7Q－6

任意の重み係数を持つ有理 Bézier 曲線・曲面の干渉処理のための再帰分割法

山田 敦
日本アイ・ビー・エム (株)　東京基礎研究所

山口富士夫
早稲田大学　理工学部

1　はじめに

有理 Bézier 曲線・曲面を対象とした干渉処理方法として再帰分割法 [1] は CAD/CAM, CG などの分野で広く利用されている．再帰分割法の処理過程でラフチェックに用いられる従来のバウンディングボックスは，重み係数が全て正の有理 Bézier 曲線・曲面に対してのみ利用可能である．本研究では，従来のバウンディングボックスに代わる図形として同次バウンディングボックスを新たに定義し，それを用いたラフチェックアルゴリズムを提案する．同次バウンディングボックスは任意符号の重み係数を持つ有理 Bézier 曲線・曲面を内部に含む特徴を持つ．そのため与えられた曲線・曲面を全て正の重み係数を持つ曲線・曲面に分割するなどの前処理を行なう必要がなく，一貫した方法で任意の有理 Bézier 曲線・曲面を処理することができる．

2　射影空間で定義した有理 Bézier 曲線・曲面

零ベクトルでない 2 つのベクトル$\boldsymbol{V}=[X\ Y\ Z\ w]$, $\boldsymbol{V}'=[X'\ Y'\ Z'\ w']$ に対して$\boldsymbol{V}=k\boldsymbol{V}'$ $(k\neq 0)$ が成り立つとき，この 2 つのベクトルは同値関係にあるとする．この同値関係により定まる同値類を点と呼び $\langle\!\langle \boldsymbol{V}\rangle\!\rangle$ で表す．全ての同値類の集合を射影空間と呼び P^3 と表す．射影空間はユークリッド空間に無限遠点を付け加えて得られる．すなわちベクトル$\boldsymbol{V}$の w成分が零でないとき，$\langle\!\langle [X\ Y\ w]\rangle\!\rangle$ は次式 (1) で表されるユークリッド座標 $(x\ y\ z)$ を持つユークリッド空間の点となり，一方 w成分が零のとき，その点は無限遠点と呼ばれる．

$$(x\ \ y\ \ z) = (X/w\ \ Y/w\ \ Z/w). \tag{1}$$

射影空間の n 次有理 Bézier 曲線は次式を満たす点の集合 $\langle\!\langle \boldsymbol{P}(s,t)\rangle\!\rangle$ であると定義される．

$$\begin{aligned} \boldsymbol{P}(s,t) &= \sum_{i=0}^{n} HB_i^n(s,t)\boldsymbol{Q}_i, \\ HB_i^n(s,t) &= \binom{n}{i} s^{n-i}t^i,\ (s,\ t\geq 0\ \text{or}\ s,\ t\leq 0), \\ \boldsymbol{Q}_i &= [X_i\ \ Y_i\ \ Z_i\ \ w_i]. \end{aligned} \tag{2}$$

Divide-and-Conquer Algorithm for Rational Bézier Curves and Surfaces with Any Signs of Weights
Atsushi YAMADA
Tokyo Research Laboratory, IBM Japan

制御ベクトル$\boldsymbol{Q}_i$の w成分を重み係数と呼ぶ．同様にして射影空間の図形として有理 Bézier 曲面も定義することが出来る．正の重み係数だけでなく零や負の重み係数を利用することにより，有理 Bézier 曲線は無限遠制御点をもつ曲線や無限遠点を通る曲線をも表現可能となる [2].

3　同次バウンディングボックスの定義

簡単のためにまず 2 次元の同次バウンディングボックスを射影空間の図形として定義する．$k+1$ 個の 3 次元ベクトル$\boldsymbol{V}_0$, $\boldsymbol{V}_1$, $\cdots$, $\boldsymbol{V}_k$を与える．3 つの補助ベクトル$\boldsymbol{V}_K=[1\ 0\ 0]$, $\boldsymbol{V}_L=[0\ 1\ 0]$, $\boldsymbol{V}_M=[0\ 0\ 1]$ を与える．X軸方向の同次バウンディングボックスを，次式を満たす点の集合 $\langle\!\langle \boldsymbol{V}_a\rangle\!\rangle$ であると定義する．

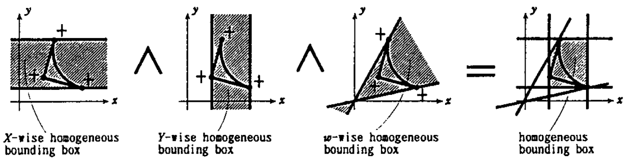
$$\begin{aligned} &\boldsymbol{V}_a = \xi_0\boldsymbol{V}_0 + \xi_1\boldsymbol{V}_1 + \cdots + \xi_k\boldsymbol{V}_k + \xi_K\boldsymbol{V}_K, \\ &(\xi_0,\xi_1,\cdots,\xi_k \geq 0\ \text{or}\ \xi_0,\xi_1,\cdots,\xi_k \leq 0). \end{aligned} \tag{3}$$

X軸方向と言う名前は，式 (3) が X軸方向の単位ベクトル$\boldsymbol{V}_K$により決定されていることを意味している．同様に Y, Z軸方向の同次バウンディングボックスを定義する．2 次元同次バウンディングボックスは，これら 3 種類の各軸方向の同次バウンディングボックスの積集合として定義される (図 2). 与えた曲線の重み係数の符号が全て正のとき，同次バウンディングボックスは従来のバウンディングボックスと似た形状となる (図 2(a)). 一方正負の重み係数が混在する場合，同次バウンディングボックスは外側に広がった図形になる場合がある (図 2(b)). この同次バウンディングボックスは，重み係数の符号に関わらず，必ず曲線・曲面を内部に含むという特徴を持つ [3].

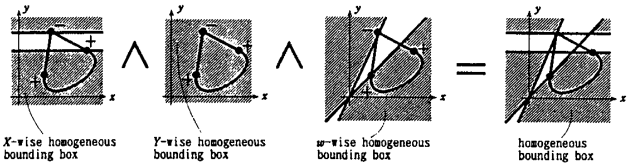
より一般的な 3 次元同次バウンディングボックスを定義する．$k+1$ 個の 4 次元ベクトル$\boldsymbol{V}_0$, $\boldsymbol{V}_1$, $\cdots$, $\boldsymbol{V}_k$及び 4 つの補助ベクトル$\boldsymbol{V}_K=[1\ 0\ 0\ 0]$, $\boldsymbol{V}_L=[0\ 1\ 0\ 0]$, $\boldsymbol{V}_M=[0\ 0\ 1\ 0]$, $\boldsymbol{V}_N=[0\ 0\ 0\ 1]$ に対して，XY軸方向の同次バウンディングボックス $\langle\!\langle \boldsymbol{V}_a\rangle\!\rangle$ は，

$$\begin{aligned} &\boldsymbol{V}_a = \xi_0\boldsymbol{V}_0 + \xi_1\boldsymbol{V}_1 + \cdots + \xi_k\boldsymbol{V}_k + \xi_K\boldsymbol{V}_K + \xi_L\boldsymbol{V}_L, \\ &(\xi_0,\xi_1,\cdots,\xi_k \geq 0\ \text{or}\ \xi_0,\xi_1,\cdots,\xi_k \leq 0). \end{aligned} \tag{4}$$

3 次元同次バウンディングボックスは，6 種類の XY, YZ, ZX, Xw, Yw, Zw 軸方向の同次バウンディングボックスの積集合として定義される．



(a) 重み係数が全て正



(b) 正負の重み係数が混在

図 1:　2 次元同次バウンディングボックス

4　ラフチェックアルゴリズム

3 次元同次バウンディングボックスを使ったラフチェックアルゴリズムを提案する．制御ベクトル$\boldsymbol{Q}_0, \boldsymbol{Q}_1, \boldsymbol{Q}_2, \cdots$で定義される有理 Bézier 曲線 (曲面) と，制御ベクトル$\boldsymbol{Q}_A, \boldsymbol{Q}_B, \boldsymbol{Q}_C, \cdots$で定義される有理 Bézier 曲線 (曲面) とを与える．ラフチェック行列 Mat_{XY},Mat_{YZ}, Mat_{ZX}, Mat_{Xw}, Mat_{Yw}, Mat_{Zw}を計算する．

$$Mat_{XY}=\begin{bmatrix} S_{A0KL} & S_{A1KL} & \cdots \\ S_{B0KL} & S_{B1KL} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, Mat_{YZ}=\begin{bmatrix} S_{A0LM} & S_{A1LM} & \cdots \\ S_{B0LM} & S_{B1LM} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix},$$
$$Mat_{ZX}=\begin{bmatrix} S_{A0MK} & S_{A1MK} & \cdots \\ S_{B0MK} & S_{B1MK} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, Mat_{Xw}=\begin{bmatrix} S_{A0KN} & S_{A1KN} & \cdots \\ S_{B0KN} & S_{B1KN} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad (5)$$
$$Mat_{Yw}=\begin{bmatrix} S_{A0LN} & S_{A1LN} & \cdots \\ S_{B0LN} & S_{B1LN} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, Mat_{Zw}=\begin{bmatrix} S_{A0MN} & S_{A1MN} & \cdots \\ S_{B0MN} & S_{B1MN} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

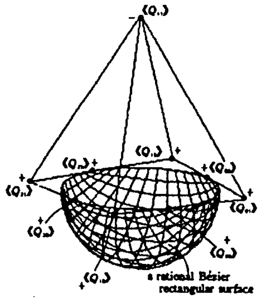
ただし例えば行列要素S_{A0KL}は，ベクトル$\boldsymbol{V}_A$, $\boldsymbol{V}_0$, $\boldsymbol{V}_K$, $\boldsymbol{V}_L$により決定される 4 × 4 行列式を表す．ここで行列 Mat_{XY}の要素が全て正であることを記号 $[Mat_{XY}]_{ij} > 0$ で表すこととすると，条件

$$\begin{array}{lllll} & ([Mat_{XY}]_{ij} > 0) & \vee & ([Mat_{XY}]_{ij} < 0) & \\ \vee & ([Mat_{YZ}]_{ij} > 0) & \vee & ([Mat_{YZ}]_{ij} < 0) & \\ \vee & ([Mat_{ZX}]_{ij} > 0) & \vee & ([Mat_{ZX}]_{ij} < 0) & \\ \vee & ([Mat_{Xw}]_{ij} > 0) & \vee & ([Mat_{Xw}]_{ij} < 0) & (6) \\ \vee & ([Mat_{Yw}]_{ij} > 0) & \vee & ([Mat_{Yw}]_{ij} < 0) & \\ \vee & ([Mat_{Zw}]_{ij} > 0) & \vee & ([Mat_{Zw}]_{ij} < 0) & \end{array}$$

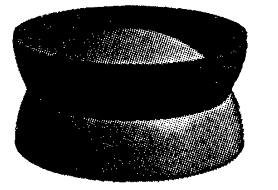
が満たされていれば，与えられた 2 つの曲線 (曲面) は交差しない．一方この条件が満たされていなければ，曲線 (曲面) どうしが交差している可能性がある．

5　曲線・曲面に対する干渉処理結果

以上に述べたラフチェックと曲線・曲面の分割を再帰的に繰り返すことにより有理 Bézier 曲線・曲面どうしの交差を算出することができる．例として無限遠制御点を持ち半球面を表す有理 Bézier4 角形曲面 [2] どうしの交線算出結果を図 2 に示す．このような例に対して従来のバウンディングボックスを使った再帰分割法では交差を求めることはできないが，提案する方法により前処理などを行うことなく交線を求めることができた．



(a) 制御ポリゴン



(b) 交線算出結果

図 2:　有理 Bézier 曲面どうしの交線算出

ある曲面どうしの交線算出問題に対して従来の処理方法との計算時間の比較を表 1 に示す．この表から，総計算時間は両処理方法ともにほぼ同程度であることがわかる．

Table 1　曲面どうしの交差問題を対象とした従来の手法との計算時間の比較

曲面の次数	従来の手法		
	分割 sec[%]	ラフチェック sec[%]	総計 sec
2nd/2nd	1.67 [79.9]	0.197 [9.4]	2.09
3rd/3rd	5.64 [87.7]	0.400 [6.2]	6.43
4th/4th	12.5 [89.1]	0.586 [4.2]	14.0
曲面の次数	提案する手法		
	分割 sec[%]	ラフチェック sec[%]	総計 sec
2nd/2nd	1.29 [67.5]	0.473 [24.8]	1.91
3rd/3rd	3.04 [67.3]	1.16 [25.7]	4.52
4th/4th	12.2 [69.7]	4.32 [24.7]	17.5

6　おわりに

本研究では，同次バウンディングボックスを新たに定義し，それを用いたラフチェックアルゴリズムを提案した．同次バウンディングボックスを用いることにより，任意符号の重み係数を持つの有理 Bézier 曲線・曲面を一貫した方法で処理できることを示した．このような拡張の一方で処理時間は従来のバウンディングボックスを用いた場合と比べてほとんどかわらないことを示した．

参考文献

[1] Houghton EG, et al. (1985) Implementation of a Divide-and-Conquer Method for Intersection of Parametric Surfaces. Computer Aided Geometric Design 2:173–183

[2] Piegl L (1987b) Less Data for Shapes. IEEE Computer Graphics and Applications 7(8):48-50

[3] 山田敦, 有理多項式曲線・曲面を対象とした同次処理の考えに基づく干渉処理, 早稲田大学博士論文, 1995.