節点削除による B スプライン曲線・曲面の簡単化

1AD−1

青野雅樹、大渕竜太郎、増田宏
日本アイ・ビー・エム (株) 東京基礎研究所

1 はじめに

近年、大規模データの取り扱い、特に表示の高速化をめざした研究がさかんになされてきた。議論の中心は、大規模ポリゴンデータの簡略化に関するものがほとんどで、様々な簡略化アルゴリズムが発表されてきた。しかしながら、曲面、とくにパラメトリック曲面の簡略化に関しては、筆者らの知る限りいまだに研究対象としてとりあげられていない。

本稿では、パラメトリック曲面として代表的な B スプライン曲面に焦点をあてて、節点削除という従来から知られている手法を、単に不要な節点を省くために用いるのではなく、積極的に曲線や曲面の簡単化に使ってみようというのが目的である。

一般に節点削除は、節点挿入と異なり、形状を不変のまま変換することは特別な条件が備わっていない限りできない。従って、節点削除により曲線や曲面は別の形状で近似される。ただし、この近似度は閾値で制御できるので、これを粗くしたり、細かくすることで、一種の曲面版の LOD も作れるという応用が可能となる。

実際のモデラやレンダラで出力された B スプライン曲面データを用いて節点削除により、どのように曲線や曲面が簡単化できるかを最後に実例をつけて示し、その考察をする。

2 B スプライン曲線・曲面の定義

1 変数パラメータ t による次数 d の B スプライン曲線は

$$\mathbf{r}(t) = \sum_{i=0}^{C-1} \mathbf{b}_i N_i^d(t) \qquad (1)$$

で与えられる [1]。ただし、$\mathbf{b}_i$ は制御点 **(control point)** を表し、C は制御点の数を表し、N は B スプラインの**基底関数 (basis function)** を表す。基底関数 N は、以下のように再帰的に定義される。

$$N_k^d(t) = \frac{t - t_{k-1}}{t_{d+k-1} - t_{k-1}} N_k^{d-1}(t) + \frac{t_{d+k} - t}{t_{d+k} - t_k} N_{k+1}^{d-1}(t) \qquad (2)$$

$$N_k^0(t) = \begin{cases} 1 & (\text{if } t \in [t_{k-1}, t_k)) \\ 0 & (\text{otherwise}). \end{cases} \qquad (3)$$

ここで、$t_k (0 \leq k \leq C + d - 2)$ は節点ベクトル **(knot vector)** と呼ばれる実数値である。節点ベクトルは、通常 $T = t_0, \ldots, t_{L+2d-2}$ のようなシーケンスで表現する。ただし、$C = L + d$ であり、C は制御点の数で d は次数を表す。

2 変数パラメータ u、v による次数 $m \times n$ の B スプライン曲面は、

$$\mathbf{r}(u, v) = \sum_{i=0}^{C_u-1} \sum_{j=0}^{C_v-1} \mathbf{b}_{i,j} N_i^m(u) N_j^n(v) \qquad (4)$$

で与えられる。ただし、節点ベクトルは $U = [u_0, u_1, \ldots, u_{L_u+2m-1}]$ 及び $V = [v_0, v_1, \ldots, v_{L_v+2n-1}]$ で与えられる。また、$\mathbf{b}_{i,j}$ は制御点を表し、C_u と C_v はそれぞれ、u 方向及び v 方向の制御点の数を表し、N は式 (2) 及び (3) で定義した B スプラインの基底関数を表す。

3 B スプライン曲面の節点削除アルゴリズム

節点削除 (knot removal) は、これまで、異なる曲線表現（たとえばベジエ (Bezier) 曲線と冪級数曲線）の間のでの変換手段や、複数の B スプライン曲線や曲面を接続する場合に、接続される隣接する曲線や曲面パッチの次数が一致する節点数を統一し、不要な節点を削除する手段などに用いられてきた。

以下に B スプライン曲線での節点削除例を述べて、節点削除アルゴリズムを簡単に解説する。

下図は、3 次の B スプライン曲線とその制御点を示したものである。もともとの制御点は $\mathbf{b}_0^0, \mathbf{b}_1^0, \mathbf{b}_2^0, \mathbf{b}_3^0, \mathbf{b}_4^0, \mathbf{b}_5^0, \mathbf{b}_6^0$ の 7 個である。節点ベクトルは、$t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8$ の 9 つである。(ここでは [0, 0, 0, 1, 1, 1, 2, 2, 2] とする)。

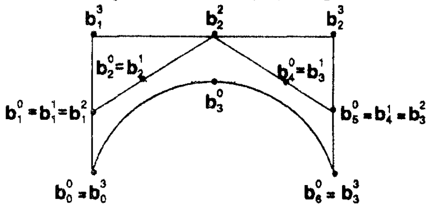


図 1: B スプライン曲線の例

節点ベクトルのパラメータ $t = 1$ で、この曲線が C^1 連続であるための条件を考えると、曲線を [0,1] と [1,2] の 2 つの区間に分けて考え、[0,1] に対応する曲線セグメントの t=1 での 1 次微分値が、[1,2] に対応する曲線セグメントの t=1 での 1 次微分値に一致すればよい。

これを $t_5 = t_3 = t$ とし、更に、$\alpha \equiv (t - t_2)/(t_6 - t_2)$ とおき、式で表すと、

$$\mathbf{b}_3^0 = \alpha \mathbf{b}_4^0 + (1 - \alpha)\mathbf{b}_2^0$$

が得られる。

簡便のため、$\mathbf{b}_4^0 = \mathbf{b}_3^1$ と置換し、$\mathbf{b}_2^0 = \mathbf{b}_2^1$ と置換し、α を α_3 と表現すると、

$$\mathbf{b}_3^0 = \alpha_3 \mathbf{b}_3^1 + (1 - \alpha_3)\mathbf{b}_2^1$$

と表現できる。

同様にして、$t = 1$ で、2 次微分可能性まで考慮すると、

$$\mathbf{b}_2^1 = \alpha_2 \mathbf{b}_2^2 + (1 - \alpha_2)\mathbf{b}_1^2$$
$$\mathbf{b}_3^1 = \alpha_3 \mathbf{b}_3^2 + (1 - \alpha_3)\mathbf{b}_2^2$$

Simplification of B-Spline curves and surfaces by knot removal
Masaki AONO (aono@trl.ibm.co.jp), Ryutarou Ohbuchi, and Hiroshi Masuda
IBM Research, Tokyo Research Laboratory

と表現される。ただし、

$$\alpha_i = \frac{t - t_{i-1}}{t_{d+i} - t_{i-1}} \quad i = 2, 3$$

である。ここで注目すべきは、$\mathbf{b}_2^2$という新しい制御点が登場していることだ。(ちなみに、$\mathbf{b}_1^2 = \mathbf{b}_1^1$で、$\mathbf{b}_3^2 = \mathbf{b}_4^1$である。) 上式から、$\mathbf{b}_2^2 = (\mathbf{b}_2^1 - (1-\alpha_2)\mathbf{b}_1^2)/(\alpha_2)$、下式より $\mathbf{b}_2^2 = (\mathbf{b}_3^1 - \alpha_3\mathbf{b}_3^2)/(1-\alpha_3)$ が得られる。もし、この２つから得られる制御点の位置が、ほぼ一致していれば、節点 $t = 1$ が更に削除できる。

同様にして、$t = 1$ で、３次微分可能性まで考慮すると、

$$\begin{aligned}
\mathbf{b}_1^2 &= \alpha_1\mathbf{b}_1^3 + (1-\alpha_1)\mathbf{b}_0^3 \\
\mathbf{b}_2^2 &= \alpha_2\mathbf{b}_2^3 + (1-\alpha_2)\mathbf{b}_1^3 \\
\mathbf{b}_3^2 &= \alpha_3\mathbf{b}_3^3 + (1-\alpha_3)\mathbf{b}_2^3
\end{aligned}$$

と表現される。ただし、

$$\alpha_i = \frac{t - t_{i-1}}{t_{d+i+1} - t_{i-1}} \quad i = 1, 2, 3$$

である。この場合、未知の制御点は $\mathbf{b}_1^3$と $\mathbf{b}_2^3$の２つである。第一の式より、得られる $\mathbf{b}_1^3$と第三の式から得られる $\mathbf{b}_2^3$を第二の式に代入して左辺と右辺がある閾値以内で等しければ、$t = 1$ で３つとも節点が削除できることになる。ただし、$\mathbf{b}_0^3 = \mathbf{b}_0^0$ で $\mathbf{b}_3^3 = \mathbf{b}_6^0$である。

一般にある t 値で節点を削除するには、上例でわかるように、多重度が m ならば、最高で m 個の等式が得られることがわかる。そこで、これをまとめると、

1. 最初と最後の式から、新しい制御点を計算する。
2. 得られた制御点を２番目と最後から２番目の等式に代入してまた新しい制御点を計算する。
3. 以上を繰り返して等式が１個または２個になるところまで繰り返す。ここで、１個になるのは等式の数が奇数個のときで、２個になるのは偶数個のときである。
   - もし、奇数個ならば、最後の１個の等式の左辺と右辺を比較し、これが閾値以内ならばその節点は削除できる。
   - もし、偶数個ならば、最後の２個になったときにこれらを比較し、これが閾値以内ならばその節点は削除できる。

B スプライン曲面の場合、たとえば u 方向の節点ベクトル $u = u_i$を削除するためには、上述のアルゴリズムを v 方向の C_v個の曲線においてすべて条件をみたしたときに $u = u_i$にて節点削除できる、というのが基本的な B スプライン曲面の節点削除アルゴリズムの概要である [3]。

4 B スプライン曲面の簡単化アルゴリズム

上述した節点削除アルゴリズムは、一般に $m \times n$ 次の B スプライン曲面に対して多重度が u 方向なら m、v 方向なら n の節点に対して最高それぞれ m 回、または n 回節点削除できるというものであった。

われわれは、この節点削除アルゴリズムを土台にして、以下の流れ図で示すような方法で B スプライン曲面の簡単化を試みた。具体的には、

1. まず、与えられた B スプライン曲面のすべての節点においてその多重度を B スプライン曲面の次数に一致させるために、節点挿入を行なう。たとえば双３次ならばすべての節点の多重度を３にする。こうすることでベジエ曲面と等価になり、後処理しやすいからである。
2. 次に任意の閾値を入力させ、節点削除を対象とする（境界を除く）すべての節点について多重度と同じ回数だけ削除可能な節点を削除する。これで簡単化された B スプライン曲面が得られ、しかも出力された B スプライン曲面もベジエ曲面と等価であるという性質を引き継ぐ。

なお、閾値をいろいろ制御することでいろいろな近似度の B スプライン曲面が得られ、曲面の LOD(Level Of Detail) 化することも可能である。

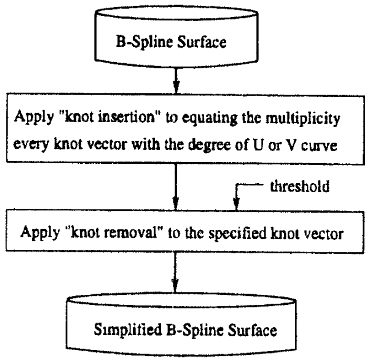
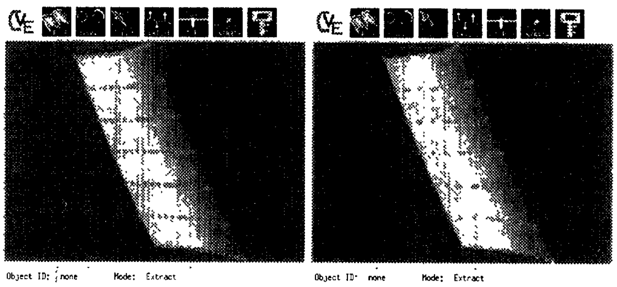


図 2: B スプライン曲面の簡単化処理の流れ

5 テスト結果と考察

下図は、Alias^{TM}でモデリングされた自動車部品の一部でこれを OpenInventor[2] の IndexedNurbsSurface で出力された B スプラインパッチの一例に対して、本手法の簡単化アルゴリズムを適用したものである。もともと、このような簡単そうにみえる部品でも、モデラーから出力された時点では、左側の図に示したように、通常ミシン目のように制御点がびっしりと出力されてしまう。(図において制御点は球状の物体で表現している。) この例では、7x19 個の制御点が元データにあったものを、前述の簡単化アルゴリズムで 7x10 個（約半分）に簡単化したもので、閾値は 0.005 を与えた。このような小さい閾値の場合、見た目には形状の変化はほとんど認識されない。しかしながら、曲面を定義するデータ量は半分近く圧縮することができる。



(a) 7x19 個の制御点 (b) 7x10 個に削減

図 3: B スプライン曲面簡単化の実行例

参考文献

[1] G. Farin, *Curves and Surfaces for CAGD*, 4th Edition, Academic Press, 1996

[2] Open Inventor Architecture Group, *Open Inventor^{TM} C++ Reference Manual* Addison-Wesley Publishing Co., 1994.

[3] W. Tiller, Knot-removal algorithms for NURBS curves and surfaces, *CAD*, Vol. 24, 445-453 (1992)