I-011

ベジエクリッピング法を用いた曲面と平面との最短距離計算法

Detection of Minimum Distance from Curved Surfaces to Plane by Using Bezier Clipping

西田 友是
Tomoyuki Nishita
広島修道大学/UEI リサーチ

1 まえがき

工業製品のＣＡＤにおいては、物体間の干渉テストが重要で、特に曲面の場合予測が難しい。従来曲面との交差の計算は種々試みられているが、どの程度曲面と平面に隙間（クリアランス）があるかの計算は困難とされている。また、曲面体のアニメーションにおいて平面との衝突判定が重要である。本稿ではベジエ曲面と平面とのギャップ（最短距離）を多角形に近似しないで計算できる方法を提案する。平面とベジエ曲面（u, v の 2 変数で表現）との最短距離は、曲面の接平面の法線が、与えられた平面の法線と一致する極値として求まる。この極値は u, v それぞれで曲面の偏微分が０となる 2 つの曲線（u, v で表現される多項式）の交点として算出できる。

Bezier Clipping 法は曲面をレイトレーシング法で多角形近似しなくて表示できる方法として開発された[1]。この方法を適用することで、与えられた平面との最短距離となる曲面上の極値を算出する方法を提案する。

2 Bezier Clipping 法

本論文で採用する Bezier Clipping 法は、当初曲面のレイトレーシングのために開発された[1]。次の特徴がある。①高次多項式や有理関数に適用できる. ②ロバストである. ③ニュートン法のように初期値を推測する必要がない. ③全ての結果は推定範囲内である. ④必要であれば最小値／最大値の解のみ取得できる. ⑤交差がないことを素早くテストできる. ⑥高次 Bezier 関数/曲面は一次方程式の繰返しで解ける。また、次のものに応用されている。多項式の解、曲線と線分との交点、直線と曲面との交差、曲線と点との最短点の抽出[2]。

レイトレーシングの場合、曲面とレイ（線分）との交点は、その線分を含む 2 平面を考え、これらの平面と曲面の２つの交線曲線同志の交点として求められる。

3 提案法

曲面と平面が交差判定の前に、どの程度距離間隔があるかを検出するのが本稿の目的である。そのため、直線（平面）と曲線（または曲面）との距離を考える。交点は距離が０の点として求まるが、ここでは、曲面上の最短点との距離を求めるのが目標である。曲面を平面からの距離関数に変換し、距離関数(曲面)の極値が最短点または最遠点を与える点となる。極値は距離関数の微分値が０の点として算出できる。

3.1 曲線の極値検出

簡単のため 2 次元平面において、高次の曲線と直線との距離の最小・最大値の検出法を説明する。制御点 P_k $(x_k\ y_k)$ をもつ n 次のベジエ曲線は下記で表せる。

$$P(x,y)=\sum_{k=0}^{n}P_k B_k^n(t) \tag{1}$$

直線と点 P (x, y) の距離は $f(x,y)=ax+by+c$ である。したがって、式(1)の x、y を代入すると

$$f(t)=\sum_{k=0}^{n}f_k B_k^3(t) \tag{2}$$

なお、$f_k=ax_k+by_k+c$ である。微分すると

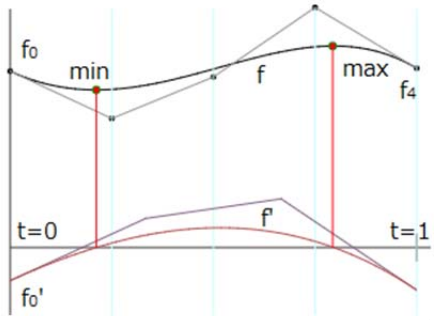

図 1： ４次の場合の距離関数の極値の算出
（下段は微分関数）

$$f'(t)=\frac{df(t)}{dt}=n\sum_{i=0}^{n-1}f'_i\ B_i^{n-1}(t) \tag{3}$$

ただし、$f_i'=f_{i+1}-f_i$ である。 3 次元の場合、平面と点との距離は $F(x,y,z)=ax+by+c$ となるだけで、距離関数に変換した後は同じ手順となる。有理曲線の場合も距離関数式の f が異なるだけである。n 次の曲線の場合、微分した距離関数は(n−1)次のベジエ関数の解を求めることになる。最短点は両端点または曲線上の極値として求まる。よって、Bezier Clipping 法を容易に提供できる。例として、図１に与えられた４次曲線を距離関数に変換したものと、それを微分した３次ベジエ関数を示す(下図参照)。この微分関数が横軸と交差(必ず制御点の凸包内)する点が、極値を与える点である。

3.2 曲面の極値検出

曲面の場合、平面からの曲面上の点の距離を考え、平面との最短距離は 4 隅の頂点を含む境界線上か、曲面内の極値となる。曲線の場合は前節で説明したので、曲面内部での最短点を考える。極値は平面の法線と一致する点（距離関数の微分が０）として求まる。曲面上（u, v のパラメータで表現されるベジエ曲面）の法線は下記外積で求まる。

$$\mathrm{N}=\frac{dF(u,v)}{du}\times\frac{dF(u,v)}{dv} \tag{4}$$

これは、u 成分と v 成分の接線ベクトルに分解して考えられる。すなわち、u に関して偏微分した曲面 $\frac{dF(u,v)}{du}=0$ となり、v に関した編微分した曲面 $\frac{dF(u,v)}{dv}=0$ になる点である。それぞれこれらを満足するのは曲線であり（図２参照）、2 つの曲線（極値曲線と呼ぶ）の交差点が求める極値（点）である。

$$\frac{dF(u,v)}{du} = n\sum_{i=0}^{n-1}\sum_{j=0}^{n} F^u{}_{i,j}\ B_i^n(u)B_j^{n-1}(v) = 0$$

$$\frac{\mathbf{dF(u,v)}}{\mathbf{dv}} = \boldsymbol{n\sum_{i=0}^{n}\sum_{j=0}^{n-1} F^v{}_{i,j}\ B_i^{n-1}(u)B_j^n(v) = 0} \qquad (6)$$

ここで $F_{i,j}^u = F_{i+1,j} - F_{i,j}$ ， $F_{i,j}^v = F_{i,j+1} - F_{i,j}$ である。式(6)の 2 式の連立多項式の解を求める。これは、レイレーシングの際、交点を含む面を凸包の性質を利用し、再分割を反復する方法が適用できる。下記の手順で算出できる（図３参照）。

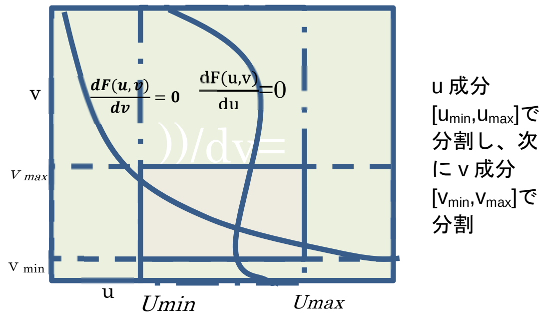

図 2: u-v 平面における2つの極値曲線

１．曲麺の制御点$P_{i,j}$から距離関数の制御点$F_{i,j}$に変換
２．境界曲線（4 辺）の極値を計算；　以下を反復
　距離関数の制御点fi'の範囲が正(または負)なら曲線上には極値はない（頂点が最短点）
　曲線を分割し、解の存在区間を抽出(区間が十分小さいなら、極値とみなし距離を計算)、区間が小さくないなら曲線を再分割
３．距離曲面の制御点$F_{i,j}^u$，$F_{i,j}^v$の範囲を計算　$F_{i,j}^u$，$F_{i,j}^v$の範囲が正（または負）なら極値無なのでスキップ、そうでないなら、曲面再分割(u か v の範囲の狭い方を分割)
　解の存在範囲 u, v 抽出し、曲面を再分割（図３左）、u, v の幅がともに微小ならその点を極値とする。あるいは$F_{i,j}^u$，$F_{i,j}^v$の範囲が共に 0 でも極値であり、その(u, v)での距離を計算(図３右は、解の存在区間を含むように再分割される過程)。なお、極値は 1 点ではないこともあるので、すべての極値の中から最短点を求める。また、極値が点ではなく、線分のすべての点のときもあるが、どの点でも距離は同じであるので、適当な代表点で距離を算出する。

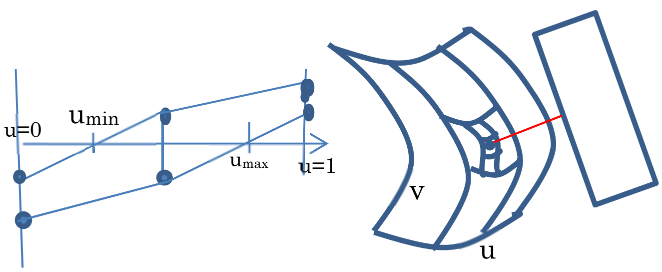

図３：曲面の分割(左；分割区間の判定、右；分割の反復)

4　実験結果

図 4,5 に提案法の計算結果を示す．携帯端末など種々のプラットホームで動作できるように、プログラムはゲームエンジンの enchant.js を利用した[3]。図 4 右は 32 個のベジエ曲面よりなるユタテーポット上の最短・最遠点（指定した線分から）を検出した例である。図 4 右はアニメーションで、衝突判定を容易にするため、判定面にそって傾いた包含箱を求めた例である。図５は 3 次元ベジエ曲面が平面に交差した例で、指定した平面と交点(赤)および最短点(最も侵入した点)を検出したものを示す。

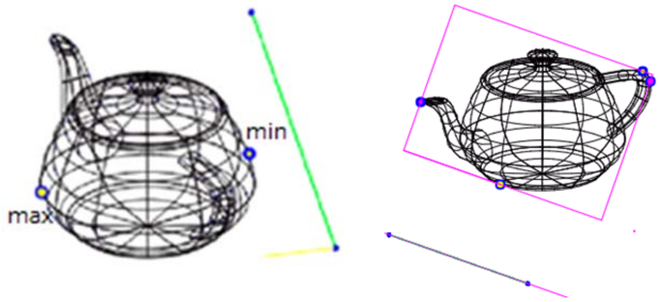

図４：指定した線分の最短最遠点の検出（左）、包含箱の抽出(右)

5　まとめと今後の課題

本稿では，ベジエ曲面と平面とのギャップ（最短距離）を多角形に近似しないで計算できる方法について提案した．

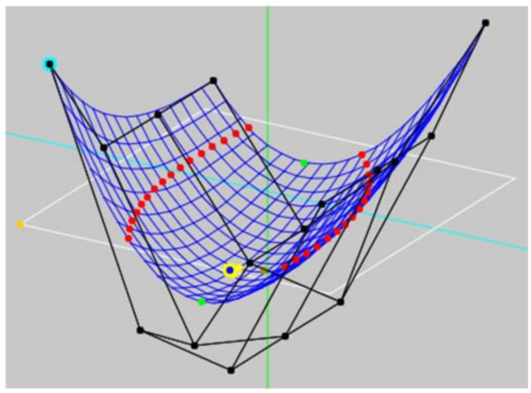

図５：曲面と平面の交差(赤)および貫通点（黄）

与えられた曲面を平面からの距離曲面に変換し、距離曲面の微分曲面の極値(最小値)を Bezier Clipping 法を用いて検出する方法である。曲面の u に関する偏微分した曲面と、v に関して偏微分した曲面が 0 となる曲線を算出し、それらの曲線の交線として求められる。この方法は、形状設計の CAD において物体間のあたり（ギャップ）の検出や、ゲームなどの衝突判定に有効であることを示した。

今後の課題としては，点と曲面あるいは曲面同志の最短距離の検出を行うことが挙げられる．

参考文献

[1] T. Nishita, T. Sederberg, M. Kakimoto, "Ray Tracing Trimmed Rational Surface Patches," Computer Graphics, Vol.24, No.4, 1990-8, pp.337-345.
[2] 西田，松田，高栄，「曲線上の最短点検出を利用した自由形状変形法」，画像電子学会誌，Vol.27，No.4，*pp.287-297，1998-8.*
[3] http://enchantjs.com/ja/