グラフィクスとＣＡＤ　40－1
（１９８９．８．１７）

ボクセル・チェーンを用いた３Ｄグラフィクス

石井郁夫　小野尚紀　大和淳二　牧野秀夫
新潟大学工学部

３Ｄシーンの実時間合成、実時間３Ｄアニメーション、対話型３Ｄグラフィクスなどに適する高速３Ｄ画像生成方式を提案する。高速化とデータ圧縮のため、３Ｄシーンをボクセル(Voxel)のチェーンで記述する。この記述法は隣接ボクセル間の位置関係を表わすベクトルのチェーンだけで３Ｄ形状を表現する。したがって、少数の隣接ボクセル間結合ベクトルのパラメータの変更だけで任意形状物体の回転やスケーリングが可能で、比較的小規模のハードウェアで実時間化できる。あらかじめ３Ｄシーンを構成する複数の物体をそれぞれ独立のサブスペース上にボクセルチェーン方式で記録しておく。色はボクセル毎に与える。物体の向きと視点に対応した結合ベクトル・パラメータならびにワールド座標上のサブスペース原点座標を物体毎に与え、それらを適宜更新することによって、物体毎に独立な回転・移動・スケーリングなどが表現できる。さらにＺバッファによる陰面消去を専用ハードウェアにより実時間で処理する。

3D Graphics Using Voxel Chains

Ikuo ISHII, Naoki ONO, Junji YAMATO, Hideo MAKINO
Faculty of Engineering, Niigata University

A method of high speed 3D imaging for realtime 3D graphics is described. For fast processing, 3D scene is represented by voxel chains. The voxel chain is a row of vectors indicating the positions of next neighboring voxels. Rotation and scaling of all over the object can be realized by changing only a few parameters of these vectors. Therefore, high speed 3D imaging is realized by small scale hardware. The objects forming 3D scene can be defined in different sub-spaces. The movement of each object can be represented independently by modification of parameters of each sub-spase according to the position and orientation of each object or visual point.

１．はじめに

３Ｄシーン中の物体の動きの実時間表現や対話的３Ｄグラフィクスの実用化が期待されている。また、我々は美術工芸品などの高品位立体像展示システムや合成シーンの視点移動実時間追従立体表示装置の実用化に向けて、高速グラフィクスの研究を進めている。しかし、これら３Ｄシーンは膨大な演算処理を必要とし実時間化の障害になっている。この解決策の１つとして、医用３Ｄ画像処理を目的としてボクセルアーキテクチャが開発され、有望視されている[1]。これはボクセルと呼ぶ単位立体セルを大容量の３Ｄメモリ上にマッピングすることによって立体形状を表現する。この表現形式は自由な立体形状が扱えるほか、スクリーンへの投影処理手続きが規則的になりハードウェア化し易い利点がある。高速化と高能率化の実現のため種々の処理方式の提案がある[2]-[4]が、いずれもメモリアドレスと空間座標を直接対応付けた巨大な３Ｄメモリが必要で、処理装置も膨大になるため縮小プロトタイプしか実現されていない。また機能上の制約も多く、より自由な表現と操作性の向上が期待されている。

我々は視点移動追従立体表示用に、ボクセル・チェーンを用いた高速３Ｄシーン描画機構を開発した[5][6]。隣接ボクセル間の位置関係を表わすベクトルのチェーンだけで３Ｄシーンを表現するため、巨大な３Ｄメモリを必要とせず、座標変換手続きが簡単になり、高速化とハードウェアの小型化に有効である。

本報ではボクセル・チェーン方式の利点を活用した新しい実時間３Ｄグラフィクス処理法を述べる。ボクセル・チェーン方式は少数の隣接ボクセル間結合ベクトルのパラメータの変更だけで自然情景を含む任意形状物体の回転やスケーリングができることを特長とする。ワールド空間上の物体の向きと視点の方向によって定まる結合ベクトルパラメータを専用のハードウェアに与えると、その透視投影像を即座にスクリーンに表示できる。複数の物体をそれぞれ独立の副空間上にボクセル・チェーンで表わし、ワールド空間上の各副空間原点座標と各物体の結合ベクトルのパラメータを適宜更新することによって各物体が独立に実時間で移動、回転、スケーリングするシーンの表示ができる。

２．座標変換方式

(1) ３Ｄ情景の表現法

複数の物体を独立の物体定義副空間 $X_iY_iZ_i$ (i=1,2,…n)上に定義し、それらをワールド空間 $X_wY_wZ_w$ 上に配置することによって３Ｄ情景を表わす。物体定義副空間座標各軸を１ボクセル間隔で分割する３Ｄ格子を考え、その交点上に複数のボクセルを配置することによって立体形状を表わす。各ボクセルの座標には絶対値を用いず、隣

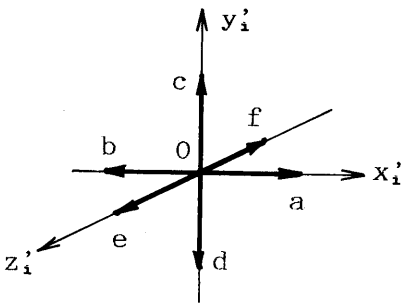


図１　隣接ボクセル間結合ベクトル

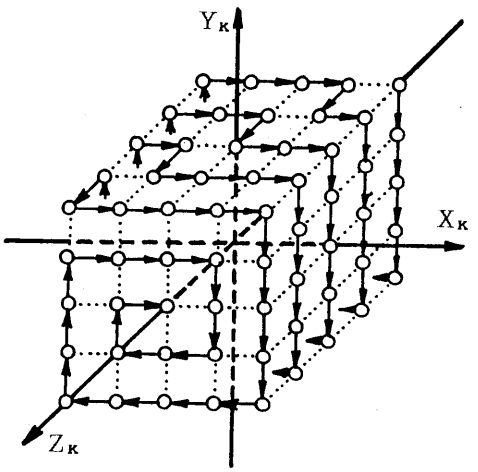


図２　ボクセル・チェーンによる立体表現の一例

接ボクセル間の相対的位置関係を識別する符号で記述する。$x'_i y'_i z'_i$ を $X_i Y_i Z_i$ 座標の差分副空間とし、図１の６種類の差分ベクトルa～fを定義する。各差分ベクトルの大きさは１格子間隔である。図２は i=kの物体定義副空間のボクセル・チェーンによる立体表現の一例である。a～fの一つを選択しながら隣接ボクセル間を順次結合して立体形状を表現する。

表示モードと移動モードのどちらかで隣接ボクセル間を結合する。表示モードでは各ボクセルの色を指定しながらボクセル間を順次結合する。移動モードは単に座標点の移動のためにボクセル間を結合して、空間位置の指定などに利用する。チェーンの先頭座標を常に物体定義副空間原点に取り、２つのモードを切替えながら結合ベクトルを順次選択し、選択符号の列で立体形状を定義する。したがって、従来のボクセル方式のようなメモリアドレスと空間座標を直接対応付けた巨大なメモリが不要である。

(2) 回転変換

ワールド空間$X_W Y_W Z_W$ 上に配置した物体の投影像を表示面$X_S Y_S$上に生成するために、表示面中央点O_S を原点とする表示空間$X_S Y_S Z_S$ を考える。Z_S 軸は視点側に向かう表示面中央点法線方向とする。各物体定義副空間上のボクセル座標値を表示空間座標値に変換した後、任意の視点に対する透視投影座標を求め、対応する画素にそのボクセルの色を与える。

各物体定義副空間ならびに表示空間の位置と向きを指定してワールド空間上に配置する。図３にi＝kの物体定義副空間と表示空間の関係の一例を示す。配置手続きは、$X_K Y_K Z_K$、$X_S Y_S Z_S$各座標の原点O_K、O_S がワールド座標原点O_Wに、またX_K, Y_K, Z_K 軸およびX_S, Y_S, Z_S 軸の３軸がそれぞれX_W, Y_W, Z_W 軸に一致している状態を基準とし、各座標を原点周りで回転して向きを決定

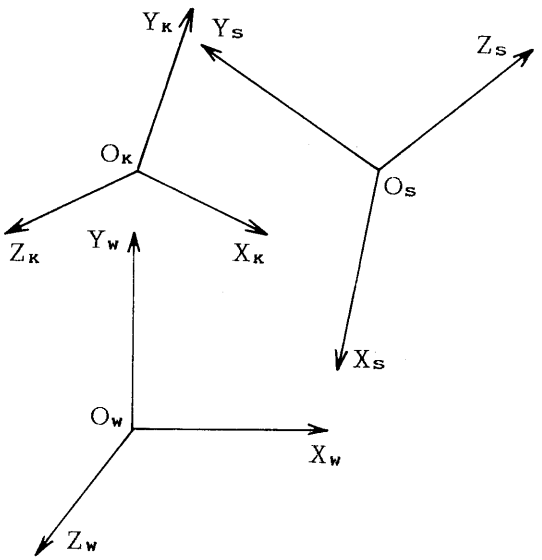
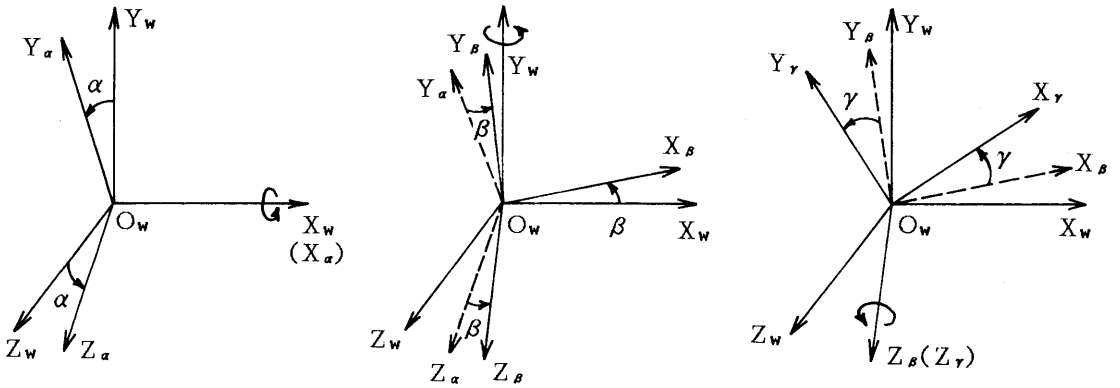


図３　i=k の物体定義副空間に対する座標変換



(a) X_W軸まわりの回転　　(b) Y_W軸まわりの回転　　(c) Z_β軸まわりの回転

図４　座標の回転

した後、目的の位置へ平行移動する。回転は各軸＋方向から原点を見て反時計回りに回転する。まず、図４(a) のようにX_w 軸の周りで角度αの回転をして$X_\alpha Y_\alpha Z_\alpha$を得る。つづいて同図(b)のように、$Y_w$軸の周りで角度$\beta$の回転をして $X_\beta Y_\beta Z_\beta$を得る。さらに同図(c)のようにZ_β 軸の周りで角度γの回転をして目的の向きの座標$X_\gamma Y_\gamma Z_\gamma$を得る。

上述の回転によって定義した物体定義副空間からワールド空間への回転変換マトリクスRは次式で与えられる。

$$R=\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \qquad (1)$$

ただし、

$$\begin{aligned} r_{11}&=\sin\alpha\sin\beta\sin\gamma+\cos\beta\cos\gamma \\ r_{12}&=\cos\alpha\sin\gamma \\ r_{13}&=\sin\alpha\cos\beta\sin\gamma-\sin\beta\cos\gamma \\ r_{21}&=\sin\alpha\sin\beta\cos\gamma-\cos\beta\sin\gamma \\ r_{22}&=\cos\alpha\cos\beta \\ r_{23}&=\sin\alpha\cos\beta\cos\gamma+\sin\beta\sin\gamma \\ r_{31}&=\cos\alpha\sin\beta \\ r_{32}&=-\sin\alpha \\ r_{33}&=\cos\alpha\cos\beta \end{aligned} \qquad (2)$$

である。また、ワールド空間から表示空間への回転変換マトリクスR'はRの転置行列になる。

$$R'=R^T \qquad (3)$$

$\alpha=\alpha_K$, $\beta=\beta_K$, $\gamma=\gamma_K$ の回転を行なったi=k の物体定義副空間に対する回転変換マトリクスをR_K 、$\alpha=\alpha_S$, $\beta=\beta_S$, $\gamma=\gamma_S$ の回転を行なった表示空間への回転変換マトリクスをR'_Sとすれば、物体定義副空間上の座標(x_K, y_K, z_K)から表示空間上の座標(x'_S, y'_S, z'_S)への変換式は次式となる。

$$(x'_S, y'_S, z'_S)=(x_K, y_K, z_K)R_K R'_S \qquad (4)$$

各ボクセルの座標変換処理に先立って、図１のa～fの６方向差分ベクトルの表示空間上の差分値表を作成する。各差分ベクトルはいずれも差分副空間座標軸方向への１ボクセル間隔の大きさであるから、式(4)の(x_K, y_K, z_K)にa～fそれぞれ(1,0,0)、(-1,0,0)、(0,1,0)、(0,-1,0)、(0,0,1)、(0,0,-1)を代入すれば表示空間上の各差分値が得られる。また、1,-1の代わりに他の値を用いることにより差分ベクトル長を可変にして、スケーリングを行なうことも可能である。

物体は各物体定義副空間の原点から始まるボクセルチェーンで定義されているので、原点の表示空間座標値に各リンク毎に差分値表から読出した表示空間差分値を順に加算して行くだけで、各ボクセルの表示空間絶対座標値が得られる。

物体定義副空間原点のワールド座標値を(x_{KO}, y_{KO}, z_{KO})、 表示空間原点のワールド座標値を(x_{SO}, y_{SO}, z_{SO})とすれば、物体定義副空間原点の表示空間座標値(x_{SK}, y_{SK}, z_{SK})は、

$$(x_{SK}, y_{SK}, z_{SK})=(x_{KO}, y_{KO}, z_{KO})R'_S-(x_{SO}, y_{SO}, z_{SO}) \qquad (5)$$

により得られる。

(3) 透視投影変換

逐次得られる各ボクセルの表示空間座標を、図５により任意の視点に対応した投影面座標値に変換する。表示空間上の点$V(x_V, y_V, z_V)$の投影面座標値$P(x_P, y_P, 0)$ は、視点を$E(x_E, y_E, z_E)$とすれば次式となる。

$$x_P=\mu(x_E-x_V)+x_E \qquad (6)$$

$$y_P=\mu(y_E-y_V)+y_E \qquad (7)$$

ただし、$\mu=z_E/(z_E-z_V)$ である。

この演算は簡単なハードウェアで実現できる。

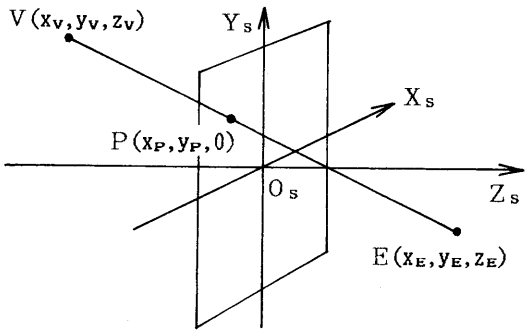


図５　透視投影変換

得られたx_P, y_P を整数化したフレームメモリアドレスにそのボクセルの色データを書き込む。

3．装置構成

(1) ハードウェアと処理方式

ハードウェアによる高速描画を容易にするためにつぎのデータを用いて３Ｄシーンを記述する。

①物体識別データ

複数物体の識別符号、描画完了符号を記述。(14)

②表示モード・データ

差分ベクトル選択符号(3)

ボクセルの色(11)

③非表示モード・データ

差分ベクトル選択符号(3)×4リンク分

実験では16ビット／語とし、()内は各情報に割当てたビット数である。

一つの物体定義副空間上の物体は、物体識別データを先頭とし、非表示と表示モードデータを適宜切替えながらチェーン記述した一連のデータ群となる。複数の物体定義副空間を用いる場合は、このデータ群を直列に並べて、末尾に描画完了符号を置く。これらのデータ群をここでは単に立体データと呼ぶ。

図６に３Ｄ描画処理ハードウェアの概略を示す。立体データは予め立体データメモリに収容しておく。ホストコンピュータでは、各物体定義副空間原点の表示空間座標値を式５によって求め、また各物体定義副空間の向きと視点に対応した図１の６種類の差分ベクトルの値を式４により求めて、それぞれ副空間原点座標テーブルおよび差分ベクトルテーブルに転送しておく。透視投影に必要な視点位置データもホストコンピュータから透視投影変換器に与える。

ホストコンピュータからの描画開始の指令により、立体データを順にメモリから読出し描画処理を行なう。物体識別データが読出されると、それを物体識別データレジスタ(ＩＲ)にロードすると共に、原点座標テーブルから対応する原点座標値

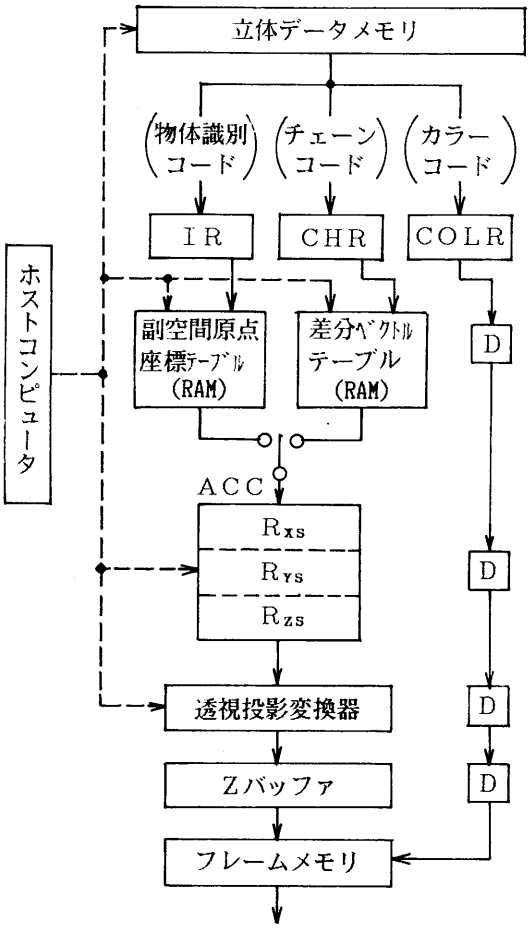


図６　ハードウェア構成

を選択して、表示空間座標アキュムレータ R_{xs}, R_{ys}, R_{zs} にロードする。表示または非表示モードデータが読出されると、差分ベクトル選択符号とＩＲのデータにより差分ベクトルをテーブルから選択して読出し、R_{xs}, R_{ys}, R_{zs} に加算して、そのボクセルの表示空間絶対座標値を求める。表示モードの場合は、求めた座標値を透視投影変換器に入力して、フレームメモリ・アドレスを求める。さらに、Ｚバッファによる奥行き値の検査の後、そのボクセルの色データをフレームメモリに書込む。ただし、非表示モードの場合はアキュムレータに表示空間座標値を求める操作だけを行なう。

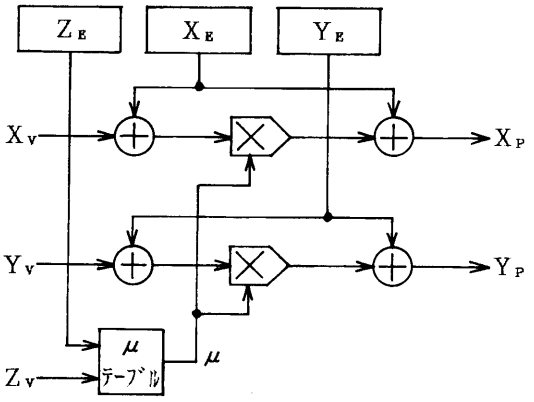


図7　透視投影変換器

透視投影変換は式6、7の演算を図7に示す構成のハードウェアで行なう。この出力x_P, y_Pは直接フレームメモリのアドレスとなり、Zバッファによる奥行き値比較の後、当該ボクセルの色データをそのアドレスに書き込む。

実験では、立体データメモリの読出し、差分ベクトルテーブルの検索、表示空間座標のアキュムレーション、透視投影変換、Zバッファ処理、フレームメモリへの書込みの計6段のパイプライン処理を行なっている。パイプライン処理による座標変換の間ボクセルの色データは図6のように遅延を与えている。この時間を利用して実時間シェーディング等の処理を行なうことを検討中である。

フレームメモリは2面分を用意し、一方の表示中に他方を書き込む方法で、描画途中の画像が表示されないようにする。

(2) 処理速度

パイプライン周期は、1パイプライン周期に2回のメモリアクセスを要するZバッファに35nSのSRAMを用いた場合を基準として、100nS に設定した。したがって、立体像の描画速度は 10Mボクセル／秒になる。

物体の空間上での移動と回転はそれぞれ副空間原点座標テーブルおよび差分ベクトル・テーブルの更新により実現できる。各テーブルのデータの導出はホストコンピュータで直接行なう。実時間で移動・回転を行なうには、これらの演算の高速化が要求される。そのため、式2に示すマトリクスRの要素はテーブルにしておく。例えばαに6ビット、βとγに各7ビットを割当てると、各軸まわりに 2.8° 刻みの回転が指定できる。 r_{11}, r_{13}, r_{21}, r_{23} は3入力変数の総ビット数が 20ビットになり、各1M語のテーブルになる。同一方向の差分ベクトルを物体定義空間の端から端まで加算しても、まるめ誤差の積算が整数部に現われないように、差分テーブルからの出力データの有効桁数を十分に設定する必要がある。

式2の演算をテーブル化した結果、1物体定義副空間原点の表示空間座標値と6方向差分ベクトルの計算と更新に要する時間は、PC9801VXを用いて約 5mSであった。原点座標や向きが前時刻の状態と変わる物体定義副空間についてのみこの処理を行なう。描画に先立ってフレームメモリの前画面を消去する必要があるが、この消去時間を利用してテーブルの更新を行なう。したがって、前時刻の状態と変更の必要がある物体定義副空間の原点座標と向きを与えてから、変更後のシーンが表示されるまでの時間は、計算時間(約5mS×変更を要する副空間数)、前画面の消去時間(約0.1mS)、描画時間 (100nS×総ボクセル数)の総和となる。変更を要する副空間数を10、総ボクセル数を100万とすれば、約0.15秒で画面が更新される。

4．検討

(1) ボクセル分解能と表示解像度

表示面解像度256×256画素のプロトタイプを試作し表示実験を行ない、前述の機能を確認した。現在、512×512画素の装置を試作中である。

本方式は3Dシーンの記述に差分表記のボクセルチェーンを用いるので、シーン定義空間の大きさについて特別な制限は設けていない。座標変換の結果表示画面外に投影されるボクセルはハードウェアでクリッピングしている。ただし、全ボクセル数によって描画速度が影響されるので、必要とする更新速度によってシーン定義範囲の制限が

必要になることがある。

本方式では３Ｄシーンの１ボクセルと表示面の１画素を１対１で対応付けている。すなわち、あるボクセルの変換後の表示面座標を画素アドレスで量子化し、該当画素にそのボクセルの色を与える。したがって、表示面に投影後の隣接ボクセル間隔が画素間隔以上になると画素欠落が起こる。例えば１次元的画素欠落は、図８(a) のように変換後の隣接するボクセルの座標点p,q がそれぞれ画素a,c に量子化され、画素d が欠落するような場合である。２次元の欠落は、同図(b) のように変換後の座標点p,q,r,s がそれぞれa,b,c,d に量子化され、画素e が欠落するような場合である。

図１の差分ベクトルが表示面に平行で、式(6)(7)のμが取り得る最大値のとき(ボクセルが視点に近いほどμが大) 表示面上の最大差分になる。また、図８(b) のような場合が最も小さい表示面の差分で画素欠落を生じる。したがって、μの取り得る最大値でこの表示面の差分を越えないようにボクセル間隔を設定すれば画素欠落が防止できる(スケーリングを行なう場合はスケーリングファクタも考慮する必要がある)。しかし、この方法では視点から遠いボクセルに対しては必要以上に細かい間隔になり、データ量の増大を招く。

本方式ではＺバッファのデータを利用した画素欠落防止法を用いた。Ｚ値に閾値を設定し、閾値より遠い場合は１ボクセルを１画素に対応付ける。閾値より視点に近い場合はつぎの方法で１ボクセルを４画素に対応付ける。まず、２進の投影面座標を小数第１位まで求め、そのXs, Ys座標値の小数第１位をxd, ydとする。投影された画素(i,j)のＺ値が閾値より視点に近い場合、xd, yd 値によって図９の規則で (i,j)の周りの４画素を塗潰す。フレームメモリはこの４画素描画を１回のアクセスで実行できる特別な構造にした。

設定した閾値によって画素欠落発生率と解像度が変化する。静止画像または静止部分に欠落が発生すると目立つが、動的部分に発生しても比較的目立たない。したがって、画像を見ながらこの閾値を適宜調整して欠落発生率と解像度の調和点を

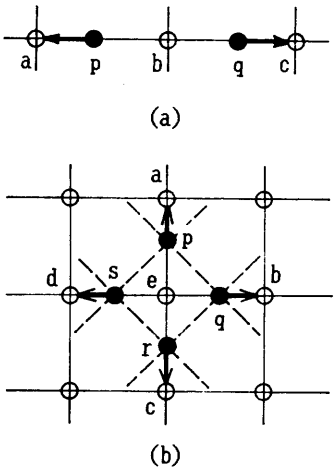


図８　画素欠落

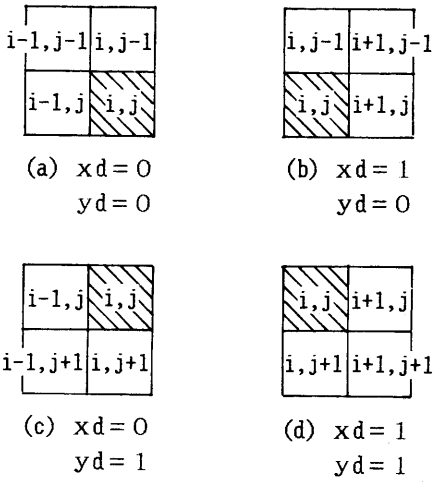


図９　画素欠落防止の一方法

捜すことができる構造にした。この方法をさらに高度化した解像度低下の少ない画素欠落防止法を開発中である。

(2) 処理の高速化

表示解像度256×256画素のプロトタイプでは、多数の任意形状物体で構成される３Ｄシーンの全体または物体毎の移動・回転・スケーリングなどがダイナミックに表示できた。表示解像度を 512×512 画素またはそれ以上にした場合、総ボクセ

ル数もそれに応じて増加するので、実時間性を維持するには一層の高速化が必要である。

描画速度の向上の一方法として、フレームメモリを含む座標変換部を複数設け、物体定義副空間をいくつかのグループに分けて並列に処理をする方法が考えられる。フレームメモリに色データと共にZ値を記憶し、その出力の優先判定をして表示する。差分テーブルと副空間原点座標テーブルの各データの演算処理も並列化すればテーブル作成時間も短縮できる。装置の簡単化のため、さらに有効な高速化法を検討中である。

5．まとめ

任意形状物体で構成される３Ｄシーンの実時間合成などの高速３Ｄグラフィクスを目的として、３Ｄシーンをボクセル・チェーンで記述し、高速に操作する方法について述べた。 試作した表示面解像度256×256画素のプロトタイプでは満足すべき実時間応答特性が得られた。

複雑な情景をさらに高解像度で表示するには、なお一層の高速化が必要であり、有効な方法について検討中である。またボクセル・チェーンデータの自動生成、実際の３Ｄシーンのデータ化法、対話的に３Ｄシーンを操作する方法などについても検討中である。

本研究は放送文化基金ならびに電気通信普及財団の助成を受けて実施した。

文　献

(1) A. Kaufman, R. Bakalash ; "Memory and Processing Architecture for 3D Voxel Based Imagery",IEEE CG&A, 8,6, pp.10-23 (Nov.1988).

(2) G.T.Herman, J.K.Udupa ; "Display of 3-D Digital Images:Computational Foundations and Medical Applications", IEEE CG&A,3, 5, pp.39-46 (Aug. 1983).

(3) S. M. Goldwasser et al;"Phisicians Workstation with Real-Time Performance",IEEE CG&A,5, 12, pp.44-57(Dec.1985).

(4) T.Ohashi, T.Uchiki, M.Tokoro ; "A Three-Dimensional Shaded Display Method for Voxel-Based Representation",Eurographics 85,North Holland, Amsterdam, Sept.1985, pp.221-232.

(5) 石井、五十嵐、大和、牧野;"視点移動に追従する立体像表示の一方法",電子情報通信学会研究報告 IE88-12 (1988年5月).

(6) 石井、五十嵐、大和、牧野;"視点移動に追従するステレオ像表示装置",第19回画像工学コンファレンス 7-2 (1988年12月).