

等関数値曲面の高速表示アルゴリズム

6P-6

土井章男・小出昭夫

日本アイ・ビー・エム株式会社

1. はじめに

本論文では、格子点データから等関数値曲面を高速に生成・表示するアルゴリズムについて報告する。ここで、等関数値曲面とは、定義域を3次元とする関数 F と定数 C に対し、

$$F(x, y, z) - C = 0 \quad (1)$$

を満たす点 (x, y, z) の集合である。

本アルゴリズムの特長は、格子点データ、四面体分割の性質を利用することにより、奥行方向の多角形ソーティングや奥行バッファを用いずに隠面消去を行うところである。

2. 格子空間および四面体分割

前提として格子点データを想定する。この格子点を頂点とする直方体を5個あるいは6個の四面体に分割する。四面体に等関数値曲面が存在するかどうかは、四面体を構成する4頂点の式(1)の左辺の符号により調べることができる(表-1)。ここで、表-1の-, 0, +の欄の数は、該当する頂点の数である。ケース7~13の四面体のみを対象に多面体近似することにより、2重に求めることなく自動的に互いに接続した多面体データを生成できる。

3. 多面体データの頂点計算および法線計算

格子点を頂点とする各四面体の辺上での線形補間により多面体近似の頂点を求めることにより、等関数値曲面を近似する多角形データを同時生成することができる。四面体内の線形補間はその頂点上の関数値だけで十分である。この多角形の各頂点 (x, y, z) は四面体の辺上にあり、辺の両端での関数値 $F_1 = F(x_1, y_1, z_1)$ と $F_2 = F(x_2, y_2, z_2)$ から

$$\begin{aligned} t &= (F_1 - C) / (F_1 - F_2) \\ x &= x_1 + t(x_2 - x_1) \\ y &= y_1 + t(y_2 - y_1) \\ z &= z_1 + t(z_2 - z_1) \end{aligned} \quad (2)$$

により、求まる。本アルゴリズムでは、多面体近似の精度を向上させるためにさらに式(2)の t を用いた2次補間も適用できる⁽¹⁾。

法線計算は、まず各格子点上での法線を関数値の差分により求め⁽²⁾、頂点上の法線は、式(2)の t を用いて線形補間により求める。頂点上の法線は、Phongシェーディングで用いる。

4. アルゴリズム

本アルゴリズムは、視点および注視点の位置が決定された時点で、格子および四面体の処理順序を決定する。ここでは、格子の処理順序を"格子処理手順"、四面体の処理手順を"四面体処理手順"と呼ぶことにする。

格子処理手順

格子処理手順は、視線ベクトル V と格子の各 x, y, z 軸の方向ベクトル I, J, K との内積の符号によりプログラム内のDOLープで表す。ここでは、右手座標系であるユーザ空間の X, Y, Z 軸にそって、 I, J, K が決められているものとする。

Step.1 ベクトル V, I, J, K を正規化する。

Step.2 内積 $V \cdot I, V \cdot J, V \cdot K$ の符号からDOLープ内のインデックスを決定する。符号が負の場合、各インデックスは1から順に増加させ、正の場合は格子数から順に減少させる。FORTRAN言語風に記述すると

```
integer*4 i1,i2,idecr,j1,j2,jdecr,
+         k1,k2,kdecr
if(V*I<0.0){i1=1;i2=Imax;idecr=1;}
else      {i1=Imax;i2=1;idecr=-1;}
if(V*J<0.0){j1=1;j2=Jmax;jdecr=1;}
else      {j1=Jmax;j2=1;jdecr=-1;}
```

High-Speed Display Algorithm for Equi-Valued Surfaces

Akio DOI, Akio KOIDE

IBM Japan Ltd.

```
if (V·K<0.0) {k1=1;k2=Kmax;kdecr=1;}
else      {k1=Kmax;k2=1;kdecr=-1}
```

```
DO 10 i=i1,i2,idecr
  DO 10 j=j1,j2,jdecr
    DO 10 k=k1,k2,kdecr
```

直方体(i,j,k)を処理しなさい。

10 CONTINUE

ここで、Imax, Jmax, Kmaxは、各x, y, z方向の直方体の数である。

四面体処理手順

一個の直方体を取り出し、その中に含まれる四面体の処理順序を、視点から四面体の重心までの距離により決定し、テーブルに蓄える。四面体の処理は、このテーブルを参照しながら行う。

等関数値曲面の生成および表示

等関数値曲面を含む四面体を格子処理順序および四面体処理順序により取り出し、多面体近似を行ない、順次、グラフィック装置に転送またはフレームバッファに書きこむ。多角形データは、必ず視点方向に対して奥から生成されるので自動的に隠面処理ができる。

複数の定数Cをもつ等関数値曲面の多重透明表示の場合は、四面体ごとに生成された多角形データのみを奥行を比べ、奥行順にフレーム・バッファにその色を加算することにより行うことができる。

5. 本アルゴリズムの評価

本アルゴリズムは、奥行計算を行わない分だけリスト・プライオリティ法やZ-バッファ法より高速である。実際のシミュレーションの解析において、色々な等関数値曲面を表示させたり、回転、拡大、縮小を対話的に行うことが多く、本アルゴリズムは、そのような用途に対して効果的である。

本アルゴリズムが適用出来ない場合は、視点が格子空間に非常に近くにあり過度の透視変換で等関数値曲面を見る時であるが、このような使用は、一般に少ないと仮定している。

6. おわりに

本アルゴリズムは、奥行バッファや生成される多角形データを保持する必要がないためにメモリー効率の面でも有効である。図1は、本アルゴリズムによるCT画像の骨の表示である。

7. 参考文献

- 1) A. Doi and A. Koide, "An Efficient Method of Triangulating Equi-Valued Surfaces by Using Tetrahedral Cells," IEICE Trans. Vol.E-74, No.1.
- 2) D. Gordon and R. A. Reynolds, "Image Space Shading of 3-D Objects," Computer Vision, Graphics, and Image Proc., 29, pp.361-376, 1985.

表1 四面体の関数値と定数Cとの大小関係

ケース	-	0	+	生成される多角形
1	4	0	0	なし
2	0	0	4	なし
3	3	1	0	なし
4	0	1	3	なし
5	2	2	0	なし
6	0	2	2	なし
7	3	0	1	3角形 (1個)
8	1	0	3	3角形 (1個)
9	2	0	2	4角形 (1個)
10	2	1	1	3角形 (1個)
11	1	1	2	3角形 (1個)
12	1	2	1	3角形 (1個)
13	1	3	0	3角形 (1個)
14	0	3	1	3角形 (1個)
15	0	4	0	3角形 (4個)

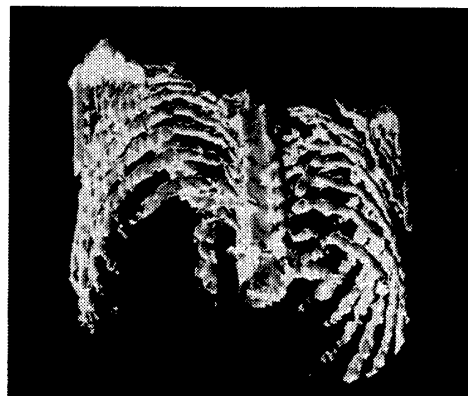


図1 CT画像の等関数値曲面表示