

双3次 Bézier パッチの4分木データ構造を用いた分割

森山 真光 原田 耕一

広島市 中区 東千田町 1 - 1 - 8 9
広島大学

あらまし

本研究では、自由曲面を効率よく平面分割する手法を確立することを考える。すなわち、4分木データ構造を用いて、いかにすれば最適な、パラメトリック曲面の平面パッチによる近似が得られるかを考える。具体的には、パラメトリック曲面を(A)パラメータ空間で一様に分割する方法、(B)パラメータ空間で中点を用いる方法、(C)曲率最大点を用いる方法、および(D)前処理として曲率最大点を検出し分割点を定める方法について調べ、各手法を適用した結果得られる分割木の深さや葉の数について評価し、得られる平面近似の近似精度について論じる。

和文キーワード Bézier パッチ , 4分木データ構造 , 曲率 , 曲面近似 , アルゴリズム

Subdivision of Bicubic Bézier Patches by Using Quad-tree Data Structure

Masamitsu Moriyama Koichi Harada

Hiroshima University

Higashisenda-machi 1-1-89 , Hiroshima-shi , Hiroshima , 730 , JAPAN

Abstract

The aim of this paper is to propose an efficient division method with which we can approximate parametric surfaces with the net of planar facets. More specifically, by using quad-tree data structure, we investigate parametric surface division methods based on (A) uniform division in parametric space, (B) setting division point at the center in parametric space, (C) using division point at the curvature maximum point, and (D) the same as (C) except the curvature maximum point is calculated as the output of pre-processing.

Each resulting quad-tree is compared regarding to the depth and the number of leaves.

英文 key words Bézier patch , quad-tree data structure , curvature , surface approximation , algorithm

1 はじめに

コンピュータ・グラフィックスの研究分野の一つである、曲面のレンダリング手法には二つの流れがある。一つは (i) 曲面を平面のパッチの集合として近似しこれをグラフィックスデータとして用い、表示や保存を行う方法である [1]。他の一つは (ii) 曲面は自由曲面としてそのままパラメトリックな形式で保存しておき、実際にレンダリングを行う際に Bézier クリッピング手法などを用いてビットマップに変換し表示する方法である [2]。

グラフィックス専用機を用いてレンダリングのための処理を極めて高速に実行できる場合には、(ii) の手法の方が適している。すなわち、(ii) を用いることにより、曲面データを作成したり、保存するのに必要な処理が簡単化される。また、使用する表示装置の解像度に応じて定まる最高の品位の曲面表示が可能となる。

しかしながら、(i) の方法を用いれば、コンピュータ・グラフィックスで広く研究されてきた平面に関する隠面消去や陰影表示手法などの成果が利用できる。また、Bézier クリッピング手法などを用いない一般の場合に高速に曲面を描くためには、曲面を前もって数多くの平面に変換し、それぞれの平面にレンダリングを施す方が適切である。なぜなら、パラメトリック曲面に直接、レンダリングのための処理を施すより、パラメトリック曲面を一旦平面パッチの集合に変換し、これをレンダリングした方が総合的にみて計算時間が短縮されることが多いからである。さらに、曲面を繰り返し表示する必要がある場合は、この傾向が著しい。

以上のことから、本研究では、自由曲面を効率よく平面分割する手法を確立することを考える。すなわち、4分木データ構造を用いて、いかにすれば最適なパラメトリック曲面の平面パッチによる近似が得られるかを議論する。具体的には、パラメトリック曲面を (A) パラメータ空間で一様に分割する方法、(B) パラメータ空間で中点を用いる方法、(C) 曲率最大点を用いる方法、および (D) 前処理として曲率最大点を検出し分割点を定める方法について調べる。各手法について、分割してできた木の深さや葉の数について評価し、得られる平面近似の精度について調べる。

なお、自由曲面には、CAGD(Compter Aided Geometric Design) において広く用いられている双 3 次 Bézier パッチを取り上げる。

2 曲面のパラメータ表現

$(n+1)(m+1)$ 個の点 $\mathbf{b}_{ij}(i = 0, 1, \dots, n; j = 0, 1, \dots, m)$ を与えて定義できるパラメトリック曲面パッチ

$$\mathbf{P}(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) \mathbf{b}_{ij}, \quad (0 \leq u \leq 1, 0 \leq v \leq 1),$$

は、 $n \times m$ 次 Bézier 曲面パッチと呼ばれる。ただし、 $\{B_{i,n}(u)\}$ は n 番目のベルンシュタイン基底関数である。また、 $\mathbf{b}_{ij}(i = 0, 1, \dots, n; j = 0, 1, \dots, m)$ の隣合う点をそれぞれ結んでいくとネットが形成され、このネットは、曲面パッチの曲面定義ネットまたは Bézier ネットとよばれる。

一般的に n と m は、5 以下に選ばれる。なぜなら、 n, m があまり大きいと、Bézier ネットを媒介とした曲面パッチの形状を予め知ることができず、CAGD において不都合となるからである。最もよく使われるのは、 $n = m = 3$ 、すなわち、双 3 次 Bézier 曲面パッチである。そこで、本研究においても、双 3 次 Bézier 曲面パッチについて、曲面分割を行っていく。

3 双 3 次 Bézier パッチを用いたデータ補間

多数のデータ点で構成されているメッシュデータを 1 つの自由曲面で補間すると、自由曲面の次数が高くなり、曲面を生成するための計算が複雑になる。そこで、データ間を順に多数のパッチでつないで、全体としてひとつの曲面を表現すれば、この問題は解決できる。本研究では、メッシュデータ間を特に双 3 次 Bézier パッチで滑らかに補間することを考える。アルゴリズムを以下に示す。

1. 基礎データとして、曲面上の点 $r_{ij}(i = 0, 1, \dots, n, j = 0, 1, \dots, m)$ を定める。
2. 点列 $r_{0j}, r_{1j}, \dots, r_{nj}(j = 0, 1, \dots, m)$ を補間し、 U 方向接線ベクトルを求める。
3. 点列 $r_{i0}, r_{i1}, \dots, r_{im}(i = 0, 1, \dots, n)$ を補間し、 V 方向接線ベクトルを求める。
4. 接線ベクトル列 $T_{0j}^u, T_{1j}^u, \dots, T_{nj}^u$ を補間し、ねじれベクトルを求める。
5. 接線ベクトル列 $T_{i0}^v, T_{i1}^v, \dots, T_{im}^v$ を補間し、ねじれベクトルを求める。

6. データ点間を、 U, V 方向の接線ベクトル、ねじれベクトルより、双3次 Bézier パッチで補間する。

メッシュデータと双3次 Bézier パッチ補間曲面の関係は図1のようになる。

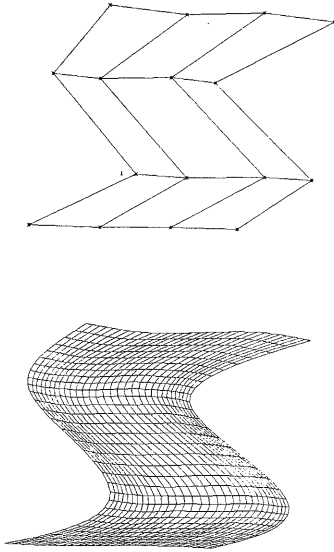


図1: メッシュデータと双3次 Bézier パッチ補間曲面

4 曲面の分割

曲面の分割には、以下のようなものが考えられる。

- (A) パラメータ空間で一様分割する。
 曲面のパラメータ値として $u_i = i/N, v_j = j/M$ ($i = 0, 1, \dots, N; j = 0, 1, \dots, M$) をとり、隣合う点どうしを直線で結んでいく。
- (B) 4分木データ構造を用いてパラメータ空間の中点で分割する。
 四隅と中央の点で、曲面を4分割することを、平面条件を満たすまで、再帰的に繰り返す。

- (C) 4分木データ構造を用いてパラメータ空間の サンプル曲率最大点で分割する。

適当な個数のサンプル点を取り、これらの点のうちで曲率が最大となる点を分割点とする。そして、四隅と分割点で曲面を4分割することを、平面条件を満たすまで、再帰的に繰り返す。

- (D) 4分木データ構造を用いてパラメータ空間の曲率最大点で分割する。

次章で述べる前処理によりパッチ内で曲率最大点を求め、この点を分割点とする。そして、四隅と分割点で曲面を4分割することを、平面条件を満たすまで、再帰的に繰り返す。

手法(A)は、最も単純な方法で曲面の平面近似によく用いられている。しかし、平らな部分も曲率が大きい部分も考慮せず分割するので、輪郭が滑らかにならないことがある。

手法(B)は、以下の手順で4分木データを再帰的に作成する。

- 各パッチのパラメータ空間上の中央の点を求めて、その点を分割点とみなす。
- 各パッチの四隅と分割点で平面条件(4.1節)を満たさないならば、分割を行ない、サブパッチ毎に最初のステップに戻る。
- 各パッチの四隅と分割点で平面条件(4.1節)を満たすならば、パッチを平面とみなす。

従来法である(B)では、パッチの形状が急激に変化するところ、つまり、パッチの曲率の大きいところを見逃してしまう恐れがある。そこで、本研究では手法(C),(D)を提案する。

手法(C)は、以下の手順で4分木データを再帰的に作成する。

- 各パッチの曲面上に適当な個数のサンプル点をとる。
- サンプル点毎に曲率を求める。
- サンプル点の内で曲率が最大になる点を分割点にする。
- 各パッチの四隅と分割点で平面条件(4.1)を満たさないならば、分割を行ない、サブパッチ毎に最初のステップに戻る。

- 各パッチの四隅と分割点で平面条件 (4.1) を満たすならば、パッチを平面とみなす。

手法 (C) ではサンプル点毎に曲率を求めているため、サンプル点間に曲率最大点がある場合見落とすことが考えられる。

そこで、手法 (D) では前処理 (5節) によって曲率最大点を求めている。手法 (D) は、以下の手順で 4 分木データを再帰的に作成する。

- 各パッチの曲率最大点を前処理によって求め、その点を分割点とみなす。
- 各パッチの四隅と分割点で平面条件 (4.1) を満たさないならば、分割を行ない、サブパッチ毎に最初のステップに戻る。

- 各パッチの四隅と分割点で平面条件 (4.1) を満たすならば、パッチを平面とみなす。

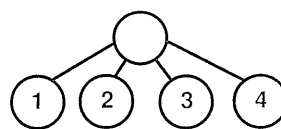
手法 (C),(D) により、双 3 次 Bézier パッチを曲率大きいところでは細かく、曲率の小さいところでは荒く平面近似することができる。

4.1 4 分木データ構造

分割されるパッチには、長方形や三角形に変換された面が使われる。どちらの面のタイプにも同一のデータ構造が使える。このデータ構造の主要な構造は、分割された面を表現する 4 分木データ構造 (quad-tree data structure) である。4 分木データ構造の節 (node) のそれぞれが、分割された面、サブパッチ (subpatch) に相当する。4 分木データ構造の葉は最終分割を表す。長方形の面の場合、ノードは長方形の面パッチの領域を表す。三角形の面の場合は、三角形の面パッチの領域を表す。4 分木データ構造とパラメータ空間と 3 次元空間との関係は、図 2 のようになる。

図 2 の 4 分木データ構造を C 言語で表現すると以下のようになる。

```
struct
node1 { int          lev;
        float        t1u[4],t1v[4],tf[8];
        struct UVcoords U[5];
        struct XYZcoords Tu[5],Tv[5],N[5],P[5];
        struct node1  *chd[4];
} Quadtree;
```



Representation of quad-tree

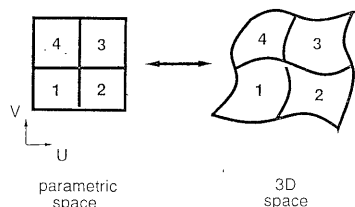


図 2: 4 分木データ構造

lev は節の深さ (レベル) を格納する (例えば、木構造の根の深さは 0 である)。t1u,t1v,tf は、edge linearity と flatness の測定値を格納している。struct UVcoords と struct XYZcoords はそれぞれ、 \mathbf{R}^2 と \mathbf{R}^3 の領域を格納する。面パッチの頂点について、U はパラメータ範囲の値、Tu,Tv は接線、N は法線、P は位置を格納する。*chd は 4 分木の子の節へのポインタである。

この分割は、直観的には、物理空間にある面の XYZ 分割というよりは、UV の局所部分の領域分割である。幾何学的には、分割は、領域をモザイク化することである。アルゴリズム的には、分割は、4 分木の生成と同じ意味をもつ。4 分木データ構造の種類には、以下のものが考えられる。

- ユニフォーム分割
(4 分木の最大レベルを最初にユーザーが選択決定するもの)
- ノンユニフォーム分割
(ある条件を満たすまで葉を生成していくもの)
- 制限付 (restricted) 分割
(木の深さの差に制限を加えたもの)

ユニフォーム 4 分木データ構造は、葉の節が全て同じ深さであり、同じ大きさの部分領域を表現している。逆に、ノンユニフォーム 4 分木データ構造は、葉の節が、違う大きさの部分領域を表現している。制限付 4 分木データ構造は、ノンユ

ニフォームに制約が加えたものである。普通は、隣合っている(辺でつながっている)葉の深さの差がある値で制限されている。ノンユニフォーム4分木データ構造は、アダプティブ(adaptive)と呼ばれる。アダプティブ分割は、flatnessとedge linearityに基いている。この3種の分割は、図3に示されている。3番目のものは、深さの差を1で制限した4分木を表現している。本研究では、以下に述べる平面条件を満たすまで葉を生成するので、アダプティブ分割を用いる。

アダプティブ分割で用いられる flatness と edge linearity は、法線ベクトルに間の角度や接線ベクトルの間の角度によって決定される(図4)。隣合う角の頂点のアイソパラメトリック曲線の方向の接線ベクトル同士(4テスト)、隣合う角の頂点の法線ベクトル同士(4テスト)、角の法線ベクトルと内部の法線ベクトル(4テスト)、をそれぞれテストする。つまり、

$$1 - T_i \cdot T_j < linearity \quad 1 - N_i \cdot N_j < flatness$$

をそれぞれ8回ずつ実行する。ここで、 i と j は、サブパッチの頂点の番号であり、linearity, flatness は平面への近似を制御するパラメータであり、必要な精度が得られるよう経験的に定める。

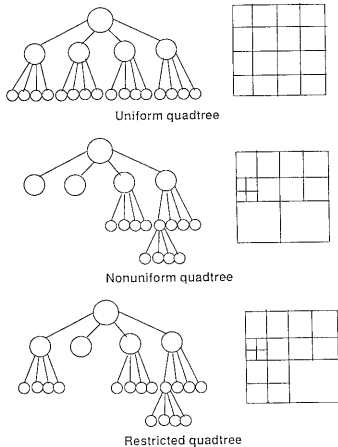


図3: 4分木データ構造の種類

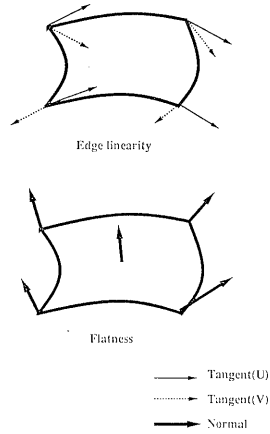


図4: 平面条件

5 曲面分割の前処理

双3次 Bézier パッチを、手法(D)により平面近似するとき、前処理として各パッチでの曲率最大点を求める必要がある。双3次 Bézier パッチの曲率最大点は、以下のようにして求める。

- 双3次 Bézier パッチ $P(u, v)$ について、 $P_u(u, v), P_v(u, v)$, 単位法線ベクトル $N(u, v), P_{uu}(u, v), P_{uv}(u, v), P_{vv}(u, v)$ を計算し、 κ を求める。

- 連立方程式

$$\begin{cases} \kappa_u(u, v) = 0 \\ \kappa_v(u, v) = 0 \end{cases}$$

を解く。

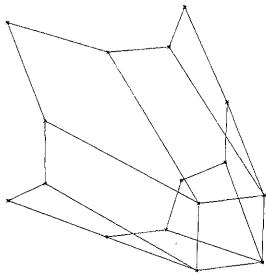
- 上の連立方程式の解

$$\begin{cases} u = u_0 \\ v = v_0 \end{cases}$$

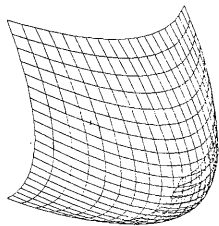
がパッチのパラメータ範囲内であれば、 $P(u_0, v_0)$ が曲率最大点になる。

6 結果

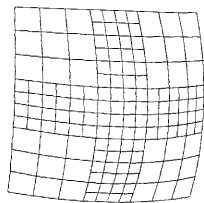
4節で述べた分割手法(A),(B),(C)について、制御パラメータ linearity, flatness を0.1の場合の結果を示す。3つのデータ(データ1, データ2, データ3)に関する適用結果を図5, 図6, 図7に示す。また、データ1, 2, 3について、linearity, flatness をそれぞれ0.3, 0.2, 0.1にしたときの深さと葉数を表に示す。



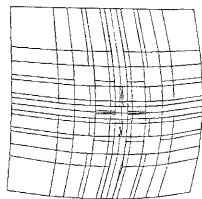
(a) メッシュデータ



(b) 手法 (A)

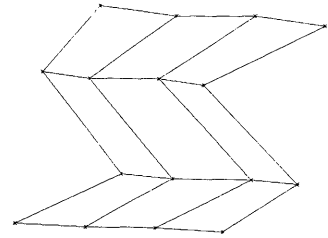


(c) 手法 (B)

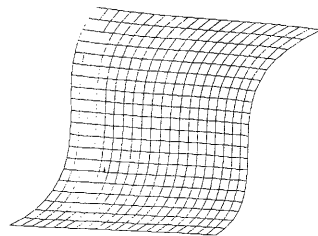


(d) 手法 (c)

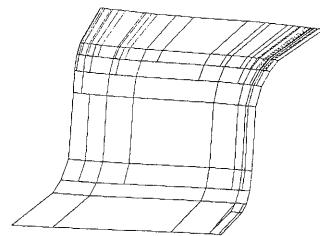
図 5: データ 1 の適用結果



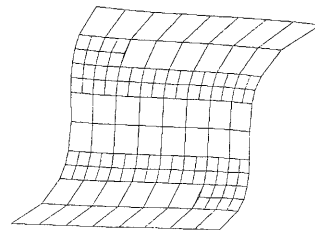
(a) メッシュデータ



(b) 手法 (A)

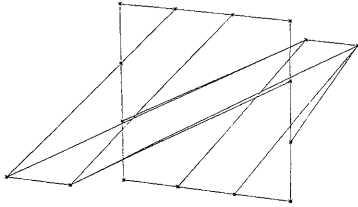


(c) 手法 (B)

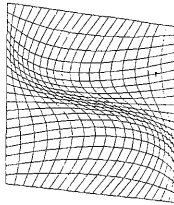


(d) 手法 (c)

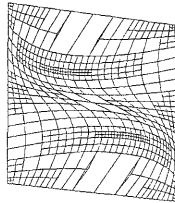
図 6: データ 2 の適用結果



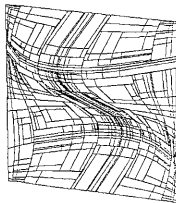
(a) メッシュデータ



(b) 手法 (A)



(c) 手法 (B)



(d) 手法 (c)

	0.1	0.2	0.3
手法 (B) : 葉数	148	112	52
手法 (B) : 深さ	4	4	3
手法 (C) : 葉数	184	64	16
手法 (C) : 深さ	4	3	2

表 1: データ 1 での手法 (B) と手法 (C) の比較

	0.1	0.2	0.3
手法 (B) : 葉数	124	64	16
手法 (B) : 深さ	4	3	2
手法 (C) : 葉数	115	52	1
手法 (C) : 深さ	5	4	0

表 2: データ 2 での手法 (B) と手法 (C) の比較

	0.1	0.2	0.3
手法 (B) : 葉数	544	256	196
手法 (B) : 深さ	6	5	5
手法 (C) : 葉数	688	1	1
手法 (C) : 深さ	7	0	0

表 3: データ 3 での手法 (B) と手法 (C) の比較

以上のことから、*linearity, flatness* を適当に小さくすると、手法 (C) のほうが深く探索しており、より細分化を行なっていることがわかる。

図 7: データ 3 の適用結果

7 おわりに

分割手法 (A),(B),(C) を適用した結果の比較を行ない、(C) では曲率の大きいところでは細かく、曲率の小さいところでは荒く平面近似することがわかった。しかしながら、(C) を適用しても処理の途中で微小な曲率最大点が見落とされることもある。これ防ぐためには、(D) の前処理を行なう必要があるが、前処理のための計算量は膨大となる。この前処理を近似的に簡便に行なう計算手法を開発することが今後の課題である。

参考文献

- [1] Snyder, J M and Barr, A H 'Ray tracing complex models containing surface tessellation' Comput. Graph. Vol 21 No 4 (1987) pp 119-128
- [2] Shantz, M and Lien, Sheue-Ling 'Shading bicubic patches' Comput. Graph. Vol 21 No 4 (1987) pp 189-196
- [3] R.E. Barnhill and S.N. Kersey 'A marching method for parameteric SSI' Computer Aided Geometric Design 7 (1990) pp 257-280
- [4] M Kusters 'Curvature-dependent parameterization of curves and surfaces' Computer Aided Design Vol 23 No 8 (1991) pp 569-578