6E-05

ベジエ曲線からの距離関数を用いたレイマーチング法の開発
A development of raymarching algorithm using distance fields from Bezier curves

西田友是
Tomoyuki Nishita
プロメテックＣＧリサーチ/デジタルハリウッド大学
nishita@shudo-u.ac.jp

1. はじめに

コンピュータグラフックスでは、種々の光学的効果を表現できる画素単位で描画するレイトレーシング法が有名で、GPU の進化によりリアルタイムレンダリングが実現できるようになった。類似した方法としてレイマーチング法がある。この方法は視線上を段階的に進み物体との交点を計算するが、この際視線上の計算点（サンプリング点）と物体との最小距離を利用して交点を抽出する。球や多角形などの基本立体の距離計算は容易だが、曲線・曲面の場合の距離計算は困難とされている。本稿では曲線と点との（2 乗）距離をベジエ関数で表現し、効率的に距離計算を行う方法を提案する。特に、曲線のオフセットを考慮した円柱曲線の描画に注目する。n 次ベジエ曲線とすると、点と曲線の距離は2n次のベジエ関数で表現でき、この関数の制御点を利用し、Bezier Clipping 法[1]を適用して最短距離を計算できる方法である。

2. 従来法および基本的な方法

基本的なレイトレーシングでは、レイをまっすぐ伸ばして物体と交差（衝突）しているかどうかを判定する。レイマーチングでは、レイを段階的に伸ばし、各物体との最短距離が十分に小さくなった時点で交差していると判断する。他方、雲、大気散乱の効果を計算するには視線上の粒子の散乱・減衰計算を要す。この方法では、視線上のサンプル点の粒子密度（光学的距離の計算）の計算にもレイマーチング法が用いられる。この場合はサンプル点の進行は、等間隔あるいは密度に依存した間隔となる。こうした散乱光の方法は、著者らは光跡（1987）[2]、大気散乱（1987）、雲（1987）、空の色（1991）の計算など古くから用いている。ただ、本稿では前者の距離関数に依存する方法を論じる。本稿では、レイマーチングの手法のひとつであるスフィアトレーシング（球面レイトレーシング）に属し、段階的にレイを伸ばして物体との交差（進み距離が０に近づく）を認識する。すなわち、最短距離を算出、レイを伸ばす、という作業を繰り返し、サンプル点と物体間の距離が十分に小さくなると物体とレイが衝突したと判断する。一般には、物体からの（符号付き）距離関数（SDF；符号付き距離場）を利用する。球は点からの距離、円柱は線分から距離のように比較的単純に距離を計算できる。本稿では、曲線から一定の太さをもつ円柱曲線（図 1）に注目する。中心線はベジエ曲線で表現できるが,この曲線からのオフセットを考慮した曲面は数式表現できないので、通常のレイトレーシングでの交差判定は難しい。著者は有理ベジエ曲面に関してはレイトレーシング法で曲面との交点を求める方法を発表

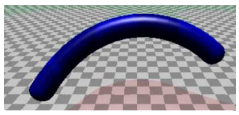

図 1 円柱曲線

[1]（Bezier Clipping 法）しているが,本稿では距離関数を用いる方法論じる。また、筆者らは、点・曲線・曲面間の最短距離（隙間、ギャップ、クリアランス）の計算法についても発表している[3,4].これらを組み合わせた効率的な距離関数計算を提案する。本稿で採用する Bezier Clipping 法は、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、有理ベジエ曲面のレイトレーシング、曲線同士の交差、平面と曲面との距離[2,3]。いずれも高次多項式であり、最短点を与えるパラメータの計算に尽きる.提案法は基本的に曲線・曲面の再分割により、解に収束する方法であり、分割区間は線形計算のみで実現できる。

3. 提案法

レイマーチング法では、物体と点との距離計算が基本であるが、物体が曲線・曲面の場合の計算が困難である。著者らは既に最短距離計算法を発表しているが[2-4]、そのまま適用するとコストがかかるので改良する。有理曲線でも適用できるが、簡単のため 2 次元で非有理曲線と点との距離についてまず説明する。まず,制御点P_k (x_k, y_k)をもつ n 次ベジェ曲線（パラメータ t）を考える。

$$\mathrm{P(t)} = \sum_{k=0}^{n} P_k B_k^n(t) \qquad (1)$$

ここで、$B_k^n(t)$はバーシュタイン多項式で、曲線上の点$\mathrm{P}(t)$、曲線 P への投影関数q、および 2 点 PQ 間の距離の 2 乗距離関数Dを考える。基本的にはベクトルの内積の形式であるが、実際に適用する際はすべてベジエ関数に変換される。

$$D(t) = \big(\mathrm{P}(t) - Q\big) \cdot \big(\mathrm{P}(t) - Q\big) \qquad (2)$$

$$\mathrm{q(t)} = \mathrm{P'(t)} \cdot (\mathrm{P(t)} - \mathrm{Q}) \qquad (3)$$

曲線上の点$\mathrm{P(x(t),y(t))}$の$\mathrm{(x,y)}$座標は式(1)のベジエ関数で定義されているものとする。点Pの$\mathrm{(x,y)}$座標にベジエ曲線の式を代入すると、これら両式ともパラメータtに関するバーシュタイン多項式(有理の場合は分数形式)となる。また、式(3)は点Pの接線とベクトルPQの内積が0になる点である(これは接線に投影した長さに相当)。

(1) 円柱曲線との距離

球(または円)と曲面(曲線)の交差判定は式(3)を利用できる。円や球でも有理曲面の場合でも、$2n$ 次の有理ベジエ関数であり、この関数が解をもつかで交差判定ができる。中心$(x_q,\ y_q)$および半径 R の円とn次ベジエ曲線(パラメータt)の場合、式(2)を修正した関数$e(t,R) = \mathrm{D}(t) - R^2$を利用する。円柱曲線(図1)の場合、曲線からのオフセットがRに等しい。

$$e(t,R) = \mathrm{D}(t) - R^2 = \Big(\sum_{k=0}^{n}(x_k - x_q)B_k^n(t)\Big)^2 + \Big(\sum_{k=0}^{n}(y_k - y_q)B_k^n(t)\Big)^2 - R^2 \qquad (4)$$

整理すると（ベジエ関数間の積の公式利用）

$$e(t,R) = \sum_{k=0}^{2n} d_k\, B_k^{2n}(t) \qquad (5)$$

上式の最小値が最短距離であるが、この高次式を精度よく解く必要があるのは距離が 0 に近い時のみである。制御点の凸包の性質から，(2n+1) 個の制御点の最小値で距離を代表できる。指定距離より小さい時は、精度よく計算するため、この関数を微分した関数$e'(t,R)=\sum_{k=0}^{2n-1}(d_{k+1}-d_k)B_k^{2n-1}(t)$（隣接した制御点の差分で構成）が 0 になる区間を求めこの区間で曲線をクリップし、距離の最小値を計算できる。この分割された区間の関数は、区間が微小（Q の方向に垂直な微小線分になる）なら制御点と関数は誤差がなくなり、分割区間の制御点の最小値で十分である。この制御点の値も点 Q と制御点P_iへのベクトル$(p_i=P_i-Q)$の内積の線形和で簡易に求まる。例えば次数 n=3 なら各制御点は下記のようである。

$$d_k=\{(p_0\cdot p_0),(p_0\cdot p_1),0.1(p_0\cdot p_2)+0.9(p_1\cdot p_1),0.4(p_0\cdot p_3)+0.6(p_1\cdot p_1),0.1(p_1\cdot p_3)+0.9(p_2\cdot p_2),(p_n\cdot p_{n-1}),(p_n\cdot p_n)\} \quad (6)$$

図 2 左はベジエ曲線と点 Q との距離計算を示す。左下は距離関数$e(t,R)$（半径 R の円からの距離）を示し、距離の最小値は関数の制御点d_kの最小値d_{min}（図中の星印）で近似できる、距離関数を微分した関数の制御点の凸包（図左の下部多角形）から、極値の存在区間 t_{min},t_{max}（図中左下）抽出し再帰計算で解に収束する、図 2 右は視点(カメラ位置)とスクリーンを含む 2 次元で、レイマーチングを計算したものである、視線上のサンプル点からの最小距離の円を示している。この例では 3 回目のサンプル点で曲線との交点に達している。右図下部では 2 回のサンプル点で距離関数を示す。1 回目の距離関数では制御点の最小値を利用し、2 回目ではクリップ（図右中の茶色枠）して最近点の精度を上げている。なお、この例は曲線からオフセット曲線との交点計算である（図 2 右上）。

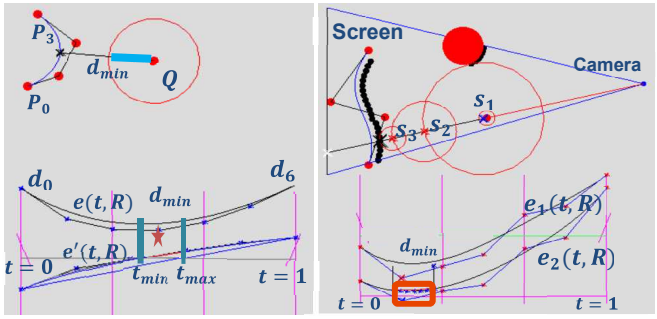

図 2 距離関数およびサンプル点からの最短距離

（2）曲面と点との距離検出

曲面の場合、2 つのパラメータで下記のように定義される。

$$P(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{n}P_{i,j}B_i^n(u)B_j^n(v) \quad (7)$$

距離関数は

$$e(u,v,R)=\sum_{i=0}^{2n}\sum_{j=0}^{2n}d_{i,j}\,B_i^{2n}(u)\quad B_j^{2n}(v) \quad (8)$$

垂直条件は；

$$\frac{\partial P(u,v)}{\partial u}\cdot(P(u,v)-Q)=0\,,\quad \frac{\partial P(u,v)}{\partial v}\cdot(P(u,v)-Q)=0 \quad (9)$$

または、$\frac{\partial e(u,v,R)}{\partial u}=0$, $\frac{\partial e(u,v,R)}{\partial v}=0$から$u,v$成分の解の区間を計算する。筆者の既発表の方法では、n 次の曲面との距離は 2n 次の距離関数の極値を求めるためu,v成分の分割をし、最近点となるu,vを含む微小曲面に収束する。これを各サンプル点で行うのはコストを要すから、次の 2 方法で曲面との距離を抽出した、一つは基準面に対してハイトフィールドとして曲面を表現する。これは基準面でのu,vを求め$H(u,v)$として n 次のベジエ曲面を表現する。サンプル点の基準面からの距離と H の差分がなくなるように進む。他の方法はu,v 2 成分の分割は時間がかかるので一成分について分割して帯状の曲面しておき、他の成分の分割区間は前述の曲線と同様に区間を抽出する。

4. 計算例

基本的には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、WebGl,JavaScript で開発した。図3に円柱曲線と球との論理演算を示す。左は和集合で中央・右は差集合である。図4左はハイトフィールドにより 3 次ベジエ曲面、中央は帯状曲面、右は 3 次ベジエ曲線の円柱曲線である。なお、球、平面、トーラスは従来の距離関数を用いており、影の計算も行っている。

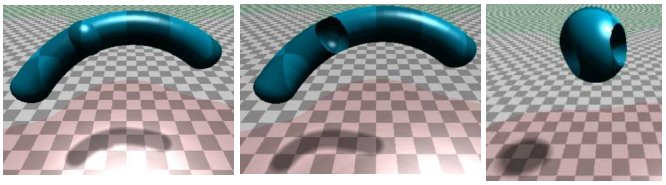

図 3　円柱曲線と球のブール演算(左;加算。右;減算)

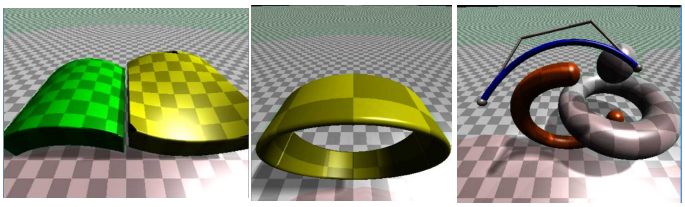

図 4　描画例（左;高さ関数,中;帯状曲面,右;円柱曲線）

5. おわりに

本稿では、ベジェ曲線/曲面と点との距離（ギャップ、最短距離）を計算する関数を提案し、この関数がベジエ関数であることを利用し、制御点および Bezier Clipping 法を用い簡易に距離計算できる方法について提案した。

n 次ベジェ曲面（あるいは曲線）と距離判定関数では、次数が n 次曲線なら、2n 次のベジエ関数である。高次多項式（曲面なら連立方程式）を解くようであるが、幾何学的性質を用いた線形計算のみで実現できる特徴がある。従来法の多くの最短距離計算には曲線・曲面を再帰的に 2 分割する、2 分探索法が用いられており、相当の分割数が必要とされるが、提案法はわずかな分割数で解が得られる。

提案法は距離計算ができるので、CG, CADにおける物体間の間隙（ギャップあるいはクリアランス）の検出や、ゲーム、VRなどの衝突判定に有効である。

参考文献

[1] T. Nishita, T. Sederberg, M. Kakimoto, "Ray Tracing Trimmed Rational Surface Patches," Computer Graphics, Vol.24, No.4, pp.337-345, 1990-8.
[2] T. Nishita and E. Nakamae, "A Shading Model for Atmosphere Scattering Considering Luminous Intensity Distribution of Light Sources," Computer Graphics, Vol.21, No.3,, pp.303-310. 1987-7
[3] 西田,出本,「曲面と多角形との最短距離検出法」 Visual Computing / グラフィクスと CAD 合同シンポジウム, 1 1 , 2017-6.
[4] 西田、「有理ベジエ曲面と球やメタボールの干渉計算」、芸術科学会論文誌 Vol.20, No.4, pp.204-209, 2021-11