多面体近似による形状モデリングのための ボリュームデータの形状解析

Shape Analysis of Volume Data for Polyhedral Shape Modeling

李春艶　剣持雪子　小谷一孔
Chunyan LI　Yukiko KENMOCHI　Kazunori KOTANI

北陸先端科学技術大学院大学 情報科学研究科
School of Information Science ,JAIST

Abstract: The polyhedrization algorithm is used for polyhedral shape modeling from a given volume data and it is well known that there are open problems for the algorithm. In this paper, we define "flatness" for each point in volume data to solve the problems and present its calculation algorithm.
Keyword: shape analysis, shape modeling, volume data, polyhedrization, flatness

1 はじめに

計算機の進歩に伴い、3 次元モデリングの技術や応用分野が急速に拡大し、3 次元物体のモデリングに関する研究が新しい情報科学の基盤の 1 つになってきている [1]。形状モデリングにおいて、形状が複雑かつ設計図がない対象物に対しては、計測したボリュームデータから形状モデリングすることを考える。形状モデリングにおける 3 次元物体の表現手法としては、ポリゴン表現、パラメトリック曲面表現、多面体表現などがある [2]。その中でも多面体表現を用いた形状モデルは対象物の境界の情報を持ちなどの特徴がある。多面体表現は多面体化アルゴリズムを用いて形状モデリングを行う。しかし、多面体化アルゴリズムには得られた多面体の平面の数、位置、大きさが一意に決まらないという問題がある [3]。そこで本稿では、この問題を解決するために、ボリュームデータのような離散的な点の配置から物体表面の平坦な度合いを表す新しい尺度である平坦度を定義し、それを用いた多面体化アルゴリズムの問題の解決方法を検討する。また、平坦度を求めるアルゴリズムも検討し、楕円体のボリュームデータに適用した実験結果を示す。

2 多面体表現

3 次元形状の多面体表現とは、与えられた対象物のボリュームデータに対し、多面体化アルゴリズムを用いて、平面の集合である多面体を用いて対象物を形状モデリングすることである。

2.1 離散平面の定義

ここでは、3 次元ユークリッド空間 $\boldsymbol{R}^3$ において座標値が全て整数値である点の集合を $\boldsymbol{Z}^3$ とし、離散空間と呼ぶ。本研究で扱うボリュームデータ $\boldsymbol{V}$ は $\boldsymbol{Z}^3$ の部分集合である。

多面体は平面の集合であり、以下に $\boldsymbol{Z}^3$ における平面を定義する。$\boldsymbol{Z}^3$ では式 (1) を満たすような点 (x, y, z) の集合を離散平面という [4]。

$$0 \leq ax + by + cz + d < w \tag{1}$$

但し、全パラメータは整数である。$w > 0$ であり、離散平面の幅という。(a, b, c) は離散平面の法線ベクトルであり、a, b, c, d は離散平面の係数と呼ぶ。 $w = \max(|a|, |b|, |c|)$ の時、離散平面を特に naive plane という [5]。今後、naive plane を略称し平面と呼ぶ。

2.2 離散平面による物体の多面体表現

対象物のボリュームデータ $\boldsymbol{V}$ からその境界部分の点集合 $\boldsymbol{B} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ は、文献 [5] のアルゴリズムによって一意に求められる。ここで、$\boldsymbol{v}_i \in \boldsymbol{Z}^3 (i = 1, 2, \ldots, n)$、$\boldsymbol{B} \subset \boldsymbol{V} \subset \boldsymbol{Z}^3$ とする。離散平面の有限部分集合 $\boldsymbol{P}_i$ を平面分と呼ぶと、$\boldsymbol{P}_i \subset \boldsymbol{B}$、かつ $\cup_{i=1}^{m} \boldsymbol{P}_i = \boldsymbol{B}$ となる複数の $\boldsymbol{P}_i$ を考えることができる。各 $\boldsymbol{P}_i$ は式 (1) を満たす点の部分集合なので、平面の係数 (a_i, b_i, c_i, d_i) で表すこともできる。そして、多面体は平面分の和集合 $\cup_{i=1}^{m} \boldsymbol{P}_i$ で表される。m, n は整数である。

3 多面体化アルゴリズム

多面体化アルゴリズムは $\boldsymbol{B}$ を $\cup_{i=1}^{m} \boldsymbol{P}_i$ に分解するアルゴリズムである。

```
Algorithm1
Input: B = {v_1, ..., v_n}
Output: ∪_{i=1}^{m} P_i
begin
1. i := 1;
2. j := 1;
3. while(B ≠ φ ) {
   3.1. Q_j ⊂ B を選ぶ;
   3.2. if( CHECK(Q_j)==0)
        3.2.1. then{ j − −; P_i := Q_j; B := B \ Q_j ; i + +; }
        3.2.2. else ;
   3.3. j + +;
}
end
```

なお、CHECK($\boldsymbol{Q}_j$) は集合 $\boldsymbol{Q}_j$ の平面判定を行う関数であり、$\boldsymbol{Q}_j$ 中の点が全て 1 つの平面に存在しうる時には 1 を返し、そうでなければ 0 を返す。本稿では、文献 [6] の平面判定アルゴリズムを用いる。

4 多面体化アルゴリズムの問題点

前章で説明した多面体化アルゴリズムは、得られた多面体の平面の数、位置、大きさが一意に決まらないという問題がある。その原因は、$\boldsymbol{B}$ から $\boldsymbol{P}_i$ への分解の仕方が一意ではないからである。言い換えると、アルゴリズム 1 のステップ 3.1 での $\boldsymbol{Q}_j$ の選択方法によって、異なった $\boldsymbol{P}_i$ が得られてしまう。この問題を解決するためには $\boldsymbol{Q}_j$ の上手な選択方法を考える必要がある。

5 平坦度による多面体化アルゴリズム問題解決の検討

多面体化アルゴリズムの問題を解決するために、ボリュームデータ点の平坦度を新たに定義する。平坦度を用いて、多面体化アルゴリズムのステップ 3.1 における $\boldsymbol{Q}_j$ の選択方法に制約をつけることで、良い $\boldsymbol{P}_i$ を得るような $\boldsymbol{Q}_j$ の選択方法を決めることを考える。

多面体化アルゴリズムでは、点の集合 $\boldsymbol{Q}_j$ に対し、平面 (planarity) であるかどうかの判定を行なっている。一般的に知られている $\boldsymbol{Z}^3$ における planarity の定義は以下のように与えられている [4][6][7]。

$$planarity(\boldsymbol{Q}_j) = \begin{cases} 0 & \boldsymbol{Q}_j\text{が平面を満たさない} \\ 1 & \boldsymbol{Q}_j\text{が平面を満たす} \end{cases}$$

それに対し、本研究では、各点 $\boldsymbol{v}$ の周りの点の分布から $\boldsymbol{v}$ の平坦の度合いを表す平坦度 (flatness) を定義する。

$$flatness(\boldsymbol{v}) = N(\boldsymbol{v}) \geq 0$$

ここで、$N(\boldsymbol{v})$ は $\boldsymbol{v}$ の周りの平面である条件を満たす点の集合の広がりを表し、平面の大きさや面積に対応する。即ち、平坦度は $\boldsymbol{v}$ 及びその周りの点の集合が平面を満たすかどうかだけではなく、平面である条件を満たす場合、平面の広がりという平坦の度合い (平面の可能性) を表すものである。但し、解像度や対象物によって、平坦度の値が異なり、一概に全対象物体の平坦度の最大値を定めることができない。そのため、現段階では、平坦度の正規化を行なわない。

そこで、平坦度を用いた $\boldsymbol{Q}_j$ の選択方法を考えると、平坦度の大きい点ほど周りの点の集合が平面である可能性が高いため、$\boldsymbol{Q}_j$ の選択には平坦度の高い点を優先することなどが挙げられる。

5.1 平坦度算出アルゴリズム

点 $\boldsymbol{v}$ における平坦度 $flatness(\boldsymbol{v})$ を以下のアルゴリズムにより求める。

Algorithm2
Input: $B = \{v_1, \ldots, v_n\}$
Output: $F = \{flatness(v_1), \ldots, flatness(v_n)\}$
begin
1. $i := 1$;
2. $j := 1$;
3. while$(i \leq n)\{$
  3.1. $O_j(v_i) \subset B$ を選ぶ
  3.2. if (CHECK($O_j(v_i)$)==0)
    3.2.1. then$\{$ $j--$; $flatness(v_i) := |O_j(v_i)|$; $i++$; $\}$
    3.2.2. else ;
  3.3. $j++$;
}
end

なお、$|\boldsymbol{O}_j(\boldsymbol{v}_i)|$ は $\boldsymbol{O}_j(\boldsymbol{v}_i)$ が含む点の数である。

5.2 順序付けアルゴリズム

アルゴリズム 2 のステップ 3.1 における $\boldsymbol{O}_j(\boldsymbol{v}_i)$ を決定するアルゴリズムを検討する。ここで、文献 [5] のアルゴリズムにより求めたボリュームデータ $\boldsymbol{V}$ の境界部分集合 $\boldsymbol{B}$ は三角形分割りされており、この三角形分割の構造を用いて、各点 $\boldsymbol{v}$ に対し星状体 [5] と呼ばれる点集合 $\boldsymbol{A}(\boldsymbol{v}) = \{ \boldsymbol{v}_1, \ldots, \boldsymbol{v}_N \}$ を求める (図 1 参照,$\boldsymbol{A}(\boldsymbol{v}) = \{ \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_5, \boldsymbol{v}_6 \}$)。

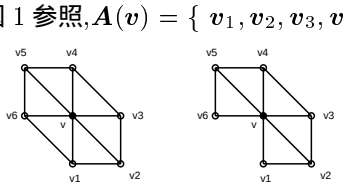

図 1 星状体 $A(v)$ が輪状である例 (左) 及び輪状ではない例 (右)

点 $\boldsymbol{v}_i$ の順番は $\boldsymbol{v}$ を中心に物体の外側から見て反時計周りに決定する。平面における星状体の位相構造は必ず輪状 [5][6] であるので、始点 $\boldsymbol{v}_1$ を決定すれば $\boldsymbol{A}(\boldsymbol{v})$ の点の順序は一意に決定することができる。そして、$\boldsymbol{v}_1$ に対しても同様に星状体 $\boldsymbol{A}(\boldsymbol{v}_1)$ を求め (図 2 参照, $\boldsymbol{A}(\boldsymbol{v}_1) = \{ \boldsymbol{v}_1', \boldsymbol{v}_2', \boldsymbol{v}_3', \boldsymbol{v}_4', \boldsymbol{v}_5', \boldsymbol{v} \}$)、図 3 にようにの $\boldsymbol{A}(\boldsymbol{v})$ の星状体との差集合に含まれる $\boldsymbol{A}(\boldsymbol{v}_1)$ の点に、やはり反時計周りに順序付けしていく。それらの点を $\boldsymbol{v}_{N+1}$、$\boldsymbol{v}_{N+2}$ とする。引き続き、$\boldsymbol{v}_2$ 以下にも同様な順序付け処理を繰り返すことで、$\boldsymbol{v}$ の周りの点を一意に順序付けすることができる。このような順序付けられた $\boldsymbol{v}$ の周りの点から集合 $\boldsymbol{O}_j(\boldsymbol{v})$ が成る ($\boldsymbol{O}_j(\boldsymbol{v}) = \{ \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_5, \boldsymbol{v}_6, \boldsymbol{v}_7, \boldsymbol{v}_8, \boldsymbol{v}_9 \}$)。

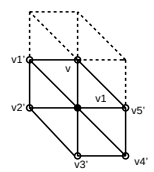

図 2 $A(v_1)$ の星状体図

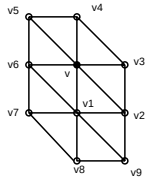

図 3 v の周りの点の順序付け図

6 実験

平坦度の妥当性を確認するために、図 4 に示す楕円体のいくつか点 $\boldsymbol{v}_1, \boldsymbol{v}_2$ に対し、平坦度をそれぞれ求めた。なお、楕円体のボリュームデータは $x^2/10^2 + y^2/15^2 + z^2/8^2 <= 1$ を満たす点 $(x, y, z) \in \boldsymbol{Z}^3$ の集合とした。それぞれの値は $flatness(\boldsymbol{v}_1) = 493, flatness(\boldsymbol{v}_2) = 11$ となり、$\boldsymbol{v}_2$ よりも $\boldsymbol{v}_1$ の平坦度が高いという予測と一致した。

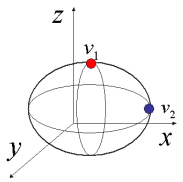

図 4 実験対象図

7 まとめ

ボリュームデータを入力とする多面体による形状モデルを実現するために、多面体化アルゴリズムの問題の原因を分析し、問題を解決するために、ボリュームデータに対して平坦度を新たに定義し、解決方法の検討を行なった。さらに、平坦度を求めるアルゴリズムを検討し、実際のデータに適応して実験した。今後の課題として、平面の法線ベクトル、幅、平面の種類を考慮し、平坦度の拡張や平坦度を用いた多面体化アルゴリズムの検討などがあげられる。

参考文献

[1] Hongbin Zha, Tomoo Mitutomi, Tsutomu Hasegawa, "Generation of Multiresolution Model of 3-D Free-From Objects from Range Data",Meeting on Image Recognition and Understanding , vol.2, pp241-246, 2000

[2] 技術編 CG 標準テキストブック編集委員会, "技術編 CG 標準テキストブック", 財団法人画像情報教育振興協会, 1999

[3] Jean Francon, Laurent Papier, "Polyhedrization of the Boundary of a Voxel Object", Discrete Geometry for Computer Imagery, LNCS 1568,pp425-434,1999

[4] J.P. Reveilles, I.Devled-Renesson, "Incremental algorithm for recognizing pieces of digital planes", 4^{th} Conference on Discrete Geometry in Computer Imagery,Strasbourg , 1994

[5] Yukiko Kenmochi, Atsushi Imiya, Akira Ichikawa,"Boundary Extraction of Discrete Objects",Computer Vision and Image Understanding,Vol. 71,pp281-293,1998

[6] Jean Francon, Jean-Maurice Schramm, Mohamed Tajine, "Recognizing Arithmetic Straight Lines and Planes", Proc. Int. Workshop on Discrete Geometry for Computer Imagery, LNCS 1176, pp142-149,1996

[7] C.E.Kim, "Three-dimensional digital planes ", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.PAMI-4, pp639-645, 1984

〒 923–1292 石川県 能美郡 辰口町 旭台 1–1
北陸先端科学技術大学院大学 情報科学研究科
Phone 0761–51–1231 E-mail {sri,kenmochi,ikko }@jaist.ac.jp