

Bezier Clipping およびその計算機支援形状設計への応用

西田 友是*

Bezier Clipping and its Application to Computer Aided Geometry Design

Tomoyuki NISHITA

ABSTRACT

Recently, computer aided geometry design for industrial products has become popular. Geometry design including free form surface is very significant in CAD/CAM system. In geometry designs there are some problems that need to be solved such as curve-curve intersection, surface-surface intersection, and curve-surface intersection. A display method of desired shapes is also very important. To display curved surfaces we have to solve a hidden line problem and a hidden surface problem. Shading models including shadowing, reflection, and refraction are required to get photo-realistic images of designed shapes.

This paper proposes *Bezier clipping method* which can solve all of the problems mentioned above. Bezier clipping is a robust iterative method which can solve functions expressed by polynomials of a high degree. This paper describes the basic idea of Bezier clipping and its application to computer aided geometry design and computer graphics.

1 はじめに

近年コンピュータ・グラフィックスの発展に伴い、自動車や電気製品等の工業製品の設計結果を表示することが多く行われるようになってきた。こうした製品は一般に自由曲面を含んでおり、曲面の処理が重要な役割を占めている。ここでは、曲面の処理に有効な方法を議論する。本稿では、 n 次多項式で表現された関数の解を安定に求める方法(Bezier Clipping法と呼ぶ)およびその解法の図形処理における種々の応用について、今までの研究成果をまとめたものである。

代数方程式の数値解法としては、Newton法がよく用いられている。しかし、Newton法は初期値が必要であり、不安定で、さらに、総ての解が求まる保証がない。形状処理およびコンピュータ・グラフィックスの分野において、曲線同士の交点等種々の問題に対して、従来Newtonが広く用いられている。しかし、前述のように

Newton法は安定でないことから、近似解を求めたり、安定に解を求めるために他の方法との組合せが必要であった。本稿では、多項式、曲線、および曲面はBezier形式の曲線(または曲面)に変換されてるものと考え、Bezier曲線の性質を利用することにより、安定に解(例えば、交点)を求める方法を提案する。この方法は、解の存在区間を反復的に減少させて、解に収束させる方法である。なお、区間を収束させることから、区間Newton法と言える。

形状設計において曲面同士の交差、曲面の表示等、種々の問題を解決する必要がある。ここで議論するBezier Clipping法は基本的問題からコンピュータグラフィックスにおける種々の問題まで広範囲に適用できる。

図形処理における基本的問題に対して、Bezier Clippingは次のものに適用できる。(a)曲線と曲線との交点、(b)直線と曲面との交点、(c)曲線と曲面との交

点、^{3),6)}(d)曲面と曲面との貫通曲線³⁾、(e)閉曲線に対する点の内外判定²⁾、(f)閉曲線の走査変換(領域の塗りつぶし)⁴⁾。

曲面を含む3次元物体の表示の問題は前述の基本的な問題を組み合わせることによって解決できるものが多い。曲面を含む3次元物体の表示に関して、著者がBezier Clipping法を適用したものは、次のようである。

- (1) 隠線消去(投影面での曲線同士の交差、シルエットの抽出、等高線の抽出を含む)⁵⁾
- (2) 隠面消去(レイトレーシング法およびスキャンライン法)⁶⁾
- (3) 影、反射、屈折を伴う陰影表示(レイトレーシング法)²⁾
- (4) 種々の光源(線光源、円柱光源、曲面光源)⁷⁾により照射された曲面の表示
- (5) 相互反射を考慮した曲面の表示⁸⁾

以上のように、Bezier Clippingは計算機支援形状設計およびコンピュータグラフィクスにおいてかなり広範囲に適用できる。曲面の表示は、曲面形状の評価、照明シミュレーション等に関して有用である。

2 Bezier Clipping法

ここで議論するBezier Clippingは、基本的にはn次多項式で表現される関数の解を求める方法である。代数方程式の解を求める方法としては、Newton法が広く使われている。しかし、このNewton法には次の問題がある。すなわち、初期値が必要で、収束しないことがある。また、総ての解が求まるとは限らない。それに対し、提案するBezier Clipping法は安定に総ての解を求めることができる。

次式のようなn次多項式で表わされる関数の解を求めることを考える。

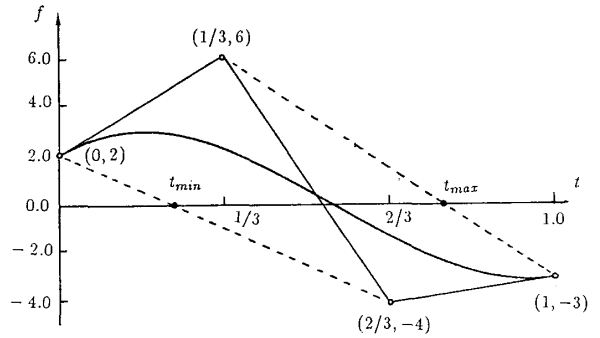
$$a^n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$$

n次多項式はn次Bezier曲線に変換できる。Bezier曲線は一般にパラメータt(0≤t≤1)を用いて表現されるから、変換後xをtに置き換えて考える。Bezier曲線に変換した後の関数をfとすると、fは一般に次式で表現できる。

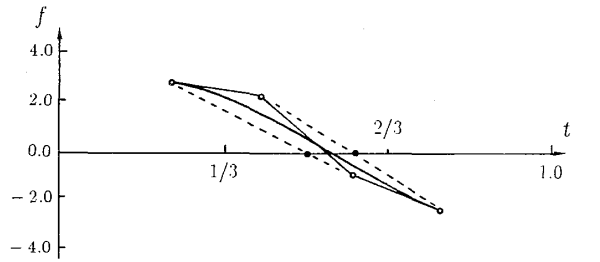
$$f(t) = \sum_{i=0}^n f_i B_i^n(t) \tag{1}$$

ここで、f_iはBezier曲線の制御点で、B_iⁿはBernsteinの基底関数であり次式で表わされる。

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$



(a) original

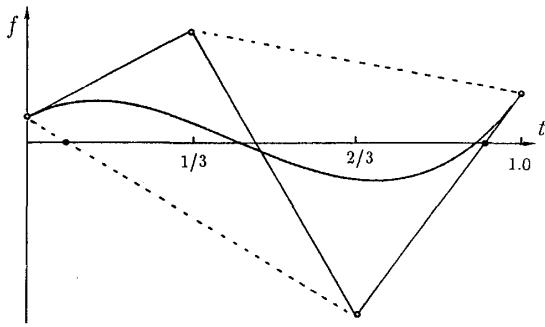


(b) after clipping

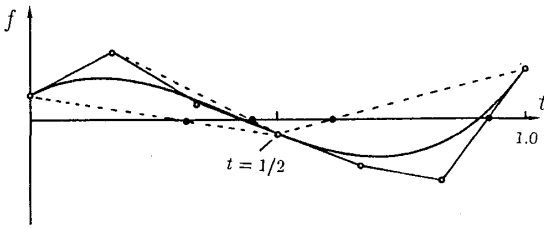
Fig.1 Bezier curve converted from polynomial. (example of 31x³-42x²+24x+2=0 (0≤x≤1))

図1に多項式をBezier関数で表現した例を示す。図のように、Bezier関数には、凸包の性質がある。すなわち、曲線は必ずその制御点(i/n, f_i)(i=0, ..., n)の凸包(制御点を包含する最小の凸多角形)に包含される。関数fとt軸との交点は、fの凸包とt軸との交差区間に存在する。凸包とt軸との交点をt_{min}, t_{max}とすると、必ず解は区間[t_{min}, t_{max}]に存在するから、それ以外の区間の曲線を切り捨てることができる。切り捨てた後は短い曲線が得られる(図(b)参照)。クリップされて短くなった曲線の制御点を利用して、再度解の区間を求めることができる。この処理を繰り返すと、曲線は直線に近ずき、解の区間は小さくなる。このパラメータ区間が指定した許容誤差(tol)より小さくなれば、収束したとみなす。図1の例では、tol=10⁻³の場合には、僅か3回で収束した。なお、最終ステップでは分割は不要である。各ステップでの解の区間幅t_{max}-t_{min}は、0.5555, 0.0083, 0.0003であり、加速度的に収束しているのが判る。

f_i(i=0, ..., n)の値が総て正、または総て負の場合は解がないことは自明である。このように、幾何学的に即座に解の有無を判定できる。区間があまり減少しないことがある。この場合は、解が複数ある場合であり、曲線を2分割して、それぞれに対して反復処理を行う(図2参照)。この方法により総ての解を求めることができる。なお、1回の反復計算で2回の分割(t_{min}とt_{max}で分割)



(a) original



(b) subdivision at midpoint of curve

Fig.2 Subdivision of curve having multiple solutions.

が必要であるが、この分割には、よく知られたde Casteljouの方法を用いる。

次に、多項式が有理関数の場合を考える（すなわち、分数で表現される多項式）。この場合、関数 f は次式で表現される。

$$f(t) = \frac{\sum_{i=0}^n w_i f_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \quad (2)$$

ここで、 w_i は制御点の重みである。この重みが大きくなると曲線が制御点に近づく性質がある。有理関数の場合での制御点の位置は非有理の場合（すなわち $w_i=1$ ）と全く同じである。したがって、非有理の場合と同じように、制御点 $(i/n, f_i)$ ($i=0, \dots, n$) の凸包を利用して解の区間を求めることができる。ただし、曲線を分割する際、有理曲線であるので f のみでなく w の成分についても分割する必要がある。

式(2)の分母は常に正なので、分子が0となればよい。したがって、

$$f'_i = w_i f_i \quad (i=0, \dots, n)$$

とすると、式(2)は

$$f'(t) = \frac{\sum_{i=0}^n f'_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} = 0 \quad (3)$$

を解くことに等しい。この場合、式(3)は式(1)と同じ形式となり、新しい関数 f' を分割するのみで解に収束する。

3 交差問題へのBezier Clippingの適用

図形処理において、2次元上での直線と曲線、曲線同士との交差問題、あるいは3次元上での曲線と曲面との交差等、種々の交差問題がある。これらの問題は、式(1)または(2)の形式に変形することができるので、前述のBezier Clippingを利用して解くことができる。

3.1 直線と曲線との交差

有理曲線の場合、円、双曲線、楕円、放物線等も表現できる。したがって、ここでは、有理曲線を考える。

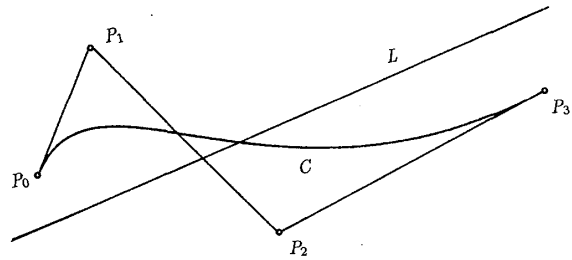


Fig.3 Intersection test between Bezier curve and line.

図3の例をもとに、 n 次Bezier曲線 C と直線 L との交点計算のためのアルゴリズムを説明する（図は $n=3$ の場合）。曲線 C 上の任意の点の座標 (x, y) は変数 t を用いて次のように表わされる。

$$\begin{aligned} x(t) &= \frac{\sum_{i=0}^n w_i x_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \\ y(t) &= \frac{\sum_{i=0}^n w_i y_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \end{aligned} \quad (4)$$

ここで、 $P_i(x_i, y_i)$ ($i=0, 1, \dots, n$) はBezier曲線の制御点の座標で、 w_i は重みである。一方、直線 L からの任意の点 (x, y) までの距離を表す関数を $d(x, y)$ とすると、 d は次式で与えられる。

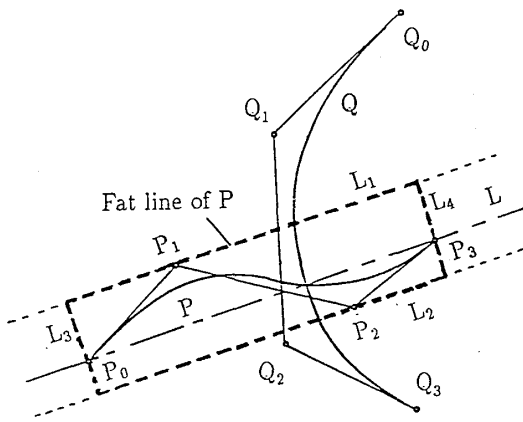
$$d(x, y) = ax + by + c \quad (5)$$

ここで、 (a, b) は L の単位法線ベクトル ($a^2 + b^2 = 1$) である。曲線 C 上の点と直線 L との距離 d は、式(5)に式(4)を代入することにより求まる (t の関数となる)。したがって、

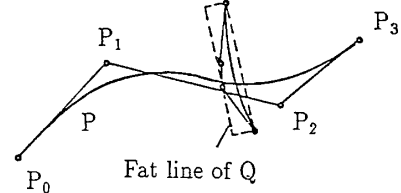
$$d(t) = \frac{\sum_{i=0}^n w_i d_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \quad (6)$$

ここで、 $d_i = ax_i + by_i + c$ 。

式(6)の d は制御点 $(i/n, d_i)$ ($i=0, \dots, n$) よりなる有理Bezier曲線となる。すなわちノンパラメトリックBezier曲線となる。したがって、式(6)の曲線を距離関



(a) Clipping Q by the fat line of P.



(b) Clipping P by the fat line of Q.

Fig.4 Calculation of the intersection point by the fat line method.

数と呼ぶことにする(図3の例ではdは図1のfと全く同一となる)。曲線C上の距離が0の点が交点である。したがって、 $d(t)=0$ を満たすtが交点を与える。なお、 $d'_i=d_i w_i$ とすると、式(6)は式(3)と同じ形式であるので、交点に相当するtの解はBezier clipping法を用いて求めることができる。また、求めたtから式(4)を用いて交点でのx, y座標を算出できる。

3.2 曲線と曲線との交差問題¹⁾

制御点 P_i よりなる曲線 $P(s)$ と制御点 Q_i からなる曲線 $Q(t)$ を考える。曲線を囲むfat lineを利用して交差判定を行う。fat lineとは図4に示すように、直線 L_1, L_2, L_3, L_4 に囲まれた長方形の領域である。すなわち、曲線を幅を持った直線とみなす。まずPのfat lineによって、fat lineと重なる曲線Qの区間をBezier Clippingにより求め、重ならないQの区間を切り捨てると、Qは短くなり直線に近づく(図(b)参照)。この切捨てる範囲は次のように求める。

曲線Pの両端点 P_0, P_n を通過する直線Lを考える。Lから任意の点(x, y)までの距離は、式(5)の $d(x, y)$ で表わされる。Fat lineの境界の2直線 L_1, L_2 内の領域は次式を満たす。

$$d_{\max} \geq d(x, y) \geq d_{\min} \quad (7)$$

ここで、 d_{\max}, d_{\min} は曲線Pの制御点 P_i の座標 (x_i, y_i) より次式によって求まる。

$$\begin{aligned} d_{\max} &= \max\{d_0, d_1, \dots, d_n\}, \\ d_{\min} &= \min\{d_0, d_1, \dots, d_n\}, \\ d_i &= d(x_i, y_i) \end{aligned}$$

一方、曲線 $Q(t)$ は式(4)で表される。曲線Qのx, y座標

を式(5)に代入すると、dは式(6)となる。したがって、式(6)の $d(t)$ が式(7)を満たすtの範囲を求めればよい。これを満たすtの範囲もBezier Clipping法により求めることができる。

図5は図4に対する距離関数d(3次Bezier曲線の場合)を示す。図のように、dの凸包性から式(7)を満たすtの範囲 t_{\min}, t_{\max} が求まる。したがって、この区間以外の曲線Qを切り捨てる。次に、曲線QのFat lineを求め、Pをクリップする。この操作を反復すると、交点に収束する。なお、複数の交点の場合については説明を省略するが、これについても安定に求められる。

幾つかの例について実験した結果、曲線同士の交差判定に効率が良いとされてきた陰関数化する方法¹⁾に比べ、4次以上の曲線に対して特に有効であった。

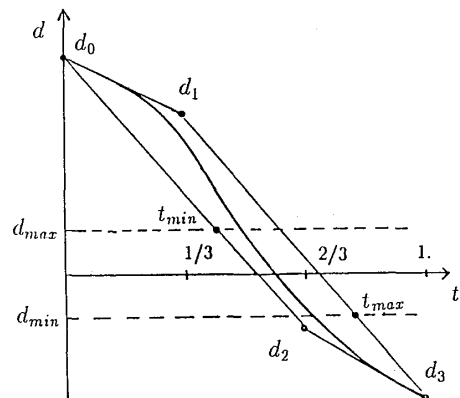


Fig.5 Distance function for P in Fig.4.

3.3 平面と曲線との交差

3次元空間における平面とBezier曲線との交差判定について説明する。平面からある点 (X, Y, Z) までの距離は次の関数で求まる（本稿では、3次元座標は大文字によって表現するものとする）。

$$d(X, Y, Z) = AX + BY + CZ + D \quad (8)$$

ここで、 (A, B, C) は平面の単位法線ベクトルである。n次有理Bezier曲線は次式で与えられる。

$$\begin{aligned} X(t) &= \frac{\sum_{i=0}^n w_i X_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \\ Y(t) &= \frac{\sum_{i=0}^n w_i Y_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \\ Z(t) &= \frac{\sum_{i=0}^n w_i Z_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \end{aligned} \quad (9)$$

この式の X, Y, Z を式(8)の距離関数に代入すると、次式が得られる。

$$d(t) = \frac{\sum_{i=0}^n w_i d_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \quad (10)$$

ここで、 $d_i = AX_i + BY_i + CZ_i + D$ である。この場合も式(2)と同じ形式となり、交点計算にBezier Clipping法が適用できる。平面と曲面との交差判定については、4で述べる。

3.4 曲線と曲面との交差問題³⁾

3次元上における有理Bezier曲線と有理曲面との交差判定について考える。前述のfat lineの考え方を発展させ、曲線を囲む3次元のfat lineと、曲面を囲むfat faceを利用する（図6参照）。n次有理Bezier曲面は (u, v) パラメータを用いて次式で表現される。

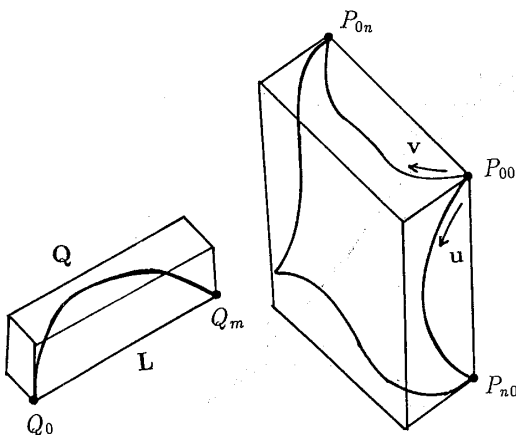


Fig.6 Intersection test between curve and surface.

$$\begin{aligned} X(u, v) &= \frac{\sum_{i=0}^n \sum_{j=0}^n W_{ij} X_{ij} B_i^n(u) B_j^n(v)}{\sum_{i=0}^n \sum_{j=0}^n W_{ij} B_i^n(u) B_j^n(v)} \\ Y(u, v) &= \frac{\sum_{i=0}^n \sum_{j=0}^n W_{ij} Y_{ij} B_i^n(u) B_j^n(v)}{\sum_{i=0}^n \sum_{j=0}^n W_{ij} B_i^n(u) B_j^n(v)} \\ Z(u, v) &= \frac{\sum_{i=0}^n \sum_{j=0}^n W_{ij} Z_{ij} B_i^n(u) B_j^n(v)}{\sum_{i=0}^n \sum_{j=0}^n W_{ij} B_i^n(u) B_j^n(v)} \end{aligned} \quad (11)$$

ここで、 (X_{ij}, Y_{ij}, Z_{ij}) は制御点の座標で、 W_{ij} は重みである。

Bezier曲面の4隅の頂点のうち3点（例えば P_{00}, P_{n0}, P_{0n} ）を通過する面を F とすると、Fat faceは次の6平面に囲まれる平行6面体である。すなわち、 F に平行で、すべての制御点を包含する2平面、ベクトル $P_{00}P_{0n}$ と面 F の法線との外積ベクトルと等しい法線をもつ面で総ての制御点を囲む2平面、および $P_{00}P_{n0}$ と面 F の法線との外積ベクトルと等しい法線をもつ面で総ての制御点を囲む2平面である。一方、2端点 Q_0, Q_m を通過する直線を L とすると、fat lineはベクトル Q_0, Q_m と $P_{00}P_{0n}$ との外積ベクトルと等しい法線をもつ平面で曲線の総ての制御点を包含する2平面、ベクトル Q_0, Q_m と $P_{00}P_{n0}$ との外積ベクトルと等しい法線をもつ平面で曲線の総ての制御点を包含する2平面、およびベクトル Q_0, Q_m と等しい法線をもつ平面で曲線の総ての制御点を包含する2平面。

まず、曲面 P のfat faceによって曲線 Q をクリップし、次にクリップされた曲線のfat lineによって曲面をクリップする。この操作を反復することにより、fat faceおよびfat lineは点に近づく。この処理において、クリップすべき領域はBezier Clipping法で抽出できる。

1次Bezier曲線は直線に等しいから、直線と曲面との交差判定も前述の方法で可能であるが、より効率の良い方法を5.2で説明する。

3.5 曲面と曲面との貫通曲線³⁾

曲面同士の交線を求める問題は計算機支援設計において重要な問題の1つである。例えば、双3次曲面同士の交線は324次の曲線となるから、ここでは近似解を求める。交線を求める方法として、まず1つの交点（例えば境界曲線と曲面との交点）から始めて、マーチング法によって交線を追跡する方法がある。この場合、交線は折れ線となり、精度がよくない。ここでは、直接2つの曲線（一方の曲面上に存在する交線と、他方の曲面上に存在する交線）を精度よく求める方法を提案する。この方法はまず交線の2端点（一般に、一方の曲面の境界線と他方の曲面との交点）を求める。これには3.4で述べた

交差判定法を用いる。この方法では2端点において2つの曲線が C^k 接続 (k 次の微分値まで連続) である曲線が得られる (詳細は文献3参照)。例えば、 C^1 接続の場合、得られた曲線はパラメータ空間において3次Bezier曲線となる。これらの曲線をそれぞれの曲面にマッピングすると2つの18次のBezier曲線となる。2つのBezier曲線同士の制御点の距離がある許容値より大きければ、曲線を細分割することにより必要な精度の曲線が得られる。

4 平面曲線の処理

最近、グラフィクエディターが多く開発されている。これらは主として2次元図形を扱っており、曲線、多角形などの基本図形を表示するといった簡単なものの他に、図形の塗りつぶし、閉曲線に対する点の内外判定、文字フォントの表示などが含まれている。これらへのBezier Clippingの適用を議論する。

4.1 閉曲線と点の内外判定

いくつかのBezier曲線により構成された閉曲線に対して、ある点はその曲線の内部か外部かの判定について説明する。多角形の場合、内外判定の方法はすでにいくつか開発されている。その中で、判定すべき点を始点とする半直線と多角形の交点の数が、奇数か偶数かにより判定する方法がある。この方法を曲線の場合にも応用する。

直線と曲線との交点は前述の3.1で述べた方法により求まるから、この交点数をカウントすればよい。しかし、ここでは、精密な交点の値が必要なのではなく、単に個数のみが必要である。そのため、次の方法を用いる。

図7に示すように、ある点Sについて判定する (図の円は4つのBezier曲線で構成されている)。点Sを端点とする半直線Rを考える。いずれの方向にRを設定して良いが、ここではx軸方向に考える。点Sを通過する垂

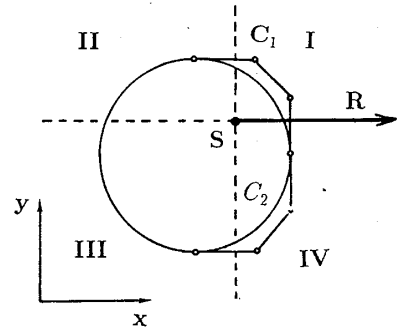


Fig.7 Classification of point.

直線、水平線によりx y平面は4分割される。それぞれ図のようにI、II、III、IVとする (点Sを原点にするように平行移動すると象限と一致)。閉曲線はいくつかのBezier曲線で構成されているとすると、次の手順によってある曲線Cと半直線Rとの交点数を判定する。

- 1) もし、Cのすべての制御点がRの後方 (II、IIIに存在) か、Rに対して同じ側に存在する (例えば図の C_2) なら交点はない。
- 2) Cの制御点がRの前方 (I、IV) であつ、1部の制御点がIとIVに存在するなら、必ず交点を持つ。もし、曲線Cの両端の制御点と同じ側なら、交点は偶数、そうでないなら奇数 (図の C_1)。
- 3) それ以外なら、Bezier Clipping法により曲線Cを直線Rによりクリップし (短い曲線となる)、ステップ1)へ戻る。

4.2 Bezier曲線で囲まれた曲線の走査変換⁴⁾

閉曲線を塗りつぶす方法としては、次の方法がある。(a)スクリーン上の各画素について、その画素が閉曲線の内部かどうかを判定する方法。(b)スクリーンに相当するフレームバッファを準備し、まず輪郭を表示し、その内部を塗りつぶす方法。(c)閉曲線を上から下への走

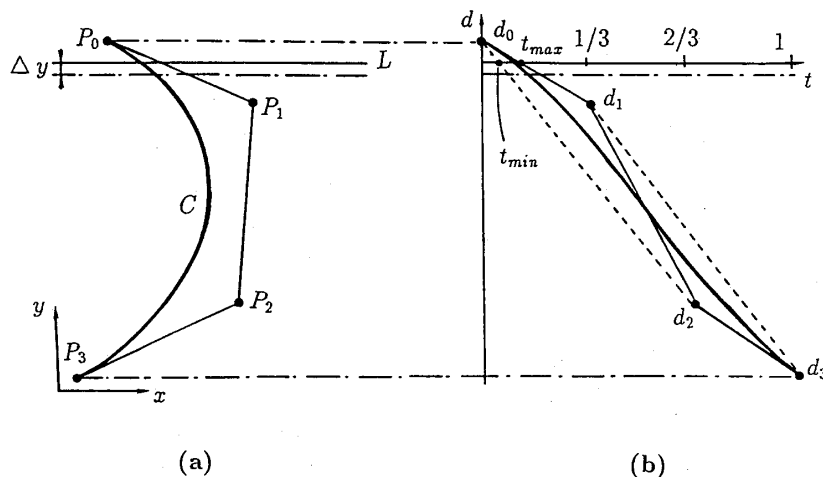


Fig.8 Intersection test between Bezier curve and scanline.

査し、走査線と閉曲線との交点を求め、交点間を塗りつぶす方法。曲線の場合(b)の方法がよく用いられるが、メモリおよびエリアシング（境界部が階段状のギザギザを生じる）の点から、ここでは(c)の方法を用いる。

走査線は直線であるから、走査線と曲線との交点は、3.1で述べた方法で求めることができるが、走査線間のコヒーレンスを利用することにより、より高速化することができる。

走査線 L は $-\Delta y$ （一般に $\Delta y=1$ ）ずつ移動するものとする（図8(a)参照）。走査線 L が最初に曲線 C と交差したときの交点における曲線のパラメータ t は0に近い値をもつ（すなわち交点は曲線 C 上の始点 P_0 の近く）。距離曲線の制御点 d_i により構成される凸包と走査線との交差区間 $[t_{\min}, t_{\max}]$ は殆どの場合狭い（図8(b)参照）。したがって、反復回数はかなり少なくても解が求まる。解 t_{\min} において距離曲線を分割し、 $[0, t_{\min}]$ の区間を切り捨て、 $[t_{\min}, 1]$ の側の制御点を新たな制御点とする。このようにすることによって、次の1段下の走査線（図中の水平な一点鎖線のように t 軸をシフト）との交点判定時にも新たな曲線の $t=0$ 付近に解は存在するから、同様に少ない反復数により解が求まる。なお、 t_{\min} における分割は、1つ前のものとの差が小さい時は、分割しておかなくてもよい。なお、この分割は y 成分についてのみ行えばよく、 x 成分については分割する必要はない。

図9(a)に3次Bezier曲線で表現された閉曲線を示す。また、(b)にアウトラインフォントへの応用例を示す。このように拡大文字についても高品位である。



(a)



(b)

Fig.9 Examples of scan conversion.

5 曲面を含む3次元物体の表示

3次元物体を表示する場合、線画で表示する方法と陰影を付けた面画によって表示する方法がある。前者では、隠れて見えない曲線を消去して表示する隠線消去問題を解決する必要がある。後者では、見えない面を消去して表示する隠面消去問題がある。¹⁰⁾後者では、さらに、光源の種類に応じた輝度計算、影の処理、反射・屈折光の処理といった問題も解決する必要がある。物体が曲面の場合、これらの問題は特に難しく、従来多角形で近似する方法が用いられている。ここでは、多角形に近似しないで精度よくこれらの問題を解決する方法を提案する。3次元物体を表示するには、曲面を透視投影した後、隠線消去または隠面消去する。

5.1 3次元物体の隠線消去⁵⁾

従来、曲面は多角形近似されて表示されていた。最近Elberらは曲面のまま表示する方法を開発した。¹²⁾しかし、効率が良いとは言えない。また、貫通した曲面を処理できない。本稿では、より効率の良い方法を提案する。提案法は、曲面上の図柄（曲線で表現できる曲面上の種々のパターン）や等高線の表示も可能な方法である。なお、曲面同士が貫通した場合も処理できる方法である。

隠線消去問題における解決すべき点には次のものがある。(a)与えられた視点から曲面を見た際の輪郭（シルエット）の抽出。(b)曲線同士の交差判定、(c)貫通点および干渉曲線の抽出。(d)曲線上の点の可視性の判定。曲線の可視性は、曲線と曲面の境界での交点、曲線とシルエットとの交点、および曲線と曲面との貫通点で変化する。したがって、上記の(a)、(b)、(c)の交点を求めればよい。また、交点がない場合で、曲面に曲線が完全に投影面上で包含されている場合、可視性を判定する必要がある（上の(d)）。これらのすべてにBezier Clippingが適用できる。紙面の都合上(a)については省略する。

(1) 曲線同士の交差判定

スクリーン上に投影された曲面は有理Bezier曲面であるので（付録参照）、その境界線および等パラメータ曲線は有理Bezier曲線である。したがって、有理曲線同士の交差判定について考えればよい。すなわち、前述3.2のFat line法を用いた交差判定法が使える。曲面上の図柄曲線については、パラメータ空間においてBezier曲線で表現するものとする。曲面が双3次、図柄曲線が3次の場合、投影後の図柄は有理18次Bezier曲線となる。この様な高次の曲線に対してfat line法は有効である。貫通曲線、等高線もパラメータ空間でBezier曲線として求められるが、同様に投影面では高次Bezier曲線となる。

(2) 貫通点および貫通曲線の抽出

曲線と曲面との貫通点は、前述の3.4の方法を利用

きる。また、貫通曲線については、3.5の方法で求めることができる。

(3) 可視区間の判定

曲線と曲面の境界との交点がない場合には、その曲線上の1点が可視かどうかを判定する必要がある。これは、視点とその点を結ぶ線分と曲面が交差するかを判定すればよい。これは、後述の5.2.1の方法を利用できる。

可視区間の抽出は、曲線の始点での不可視数(何枚の曲面にその点が隠されるかを示す)を求めておき、交点を通過する毎に±1を加え(曲面に進入時1、退出時-1)、不可視数0の区間をプロットする。

5.2 曲面の隠面消去

多面体に対する、隠面消去法として、Zバッファ法、レイトレーシング法、およびスキャンライン法が開発されている。なお、前2者は画素単位に可視面を判定する方法であり、後者は走査線単位に可視面を判定する。ここでは、レイトレーシング法とスキャンライン法とに属す曲面の隠面消去法を提案する。スキャンライン法は処理速度が速い特徴があり、レイトレーシング法は、反射屈折などの種々の光学的特性を取り扱うことができる。

5.2.1 レイトレーシング法による陰影表示²⁾

レイトレーシング法は、スクリーン上の各画素と視点とを結ぶ直線、すなわち視線(レイ)と曲面との交点を求め、最も手前の面を表示する方法である。もし、反射があるなら反射方向への直線に交差する曲面を求める。影については、曲面上の計算点と光源との間に他の曲面があるかどうかを調べる。いずれにしても曲面と直線との交差判定が基本となる。

(1) レイと曲面との交差判定

曲面を多角形近似しないでレイトレーシングする方法を、最初Kajiyaが開発した。彼は、レイを2平面の交線

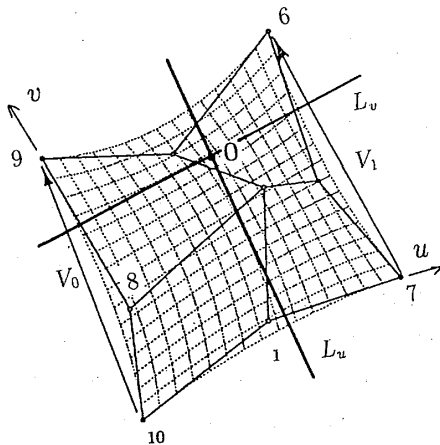


Fig.10 Projected surface.

として定義した。この方法では、双3次曲面の場合18次式を解く必要がある。本稿ではより効率のよい方法を提案する。

視点を原点とした視点座標系において曲面を定義するものとし、レイがZ軸となるように曲面を回転する。Z軸(すなわちレイと一致)を通過する2つの平面をF1、F2とすると、これらの平面式は次式となる。

$$d^k(X,Y) = a^k X + b^k Y = 0 \quad (k=1,2) \quad (12)$$

(a^k, b^k, 0)は面の単位法線ベクトルである。平面と曲面との交点は、平面式(12)に式(11)の(X,Y,Z)を代入し、分母を消去すると、次式が得られる。

$$d^k(u,v) = \sum_{i=0}^n \sum_{j=0}^n d_{ij}^k B_i^n(u) B_j^n(v) = 0, k=1,2 \quad (13)$$

ここで、

$$d_{ij}^k = w_{ij}(a^k X_{ij} + b^k Y_{ij}).$$

d^k_{ij}は平面から制御点への距離に比例している。上式ではz成分は含まれていない。したがって、x y平面に投影したものとして考えることができる。図10は、投影された曲面Pの(例は2次のBezier曲面)である。

図10に示すように、原点を通過する2直線Lu, Lvからの距離が共に0となる点が交点である。まず、ベクトル(V₀+V₁)に平行で原点を通過する直線Luからの距離dを考える。Luの法線を(a,b)とすると、dは式(13)により求まる。

v軸方向から見たd(u,v)の側面図を図11に示す。この例では、u_{min}とu_{max}においてu軸とdの凸包が交差している。u < u_{min}、u > u_{max}の範囲ではd(u,v) ≠ 0

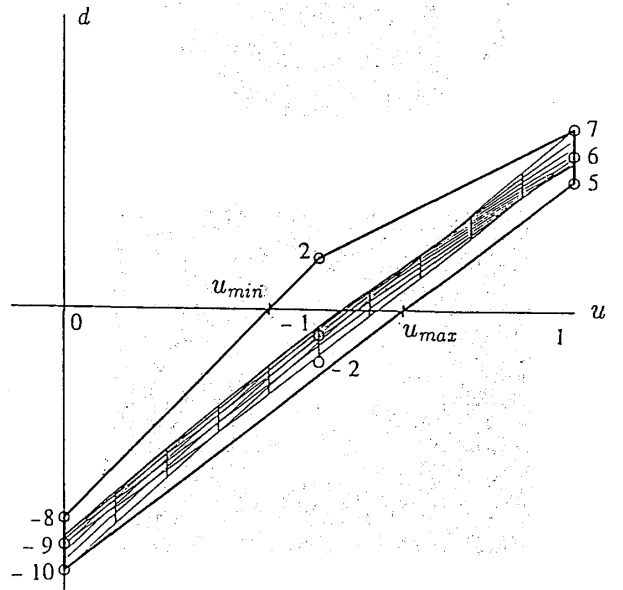


Fig.11 Side view of d(u,v) patch.

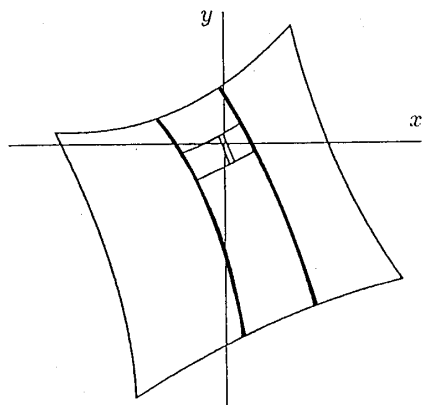


Fig.12 Iterating to the solution.

と結論づけられるから、この区間を切り捨てる。分割後、図12の太線の曲面が得られる。同様に、 v についても Bezier Clippingが適用できるのは明らかである。この交点計算法は u と v について交互に行われる。図12は u と v に関してBezier Clippingを適用した結果を示している。なお、最終段階では、実際には分割しないで許容誤差内に入る交点を求めることができる。この図の例では、 u_{\min} , u_{\max} および v_{\min} , v_{\max} の抽出は5回反復され、3組の分割が実行された。

走査線ごとに前処理することによりBezier Clippingをより効果的にすることができる。走査線に対して曲面を u , v の両方についてBezier Clippingすることにより、走査線と交差しない曲面の領域を切り捨てることができる。図13は図10の例を走査線でBezier Clippingしたものである。

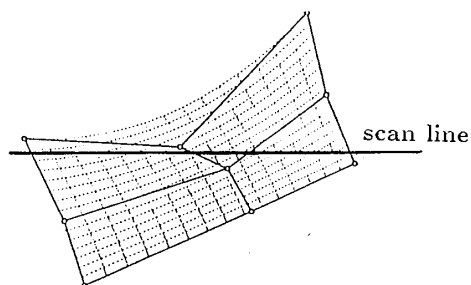


Fig.13 Trimming to the scanline.

(2) トリムド曲面の表示

曲面の一部が切り取られているいわゆるトリムド曲面を考える。トリミング領域は、パラメータ領域においてBezier曲線によって囲まれた閉領域とする。パラメータ領域 (u, v) で与えられた点 S （前述の方法によって求めた交点に相当）がトリミング曲線によって囲まれた範囲内にあるどうかを判定する。これは、4.1で述べた点と閉曲線の判定法が適用できる。

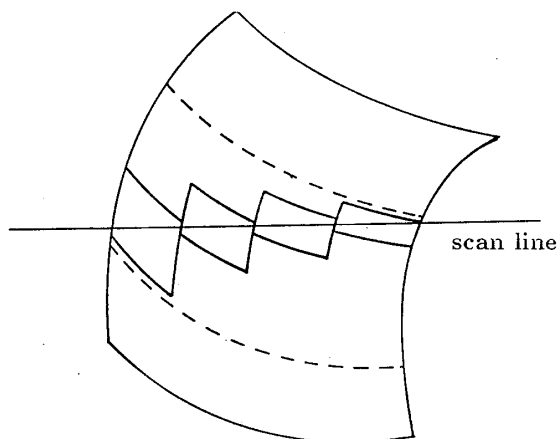


Fig.14 Subpatches on the scanline.

5.2.2 スキャンライン法による陰影表示⁶⁾

スキャンライン法による隠面除去の方法として、Laneらの方法が開発されている¹⁴⁾。彼らの方法は走査線単位に曲面を多角形に細分し、隠面除去は従来の多角形に対する方法を用いた。しかし、この方法は多角形間に隙間を生じる欠点がある。提案法はこの方法を改良したものである。すなわち、提案法は多角形でなく曲線の境界をもつサブパッチを発生し、走査線とサブパッチとの境界線との交点を求める方法である（図14参照）。

走査線と交差しない曲面の領域を切捨てた後、曲面を分割する操作を繰り返すことにより効率的に走査線上のサブパッチを求めることができる。 n 次Bezier曲面は式(11)で表わされる。制御点 $P_{ij}(X_{ij}, Y_{ij}, Z_{ij})$ は視点を原点とした視点座標系で与えられているものとし、投影面を $z=R$ 平面とすると、投影後の曲面の y 座標は付録式(1)となる。

走査線の投影面での y 座標を y_s とすると、 $y(u, v) = y_s$ が成立する。したがって、これに付録式(1)を代入し、分母を消去すると、

$$\sum_{i=0}^n \sum_{j=0}^n d_{ij} B_i^n(u) B_j^n(v) = 0, \quad (14)$$

$$d_{ij} = W_{ij}(Y_{ij} - (y_s/R)Z_{ij})$$

が得られる。これは式(13)と同一形式であるから、Bezier Clipping法を用いて、走査線と交差する u , v の区間を抽出できる。この区間で曲面をクリップする。サブパッチがほとんど平坦になるまで、2分割しさらに同様の操作を反復することにより走査線上のサブパッチを求められる（図14参照；点線は1回目の分割）。

次にサブパッチの境界曲線との交点を求める。境界曲線もBezier曲線であり、例えば $v=0$ の境界曲線の場合、

式(14)で $v=0$ とすると、式(1)と同一形式となり、Bezier Clipping法により走査線と境界曲線との交点に相当する u が求められる。

サブパッチはほとんど平坦であるから、サブパッチと走査線との交線は直線分であり、従来の多角形に対する隠面消去法を用いることによって容易に可視区間を判定できる。この方法は、影やアンチエイリアシング処理も可能であるが、それらの方法は省略する。

5.3 照明シミュレーション

従来の照明予測では、点光源に限られ、影の計算も単純な形状に限られていた。本稿では、曲面を含む種々の光源によって照射された曲面の照明シミュレーションを論じる。曲面の場合影の計算が困難である。大きさを持つ光源の場合、影は本影と半影からなる。特に半影中での輝度の計算が重要となる。

5.3.1 円柱光源の処理⁷⁾

オフィス、工場、事務所、教室等では、蛍光灯（円柱光源）が多く使用されており、円柱光源による照明計算の必要性は高い。一方、自動車の設計において、試作後、平行に並べられた蛍光灯の写り込みを利用して曲面形状の評価が行なわれている。このためにも、複数の平行円柱光源に照射された曲面の表示は重要である。

披照点から円柱光源を見た際は、その輪郭は長方形となる。したがって、長方形光源に対しては、輝度計算には境界積分法が適用できる。計算点と光源がある程度離れている場合、円柱光源は線光源と見なすことができる。ここで問題となるのは、光源と計算点との間を他の曲面に遮る場合である。これは、図15に示すように、計算点

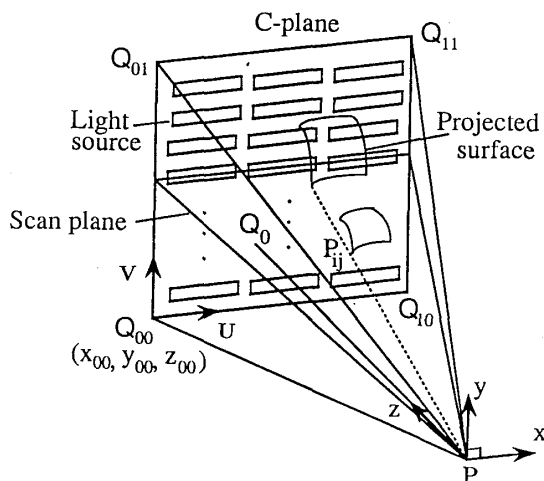


Fig.15 Shadow detection due to cylindrical sources.

Pを視点、光源（線光源）を走査線と見なすことにより、前述のスキャンライン法による隠面消去と同じとみなせる。ここで、天井に相当する面（C面と呼ぶ）に光源は平行に配置されているものと考え、この面が投影面、光源を走査線、計算点を視点とみなすことができる。この隠面消去法により求めた光源の可視区間を新たな光源と見なすことによって半影中の輝度を求めることができる。

5.3.2 曲面光源による陰影表示⁸⁾

多角形光源に対しては前述のように境界積分法により輝度を求めることができる。¹⁰⁾しかし、この方法は、曲面には適用できないので、曲面光源を帯光源の集合に分解し、それらによる輝度の総和により輝度を求める。面光源による輝度は計算点を中心とした単位半球表面に光源を投影し、さらに半球の低面に正投影したときの面積に比例する（立体角投射率と呼ばれる）。この投影した面積を効率よく求める方法を提案する。

図16に示すように、曲面をx軸（計算点を原点とする）を含む2平面でクリップすると、その投影した面積（図のクロスハッチング部）は簡単な式で求まる（文献(8)参照）。この面積を加算することで、曲面光源全体からの輝度が算出できる。平面と曲面との交差判定に5.2.2述べたBezier Clippingを利用した方法が利用できる。

5.3.3 相互反射を考慮した陰影表示⁹⁾

相互反射を計算するには、まず室内および物体表面をいくつかの要素に分割し、要素間のForm Factorを求める。このForm Factorとは、ある要素から発せられた光の他の要素への受光の割合を示す。このForm Factorを要素とするマトリックス方程式を解くことにより、各要

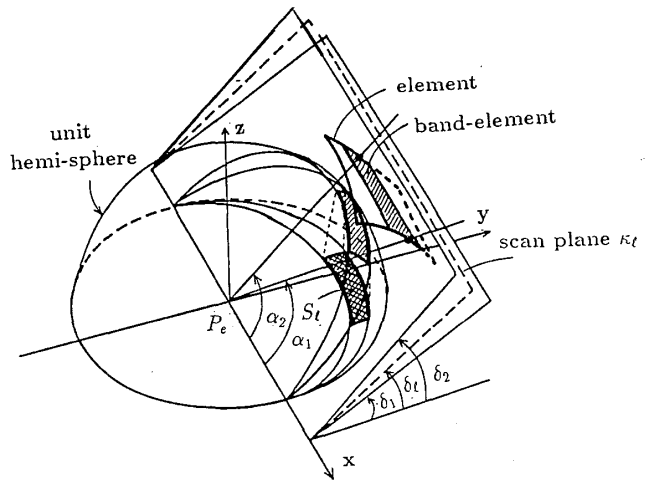


Fig.16 Sampling method of element using scan planes.

素での輝度を求めることができる。この方法をコンピュータグラフィクス分野ではラディオシティ法と呼んでいる。ここで重要なのはform factorであるが、これは前述の立体角投射率を要素内で積分したものと等しい。したがって5.3.2.の方法で解決することができる。

5.4 自由曲面の表示例

図17に隠線消去の例を示す。(a)はドーナツ形状が貫通したもので、(b)はティーポットおよびグラス表面上に曲線を含んだ例である。図18は、レイトレーシング法により表示したものである。(a)は、CGの分野ではスタンダードとなっているNewellのティーポットである(33パッチで6.9分、影なし時3.9分)。さらに、3で述べたアウトラインフォントを重ねたものである。(b)はパッチワークキルト上の鎖(4024パッチで29分)。両者とも影及び反射(写り込み)を計算している。ここで、計算時間はIRIS-4D/70GT使用時を示している。図19はスキャンライン法による陰影表示である。(a)は工業製品の例として、ハンデタイプの掃除機、(b)は熱気球である。フラッグ中の文字はトリミング曲線として定義されている。これらの例は曲面に穴のあるトリムド曲面を含んでいる。

図20は円柱光源で照射された車を示す。この図のように写り込んだ光源で曲面のクオリティを評価でき、また、影を含んだ照明予測も可能である。図21は相互反射を考慮したホール(実際の例で、共済会館:データはヤマギワ提供)である。楕円状の曲面の境界を持つ光源を曲面が囲んでいる。

6 おわりに

コンピュータグラフィクスにおいて、3次元物体を表示するには、隠線消去、隠面消去は不可欠な技術である。さらによりリアルな画像を得るには、影、反射、屈折を取り扱えるシェーディングモデルが重要である。これらのいずれにもBezier Clipping法が適用できる。

ここで述べた方法は、実際には、(a)隠線消去、(b)スキャンライン法による陰影表示、(c)円柱光源に対する陰影表示、(d)相互反射を考慮した陰影表示の4つのプログラムからなり、それぞれ1万ステップ程度のかなり大きなシステムである。これらのプログラムに対して、形状データの入力形式は総て同じである。したがって、同じ形状データに対して種々の表示法を選択できる。ここで、基本的な部分はかなり重複している。また、(c)、(d)の隠面消去は(b)を利用している。

本稿では、形状処理およびコンピュータ・グラフィクスに関連するものに限定して議論したが、Bezier Clippingは種々の数学的問題に適用でき、かなりパワフルな解析

法である。

謝辞

Bezier Clipping法は、本学の留学制度により著者が米国Brigham Young大学に滞在した際に、Sederberg博士との議論を通して開発したものである。本稿は、留学時およびその後の研究成果をまとめたものである。留学の機会を与えた下さった宮地茂学長を始め、学科長佐野教授、および教室の諸先生方に感謝致します。また、Brigham Young大学のコンピュータ・グラフィクス研究室(Christiansen教授が代表)への滞在を勧めて下さった広島大学中前教授に感謝する。

[参考文献]

- 1) Sederberg, T., Nishita, T., "Curve Intersection using Bezier Clipping", CAD, Vol.22, No.9, pp.538-549 (1990).
- 2) Nishita, T., Sederberg, T., Kakimoto, M., "Ray Tracing Rational Truncated Surface Patches," Computer Graphics, Vol.24, No.4, pp.337-345 (1990).
- 3) Sederberg, T., Nishita, T., "Geometric Hermite Approximation of Surface Patch Intersection," Computer Aided Geometry Design, Vol.8, No.2, pp.97-114(1991).
- 4) 西田、中前: Bezier曲線に囲まれた領域の走査変換、情報処理学会、グラフィクスとCAD研究会、Vol.90, No.65, 90-CG-45 (1990).
- 5) 西田、田北、中前: パラメトリック曲面の高精度隠線消去表示、情報処理学会、グラフィクスとCADシンポジウム、Vol.90, No.6, pp.129-138 (1990).
- 6) Nishita, T., Kaneda, K., Nakamae, E., "A High Quality Rendering by using Robust Scan Line Algorithm," CG International'90, pp.493-506 (1990).
- 7) 西田、田北、中前: 曲面のスムーズネステストのための平行円柱光源のシェーディングモデル、第42回情報処理学会全国大会、1p-4 (1991)
- 8) 西田、中前: 曲面光源に照射されたパラメトリック曲面の相互反射、情報処理学会、グラフィクスとCAD研究会、Vol.90, No.65, 90-CG-45 (1990).
- 9) 山口: コンピュータディスプレイによる形状処理工学[II]、日刊工業新聞社、pp.10-15.
- 10) 中前、西田: 3次元コンピュータグラフィクス、昭晃堂、(1986)
- 11) Sederberg, T., Parry, S.R., "Comparison of

- Three Curve Intersection Algorithm,” CAD, Vol.18, pp.58–63 (1986).
- 12) Elber G., Cohen E., “Hidden Curve Removal for Free-Form Surfaces”, Computer Graphics, Vol.24, No.4, pp.95–104 (1990).
- 13) Kajiya, J., “Ray Tracing Parametric Patches,” Computer Graphics, Vol.16, No.3, pp.245–254 (1982).
- 14) Lane, J., Carpenter, L., Whitted, T., Blinn, J., “Scan Line Methods for Displaying Parametrically Defined Surfaces,” CACM, Vol.23, No.1, pp.23–34 (1980).

付録 1 曲面の透視投影

n 次のBezier曲面について考える。制御点の座標 (X_{ij}, Y_{ij}, Z_{ij}) が視点座標系（視点が原点）で与えられているとすると、曲面上の点 $(X(u, v), Y(u, v), Z(u, v))$ は式(11)により表現される。視点座標系を考えているので Z は奥行きに相当する。投影面を $Z = R$ 平面とすると（ R は視点と投影面との距離）、投影面上の曲面は次式のように有理Bezier曲面となる。（透視変換後の座標は小文字で表現する）。

$$x(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^n w_{ij} x_{ij} B_i^n(u) B_j^n(v)}{\sum_{i=0}^n \sum_{j=0}^n w_{ij} B_i^n(u) B_j^n(v)} \quad (\text{付1})$$

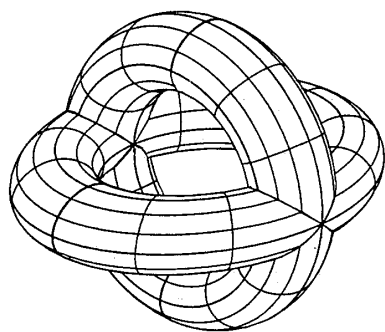
$$y(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^n w_{ij} y_{ij} B_i^n(u) B_j^n(v)}{\sum_{i=0}^n \sum_{j=0}^n w_{ij} B_i^n(u) B_j^n(v)}$$

ここで、曲面の制御点 P_{ij} の投影面上の同次座標 (x_{ij}, y_{ij}, w_{ij}) は次式となる。

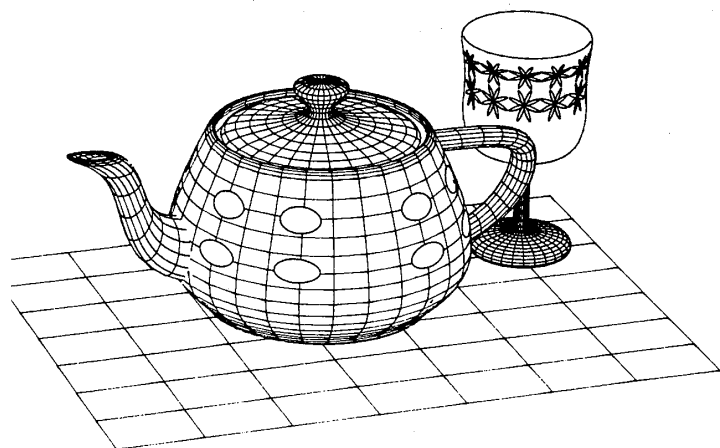
$$x_{ij} = R X_{ij} / Z_{ij}, \quad y_{ij} = R Y_{ij} / Z_{ij}, \quad (\text{付2})$$

$$w_{ij} = W_{ij} Z_{ij}$$

なお、 w_{ij} は制御点の重みで、曲面が非有理の場合（すなわち $W_{ij}=1$ ）、制御点の視点からの奥行きに相当する。このように、単に制御点を投影することによって曲面の透視投影が実現できる。

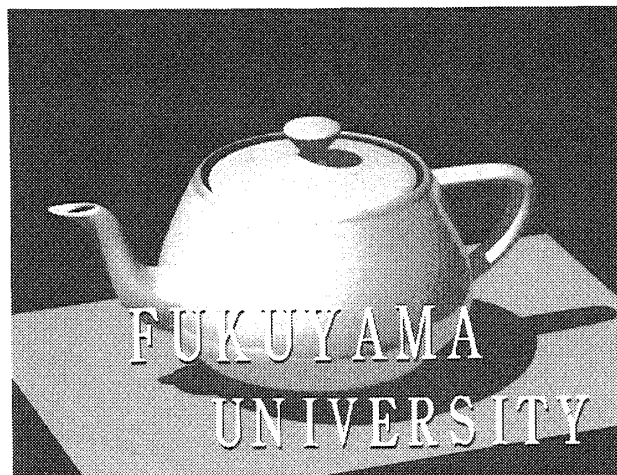


(a)

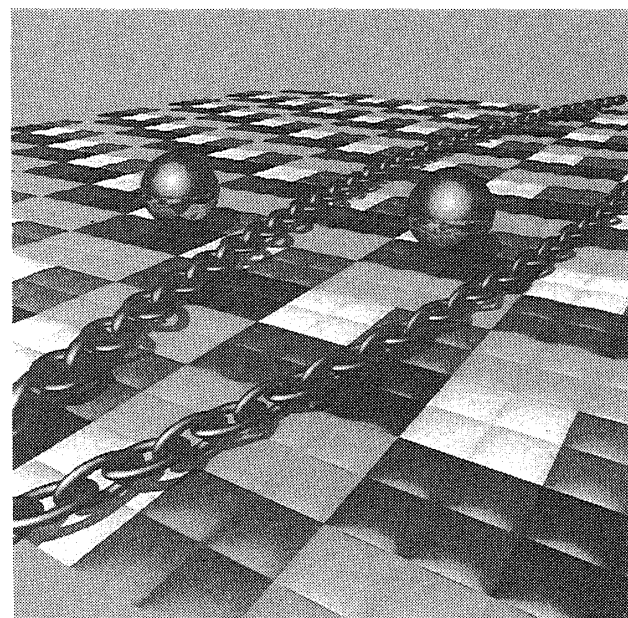


(b)

Fig.17 Examples of hidden curve removal.

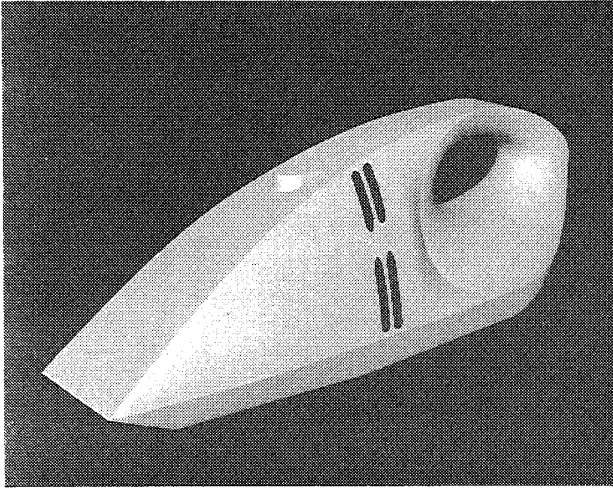


(a)

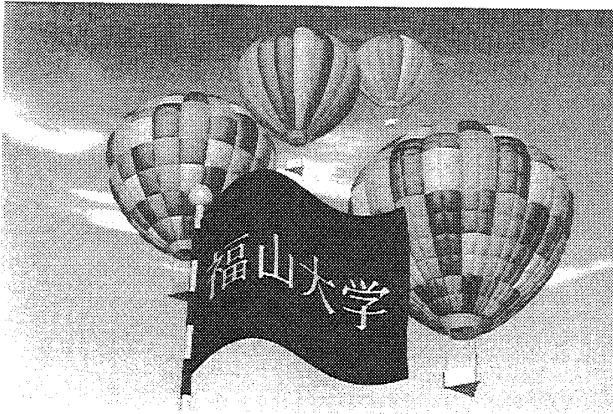


(b)

Fig.18 Examples of hidden surface removal using ray tracing.



(a)



(b)

Fig.19 Examples of hidden surface removal using scanline algorithm.

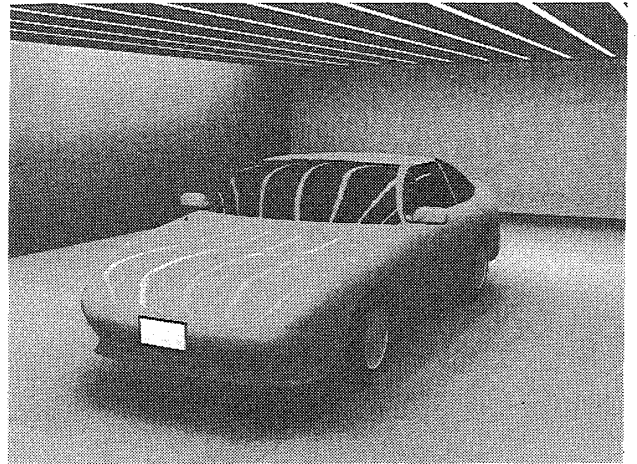


Fig.20 Example of smoothness test for curved surfaces illuminated by cylindrical sources.

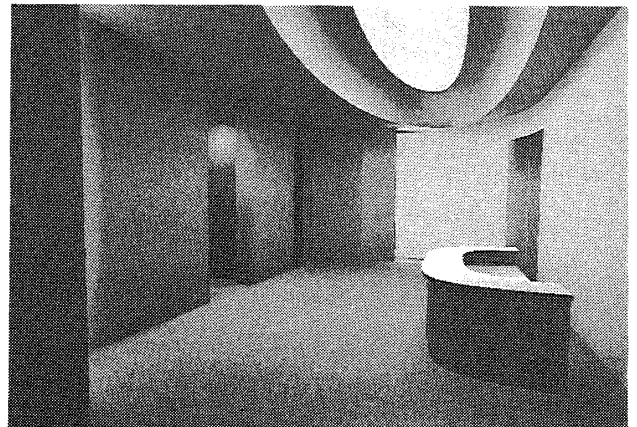


Fig.21 Examples of illumination simulation taking into account of interreflection of light.