

可変歩幅 DDA による 3 次元メモリ空間での 高速なボクセル追跡†

田山典男† 清水則明†† 漆間文俊††

本論文は、3次元状の計測データや計算結果を高速に立体描画するための光線追跡手法に関するものである。3次元デジタル物体の格納されている3次元メモリ空間に対し空白部分を階層表現するピラミッド構造の3次元メモリを導入して、歩幅を変える3次元 DDA により高速にボクセル追跡を行うハードウェア向きの立体画像生成法を提案する。3次元メモリ空間(キューブ)において光線が通過するボクセルを欠落なく生成するのは、画像をより正確に生成するためであったが、ピラミッド構造キューブにおいては、上層キューブでのわずかなボクセルの欠落が下層キューブでは大きなボクセル列の欠落誤りを起こすので、特に重要になる。本文では、ピラミッド構造のキューブに対して光線が通過するボクセルを欠落なく生成する可逆歩進型の3次元 DDA を適用して、可変歩幅でボクセル追跡するハードウェア向きのアルゴリズムを導出している。次に、ここからデジタル物体表面を滑らかに立体描画するために法線ベクトルの算出法が重要なので、物体の表面近傍に存在濃度を設定しその濃度分布の勾配から法線を算出するハードウェア向きの手法を示す。最後に、コンピュータシミュレーションにより人体などの滑らかに陰影付けされた立体画像を生成し、画像生成時間の比較例を示す。可変歩幅3次元 DDA により高速化が図れる。ピラミッド構造キューブの付設に必要なメモリ容量の増加はわずかでよいことに留意する。

1. はじめに

最近、医学や物理学、産業の分野で3次元状の測定データや計算結果等を人間に直感的に分かりやすい形に視覚化する方法として、『3次元ビジュアルライゼーション』¹⁾ 技術が注目されている。特に医用工学の分野では、X線 CT 等の多数枚の断層画像から人体内部の形状を立体描画することが求められており、3次元画像データを直接に立体描画する種々の方法が研究されている⁷⁾⁻¹⁶⁾。

3次元状の画像データは、その容量が大きいので、データ処理量が膨大であり、長時間を要する。そこで、大容量3次元データに対する3次元処理や立体表示を高速化する研究も重要である。これまでに、3次元メモリ内のデータを高速に立体表示する機構が幾つか報告されている。GODPA²⁾ では3次元空間の8等分割を繰り返し前後関係から陰面消去して画像生成する。SCOPE³⁾ ではスクリーンの各画素から物体までの距離を求めて奥行配列(Z-バッファ)を作成し、上下左右の十字型を重ねて十字型の垂直方向を法線とする。CUBE⁴⁾ ではスクリーンを固定し光線追跡を簡単にして高速な物体回転部を別に設ける。ARTS⁵⁾ では、物体を関数式や多角形で記述する光線追跡法にお

いて交点探索を削減するために等分割空間を3次元 DDA で歩進する。さらに文献6)や7)の3次元表示システムが研究されている。筆者らも、3次元画像の処理と表示を高速化する観点から同時並列処理を行う3次元並列画像処理計算機 PIPE-II のシステム開発を進めており¹⁷⁾⁻¹⁹⁾、3次元状の画像データを高速に立体描画するボクセル追跡法 SPOT を報告した²⁰⁾。このように種々の試みがなされてはいるが、3次元メモリ内の3次元物体を即座にしかもその内部を自在に視覚化する装置を実用化するには、さらに、高速な画像生成法や機構の研究が必要と思われる。

そこで本研究の目的は、ボクセル追跡法 SPOT をさらに高速化する観点から、3次元メモリ空間において物体のない所とある所で3次元 DDA の歩幅を大きくしたり小さくしたり適応して変えることで光線追跡を高速化する方法、つまり歩進速度を適応し変えることで能率よく高速に光線追跡する新しい方法を検討する。ボクセル追跡法 SPOT は、物体の形状を3次元画像データ(ボリューム)として記憶している主メモリ内の(または専用3次元 IC メモリ装置内の)3次元メモリ空間において、光線が直進する時に通過するボクセルのアドレスを3次元 DDA により生成して、そのボクセル列を読み出すことによりボリュームを透明や半透明で切り出して立体描画する手法である²⁰⁾。

本論文では、3次元メモリ空間のどの部分に物体が存在するか否かを能率よく扱えるようにするために、

† A Speedy Voxel-Tracing Method by Use of a Variable Stepping DDA in 3D Memory Space by NORIO TAYAMA, NORIAKI SHIMIZU and FUMITOSHI URUMA (Department of Computer Science, Faculty of Engineering, Iwate University).

†† 岩手大学工学部情報工学科

するか否かをあらかじめ調べておき、光線追跡にその情報を用いて物体が存在しない所では3次元 DDAの歩幅を大きくして能率よく読み飛ばし歩進させることにより、ボクセル読出し回数を削減し高速化することを考える。

2.1 ピラミッド構造キューブの生成

3次元メモリ空間で物体周辺には通常空白があると考えられる。そこで1個の3次元メモリ空間を8等分割しさらにそれらを繰り返し8等分割して、それらの各部分空間に物体が存在しているか空白かを調べる。それらの2値情報を構造的に取り扱いやすく格納するために、2値の3次元メモリをピラミッド状に階層構成する。この考え方を底層から上層へ逆に見ると、立方体状の隣接する8個のボクセルを1個のボクセルにまとめて、これを階層化することになる。この8個のボクセルをまとめて1個のボクセルに縮めることを“圧縮操作”と呼ぶ。3次元画像が格納されている最底辺の3次元メモリ空間を“基礎キューブ”と呼び、これに圧縮操作を施すことにより生成される2値3次元メモリ空間を“圧縮キューブ”と呼ぶ。このように圧縮操作を繰り返し施すことにより、ピラミッド状に階層構成された3次元メモリ空間が生成される。この基礎キューブと圧縮キューブの全体を“ピラミッド構造キューブ”（または“ピラミッド構造3次元メモリ空間”）と呼ぶ。

始めに、ピラミッド構造キューブを主メモリ上に生成する手順を示す。なお、存在濃度は3章で述べる。

《ピラミッド構造キューブの生成手順》(図2参照)

① 元の3次元画像が格納されている基礎キューブに対して圧縮操作を施し、上層に圧縮キューブ1を生成する。圧縮キューブのボクセル値は、基礎キューブの8個のボクセルの存在濃度がすべて存在閾値 t_c より小さい場合のみ‘0’とし、存在閾値 t_c より大きい場合等しいボクセルを1個以上含む場合には‘1’とする。

② 圧縮キューブ n ($n=1, 2, \dots$) に対し圧縮操作を施し、その上層に圧縮キューブ $n+1$ を生成する。上層キューブのボクセル値は、下層キューブの8個のボクセルの値がすべて‘0’の場合にのみ‘0’で、‘1’のボクセルを含む場合には‘1’とする。

③ ②の圧縮操作を、圧縮キューブのボクセル値がすべて‘1’になる一歩手前まで繰り返す。この時の圧縮キューブを“最上層キューブ N ”と呼ぶ ($n=1,$

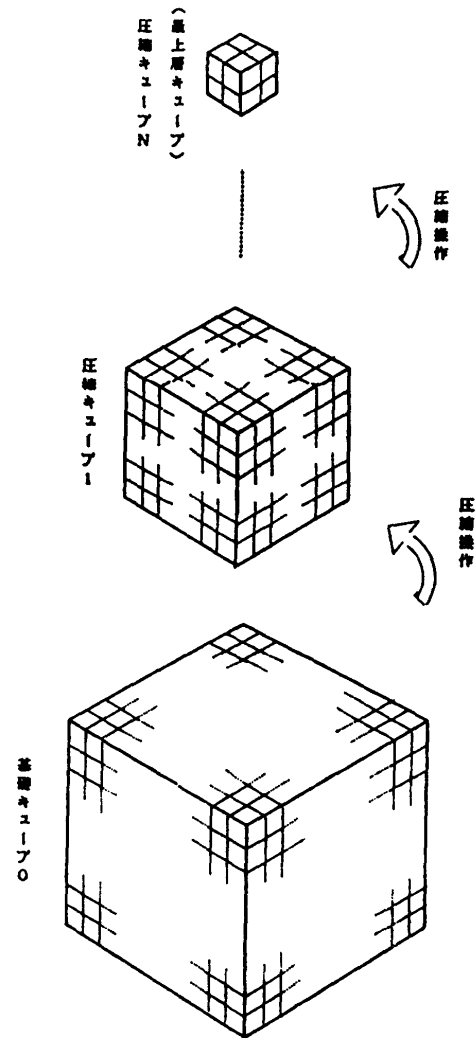


図2 ピラミッド構造キューブの生成

Fig. 2 Generation of pyramidal hierarchy of 3D memory space.

2, ..., N).

2.2 ピラミッド構造キューブでのボクセル追跡の考え方

ここでは、ピラミッド構造における各部分空間ごとの“物体が存在しない”という空白情報をピラミッド構造キューブに基づいて階層表現し、このピラミッド構造キューブに3次元 DDA を適用し歩進の仕方を工夫することで、歩幅を可変にしボクセル読出し回数を削減する。以下に、空白情報が階層表現されたピラミッド構造キューブにおけるボクセル追跡の考え方を示す。

まず、光線追跡の対象を最上層キューブにし、光線が通るボクセルの値を調べる。もしボクセル値が‘0’

ならば、そのボクセルに対応する基礎キューブの領域には物体が存在しないので、次のボクセルへ光線を進める。もし '1' ならば、その対応する領域には物体が存在し光線と交差する可能性があるので、1 段下層の圧縮キューブへ降りて該当のボクセルに対して同様の操作を繰り返す。そして最終的に最下層の基礎キューブのボクセルに対して光線が交差するかどうか判定する。途中の圧縮キューブで交差しないことが判明した場合には上層の圧縮キューブへ昇れる位置にきた時、可能なだけ上層の圧縮キューブへ昇り、該当のボクセルに対して同様の操作を繰り返す。このようなピラミッド構造キューブの中を昇り降りしながら光線を直進させ、物体と交差するか基礎キューブの外側に出るまで繰り返す。

なおピラミッド構造キューブにおいて圧縮キューブの階層を移動する時には、1 段下層へ降りる時に 3 次元 DDA の小数点位置を右へ 1 桁移し、1 段上層へ昇る時に 3 次元 DDA の小数点位置を左へ 1 桁移す。このような簡単な桁移動だけで、3 次元 DDA の整数部がピラミッド構造キューブの読み出そうとしている階層の圧縮キューブにおける正しいアドレスを与えており、即座に該当のボクセルを読み出すことができる。

具体的な例を図 3 に示す。分かりやすくするために 2 次元平面上で光線追跡する例を示す。まず、光線追跡の対象を最上層キューブにして、最初にボクセル①をアクセスする。するとボクセル値が '0' なので、次のボクセル②をアクセスする。するとボクセル値が '1' なので光線追跡の対象を下層の圧縮キューブ 1 へ移して該当のボクセル③をアクセスする。そのボクセル値が '0' なのでボクセル②の領域では物体と交差しないことが分かり、上層キューブへ昇れる位置にきたので光線追跡の対象を最上層キューブへ移して、ボクセル④をアクセスする。するとボクセル値が '1' なので光線追跡の対象を下層の圧縮キューブ 1 へ移して、ボクセル⑤からボクセル⑥へと進む。そしてボクセル値が '1' なので対象を最下層の基礎キューブへ移して、ボクセル⑦からボクセル⑧へと進む、ここで光線がついに物体と交差することを判定する。

この手法では、物体が存在していない空白ボクセルの集まりを、ピラミッド構造キューブの生成によって大きさの異なる階層ごとの空白ボクセルにまとめ、そ

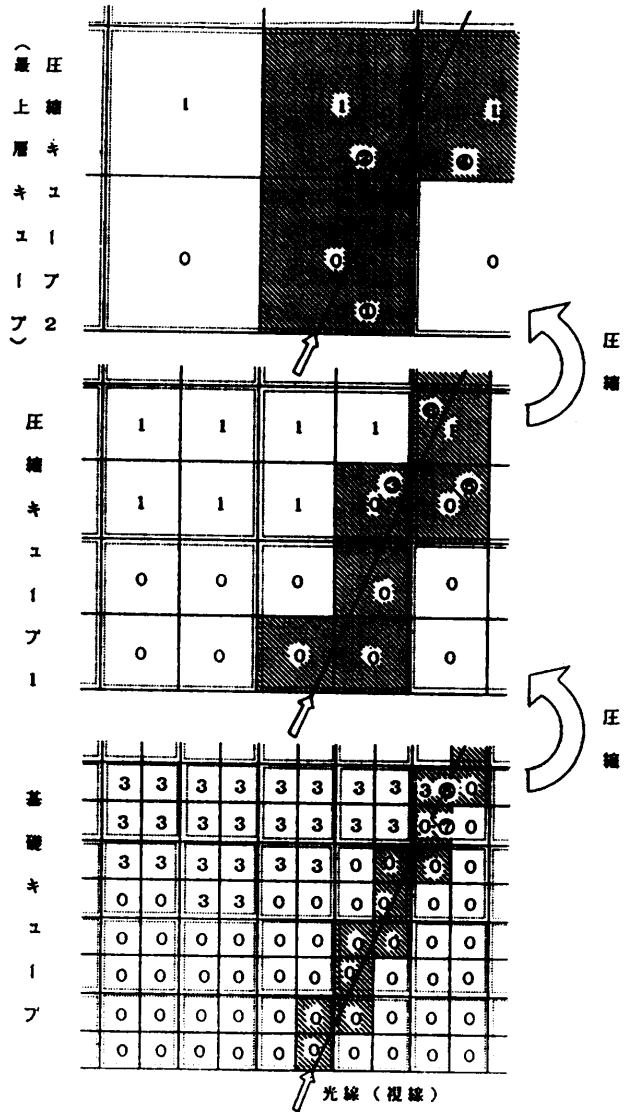


図 3 ピラミッド構造キューブにおける光線追跡
Fig. 3 Ray tracing in pyramidal hierarchy of 3D memory space.

のピラミッド構造キューブの中を 3 次元 DDA によって歩進させる。このようなピラミッド構造キューブの中を 3 次元 DDA が昇り降りしながら歩進している様子を、元の 3 次元メモリ空間に投射して眺めるならば、物体が存在しない所では 3 次元 DDA が歩幅を大きくしてボクセルを読み飛ばしながら歩進しているように見える。ボクセルの読み出し回数が相当に削減されるので、光線追跡を高速化できる。

2.3 可変歩幅のボクセル追跡アルゴリズム

以下では 2.1 節で生成したピラミッド構造キューブに対して、筆者らが提案したボクセル追跡法 SPOT²⁰⁾

により光線追跡する手順を示す。

《ピラミッド構造キューブにおけるボクセル追跡手順》

① 光線の進行方向ベクトルを (R_x, R_y, R_z) とすると、 R_x と R_y と R_z の中で値の一番大きい方向を基準方向と定める。進行方向の傾斜から、基準方向へ増分幅を1とする時の他方向の増分幅を求め、それを1歩進の各方向増分幅とする。

② スクリーン上の次に求める画素(ピクセル)を定め、その光線が基礎キューブに交わる入口点と出口点の座標を計算して、入口点座標の整数部と小数部とを3次元 DDA に設定する。透過レジスタをクリアする。もしすべてピクセルの輝度を生成した時には、終了する。

③ 最上層キューブが N 回の圧縮操作により生成された圧縮キューブ N である場合には、3次元 DDA の各方向レジスタの小数点位置を左に N 桁だけ移動して、各方向レジスタの整数部を圧縮キューブ N への読出しアドレスとして出力する。

④ 現在の光線追跡の対象になっているキューブ n ($n=0, 1, 2, \dots, N$) に対して、読出しアドレスによりボクセル値を読み出す。ボクセル値が '0' の場合には、そのボクセル領域に物体が存在しないことが分かったので⑤へ行く。ボクセル値が '1' の場合には、現在の光線追跡の対象が基礎キューブかどうか調べる。

i) 対象が基礎キューブでない場合には、そのボクセルの領域に物体が存在することが分かったので、下層の圧縮キューブへ移り調べる準備をする(3次元 DDA の小数点位置を右へ1桁移動し、その整数部を圧縮キューブ $n-1$ への読出しアドレスとして出力する)。そして④に戻る。

ii) 対象が基礎キューブである場合には、光線が物体と交差することが判明したので《物体交差処理》を行う。つまり、ボクセルの不透過率を読み出して、1未満の場合には(光線が半透明中をさらに進むので)そのボクセルでの不透過の輝度と透過光の分量とを計算し透過レジスタを修正して⑤へ行く。1の場合には透過レジスタを調べ輝度を計算してピクセルに格納し、②へ戻る。

⑤ 3次元 DDA を1歩進ませて(3次元 DDA の各方向レジスタに小数点位置を配慮し各方向増分幅を足し込み)、各整数部を次候補レジスタに格納する。非基準方向の整数部に変化がない場合には、⑦へ行く。

く。変化がある場合には、1歩進の途中で他のボクセルを通過する可能性があるので《途中通過ボクセルの判定》を行いそのアドレスを途中通過レジスタに設定更新する。(これは、文献²⁰⁾の可逆歩進動作により途中の通過ボクセルを判定するもので、その概略を以下に示す)。

i) 非基準方向の整数部が1方向だけ変化した場合には、非基準方向小数部が0でなければ、基準方向の整数部だけ歩進前の値のボクセルを途中通過したと判定しそのアドレスを途中通過レジスタへ設定更新し、⑥へ行く。非基準方向小数部が共に0ならば、途中通過しなかったと判定して、⑦へ行く。

ii) 非基準方向の整数部が2方向とも変化した場合には、図4の示すように6通りの通り方があるため、3次元 DDA を可逆歩進させ途中に通過する1個か2個のボクセル列を判定してそのアドレスを途中通過レジスタへ設定更新し、⑥へ行く。途中に通過するボクセルがなかった時には、⑦へ行く。

⑥ 途中通過レジスタのボクセル列を読み出し調べる。途中通過のボクセル値がすべて '0' の場合には、⑦へ行く。途中通過のボクセル列に '1' が存在する場合には、

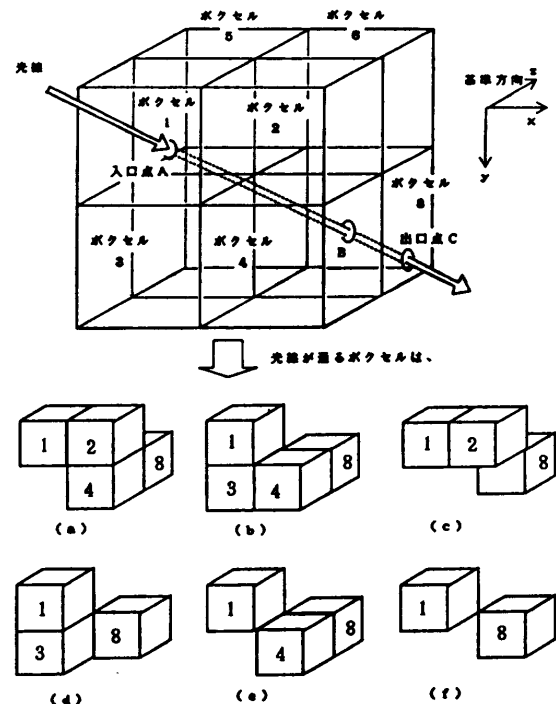


図4 光線が通過するボクセル列の例
Fig. 4 Example of voxel orders through which rays pass straight.

i) 対象が基礎キューブでない時には、光線が物体と交差する可能性があるため、3次元 DDA を1歩後戻りさせてから1段下層の圧縮キューブへ移って調べる準備をする(3次元 DDA レジスタの小数点位置を右へ1桁移し、整数部を圧縮キューブ $n-1$ への読出しアドレスとして出力する)。そして④へ戻る。

ii) 対象が基礎キューブである時には、光線が物体と交差することが判明したので、ボクセル列に対し《物体交差処理》を行い、不透過率が1未満の時に⑦へ行き1の時には輝度計算して格納し②へ戻る。

⑦ 光線が基礎キューブの外側に達した場合には、光線が不透過率1の物体と交差しないことが判明したので、透過レジスタを調べ背景と合算した輝度を計算してピクセルに格納し②へ戻る。光線が基礎キューブの外側に達していない場合には、

i) 光線追跡の対象が最上層キューブである場合または基準方向アドレスの最下位桁が‘1’である場合には、次候補レジスタのアドレスをキューブ n ($n=0, 1, 2, \dots, N$) への読出しアドレスとして出力して、④へ戻る。

ii) 光線追跡の対象が最上層キューブではなく、基準方向アドレスの最下位桁が‘0’である場合には、光線が上層の圧縮キューブへ昇れる位置にきたので、可能なだけ上層の圧縮キューブへ移って調べる準備をする(基準方向アドレスの下位 m 桁 ($m=1, 2, \dots, N-n$) が‘0’の時には、3次元 DDA レジスタの小数点位置を左へ m 桁移し、整数部を圧縮キューブ $n+m$ への読出しアドレスとして出力する)。そして④へ戻る。

3. 存在濃度の設定と法線ベクトルの算出

立体画像をリアルに陰影付けするには、物体表面の法線が重要である。そこで、基礎キューブに格納されている3次元デジタル物体の表面近傍に存在濃度を設定し、その勾配から法線を算出する方法を示す。

積み木を積み重ねたような3次元デジタル物体の表面に近似的に滑らかに変化する法線を算出するためにデジタル物体の表面近傍に存在濃度を設定して、“表面のぼやけた物体”を考える。ここで存在濃度とは、本来の物体が存在する割合を表す指標とする。つまり2値のデジタル物体に対して、その内側や外側の表面近傍で存在濃度が1から0に連続的に変化した空間的に連続に分布する表面のぼやけた物体を考える。ここで物体表面は、存在濃度がある一定の等濃度面と考

えられ、法線方向は、この等濃度面に垂直なので濃度勾配の逆向きと考えられる。存在濃度が連続的に変化する存在濃度空間では、この法線ベクトルも表面に沿って滑らかに変化的ことが予想される。そこで、表面のぼやけた物体の法線ベクトルをデジタル物体の法線ベクトルとみなす。

このような考え方から、表面のぼやけた物体に変換するための存在濃度を簡単な計算で求める方法を以下に示す。ここでは指定する物体にだけ着目して、その物体が入っているボクセルでは値を1に、入っていないボクセルでは値を0とする。注目しているボクセル (i, j, k) の値を $v(i, j, k)$ とし、注目しているボクセルの存在濃度値 $S(i, j, k)$ を次式で求める。

$$\begin{aligned} S(i, j, k) &= \{2v(i, j, k) + v(i-1, j, k) + v(i+1, j, k) \\ &\quad + v(i, j-1, k) + v(i, j+1, k) + v(i, j, k-1) \\ &\quad + v(i, j, k+1)\} / 8 \end{aligned}$$

このオペレータでは、ボクセル値を2倍するのに値を1ビット左シフトし、平均値を求めるのに3ビット右シフトすればよい。このような存在濃度の計算は、加算とシフトだけの整数演算なので高速に実行できる。

このオペレータを3次元デジタル物体に作用させると物体の十分内側では1に、十分外側では0になり、物体表面の内側近傍や外側近傍では1と0との中間の値になり、表面のぼやけた物体に変換される。このオペレータをさらに作用させると、表面のさらに滑らかなぼやけた物体になる。ただしこのオペレータでは、存在濃度値を空間的な立方格子点においてだけ求めているので、もしボクセルの内部空間における存在濃度値を欲しい場合には、ボクセルの存在濃度をボクセルの中心点での存在濃度と考えて、8個の頂点からなる立方格子点の領域内での任意点における存在濃度を3次元線形補間により求めればよい。

このような存在濃度空間において、物体表面の内側や外側の近傍における存在濃度の分布から勾配の方向成分を計算して、その逆向き方向を法線ベクトル N として算出する。

$$\begin{aligned} N_x(i, j, k) &= S(i-1, j, k) - S(i+1, j, k) \\ N_y(i, j, k) &= S(i, j-1, k) - S(i, j+1, k) \\ N_z(i, j, k) &= S(i, j, k-1) - S(i, j, k+1) \end{aligned}$$

法線ベクトルは物体表面の近傍にだけ存在するので、物体の十分内側や十分外側で計算するのは無駄であり、存在濃度値が小数値でかつ(物体を2値化し

た) 物体ビットが 1 である部分でのみ法線ベクトルの算出を行えばよい。また、物体の全表面に対する法線ベクトルをあらかじめ算出し格納しておく方法は、メモリ容量が多くなるが、種々の方向から視覚化する場合の画像生成時間を短縮できるので、3次元物体を任意に回転して即座に視察する物体内部視覚化装置の高速化に役立つ。これらの存在濃度や法線ベクトルを求める時、筆者らが検討している各種形状で並列アクセス可能な 3 次元画像メモリ¹⁰⁾を使用すると、相当に高速化が図れる。

以上のように、本文で述べた手法では、3次元デジタル物体の格納されている基礎キューブに対して、ピラミッド構造キューブを導入し 3 次元 DDA により、物体が存在しないボクセルを一まとめにして読み飛ばし歩幅を変えて歩進するので、ボクセル読み出し回数を削減でき、ボクセル追跡を高速化することができる。本手法は、光線の進み幅(歩幅)を大きくしたり小さくしたりして空白状況に適応し光線をより速く進めるという点に特徴がある。また本手法では、圧縮キューブの間を移動してボクセルをアクセスする時、3次元 DDA レジスタの小数点位置を移動するだけで対象のキューブの必要なアドレスがすぐ得られ、簡潔である。3次元 DDA の歩進は増分の整数加減算で実行され、存在濃度や法線ベクトルの算出は整数加算やシフトで実行される。このように本手法には、高速なハードウェアの実現を容易するという利点がある。歩幅の変わる 3 次元 DDA により 3 次元メモリ空間をボクセル追跡する手法を“可変歩幅ボクセル追跡法 VSPOT”(Variable Stepping SPOT)と呼ぶ。

4. 画像生成のシミュレーション

以上述べたアルゴリズムを評価するために、ソフトウェアによりシミュレーション実験を行った。ワークステーション NEWS (UNIX 4.2 BSD) 上に C 言語を用いてシミュレータを実装した。実験に使用した 3 次元メモリ空間は、 $128 \times 128 \times 128$ ボクセルの粗いものである。モデルとしては、半径 60 ボクセルと 30 ボクセルの球および人体である。人体の原データは、(48 歳女性)の上部胴体を 2 cm 幅でスライスした 16 枚の CT 画像データである。医師が臓器分けトレースをしたものを 8 種類の主な臓器の物体要素に色分けして、カラスキャナで取り込んだ。図 5 に生成画像の例を示す。(a) は 6 種類の臓器だけを色分けしカラー立体表示したもので、他を透明にしている。(b)

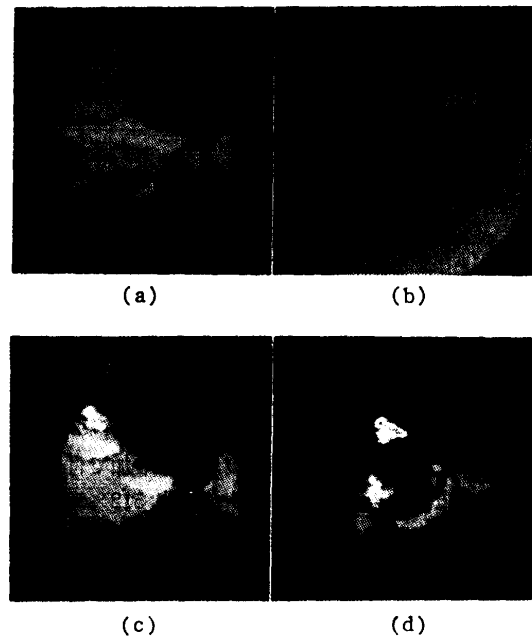


図 5 人体の立体表示シミュレーション

(a) 臓器の立体表示, (b) 胴体切出し立体表示, (c) 胃の半透明立体表示, (d) 患部の透明立体表示。

Fig. 5 Simulation results of 3D shading display of a body.

(a) 3D shading display of a viscera, (b) 3D shading display of cut of a body, (c) Semitransparent 3D shading display of a stomach, (d) Transparent 3D shading display of a diseased part.

表 1 光線追跡時間の比較例
Table 1 Comparison of ray tracing times.

手法	モデル	半径60の球	半径30の球	人体
DDA を用いない通常の方法		11分27秒	22分25秒	11分20秒
ボクセル追跡法 SPOT		3分28秒	6分27秒	5分50秒
ピラミッド構造キューブを用いるボクセル追跡法		1分22秒	41秒	3分29秒

は胴体を斜めに切り出し臓器を残して立体表示したものである。(c)と(d)では不透過率により臓器を半透明や透明にして患部を分かりやすく立体表示している。

表 1 には、立体画像の生成時間の例を示している。ピラミッド構造キューブを用いた可変歩幅の 3 次元 DDA によりボクセル追跡法の高速化が実現されていることが分かる。特に、対象物体が小さい場合に大幅

に時間が短縮されている。これは空白部分が大い時に、ピラミッド構造キューブにより基礎キューブの多数のボクセルを読み飛ばして歩進するためと考えられる。

このピラミッド構造3次元メモリを付設するためには余分にメモリが必要になる。その増加量は、キューブのデータ幅を同じにすると、ピラミッド状なので1段昇ると1/8に縮小してしまうために、物体を格納する基礎キューブ容量の14.3%でよい。このピラミッド構造キューブは2値情報を格納できればよいので、2値ピラミッド構造キューブの場合には、基礎キューブのデータ幅を1バイトにすれば、1.8%、2バイトにすればわずか0.9%程度のメモリ増加で付設できる。例えば、基礎キューブのサイズが $512 \times 512 \times 512$ で物体のデータ幅を2バイトにすると、基礎キューブの容量256MBに対して、ピラミッド構造キューブは容量が2.3MBで付設できる。

5. おわりに

本文では、3次元メモリに格納されている3次元状の計測データや計算結果を高速に立体描画する一手法として、“可変歩幅ボクセル追跡法 VSPOT”を提案した。ピラミッド構造の3次元メモリを導入付設して、空白状況に適応し可変歩幅で能率よく歩進する3次元 DDA により欠落のない正確な光線追跡を行う。簡単な計算で求める存在濃度から法線ベクトルを算出して立体画像を生成する。

この手法の特色を以下に列記する。

- 1) ピラミッド構造3次元メモリ空間に3次元 DDA を適用して、物体の存在しないボクセルを一まとめにし歩幅を変えて歩進するので、ボクセル読み出し回数を削減できボクセル追跡法を高速化できる。空白状況に適応して光線が速くなるという点に特徴がある。
- 2) ピラミッド構造キューブ間を移動する時、3次元 DDA レジスタの小数点位置を移動するだけで対象のキューブの必要なボクセルアドレスがすぐに得られ、簡潔である。3次元 DDA の増分加算や判定の機構、存在濃度や法線ベクトル算出でのシフトや整数加算の利用など簡潔で高速であり、ハードウェア向きの手法という点に特色がある。この手法は、大容量の3次元メモリを扱う高速ボリュームレンダリング専用ハードウェア装置の実現に一つの可能性を高める。
- 3) ピラミッド構造キューブを付設するために必要

なメモリ容量は、基礎キューブのデータ幅を2バイトにすると0.9%であり、非常に少ないことに留意する。このピラミッド構造3次元メモリ概念や可変歩幅で歩進する3次元 DDA の概念は、よく利用されているオクトトリーの概念に対応類似する面があり、必要なメモリ容量が少ないので、今後の活用が期待される。

今後、この VSPOT に基づく高速化アーキテクチャを検討する。3次元物体の内部を自在に実時間で観察できる専用装置が実現するならば、医療やその他の分野で3次元ビジュアルイゼーションツールとして有効なものになると思われる。

謝辞 日頃御支援をいただく本学の太田原功教授、御検討いただいた千葉則茂助教授に深謝します。

参考文献

- 1) 特集: Scientific Visualization, *IEEE Comput.*, Vol. 22, No. 8, pp. 10-101 (1989).
- 2) Goldwasser, S.M.: A Generalized Object Display Processor Architecture, *IEEE CG & A*, Vol. 4, No. 10, pp. 43-55 (1984).
- 3) 内木, 所: 3次元メモリを用いた立体図形表示機構—SCOP—, 信学論, Vol. J86-D, No. 4, pp. 741-748 (1985).
- 4) Kaufman, A.: Memory Organization for a Cubic Frame Buffer, *Proc. EUROGRAPHICS '86*, pp. 93-100 (1986).
- 5) Fujimoto, A., Tanaka, T. and Iwata, K.: ARTS: Accelerated Ray-Tracing System, *IEEE CG & A*, Vol. 6, No. 4, pp. 16-26 (1986).
- 6) 高橋, 坂本, 青木: 雪中レーダシステムのための3次元データ表示手法, 信学論, Vol. J71-D, No. 10, pp. 2002-2009 (1988).
- 7) 英保, 関口: 3次元画像メモリを用いた実体断面・3次元表示システム, 信学論, Vol. J72-D-II, No. 4, pp. 577-585 (1989).
- 8) Herman, G.T. and Liu, K.: Three-dimensional Display of Human Organs from Computed Tomograms, *Comput. Gr. Image Process.*, Vol. 9, No. 1, pp. 1-21 (1979).
- 9) 鳥脇: 最近の医用画像三次元表示の基本手法, *ME 誌*, Vol. 24, No. 5, pp. 293-303 (1986).
- 10) 伊東, 井上, 山本: 生体組織の濃淡画面による3次元表示システム, 信学論, Vol. J66-D, No. 9, pp. 1031-1038 (1983).
- 11) 仁木, 樋口, 高橋: 頭部 CT 画像からの脳形状の高精度構成とその濃淡3次元表示, 信学論, Vol. J69-D, No. 10, pp. 1518-1527 (1986).
- 12) 安田, 橋本, 横井, 鳥脇: CT 画像を用いた形成外科手術計画支援システム, 信学論, Vol. J70-D, No. 11, pp. 2134-2140 (1987).

- 13) Levoy, M.: Display of Surfaces from Volume Data, *IEEE CG & A*, Vol. 8, No. 3, pp. 29-37 (1988).
- 14) Drebin, R. A., Carpenter, L. and Hanrahan, P.: Volume Rendering, *Comput. Gr.*, Vol. 22, No. 4, pp. 65-74 (1988).
- 15) 周藤, 黄, 古旗, 内野, 河野: 改良型ボクセル法による脳腫瘍の 3 次元表示, *MIT*, Vol. 6, No. 3, pp. 280-291 (1988).
- 16) Fuchs, H., Levoy, M. and Pizer, S. M.: Interactive Visualization of 3D Medical Data, *IEEE Comput.*, Vol. 22, No. 8, pp. 46-51 (1989).
- 17) 田山: 3 次元並列画像処理計算機の開発研究, 文部省科研費 (試験研究) 報告書 (1984. 3).
- 18) 田山, 孫, 太田原: 並列アクセスの可能な 3 次元画像メモリシステムの構成理論, *信学論*, Vol. J71-D, No. 6, pp. 984-993 (1988).
- 19) 田山, 田中: 3 次元物体内部構造の記述と切出し立体表示, *信学論*, Vol. J70-D, No. 5, pp. 1043-1046 (1987).
- 20) 田山, 清水, 千葉, 太田原: 切出し立体画像を高速に生成するボクセル追跡法, *信学論*, Vol. J72-D-II, No. 9, pp. 1332-1340 (1989).
- 21) 田山, 漆間, 清水, 千葉: 可変 DDA による 3 次元メモリ空間での光線追跡, *情報処理学会グラフィクスと CAD シンポジウム*, pp. 127-136 (1989).

(平成 2 年 10 月 11 日受付)

(平成 2 年 12 月 18 日採録)



田山 典男 (正会員)

昭和 41 年岩手大学電気卒業。昭和 43 年東北大学大学院電子修士課程修了。同年日本信号(株)入社。昭和 47 年岩手大学電子講師。昭和 53 年同大情報助教授。列車群実時間制御システムのソフト開発, 対称 3 値論理系の提案。現在, 3 次元画像並列処理コンピュータのシステム開発, 3 次元画像再構成と 3 次元画像処理, 立体表示の研究に従事。電子情報通信学会, 計測自動制御学会, ME 学会, IEEE 各会員。工学博士。



清水 則明

昭和 61 年岩手大学電気卒業。昭和 63 年同大大学院情報修士課程修了。同年(株)東芝に入社(画像情報機器部)。在学中に 3 次元デジタル物体の立体表示について研究。



漆間 文俊

昭和 63 年岩手大学情報卒業。平成 2 年同大大学院情報修士課程修了。同年 IBM に入社。在学中に 3 次元データを立体描画する手法について研究。