LINKS-1における画像生成手法

吉村 浩 辰己敏一 西村仁志 河田亨
白川 功 大村皓一 （大阪大学 工学部）

[1] まえがき

近年、コンピュータを用いてCRTディスプレイ上に3次元画像を生成するコンピュータグラフィックスが注目を集め、アニメーションや各種シュミレーション、機械設計などさまざまな分野への応用が拡まりつつある。この3次元画像は、物体を線で表わすワイヤーフレーム表示と、陰影をほどこした面で表わすシェイディング表示とに大別される。

両者のうち物体の3次元形状や材質感をリアルに表現するためにはシェイディング表示が優れている。このシェイディング表示による画像生成は、次のような性質をもっている。

i) 基本的な処理は、与えられた視点、光源、物体の位置関係から、表示画面を構成する各画素における輝度を計算することである。

ii) 各画素における計算は、どの物体がその画素に見えるかを求める計算（交差判定）と、この計算により求まった物体表面の向きと、光源、視点の位置関係から物体の表面の輝度を求める計算（輝度計算）とに分かれる。いずれも比較的簡単な演算（3次元の行列、ベクトル演算）から成り立っている。

iii) 画面を構成する画素は、通常数十万～数千万点あるため、たとえiiで述べたように各画素に対する演算が簡単であったとしても、画像生成には膨大な演算が必要である。たとえば5/2×5/2の分解能である画面をリアルタイム（30枚/秒）に生成するためには、1画素平均の演算時間は130ns以下でなければならない。

シェイディング表示画像を効率よく生成するためのアルゴリズムとして、すでに多くの文献[6][7]が発表されているが、多くは既存のコンピュータを用いてii)における演算をいかに少なくするかを述べたものである。

これに対し、われわれは画像生成処理が本質的にもっている並列処理の可能性、すなわち各画素単位の演算を独立に行なうことができる点に注目し、LINKS-1[1]における画像生成システムとしてマルチコンピュータ構成をとり、1枚の画像生成を複数台のマイクロコンピュータで並列に行なっている。

本報告では、LINKS-1上で実現した画像生成の並列処理アルゴリズムについて述べる。

第2章では画像生成手法の概要について述べる。第3章では基本となるアルゴリズムとして採用した視線探索法について述べる。第4章では複数台のコンピュータへの画面分割方式について述べる。

[2] 画像生成手法の概要

2-1 画像生成の満たすべき条件

われわれは、LINKS-1における画像生成手法として、次の条件を満足するものを考えた。

i) 並列処理の効率を上げるために単位コンピュータ間の通信、データ転送が少なく、各コンピュータが独立に演算を行なう割合が大きい

ii) 単位コンピュータの台数の変化

（故障や将来の拡張）に即応できる。

iii) 各単位コンピュータの処理時間が均等になる。

iv) 基本となる画像生成アルゴリズムはいろいろな物体表現形式に対して適用できる。

v) 生成画像は、3次元陰影処理をほどこしたものとする。そして物体の表面状態や影あるいは透明感も表現できる。

2－2 システムの構成

まず、i)の並列処理の効率であるが第1章で述べたように画像生成は各画素に対する処理が独立に行なえるのであるから、画面を分割して各コンピュータへ割り当てる方式がよい。またii)とiii)を考慮すれば、画面分割はハードウェアにより固定されたものであってはならず、生成すべき画像によって可変でなければならない。

そこで、LINKS－1では図1のように、1台のルートコンピュータ（以下RC）と複数台のノードコンピュータ（以下NC）を接続した構成をとった。

RCは、画像生成に必要なデータをディスクから読み、各NCにおける演算負荷が均等になるように画面を分割し、各NCに割り当てる。NCは与えられた分割画面上の全画素の輝度を計算し、結果をフレームメモリ（FM）へ格納する。

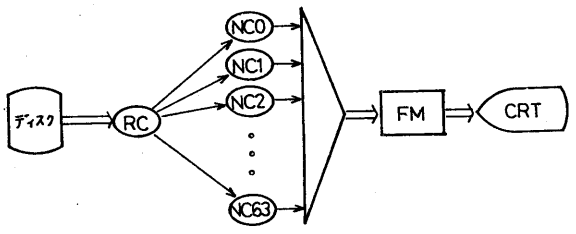

図1. システム構成

[3] NCにおける画像生成手法

3－1 視線探索法

われわれはLINKS－1の画像生成アルゴリズムとして視線探索法[5][8]を採用した。

視線探索法とは、図2に示すように視点と画面の各画素を通る半直線（視線）が、各物体と交差するかどうかを求めていく方法である。

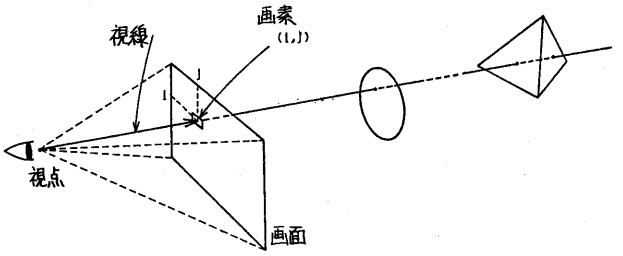

図2. 視線探索法

この視線探索法は次のような特長をもっている。

i) 視線と各物体との交点までの距離を比較し、最も視点に近い物体表面を決定することで、隠面消去を簡単に行なえる。

ii) 他のアルゴリズム（スキャンラインアルゴリズム[6][7]、Zバッファ法[9]）などのように、物体同志の位置関係を求めるのではなく、常に視線と1物体の関係を求めるので、各画素における処理は全く独立に行なえ、効率よく並列処理を行なえる。

iii) 物体の表現形式は、視線との交差判定が容易に行なえるものであればよい。したがって、さまざまな形式で与えることができる。

iv) 視線と物体の交差判定と同じ処理ルーチンを用いて、影処理や光の反射、透過を表現することができる。

また欠点としては、他のアルゴリズムと比べた場合、1画素を処理するた

めに要する演算量が多いことであるが、この点は画面分割による各NCへの負荷分散と、NCに演算専用ユニット(APU)[2]を付加することによって補っている。

3－2　物体表現形式

先に述べたとおり、視線探索法では画面の各画素に対応した視線と物体との交点を求めることが基本となる。したがって物体形状の表現形式は直線との交点が簡単な式で解けるものが望まれる。

現在われわれは以下に示す3種類の物体形状の表現形式を用いている。

3－2－1　三角形分割法

LINKS－GMS[3]を用いて入力された多面体は、内部で三角形に分割して処理を行なう。これは、いかなる多角形も三角形に分割が可能であり、処理を一定にすることができるためである。

三角形と視線との交点は以下のようにして求める。

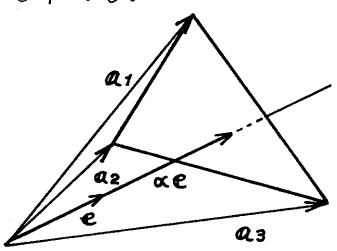

図3　三角形との交差判定

図3に示すように、視線方向を示す単位ベクトルをe、三角形の3頂点の座標をそれぞれa_1、a_2、a_3とするとき、

$$h = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = [a_1, a_2, a_3]^{-1} e \qquad (3\cdot1)$$

で求められるhが非負の要素からなる時、視線は三角形と交点をもつ。

この時、視点から三角形までの距離αは、

$$\alpha = \frac{1}{h_1 + h_2 + h_3} \qquad (3\cdot2)$$

で求まる。

また、Bースプライン曲面や双3次曲面などにより記述された自由曲面と直線との交差判定を厳密に行なうにはかなりの演算量を必要とする。

そこで、多面体と同様に三角形に分割して処理を行なっている。ただし、面の法線ベクトルを一定とするのではなく、3頂点の法線ベクトルn_1、n_2、n_3を別々に与えておき、式(3·1)におけるhを用いて、

$$n = h_1 n_1 + h_2 n_2 + h_3 n_3 \qquad (3\cdot3)$$

としている。

3－2－2　球、楕円、円柱

簡単な式で曲面を表わせるものとしては球や楕円、円柱がある。

これらは2次形式で次のように表わせる。

$$|A(x-m)|^2 = 1, \quad A = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \qquad (3\cdot4)$$

式(3·4)に、視線$x = \alpha e$を代入してαに関する2次方程式ができる。

$$|Ae|^2 \alpha^2 + (Ae \cdot Am)\alpha + (|Am|^2 - 1) = 0 \qquad (3\cdot5)$$

交点が存在するのは、式(3·5)が正の2実根を持つ場合である。

さらにオプションとして平面の方程式を与え、物体を任意の平面で切断できるようにしている。

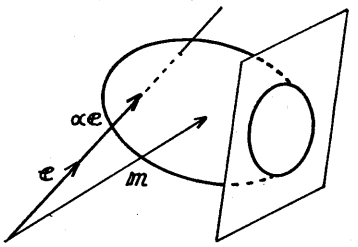

図4　楕円との交差判定

3－2－3　重み関数による表現

上に述べた三角形分割や2次形式だけでは表現できる物体の形状が制限されてしまう。そこで、より自由に物体形状を表現する手段として以下に述べる重み関数を考案した。

まず基本となる関数として、有界な台を持ち、$\rho(0)=1$で、なめらかに減少する関数を1つ選ぶ。そして、空間上のある点Pに重みwを与え、任意の点はPの影響によって、$w\rho(r)$（rはPからの距離）の重みを持っていると考える。

このような重みの分布を空間上に複数個置けば、任意の点はこれらの総和

$$W=\sum_i w_i \rho(r_i) \qquad r_i:\ i\text{番目の重み中心からの距離} \tag{3·6}$$

の重みを持つことになる。そして、$W=$一定となる点は空間上でなめらかな曲面を作るので、あらかじめ物体の表面における重みの値W_{surf}を定めておけば、表面がなめらかにつながった物体を表現することができる。

Wは3次元座標値を変数とする関数であるから、視線探索法で用いる視線$\alpha\boldsymbol{e}$（$|\boldsymbol{e}|=1$）上の重みはαをパラメータとして

$$W(\alpha)=\sum_i w_i \rho(r_i) \tag{3·7}$$

$$r_i=\sqrt{(\alpha e_x-P_{ix})^2+(\alpha e_y-P_{iy})^2+(\alpha e_z-P_{iz})^2}$$

$$\boldsymbol{P}_i=\begin{bmatrix}P_{ix}\\P_{iy}\\P_{iz}\end{bmatrix}:\ i\text{番目の重み中心位置}$$

で与えられる。

したがって、視線との交点を求めるためには、

$$W(\alpha)=W_{surf} \tag{3·8}$$

をαについて解けばよいことになる。

さて、基本となる関数$\rho(r)$の定め方であるが、われわれはrについての2次式を組み合わせ、

$$\rho(r)=\begin{cases}\dfrac{3S^2-r^2}{3S^2} & (0<r<S)\\[2ex] \dfrac{(r-3S)^2}{6S^2} & (S<r<3S)\end{cases} \tag{3·9}$$

S：重みの有効範囲

を用いた。（図5参照）

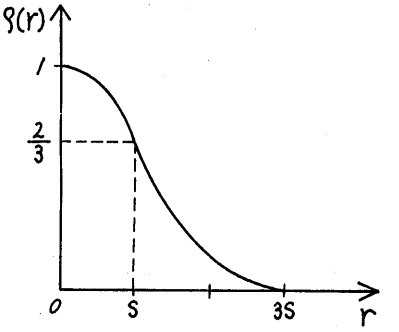

図5 基本関数

例として、3つの重み分布がある場合の$W(\alpha)$の概略図は図6のようになる。

図6(b)では、1つの重み分布に負の重み分布を与えている。負の重み分布は他の重み分布を減じる作用をするため、物体の表面にくぼみを作ったり、表面を削ったりすることができる。

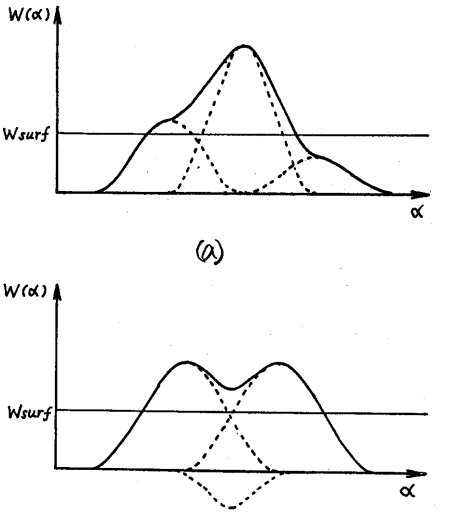

図6 重み関数の例

次に式(3·8)の解を求めるのであるが、式(3·7)から分かるように、W(α)はαの1/2乗項を含んでいるため簡単に解くことはできない。そこで1つの方法として近似法を用いた。

求解の手順としては、

i) W(α)が持つ局所的な台(W(α)≠0の区間)を求める。

ii) この区間を重み分布の個数に応じたサンプル数で等間隔にサンプリングする。

iii) これらのサンプル点間を、内挿法を用いて2次関数W*(α)で近似しW*(α)=Wsurfを解く。

となる。(図7参照)

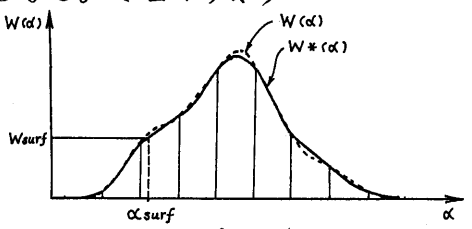

図7 近似法

以上のような重み関数W(α)を用いた場合の物体形状の決定は、気に入った形状になるまで重み中心点を付加、移動あるいは重み値を増減させたりする処理を繰り返すことによって行なう。この処理は、変化させるパラメータと物体の形状との間が直接対応し、かつ局所的な修正ができるため、人間にとって操作しやすいものである。

ただし、重み関数と視線の交差判定は、三角板や楕円などと比べ演算量が多いため、演算回数を減らすために、近似多項式やサンプリング点の個数の決定などについて、さらに検討する必要がある。

3-2-4 物体データの階層化

視線と各物体との交差判定は、それ自体さほどの演算を必要としない。しかし、この処理ルーチンの繰り返し回数は(物体数)×(画素数)回となる。

そこで、RCにおいて物体を幾つかのグループに分け、グループ全体の外接直方体を作成しておき、NCにおいてはまず視線がこの直方体と交差するかどうかを判定し、交差する場合のみ内部の物体それぞれとの交差判定を行なうことにして、交差判定の回数を減らしている。

さらに、幾つかの外接直方体を1つのグループとし、このグループに対する外接直方体を作成することにより、物体データを階層的に構成している。

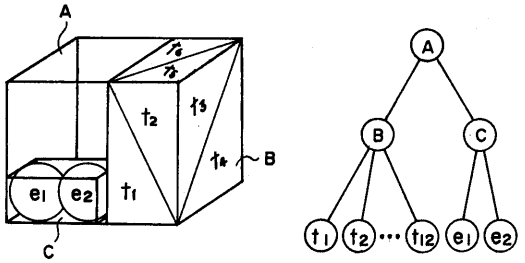

図8 物体データの階層化

3-3 輝度計算

3-3-1 光線の定義

視線と物体との交差判定が終われば次に画素の輝度を計算しなければならない。まず、光源としては以下に示す3種類のモデルを用意している。

i) 平行光線

この光源は全空間いたるところで同じ強度を持ち、平行である。太陽光線などがこれにあたる。

ii) 点光源

街灯のように、ある場所から全方向へ照射される光である。ただし強度分布は全方向に一様ではな

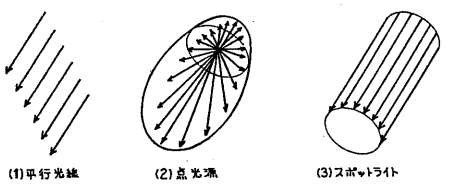

図9 光源の種類
(矢印の長さは強度を表わす)

く、式(3・10)のような分布をもつ。

$$\text{強度} L_s = \begin{cases} L_1 + L_2 \cos\theta & -90° \le \theta \le 90° \\ L_1 & \theta < -90°,\ \theta > 90° \end{cases}$$

ただし θ：最高強度方向からの角度
L_1, L_2：定数　　(3・10)

iii)スポットライト

i)と同じく平行光線であるが、空間内で半無限な円筒内のみ強度をもつ。テレビや映画のスタジオで使うスポットライトを想定する場合に用いる。

なおすべての光源の強度は光の3原色の値で与える。したがって、色つきの光源を定義することも可能である。

3-3-2 輝度計算式

画素の輝度は、視線と最初に交わった物体の面の法線ベクトル n、視線 e 各光線の入射方向 L_i と、面の状態を表わすいくつかの係数との関係から計算する。われわれはこの関係式として、phong の提案した次式のモデル[10]を用いている。

$$I_o(P) = dic + drc \sum_{i=1}^{N} L_{si} \cos\theta_i + \sum_{i=1}^{N} src(\theta_i) L_{si} (\cos\theta_e)^n \qquad (3・11)$$

ただし、$I_o(P)$：点Pにおける面の輝度
dic：物体表面が周囲から一様に照らされている強度
drc：入射光の散乱強度
$src(\theta_i)$：入射光の表面反射強度で θ_i の関数
n：表面反射の鋭どさを決める定数
L_{si}：i番目の光線の強度 $(i = 1 \sim N)$

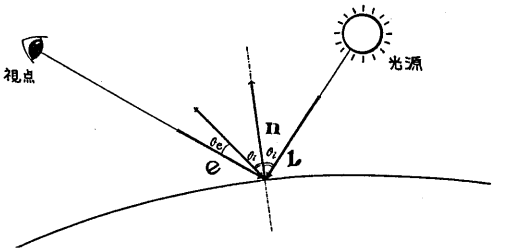

図10　視線、法線、光線の関係

なお式(3・11)の dic, drc, L_{si} は、光の3原色の成分をもつため、面の輝度も3原色それぞれについて計算する。

3-3-3 影、反射、透過処理

(1)影処理

式(3・11)では、すべての光線が物体表面に達するものとしているが、実際には光源位置と物体表面の間の空間に、他の物体がある場合には、光線が表面には達せず影ができる。この影処理に対しては、視線探索法の特長をいかして次のような手法を用いている。

i) 視線と物体表面の交点Pから、光線の入射方向へ仮想的な視線 e_L ($e_L = -L$) を想定する。

ii) e_L と各物体との交差判定を行なう。この処理は3.2節で述べた処理と同一である。

iii) e_L が不透明な物体と交差した場合はその光線を、式(3・11)の計算から除外する。

iv) 交差した物体がすべて光の透過率 t_i ($0 < t_i < 1$) をもつ場合には、光線の強度 L_s を $L_s' = t_1 t_2 \cdots t_n L_s$ でおきかえる。

(2)反射処理

物体の表面が鏡や金属である場合は他の物体がこの表面に映って見える。この現象は物体の表面反射係数が大きい場合におこる。そして映って見える物体は、視線 e を物体表面で正反射した方向 ($\theta_r = \theta_e + \theta_i$) にある物体である。

そこで θ_r 方向に仮想的な視線 e_R を考え、各物体との交差判定を行なう。

e_R は、

$$e_R = e - 2(e \cdot n) n \qquad (3・12)$$

で与えられる。e_R が物体と交差した場合はその交点 P_R における輝度 $I(P_R)$ を求め、これを $I_o(P)$ に加え

$$I(P) = I_o(P) + src(\theta_R) I(P_R) \qquad (3・13)$$

とする。

(3) 透過処理

物体が、ガラスのように光を透過させる材質である場合、物体のもつ屈折率 n ($n=\frac{\cos\theta_t}{\cos\theta_e}$) で定まる方向にある物体が見える。

そこで、この θ_t 方向へ仮想的な視線 e_t を考え、各物体との交差判定を行なっている。

透過方向への視線 e_t は、

$$e_t = S - \sqrt{1-|S|^2}\, n \qquad (3\cdot14)$$

で求められる。ただし S は、e_t が物体に侵入する時

$$S = n\{e-(e\cdot n)n\} \qquad (3\cdot15a)$$

e_t が物体から出る時

$$S = \frac{1}{n}\{e-(e\cdot n)n\} \qquad (3\cdot15b)$$

である。

そして、e_t が物体と交点をもつ場合反射処理の場合と同様、交点 P_t における輝度 $I(P_t)$ を計算して $I(P)$ に加える。

$$I(P) = I_0(P) + src(\theta_R)I(P_R) + tI(P_t) \qquad (3\cdot16)$$

ただし、t は物体のもつ透過率

$I(P_R)$ や $I(P_t)$ を求める段階で、これらの点 P_R、P_t から新たに光源、反射、透過方向へ視線を出すことにより、輝度計算は再帰的に行なわれる。この処理のようすを図11に示す。

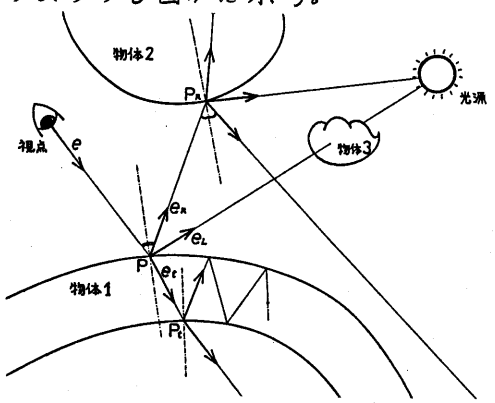

図11 再帰的な輝度計算

[4] RCにおける画面分割方式

画像生成は複数のNCにより並列に行なわれるが、並列処理の効率をよくするためには第2章で述べたように、各NCにおける演算負荷が均等になることが要求される。

しかしながら、画面上の各画素における演算は一定ではなく、多くの物体と交差する画素では当然多くの演算が必要である。理想的には、あらかじめ各画素ごとの演算負荷を予想し、これに応じて割り当てる画素の個数を定めればよいが、これではRCにおいての演算が多くなりすぎる。

そこで現在われわれは、画面分割を画素単位ではなく、より大きな領域(小画面、走査線)を単位として行なう方式について検討している。次に検討中の画面分割方式について述べる。

4-1 小画面分割方式

(処理手順)

i) 画面を均等な大きさの小画面に分割する。分割数はNC台数を上回る数(16, 64倍など)とする。

ii) 各物体データ S_j の演算負荷 w_j を決める。w_j は物体の表現形式および表面の条件(反射や透過の有無)により決定する。

iii) S_j が画面に透影された図形の外接長方形を求める。なお、物体数が多い時は、3-2-4で述べた外接直方体を処理の単位とする。

iv) 各小画面の負荷を次のように計算する。

$$W_i = \sum_{j=1}^{M} w_j A_j \qquad (4\cdot1)$$

M: 物体数　w_j: j 番目の物体の負荷
A_j: j 番目の物体の外接長方形と小画面 i との交差する面積

v) 全画面の負荷を計算し、これをNC台数で割って1台当たりの平均負荷 $\overline{W}$ を求める。

vii) 小画面を画面の左下から右上に向かって負荷の和がWに達するごとにグループに分け、NCに対して1つのグループとその領域内に存在する物体データを割り当てる。

4－2　走査線分割

画面を走査線に分割し、最も下の走査線から順に1本ずつNCへ割り当てる。したがって各NCにはNC台数本おきに1本の走査線が割り当てられる。

この方式では、隣り合った走査線間ではほぼ同じ物体が存在することから各NCの負荷がほぼ均等になる。

しかしながら、1台のNCに割り当てられる領域が全画面に分布してしまうので、割り当てられる物体データが他の方式と比べて多くなってしまう。

4－3　動的分割

4－2の方式は、負荷がほぼ均等になるが、物体データがNCのメモリ容量を超えてしまう可能性がある。そこで走査線をNC処理中に動的に割り当てる方式を考える。

まずRCは、各NCに対し1本の走査線（あるいは1つの小画面）だけを割り当て、その後各NCの状態を監視する。NCは割り当てられた走査線（小画面）の処理が終了するとRCに対して要求を出す。RCは、まだ処理していない走査線（小画面）が存在する場合には次の走査線（小画面）とその領域内に存在する物体データを割り当てる。

4－4　影、反射、透過処理に対する処置

3-3-3で述べたような、光の反射透過、影処理をほどこした画像を生成するためには、各小画面の領域内には存在しない物体のデータも必要である。

このような場合は、全物体データを各NCへ割り当て、領域内に入る物体にはフラグを付けて区別することにする。もし、物体データ量がNCのメモリ量を上回る時は、領域内に入る物体データを優先的に割り当て、NCの処理中に割り当てられていない物体データが必要となった場合にはRCに対して要求を出し、新たに物体データを受けとる方式をとっている。

[5] あとがき

LINKS－1における画像生成手法について、第3章ではNCにおける交差判定、輝度計算、第4章では画面分割方式を報告した。

以上のソフトウェアはLINKS-1用に開発された拡張C言語を用いて記述されている。[4] また、行列、ベクトル演算は各NCに付属のAPUを起動して行なっている。

現在われわれは、本報告で述べた交差判定や輝度計算などの処理ルーチンをファンクション化し、APUでまとめて演算するソフトウェアを開発中であり、画面分割のアルゴリズムや新しい物体表現形式について、実験・検討を行なっている。

[参考文献]

(1) 西村、出口、大野、河田、白川、大村、尾崎「LINKS－1：コンピュータグラフィックスシステム」
(2) 中山、平井、大野、西村、江木、河田、白川、大村「画像生成用マルチマイクロコンピュータ」
(3) 河合、吉村、出口、西村、河田、白川、大村「画像データ操作システム LINKS－DMS」
(4) 出口、河合、中西、西村、河田、白川、大村「プログラミングシステム LINKS－C」
以上 マイコン研究 1982, 11
(5) 西村、「対話形三次元画像生成システムに関する研究」、大阪大学工学部修業論文、1982, 3
(6) Ivan E. Sutherland, Robert F. Sproull, and Robert A. Schumacker, "A Characterization of Ten Hidden-Surface Algorithms", Computing Surveys, vol6, no.1, march 1974
(7) Carpenter, Blinn, Whitted, "Scan Line Methods for Displaying Parametrically Defined Surfaces.", CACM, 23-1, January 1980
(8) Whitted, "An Improved Illumination Model for Shaded Display", CACM, 23-6, June 1980
(9) Csuri, Hackathorn, Parent, Carlson, Howard "Towards an Interactive High Visual Complexity Animation System", Siggraph79 proceedings
(10) Phong, "Illumination for Computer Generated Images", CACM, 18-6. June 1975