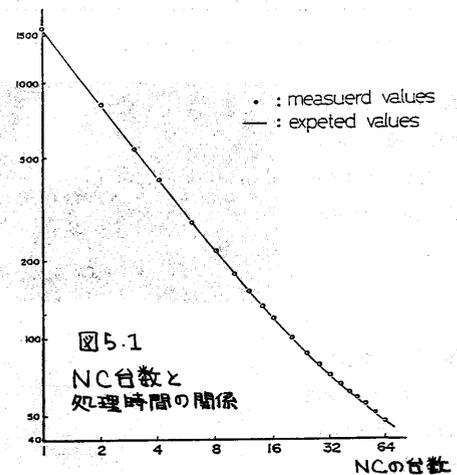
§5. システムの性能評価

前章までに、LINKS-1の画像生成システムを高速化するために採用したいくつかの手法を紹介したが、本章では実験データからその効果を評価する。

5.1 ハードウェア的手法の評価

図5.1は、写真1の画像を生成するために要する時間 T を、並列処理するNCの台数を変えて測定したものである。この結果では、1台の時と64台の時の速度比は1:63.07であり、高い並列性が実現されている。この結果から、

画像生成時間 (sec)



†: 透過・反射・影処理の必要があれば、着しく計算量が増大する。

‡: DMにおける前処理の時間は測定の対象外とする。

NCの台数を増やすことにより、確実に処理速度が上がるのがわかり、将来VLSI技術の発達によって大規模な並列処理が行なわれるならば、非常に高速な処理が可能となることが予想される。

表5.1は、浮動小数点演算を $\Sigma 8001$ を用いて行なった場合と、 $APX86/20$ を用いて行なった場合の速度 \dagger を測定したものである。この結果によると、単純な演算を $\Sigma APX86/20$ を用いて行なった場合、通信のためのオーバーヘッドによりむしろ速度が低下している。すなわち、数値演算プロセッサを有効に使用するためには、多くの行列演算を含んだ計算等に関数単位で行なわせるべきだと結論できる。われわれはこの結果に基づき、このような $\Sigma APX86/20$ のソフトウェアモジュールを作成中である。

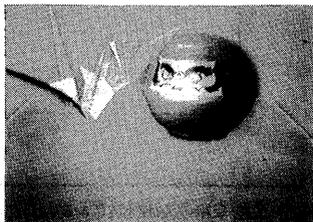
表5.1 数値演算プロセッサによる演算速度の改善 (単位 mSec)

演算 CPU	バケトルと バケトルの和	バケトルと バケトルの積	行列と 行列の和	行列と 行列の積
$\Sigma 8001$	0.43	0.79	1.4	8.4
$\Sigma APX86/20$	0.51	0.57	0.80	1.8

写真1 738枚の三角板で構成されるカメラ、監視カメラの動的画像分割を行なっている。



写真2 写真1のデータに、170枚の三角板で構成された建物の断面を加え、床のマスキングを施したものである。



5.2 ソフトウェア的手法の評価

表5.2は、写真2の画像を生成するために要する時間が、使用するソフトウェア的手法によってどのように変化するかを測定したものである。

- ① type 1 : 4章で説明したすべての手法を用いたもので、透過はすべて非屈折透過処理を用いている。
- ② type 2 : type 1からX方向でのクリップング処理を省いたもの。
- ③ type 3 : type 2からさらにY方向でのクリップング処理を省いたもの。
- ④ type 4 : type 3からさらに外接直方体によるクリップング処理を省いたもの。

この結果から、各々の手法によって確実に処理速度が数倍から数十倍に上がっており、最終的にはアルゴリズムをそのままプログラムしたとき（type 4）の約100倍になっていることがわかる。特に外接直方体によるクリップングは効果が大きく、これだけで25倍の処理速度を得ている。

また写真2のデータに対して非屈折透過処理を行なわずに通常の透過処理を行なったときの処理速度を測定した結果、分割数2400のとき212秒であった。これは非屈折透過処理を用いた場合の2.59倍の時間を要していることになる。実際の作品において、透過は必要だが屈折は特に必要としないことがしばしばあるので、非屈折透過処理は表5.2 ソフトウェア手法による速度の改善 (単位 sec)

分割数 type	type 1	type 2	type 3	type 4
2400	82	111	340	8499
480	86	200	340	8430

†: 通信オーバーヘッドを含む。行列は3×3。バケトルは3元のもの。

‡: 測定はNCを16台使用し、動的画像分割法を行なっている。

‡‡: 動的画像分割を行なうときの小画像の数を、7×7と同じ面積とした。

実用面において非常に有用な手法であると言える。

5.3 動的負荷分散の評価

動的負荷分散において、分割する小画面の数と処理時間の関係を測定した結果を図5.2に示す。実験に用いたデータは写真1であるが、分解能は512×480とし、影処理を省略した。また処理時間についてはその分布範囲を調べた。

実験の結果、64台のNCを有する本システムの場合、全画面を1024(32×32)に分割したとき処理速度が最高に行っている。すなわち、分割数が少ない場合は各NCの処理時間のばらつきが大きく、逆に分割数を多くすると処理時間のばらつきはなくなるが、通信オーバーヘッドが増加するため処理速度がかえって遅くなるわけである。

分割数64(8×8)の場合は実質的に動的負荷分散を行なっていない時のデータと考えることができる。これと比較すれば、動的負荷分散の効果によって処理速度が約1.9倍になっており、動的負荷分散の有用性がわかる。

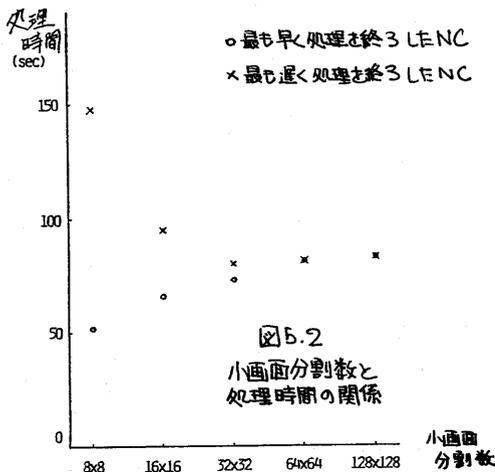


図5.2
小画面分割数と
処理時間の関係

5.6 おわりに

さまざまな隠面消去アルゴリズムの中で低速な部類に属する視線探索法を、

実用的な速度で実現するには多くの手法を必要とした。これを実現した現在、視線探索法の持つ卓越した実在感・表現力は、非常に魅力的な映像を生み続けている。しかしこれでも満足できる速度ではない。

映像芸術家にとっての創作活動は試行錯誤の繰り返しであると言われ、入出力間の応答が速いほど円滑な創作が可能になる。われわれは視線探索法の表現力を落とすことなくさらに処理速度を高めることによって、映像芸術分野におけるCGの地位を確立することかできると確信し、さらにLINKS-1の能力を向上させる研究を続けている。

参考文献

- [1] 面村仁志他 "LINKS1: コンピュータグラフィックスシステム", 1982.11
- [2] 河合利幸他 "画像データ操作システム LINKS-DMIS", 1982.11
- [3] 中山賢司他 "画像生成用マルチマイクロコンピュータシステム", 1982.11
- [4] 大野廣司他 "採用BIOSマシン", 電気学会情報処理研究会資料 1982.4
- [5] I. E. Sutherland, et al., A Characterization of Ten Hidden-Surface Algorithms. Computing Surveys, Vol. 6, No. 1, Apr. 1974.
- [6] 面村仁志, "対話型画像生成システムに関する研究", 1982.3
- [7] 吉村浩他 "LINKS1における画像生成手法", 1982.11
- [8] 吉村浩 "コンピュータグラフィックスシステムLINKS1のソフトウェアの作成とその評価", 1983.3
- [9] P. Bui-Tuong, Illumination for computer generated pictures. Commun. ACM, Vol. 18, No. 6, June 1975.
- [10] J. F. Blinn, et al., Texture and reflection in computer generated images. Commun. ACM, Vol. 19, No. 10, Oct. 1976.
- [11] W. M. Newman, et al., Principles of Interactive Computer Graphics. McGraw-Hill, 2nd Edition, 1979.
- [12] 大野廣司 "コンピュータグラフィックスシステムLINKS1のハードウェアの作成とその評価", 1983.3
- [13] 竹田史 "高解像度カラーディスプレイシステム", National Technical Report 25-3 1979.6
- [14] 面村仁志他 "マルチマイクロコンピュータシステムにおける画像生成のための画面分割方式", 信学技報 CAS82-144 1983.2

†: 情報処理学会マイクコンピュータ研究会資料

‡: 大阪大学工学部研究科電気工学専攻特別研究報告