

## ボクセル・データと面データを画像化する ボリューム・レンダリング・アルゴリズム

宮澤 達夫 杉本 和敏

日本アイ・ビー・エム株式会社 東京基礎研究所

ボクセル・データと面データを統一された枠組みで処理するレンダリング・アルゴリズムを提案する。本手法により、物体等の3次元幾何モデルを含んだ3次元空間のスクリーン場の可視化において、情報量の多い3次元ボリューム・データの表示とともに、高解像度の3次元幾何モデルの表示が実現できる。本手法では、「レイ・キャスト法」を用いて、ボクセル・データと面データを統一的に処理する。物理量を表すボクセル・データとして3次元直交格子点上で定義されるスカラー・データ、物体の形状を近似する面データとしてポリゴンを用いるものとする。本手法は、ボリューム・レンダリングとサーフェス・レンダリングの長所を統合した手法として位置づけることができる。

本論文では、3次元ボリューム・データの可視化技術、本手法のアルゴリズム、本手法の適用例について述べる。

### A Rendering Algorithm for Visualizing Voxel Data and Surface Data

Tatsuo Miyazawa, Kazutoshi Sugimoto

IBM Research, Tokyo Research Laboratory

5-19, Sanbancho, Chiyoda-ku, Tokyo 102, Japan

This paper presents a new rendering algorithm for visualizing volumes comprised of voxel and surface data by means of processing both the data in a unified way. The algorithm is very versatile and effective in interpreting 3-D volumetric data including 3-D geometric objects. In the algorithm, ray casting method is applied as a global illumination model, and voxel data defined in 3-D orthogonal grids and surface data composed of polygons are processed. The algorithm continuously covers the gap between surface and volume rendering techniques.

This paper describes the general techniques for visualizing 3-D volumetric data, the proposed rendering algorithm, and its applications.

## 1. はじめに

近年、医療、科学・工学などの様々な分野において、3次元ボリューム・データが容易に生成されるようになり、これらのデータをいかに効果的に視覚化するか重要な課題となっている。医療、非破壊検査 (NDE) などの分野で幅広く使用されているCT、MRI スキャナからは、多数の断層画像よりボリューム・データが簡単に構成できる。また、差分法 (FDM)、有限要素法 (FEM) による構造解析、流体解析などのコンピュータ・シミュレーションの分野では、コンピュータ・パワーの向上、数値解析技術の進歩に伴い、解析の対象が2次元問題から3次元問題に進展し、ますます膨大なデータが生成されるようになってきた。

このような3次元ボリューム・データの可視化技術として、サーフェス・レンダリングによる方法とボリューム・レンダリングによる方法がある。サーフェス・レンダリングは、従来からの表示技術で、3次元物体の表面をポリゴンなどの幾何プリミティブで多面体近似してその表面の可視面のみを表示する技術である。この場合、3次元ボリューム・データから表示すべき表面を面データとして抽出する必要がある。一方、ボリューム・レンダリングは、ボリューム・データから面データを抽出することなく、ボクセルから構成されるボリュームを直接表示 (スクリーン上へ投影) する技術である。サーフェス・レンダリングは3次元幾何学モデルの表面を可視化することを目的とするのに対し、ボリューム・レンダリングはボリューム表面だけではなくボリューム内部の3次元構造の可視化、あるいは連続的な分布を連続的に表示することを目的としている。したがって、3次元ボリューム・データの可視化においては、ボリューム・レンダリングによる方法の方がより多くの情報が得られる。また、表示すべき部分の面がはっきりしない場合にも、ボリューム・レンダリングによる方法は適している。

コンピュータ・シミュレーション、実験・計測等から生成される3次元ボリューム・データの多くは、そのボリューム中に物体等の3次元モデルを含んでいる。たとえば、ある物体の回りの熱流体解析・計測の結果を表示する際、より人間の理解を容易にするために、その物体の回りの物理量 (温度、密度、濃度等) の分布と同時に物体自身の形状、あるいは物体上での分布を表示する必要がある。予め

形状が決まった物体をボリューム・レンダリングで表示するには、物体の解像度と計算時間の観点から、問題がある。しかし、このような物体の形状はCAD等からの幾何学データをもとにサーフェス・レンダリングすることにより、簡単に表示することができる。

本論文では、上記の問題点を解決する表示法を提案する。本手法は、大域照明モデルとして「レイ・キャスティング法」を採用し、ボリューム・レンダリングのアルゴリズムの中にサーフェス・レンダリングのアルゴリズムを導入して、ボクセル・データと面データを統一された枠組みで処理できるよう、そのアルゴリズムを拡張したものである。本論文では、物理量を表わす3次元ボリューム・データとして不等間隔直交格子系 (等間隔直交格子系を含む) で定義されるスカラー・データ、物体の形状を近似する面データとしてポリゴン・データを扱うものとする。

## 2. 3次元ボリューム・データの可視化技術

本章では、3次元ボリューム・データの可視化技術として、ポリゴンなどの幾何プリミティブを処理するサーフェス・レンダリングによる方法、ボクセルから構成されるボリュームを直接処理するボリューム・レンダリングによる方法を取り上げ考察する。

### 2. 1. サーフェス・レンダリングによる方法

サーフェス・レンダリングは、3次元物体の表面をポリゴンなどの幾何プリミティブで多面体近似してできた連続的な3次元幾何学モデルを用い、その表面の可視面のみをシェーディングすることによって、より3次元的な現実感のある画像を生成するものである。

この方法を3次元ボリューム・データの可視化に適用したのとして次の2つ表示法がある。

#### 1) クロスセクション・レンダリング

(切断面表示法)

$xy$ 面、 $yz$ 面、 $zx$ 面に平行な任意の断面でボリュームを切断し、その切断面のみを表示する方法である。表示に際しては、切断によって切り出された直方体の各面に切断面上のデータをカラー・マッピングし、その直方体を3次元的に表示する。

この方法の欠点は、切断面だけしか表示されず、内部情報が画像に反映できないことである。そのため、人間が切断する位置を対話的に変更しながら、ボリューム内部の3次元構造を自分の頭の中で再構築しなければならない。

## 2) スレッシュホールド・レンダリング

### (閾値面表示法)

閾値(スレッシュホールド)を設定して、3次元ボリューム・データの内、閾値と同じ値を持つボクセルを探し、それらを接続してできる形状を面で再構成(近似)し、その面(閾値面)をシェーディングして表示する方法である。表示のための面を生成する方法として、各断面層から閾値にしたがって輪郭線を抽出し、隣接した断面層間の輪郭線をつないでパッチを生成する方法<sup>1)</sup>と、8個のボクセルから形成されるセル毎にポリゴンを生成する方法<sup>2)</sup>とがある。

この方法では、複雑なデータの場合、閾値面を構成する面データ(幾何プリミティブ)の量が膨大になったり、面データの生成が困難になったりする。しかも、計測データにエラーがある場合やそのデータの量が少ない場合、面データが求まらないこともある。また、与えられた3次元ボリューム・データをわずかな閾値面データに落としてしまっている。

以上のことから、サーフェス・レンダリングによる方法は、予め形状の定義された3次元幾何学モデルの表面の表示には適しているが、3次元ボリューム・データの可視化には必ずしも適しているとは言えない。

## 2.2. ボリューム・レンダリングによる方法

この方法は、幾何プリミティブで多面体近似することなく、3次元ボリュームのボクセル・データをその色と不透明度(寄与率)を考慮して直接表示(スクリーン上へ投影)することにより、表面だけでなく内部の様子も観察できるようにしたものである。各ボクセル・データをスクリーンへ投影する方法として、レイ・キャスティング法<sup>3)</sup>(イメージ順アプローチ)と多画面合成法<sup>4)</sup>(オブジェクト順アプローチ)などがある。

レイ・キャスティング法では、視点とスクリーン上のピクセルを通るレイがボリューム中をサンプリングしながらトレースし、レイ上のサンプリング点

でのシェーディングされたボクセル・データを、不透明度を考慮して手前から順に加算(積分)しているとき、加算した結果をそのスクリーン上のピクセルの色として表示する。また、サンプリング点でのデータ値をその周囲8個の格子点でのデータ値から補間することが多い。この方法は、メモリーに対してランダム・アクセスであり、計算コストも高いが、幾何プリミティブとの統合が容易である。

多画面合成法は、シェーディングされたボクセル・データを座標変換し、視点座標系に合うように再サンプリングして出来た視線に垂直な画面を視点から遠い順に不透明度を考慮して合成( $\alpha$ チャンネル・アルゴリズム)して画像を生成する方法である。この方法は、ハードウェア化しやすいが、再サンプリングに時間を必要とし、また幾何プリミティブとの統合が困難である。

これら方法は、ボクセル・データのシェーディングの方法、不透明度の選択によりボリューム全体の表示を制御できるため、ボリューム内部の3次元構造の可視化に適している。また、サーフェス・モデルなどの3次元モデルに比べて、データ構造が簡単で、CT画像などから3次元ボリューム・データを容易に構成できる。しかし、この方法の欠点は、非連続的な3次元データを扱うため、格子間のデータに補間法を用いても画像が平滑化がされるだけで、その格子間隔以下の解像度は望めないことである。

## 3. レンダリング・アルゴリズム概要

ボリューム・レンダリングは3次元ボリューム・データの表面及び内部の可視化に適した技法であるが、ボリューム内に物体が存在する場合、その物体形状をもボリューム・レンダリングにより表示することは、物体の解像度と計算時間の観点から見ると、適切な手段とは言えない。したがって、物体の形状を効率的に表示するためには、物体形状を近似する3次元幾何学モデルを定義し、それをレンダリングした方が以上の観点から望ましい。そこで、本論文では、大域照明モデルとして「レイ・キャスティング法」を採用し、ボクセル・データと面データを統一された枠組みで処理する表示方法を提案する。この方法により、アルゴリズムの簡単な拡張で、ボリューム・レンダリングのアルゴリズムの中にサーフェス・レンダリングのアルゴリズムを自然に導入することが可能となった。次にレンダリン

グ・アルゴリズムの概要を示す。

前処理として、ボクセルーポリゴン・リストを作成する。その後、視点からスクリーン上のピクセルを通るレイを発生し、そのレイと交差するボクセルを順に求めていく。その際、レイが交差するボクセル毎に、そのボクセルのポリゴン・リストを調べ、リストが空でない場合は、レイとポリゴンの交差判定からそのボクセル内でレイと交差するポリゴンを選び出す。ボリューム中のボクセル・データは、レイに沿って手前から順に等間隔にサンプリングし、シェーディングした結果を不透明度を考慮して、スクリーン上のピクセルの色に加算していく。ただし、レイがポリゴンで囲まれた物体の内部を通過中はボクセル・データのサンプリングは行わない。また、レイがポリゴンと交差した場合、シェーディングした後、ボクセル・データと同様に不透明度を考慮して、ピクセルの色に加算する。

#### 4. レンダリング・アルゴリズム

本論文で使用したデータ構造の概念図を図1に示す。その他、ボクセル・データ、3次元不等間隔直交格子の座標データは配列データとして保持する。次に、視点から出発したレイが3次元不等間隔直交格子中のボクセル・データとポリゴン・データをレンダリングしていくアルゴリズムを図2に示す。この処理は、各レイに対して、レイが3次元格子の外部に出るか、不透明度が1に達するか、あるいは、レイが不透明なポリゴンと交差するまで繰り返される。

##### 4. 1. ボクセルーポリゴン・リストの作成

前処理として、各ボクセル毎にそのボクセル内に存在するポリゴンのリストを作成する。(図1参照)各ポリゴンの頂点の座標よりバウンディング・ボックスを考え、このバウンディング・ボックスと3次元不等間隔直交格子とのMINMAXテストにより、ポリゴンの存在するボクセルを決定する。このリストを利用することにより、レイとポリゴンの交差判定を局所化し、計算の高速化を計る。

##### 4. 2. レイが最初に交差するボクセルの決定

視点から出発したレイ (0+Dt) が最初に交差する

ポリゴン・データ

面ID | ↑面属性ID | 頂点の数 | ↑頂点ID<sub>1</sub> | ..... | ↑頂点ID<sub>n</sub>

頂点データ

頂点ID | ↑座標ID | ↑法線ベクトルID

座標データ

座標ID | X座標 | Y座標 | Z座標

法線ベクトル・データ

法線ベクトルID | 法線ベクトルX成分 | 法線ベクトルY成分 | 法線ベクトルZ成分

面属性データ

面属性ID | 面タイプ | ↑色値ID | ↑光学属性ID

色属性データ

色値ID | 赤値 | 緑値 | 青値

光学属性データ

光学属性ID | 周囲光係数 | 拡散反射係数 | 鏡面反射係数 | 鏡面反射色係数 | 透過率

ボクセルーポリゴン・リスト・データ

ボクセルID | ↑リストID | リストID | ↑面ID | ↑次のリストID

レイーポリゴン交差判定履歴データ

レイID | ↑交差ID

交差ID | ↑面ID | 交点でのレイパラメータ | 法線ベクトルX成分 | 法線ベクトルY成分 | 法線ベクトルZ成分

IN/OUT判定 | ↑次の交差ID

- 注1) nil : 該当するボクセルに存在しうるボクセルなし
- 注2) nil : 交差なし  
t<sub>s</sub> : 交点でのレイ(0+Dt)のtパラメータ(t<sub>s</sub>)
- 注3) 0 : レイがポリゴンの裏側で交差  
1 : レイがポリゴンの表側で交差

図1 データ構造の概念図

ボクセルは、適当なクリッピング・アルゴリズムとバイナリー検索によって求める。

##### 4. 3. レイとボクセル内に存在するポリゴンとの交差判定

検査中のボクセルのボクセルーポリゴン・リストが空でない場合は、リスト中のすべてのポリゴンについてレイとポリゴンを含む平面との交点におけるtパラメータ (t<sub>s</sub>) を計算しその交点の内外テストを行い、レイとポリゴンの交差判定をする。各ポリゴンの交点におけるtパラメータ (t<sub>s</sub>) から、ボクセル内 (t<sub>0</sub>~t<sub>1</sub>) でレイと交差するポリゴンの中で、視点に最も近い位置でレイと交差するポリゴン、すなわちt<sub>s</sub>が最小であるポリゴンを決定する。(図3、図4参照)

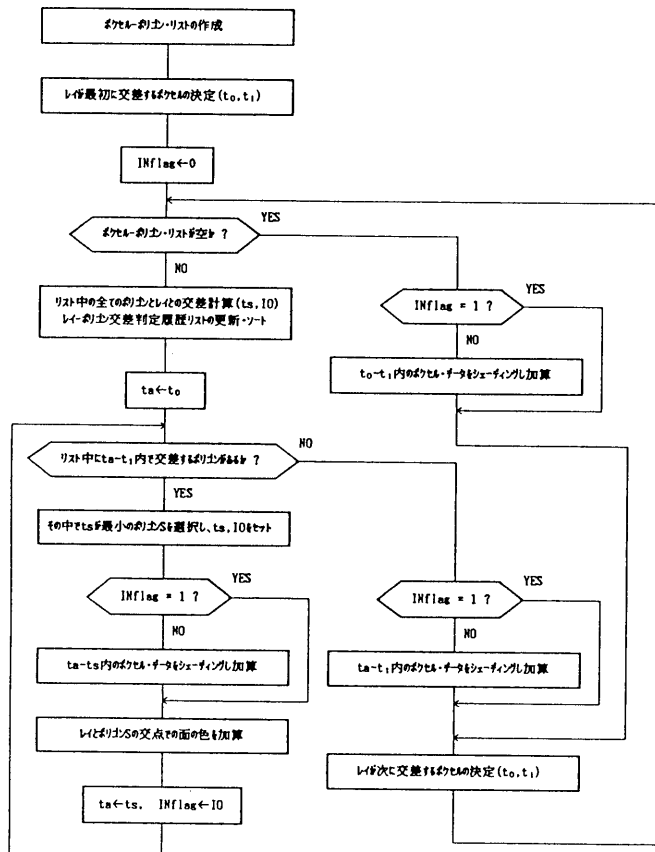


図2 処理の流れ

ここで、レイとポリゴンの交差判定を局所化するために利用している3次元直交格子は、本来ボクセル・データを定義するものであるため、ポリゴンのサイズと格子間隔を最適に制御できない。したがって、ポリゴンが多くのボクセルにまたがることも予想され、その場合には一つのポリゴンに対して何回も交差判定を行うことになる。この重複を避けるため、一度行った交差判定の結果はレイ・ポリゴン交差判定履歴データ(図1参照)として、交点におけるパラメータ( $t_s$ )の小さい順にソートして記憶しておく。このソート操作を行うことにより、ボクセル内でレイと交差するポリゴンが効率的に検索できるようになる。また、交差判定の際、レイの方向ベクトルDとポリゴンの法線ベクトルNsから、レイがポリゴンの表面から交差するか( $I0=1$ )、裏面から交差するか( $I0=0$ )の判定も行い記憶しておく。このデータは、レイが物体の内部を通過中か( $INflag=1$ )の判定に使う。

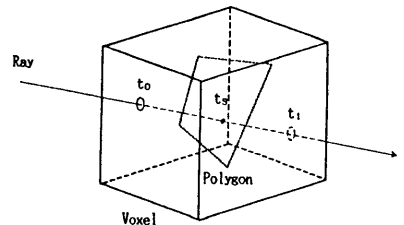


図3 レイとボクセル内のポリゴンの交差

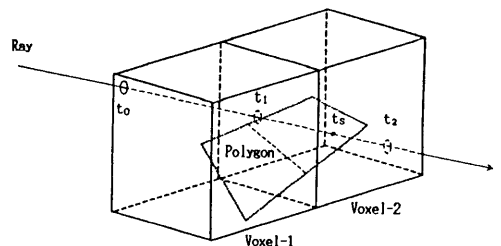


図4 レイと複数のボクセルにまたがるポリゴンの交差

#### 4. 4. ボクセル・データのシェーディングと加算

レイに沿って手前からボリューム中のボクセル・データを等間隔にサンプリングしてしるが、レイがポリゴンで近似された物体の内部を通過中( $INflag=1$ )は、ボクセル・データのサンプリングはしない。したがって、レイが通過中のボクセルを考えると、ボクセル・データのサンプリング範囲は、図5から明らかのように次の4通りとなる。

- 1) ボクセルの入口( $t_0$ )から出口( $t_1$ )まで
- 2) ボクセルの入口( $t_0$ )からレイとポリゴンの表面交点( $t_s$ )まで
- 3) レイとポリゴンの裏面交点( $t_{s0}$ )からポリゴンの表面交点( $t_{s1}$ )まで
- 4) レイとポリゴンの裏面交点( $t_s$ )からボクセルの出口( $t_1$ )まで

なお、サンプリング点( $t=t_1$ )でのボクセル・データの値 $f_i$ 、勾配ベクトル $N_i$ は、ラグランジェ補間関数<sup>5)</sup>を用いて、そのボクセルの周囲8個の格子点の座標と格子点における値、勾配ベクトルから直接求めることができる。

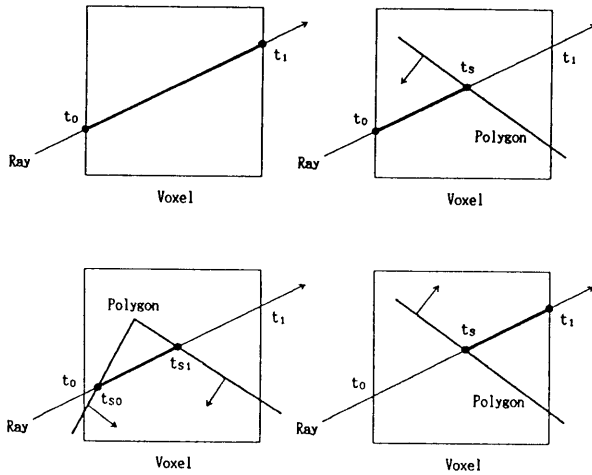


図5 ボクセル内のレイとポリゴンの関係

サンプリング点でのボクセル・データの値 $f_i$ 、勾配ベクトル $N_i$ が求まると、色・不透明度ルックアップ・テーブルを参照することにより対応する色 $c_i$ と不透明度 $\alpha_i$ を決定する。

さらに、カラー・マッピングされたボクセル・データ $c_i$ に対して、(1)式によりシェーディングし、サンプリング点での色 $C_i$ を求める。

$$C_i = c_i K_e \quad (1)$$

$$+ \frac{c_i}{k_1 + k_2 t_i} \left\{ \sum_{j=1}^m K_d (N_i \cdot L_j) I_j + \sum_{j=1}^m K_s (R_{ij} \cdot V)^n I_j \right\}$$

ここで、 $K_e$ 、 $K_d$ 、 $K_s$ は発光、拡散反射、鏡面反射の係数、 $k_1$ 、 $k_2$ はデプス・キューイングの係数、 $L_j$ 、 $R_{ij}$ 、 $V$ は光源方向、反射光線、視線方向の単位ベクトル、 $I_j$ は光源強度を表す。

最後に、サンプリング点でのシェーディングされた色 $C_i$ と不透明度 $\alpha_i$ より、(2)式によってピクセルの色 $C_{pix \times o}$ に加算する。

$$C_{pix \times o} = \sum_{i=0}^n C_i \alpha_i \prod_{j=0}^{i-1} (1 - \alpha_j) \quad (2)$$

これは、簡単な半透明表示の(3)式に基づく。

$$C_{out} = C_i \alpha_i + C_{in} (1 - \alpha_i) \quad (3)$$

#### 4.5. ポリゴンの交点での色の加算

面の色 $c_s$ を基にレイとポリゴンの交点( $t_s$ )でのシェーディングされた色 $C_s$ を(4)式から求める。なお、交点での法線ベクトル $N_s$ は、ポリゴンの各頂点での法線ベクトルから補間する。

$$C_s = c_s K_a \quad (4)$$

$$+ \frac{c_s}{k_1 + k_2 t_s} \left\{ \sum_{j=1}^m K_d (N_s \cdot L_j) I_j + \sum_{j=1}^m K_s (R_{sj} \cdot V)^n I_j \right\}$$

最後に、交点でのシェーディングされた面の色 $C_s$ を、シェーディングされたボクセル・データ $C_i$ と同様に(2)式によって、ピクセルの色に加算する。なお、面が不透明な場合は、不透明度 $\alpha_s = 1$ とするが、面が透明、半透明な場合には、(5)式より求まる透過率 $Kt$ <sup>6)</sup>より不透明度 $\alpha_s = 1 - Kt$ とする。

$$Kt = Kt_0 - (Kt_0 - Kt_1) (1 - (N_s \cdot V))^p \quad (5)$$

ここで、 $Kt_0$ 、 $Kt_1$ は面の法線ベクトル $N_s$ と視線方向ベクトル $V$ との角度が $0^\circ$ 、 $90^\circ$ の時の透過率、 $p$ は透明度強調パラメータである。

また、交点での面の色 $c_s$ として、交点でのボクセル・データに対応する色を用いることにより、面上でのボクセル・データの分布をカラー・マッピングすることができる。

#### 4.6. レイが次に交差するボクセルの決定

3次元直交格子中のレイの追跡は3DDDAアルゴリズム<sup>7)8)</sup>に基づくが、不等間隔格子を対象とするため、レイの $t$ パラメータの計算については単純な増分計算とはならず、各格子点と視点の座標値とレイの方向ベクトルより計算する。また、各ボクセルに対してレイの入口と出口における $t$ パラメータ( $t_0$ 、 $t_1$ )、すなわちそのボクセル中のレイの範囲を計算しておく。この範囲は、ボクセル・データの加算範囲、レイとポリゴンとの交差位置のチェックに利用する。

## 5. 表示例

本手法を用いた表示例を示す。作業用テーブル、バランスーを含むクリーン・ブース内の塵埃濃度分布を示したものである。塵埃濃度を $45 \times 37 \times 37$ の3次元不等間隔直交格子で定義されるボクセル・データ、また、作業用テーブル、バランスー、天井の格子および蛍光灯を約100個のポリゴンで近似した面データとして処理している。

図6、図7、図8は、3次元的な塵埃濃度分布を等濃度面表示し、作業用テーブルなどの物体を近似するポリゴンの面属性を変えて表示したものである。図6は、すべてのポリゴンを不透明な面として表示したものである。図7は、すべてのポリゴンを半透明な面として表示したものである。図8は、作業用テーブル上面での塵埃濃度分布をカラー・マッピングして表示したものである。空間の分布と同時にテーブル上での分布が容易に把握できるのが分かる。

なお、実際に表示する際には、IBM5080による立体画像を用いて、奥行き情報をも再現している。

## 6. おわりに

本手法により、簡単なアルゴリズムの拡張でボクセル・データと面データを統一された枠組みで処理することが可能となった。本手法は、種々の分野から生成される3次元ボリューム・データの可視化において、ボリューム中に複雑な物体が存在する場合でも、情報量の多い3次元ボリューム・データの表示とともに、高解像度の物体形状の表示を実現する。生成される画像は、3次元空間のスカラー場の把握において、人間の理解を支援する有効な手段となる。

なお、レイが物体の内側を通過中か、外側を通過中かを区別するフラグINflagの取扱いを変更することにより、物体内部のボクセル・データの表示も可能である。

本手法を、面による反射、影などの影響を考慮したレイ・トレーシング法に拡張すること、さらには、本手法に物体表面上へのテクスチャー・マッピングの機能を付加することなどは容易である。しか

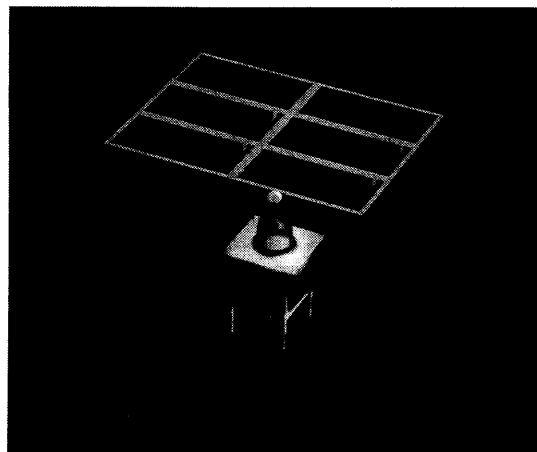


図6

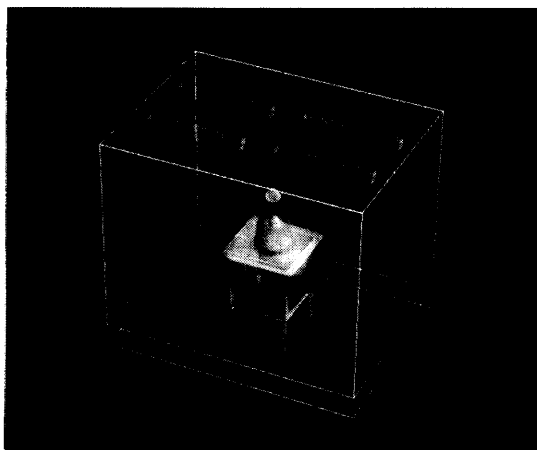


図7

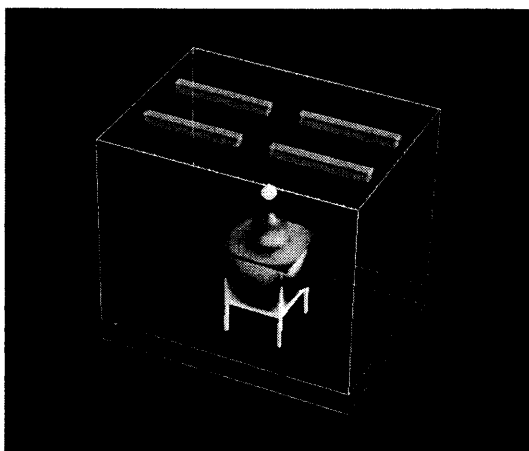


図8

し、本手法の目的はリアリズムの追及ではなく、3次元ボリューム・データの可視化であり、また、計算コストの観点からも適切な発展の方向とは言えない。

今後の課題として、計算時間の短縮が挙げられる。そのためには、計算の並列化、交差計算の高速化などを再検討する必要がある。さらに、3次元ボリューム・データの可視化においては、アニメーション化することが重要な意義を持っているが、このアニメーション作成をも含めた計算時間の短縮が必要となるであろう。

[参考文献]

- 1) H.N.Christianson,etc,"Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics" Computer Graphics,Vol.12, No.3,1978,pp.187~192
- 2) W.E.Lorensen,etc,"Marching Cubes: A High Resolution 3D Surface Construction Algorithm" Computer Graphics,Vol.21,No.4,1987,pp.163~169
- 3) H.Levoy,"Display of Surfaces from Volume Data" IEEE Computer Graphics & Applications,May,1988,pp.29~37
- 4) R.A.Drebin,etc., "Volume Rendering" Computer Graphics,Vol.22,No.4,1988,pp.65~74
- 5) 鷲津久一郎、他、"有限要素法ハンドブック I 基礎編" 培風館,1981
- 6) 中前栄八郎,etc., "3次元コンピュータグラフィックス" 昭晃堂,1986
- 7) A.Fujimoto,etc., "ARTS: Accelerated Ray-Tracing System" IEEE Computer Graphics and Applications,Vol.6,No.4,1986,pp.16~26
- 8) J.M.Snyder,etc., "Ray Tracing Complex Models Containing Surface Tessellations" Computer Graphics,Vol.21,No.4,1987,pp.119~128