

自由曲面のレンダリング手法

西田 友是

福山大学工学部

計算機支援形状設計の分野では、単に曲面を表示できればよいというものではなく、精度が重要である。したがって、従来のように曲面を単に多角形近似するのではなく、曲面形状をなるべく忠実に表現できるレンダリング技法が必要である。また、曲面を多角形近似する場合でも、ある程度の精度が保証されるレンダリング技法が望まれる。そこで、本稿は、チュートリアルとして曲面形状をなるべく忠実に表現できるレンダリング技法をいくつか紹介する。

Rendering Algorithms for Free-form Surfaces

Tomoyuki Nishita

Faculty of Engineering

Fukuyama University

Higashimura-cho

Fukuyama, 729-02 Japan

Displaying objects with high accuracy is necessary for CAGD (Computer Aided Geometric Design) and for the synthesis of photo-realistic images. Traditionally, polygonal approximation methods have been employed to display free-form surfaces. They bring on low accuracy of display not only in shape but also in intensity of objects. In this paper hidden-surface algorithms to display parametric surface patches are surveyed.

1 はじめに

自由曲面の表示技術はCAD/CAMの分野における必要性は言うまでもない。CGアニメーション、サイエンティフィック・ビジュアライゼーション分野においてもこの技術が注目を集めている。曲面の記述法としては、Bezier曲面、B-スプライン曲面、NURBS、スタイナ曲面など種々の方法があり、これらの曲面の表現形式によりそのレンダリング法も異なってくる。多面体の表示の場合と同じように、曲面の隠面消去は、Zバッファ法、レイトレーシング法、スキャンライン法に分類されるが、それぞれに属す代表的な手法を紹介する。

最近ではNURBS(Non-uniform Rational B-spline Surface)を主としたCADシステムが増加している傾向にある。しかし、NURBSを始め殆どの曲面が有理Bezier曲面に変換できるので、有理Bezier曲面に対する表示法で十分ともいえる。したがって、有理Bezier曲面のレンダリング法については特に詳しく述べる。

2 精度のよい曲面表示の必要性

従来、曲面のレンダリング法として、曲面を多角形で近似する方法が使われてきた。この方法の場合、次の問題点がある。1) 表示された物体の輪郭が直線で近似されているため、スムーズでない。また、曲面の連続性が失われる。2) 一般に、スムーズシェーディングが施されるが、この場合でも輝度の変化がスムーズでないことがある。3) 曲面同士が貫通する場合、交線がスムーズでない。4) 影を表示する場合、影の境界が折れ線となりスムーズでない。5) 精度を向上させるには、多角形数を増加する必要があり、その場合記憶容量が問題となる。

以上のように多くの問題があるので、曲面を多角形近似せずに直接表示する技術が必要となる。多角形で構成される3次元物体と同じように、曲面の隠面消去法としては、Zバッファ法、スキャンライン法、レイ・トレーシング法等がある。いずれの方法にしろ、次の技術が必要である。一般にパラメトリック曲面は、パラメタ(u, v)が与えられて、3次元座標(x, y, z)が求まる。それに対して曲面を表示するには、スクリーン座標(x, y)が与えられると(u, v)を求める必要がある。この処理は逆変換と呼ばれる。この逆変換は困難とされ、解析解を得るのは難しい。そこで、種々の反復法が開発されている。これらの方法のうち、必ずしも安定に交点を求められるとは限らないものも少なくない。したが

って、処理時間だけではなく、安定な方法が強く望まれている。

曲面のレンダリング技法としてこれまでに表1のように20種類以上の方法が開発されており、本稿で総てを解説することはできないので、歴史的に重要と思われるものをとりあげて説明する。本稿では、隠面消去の方法を中心に議論するが、輝度計算、影等のシェーディング手法については、文献[1]等を参照されたい。

3 Z-バッファ法による曲面表示

Z-バッファ法は、最も簡単な隠面消去といえる。この方法の欠点は記憶容量が多く必要で、影付けやエアリングの除去が困難である。

3.1 Catmullの方法[2]

この方法は、曲面を最初に表示可能とした方法であり、双3次曲面を再帰的に分割を行う方法である。曲面

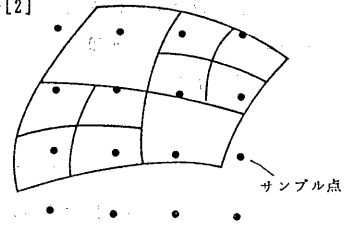


図1 曲面の分割 (Catmullの方法)

を再帰的に小さな曲面(サブパッチと呼ぶ)に分割し、1つのサブパッチがスクリーン上の最大1つの画素の中心(サンプル点と呼んでいる)をカバーするまで、曲面を再分割する。図1は曲面を分割した結果を示している。サブパッチが画素の中心を含むかどうかは、サブパッチを多角形近似して判定した。こうして、各画素上に存在するサブパッチが判明するから、そのサブパッチの輝度をフレームバッファに書き込み、奥行きをZ-バッファに書き込む。

3.2 Leien/Shantzらの方法[3,4]

抜けがでない程度のステップ幅で曲線を描くことにより曲面を高速表示する方法である。スクリーン座標から曲面上のパラメータ座標を求めることは容易でない。そこで、等パラメータ曲線を微小間隔で表示し、曲面がカバーする総ての画素を抜けがなく塗りつぶす。そして、隠面消去は、ハードウェアを利用したZバッファ法を用いている。等パラメータ線は、 v (または u)を固定して、 u をステップ幅 Δu で変化させ、各 u における3次元座標を計算し、これらを結ぶ。なお、時間を節約するため、前進差分法を利用し加算のみで計算する。その際、ステップ幅を適応的に変化させるAFD(Adaptive Forward Differencing)法を利用した。

3.3 Rockwoodらの方法[5]

ハードを用いたZ-バッファ法によりリアルタイム表示が可能な方法である。トリムドNURBSを取り扱っているが、内部ではトリムドBezier曲面に変換され、最終的には微小面に分割され(スクリーン上である許容値内となるように、微小面への分割幅を決定)、多角形に対するZバッファ法で表示される。なお、パラメトリック空間において、トリミング領域を格子状に分割し、周辺部は3角形に分割された。この方法は、曲面をほぼ忠実に表示しているが、基本的には多角形近似なのでやや境界部のスムースさにやや欠ける欠点がある。

4 レイ・トレーシング法による曲面表示

影、反射などいずれの処理においても、レイと曲面との交点計算が基本であるので、ここでは、1次レイと曲面との交点計算のみについて考える。パラメトリック曲面の場合、レイと曲面との交点を直接解析的に求める方法はなく、大別して次の2つの反復法が開発されている。すなわち、曲面の分割法(画素のサイズになるまで細分)および数値解析法(交点計算ができるような形式になるよう問題を変形し、数値解析法を利用して解く)があり、以下に代表的方法を説明する。

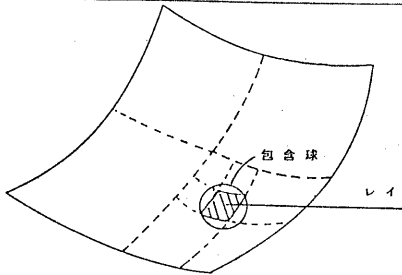


図2 Whittedの方法

4.1 Whittedの方法[6]

レイと曲面との交点を直接求めるのは困難なので、分割された曲面(パッチ)を囲む球(包含球)とレイとの交点の有無を利用する。包含球とレイが交差しない場合は、包含球に含まれるパッチとレイとの交点はないと言える。交差する場合にはパッチと交差する可能性があるため、パッチを細分割し、分割後のサブパッチの包含球との交差判定を行う。この操作を包含球が小さく(1画素程度)なるまで繰り返す(図2参照)。十分小さく分割された後もその包含球とレイとの交点がある場合には、パッチとレイとの交点が存在しているものとみなす。ここで包含球は、Bezier曲面の制御

点の凸包の性質を利用して求める。基本的に、この方法では解は線形に収束するが、画素のサイズ程度にするまではかなりの分割数が必要となる。

4.2 Kajiyaの方法[7]

レイを2つの平面の交線と考える方法である。曲面と2つの平面との交線として2つの空間曲線が得られる。これらの曲線同士の交点がレイと曲面との交点である。いま、2平面を考えると、これらの陰関数表現は、

$$a^kx + b^ky + c^kz + e^k = 0, \quad k=1, 2 \quad (1)$$

となる。双3次曲面は次式で表わされる。

$$\begin{aligned} x &= \sum_{i=0}^n \sum_{j=0}^n x_{ij} u^i v^j \\ y &= \sum_{i=0}^n \sum_{j=0}^n y_{ij} u^i v^j \\ z &= \sum_{i=0}^n \sum_{j=0}^n z_{ij} u^i v^j \end{aligned} \quad (2)$$

ここで、 (x_{ij}, y_{ij}, z_{ij}) は3次多項式の係数である。上式を平面式(1)に代入すると、

$$\begin{aligned} \sum_{i=0}^n \sum_{j=0}^n d^k_{ij} u^i v^j &= 0, \\ \sum_{i=0}^n \sum_{j=0}^n d^k_{ij} u^i v^j &= 0, \end{aligned} \quad (3)$$

ここで

$$d^k_{ij} = x_{ij}a^k + y_{ij}b^k + z_{ij}c^k + e^k, \quad k=1, 2$$

曲面とレイとの交点は、上の2曲線の共通解として求めることができる。上式は、未知数2個(uとv)の連立方程式であるが、vの多項式となるように変形することができる。さらに、これらの式はuのみの1つの多項式に変形することができる。その結果、uに関して18次多項式が得られる。uの解が求めれば、式(3)からvを求めることができる。ここで、実際に有効な解は $0 < u, v < 1$ を満たすものである。この18次式を解くために、Newton法より安定な方法を用いている。この方法では、1つの交点を計算するのに約6000回の実数演算を要すから、時間がかかる方法といえるが、数値解析的に解が求められること示した最初の方法であることに意味がある。

4.3 西田らの方法[8]

トリムド有理Bezier曲面を対象としており、安定にレイと曲面との交点が求められる方法である。この方法は、Bezier曲線の凸包性を利用して解の存在区間を収束させる方法(Bezier Clipping法と呼ばれる)を用いる。Newton法は1つの初期から始めて解に向けて収束

するが、総ての解が求まる保証がない。それに対し、Bezier Clipping法は、解が存在する可能性のある区間を縮小する操作を反復するものであり、安定に総ての解が求まる。

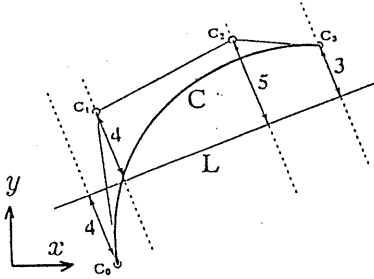


図3 Bezier曲線と直線との交差

(1) Bezier Clipping法

図3のように(x,y)平面上の曲線Cと直線Lとの交点を考える。制御点 $C_i(x_i, y_i)$ をもつ、n次のBezier曲線Cは次式で定義される。

$$C(t) = \sum_{i=0}^n C_i B_i^n(t) \quad (4)$$

一方、直線Lは次式で表される。

$$ax + by + c = 0, \quad (a^2 + b^2 = 1) \quad (5)$$

LとCとの交点は式(4)を式(5)に代入することによって求まるから、次式が得られる。

$$d(t) = \sum_{i=0}^n d_i B_i^n(t) = 0, \quad (d_i = ax_i + by_i + c) \quad (6)$$

ここで、 d_i はLから C_i までの距離に相当する。式(6)の関数 $d(t)$ は、制御点 $D_i(t_i, d_i)$ よりなる曲線で、制御点は t に関して等間隔($t_i = i/n$)である。図4はこの曲線を示す(図3に対応)。

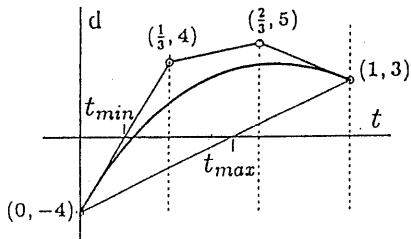


図4 距離曲線

図4のように、 D_i の凸包はt軸と、点 t_{min} と t_{max} で交差する。 $D(t)$ はその制御点の凸包の内部に存在する

から、パラメータ領域 $0 \leq t < t_{min}$, あるいは $t_{max} < t \leq 1$ ではCとLは交点をもたないと断言できる。したがって、曲線Cの両側の曲線を捨ててもかまわない。これにより解の存在区間を狭くすることができる。この操作を反復することにより、解は収束する。有理Bezier曲線の場合、制御点 P_i の座標に加えて重み w_i を考慮することになる。この場合、式(6)の d_i を $d_i = w_i(ax_i + by_i + c)$ のように修正することで適用できる。

(2) レイと曲面との交差判定

有理Bezier曲面は、次式によって表わされる。

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

ここで、 $P_{ij}(X_{ij}, Y_{ij}, Z_{ij})$ は重み w_{ij} をもつ制御点である。Kajiyaの方法と同じように、レイは2平面の交線として定義する。式(1)において、 (a^k, b^k, c^k) ($k=1, 2$)は面の単位法線とする。また、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すると、 $e^k=0$ である。平面kと曲面との交点は、平面式(1)に式(7)を代入し、分母を消去すると、次式が得られる。

$$d^k(u, v) = \sum_{i=0}^n \sum_{j=0}^n d^{kij} B_i^n(u) B_j^n(v) = 0, \quad k=1, 2 \quad (8)$$

ここで、 $d^{kij} = w_{ij}(a^k X_{ij} + b^k Y_{ij} + c^k Z_{ij})$ 。

d^{kij} は平面kから制御点への距離に比例している。さらに、Z軸とレイが一致するように曲面を回転すると、 $c^k=0$ となり、z成分は不要であり、分割に要す時間を節約できる。ある曲面の総ての制御点に対して d^{kij} が正(または負)なら、レイと曲面は交差しない。それ以外の場合、式(8)を満たす (u, v) が交点として求めることができる。

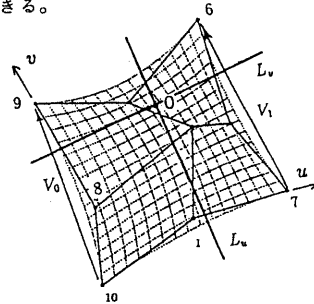


図5 直線L_uと曲面との交差

いままでは、3次元空間での議論であったが、投影した2次元平面で交点を求めることができる。図5は、z軸がレイと一致させた後、投影された曲面Pの例(2次のBezier曲面)である。

図5に示すように、ベクトル (V_0+V_1) に平行で原点0を通過する直線 Lu を定義する。 Lu は原点を通過するから $c=0$ である。したがって、直線 Lu から制御点までの距離 D_{ij} は

$$D_{ij} = ax_{ij} + by_{ij}. \quad (9)$$

投影された曲面の任意の点 $P(u, v)$ の Lu からの距離 $D(u, v)$ は、次式となる。

$$D(u, v) = \sum_{i=0}^n \sum_{j=0}^n D_{ij} B_i^n(u) B_j^m(v) \quad (10)$$

関数 $D(u, v)$ は、制御点 $D_{ij}(u_{ij}, v_{ij}, D_{ij})$ (ただし $u_{ij}=i/n, v_{ij}=j/m$)をもつ曲面として表現される。

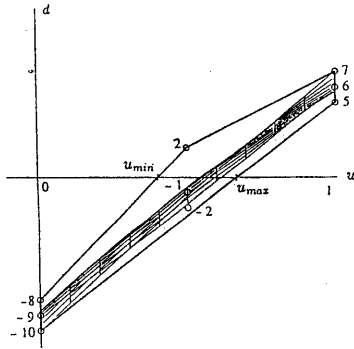


図6 D曲面の側面図

v軸方向から見た $D(u, v)$ 曲面の側面図を図6に示す。投影された制御点の凸包が $D(u, v)$ 曲面の投影を包含している。この例では、 u_{min} と u_{max} においてu軸と凸包が交差している。 $u < u_{min}$ 、 $u > u_{max}$ の範囲では $d(u, v) \neq 0$ と結論づけられる。分割後、図7に示す太線の2次元曲面が得られる。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7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。

もし、解のパラメータ区間が減少しないなら、解は複数存在するので、曲面は2分され、各々にBezier Clipping法が適用される。

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ことができる。図8

は図6の例を走査線でBezier Clippingしたものである。この処理により2回の分割を節約することができる。同時に、交点ををもたない曲面も発見される。

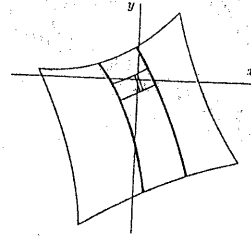


図7 3回のBezier Clipping

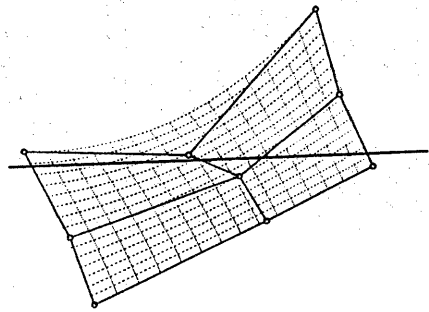


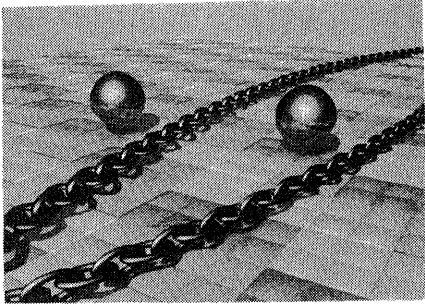
図8 走査線に対するトリミング

(3) 自由曲面の実際の表示例

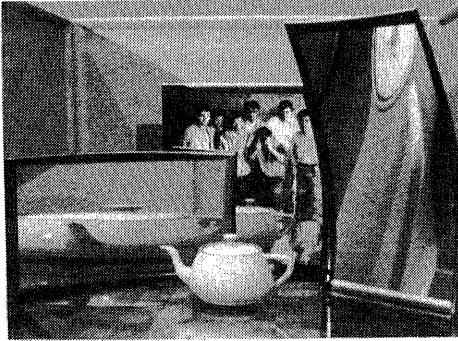
図9は本手法により曲面を表示したものである。(a)は変形したteapot(2304パッチで12.5分)。(b)は帽子の形をした鎖(4024パッチで29分)。鎖は、ドーナツ形状を複製したものをSederbergのFFD(自由形状変形理論)を用いて変形して作成した。(c)は、鏡の部屋である。人間の像は「フタタツ」によるものである(富士通の柿本氏の作品)。ここで、計算時間はIRIS-4D/70GT使用時を示している。これら例のように、4000パッチ以上の場合においても、影、反射付きで30分以内で処理できるのは、多角形に近似しない方法のなかではかなり実用的なレベルに達したと言える。



(a)



(b)



(c)

図9 レイトレーシングによる曲面の表示例

5 スキャンライン法による曲面表示

スキャンライン法は計算時間に関してレイトレーシング法より有利である。スキャンライン法は、視点と走査線とにより構成される走査面と曲面との交線を求め、最も手前の交線を表示することである。多角形なら走査面との交線は直線で、処理は簡単であるが、曲面の場合、走査面と曲面との交差は曲線となる。また、この曲線は1つとは限らない。

5.1 Blinnの方法 [9]

Blinnは、曲面の境界曲線および輪郭線と走査線との交点をNewton法により求めた。これらの交点からスキャンラインセグメントと呼ばれる曲線を求め、これらの曲線に対して隠面消去処理を行う。曲面が $F(u, v)$ で表わされ、視点は z 軸方向にあるとする。曲面の境界曲線(例えば、 $v=0$ の境界)と走査線 $y=y_s$ との交点は次式を解くことによって得られる。

$$Y(u, 0) = y_s \quad (11)$$

また、曲面の法線ベクトルの z 成分が $Z_n(u, v)$ とすると、走査線と輪郭線との交点は次式を満たす。

$$Z_n(u, v) = 0, \quad Y(u, v) = y_s \quad (12)$$

これらの式は、 u, v に関して高次式となるため、これらの解を求めるには2変数Newton法による収束計算が必要となる。走査線の移動範囲、すなわち曲面の存在範囲(極値で求める)の y 成分は次式によって求める。

$$Y_u(u, v) = 0, \quad Y_v(u, v) = 0 \quad (13)$$

ここで、 Y_u, Y_v はそれぞれ Y の u 成分、 v 成分の微分である。

5.2 Whittedの方法[10]

Whittedは、いくつかの等パラメータ曲線を追加し(曲面を分割したことに等しい)、さらに輪郭線を3次のヘルミート補間曲線によって近似した。境界線(等パラメータ線も含む)と輪郭線によって曲面は必ず囲まれているから、これらの曲線を走査変換した。この曲線と走査線との交点はNewton法により求められた。スキャンラインセグメントの両端点以外の区間の法線および奥行きは両端点の値から線形補間により求められた。この方法では、走査の前に輪郭線を求めるが、境界線と輪郭線との交点を基に輪郭線を抽出する。このため、輪郭線が境界線を通過しないような場合は処理できない欠点がある。

5.3 Lane/CarpenterおよびClarkの方法

Lane/Carpenter[11]およびClark[12]の方法は曲面の多角形への分割法を利用している。前者の方法では、面は平坦とみなせるまで分割され、平坦とみなされたパッチは多角形として扱われ、多角形のスキャンライン法を使って表示される。各走査線において、走査線と交差するパッチは、平坦と見なせるか、またはこれ以上走査線と交差しなくなるまで細分される。平坦とみなされたサブパッチは多角形(4辺形)として扱われ、従来の多角形に対する隠面消去法が適用できる。この方法は、近似多角形と曲面パッチとの間の不一致から多角形間の隙間が生じる欠点がある。これは多角形の辺が直線のため生じる。

ClarkとLane/Carpenterの方法との違いは、前者が走査の前に曲面分割を行うのに対し、後者は走査線毎にパッチ毎に分割する。Clarkの方法では、曲面が平坦かどうかの判定ではなく、投影後のスクリーン上において境界線が直線と見なせるか(曲率で判定)どうかに主眼を置いてチェックした。そのため隙間を生じない。

5.4 西田らの方法[13]

この手法は、Lane/Carpenterの方法を拡張したものである。すなわち、Lane/Carpenterの方法は走査線単

位に多角形に分割するが、この方法は曲面(ツブ)に分割し、走査線とツブとの境界線との交点を求める方法(図10参照)であり、隙間を生じる欠点を改善したものである。

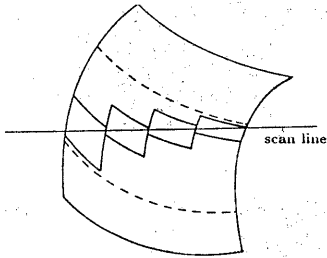


図10 走査線上のサブパッチ

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(7)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する、投影後の曲面のy座標は次式となる(簡単のため投影面を $z=1$ 平面とする)。

$$y(u, v) =$$

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

走査線の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、これに式(14)を代入し、分母を消去すると、

$$\sum_{i=0}^n \sum_{j=0}^n d_{ij} B_i^n(u) B_j^n(v) = 0, \quad d_{ij} = Y_{ij} - y_s Z_{ij} \quad (15)$$

が得られる。これは式(10)と同一形式であるから、前述のBezier Clipping法を用いて、走査線と交差するu, vの区間を抽出し、この区間で曲面を分割することにより走査線上のサブパッチを求められる(図10参照)。

次にサブパッチの境界線との交点を求める。境界線もBezier曲線であり、例えば $v=0$ の境界線の場合、式(15)で $v=0$ とすると、

$$\sum_{i=0}^n d_i B_i^n(u) = 0, \quad d_i = Y_{i0} - y_s Z_{i0} \quad (16)$$

が得られる。これも、Bezier Clipping法により走査線と境界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ができる。

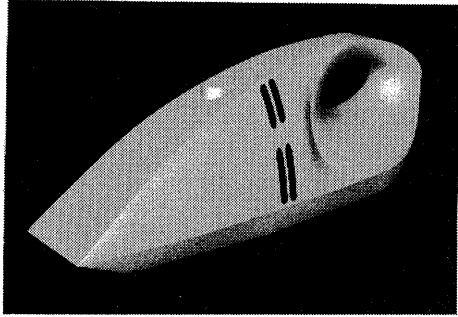
図11にこの方法によるBezier曲面の表示例を示す。(a)はトリムド曲面の例としてハンディタイプの掃除機の表示である。(b)は熱気球で、手前の旗の柄はトリムド曲線で表現されている。(c)は車の表示例である。この方法は、文献[8]のレイトレーシング法に比べ約5倍速い結果が得られている。

6 おわりに

本稿では、曲面を比較的精度良く表示する方法をいくつか紹介した。色々な方法があり、結局どのレンダリング法がベストかは決めかねる。というのは、どんな曲面を表示したいか、あるいは、スピードを優先するか、画質(あるいは精度)を優先するかにより異なるであろう。リアルタイムの処理が必要ならZバッファ法がよいかもしいない。ここで紹介したZバッファ法には、すでにワークステーションにライブラリとして標準装備されているものもある。自由曲面の表示法をいくつか紹介したが、読者の参考になれば幸いである。なお、本稿における計算例は広島大学の中前研究室において作成したものであり、中前教授に感謝致します。

参考文献

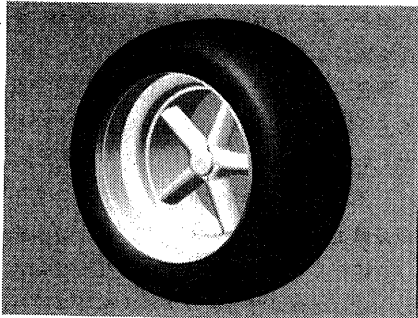
- 1) 中前、西田：3次元コンピュータグラフィクス、昭晃堂、1986。
- 2) Catmull, E., "Computer Display of Curved Surfaces," Proceedings of IEEE Conf. Computer Graphics, Pattern Recognition and Data Structures, 1975, p. 11.
- 3) Lien, S., Shantz, M., Pratt, V. "Adaptive Forward Differencing for Rendering Curved Surfaces," Computer Graphics, Vol. 21, No. 4, 1987, pp. 111-118.
- 4) Shantz, M., Ling, S., "Rendering Trimmed NURBS with adaptive Forward Differencing", Computer Graphics, Vol. 22, No. 4, 1988, pp. 189-198.
- 5) Rockwood A., Heaton K., Davis T., "Real-Time Rendering of Trimmed Surfaces, Computer Graphics, Vol. 23, No. 3, 1989, pp. 107-116.
- 6) Whitted, T., "An Improved Illumination Model for Shaded Display," Comm. ACM 23, 1980, pp. 343-349.



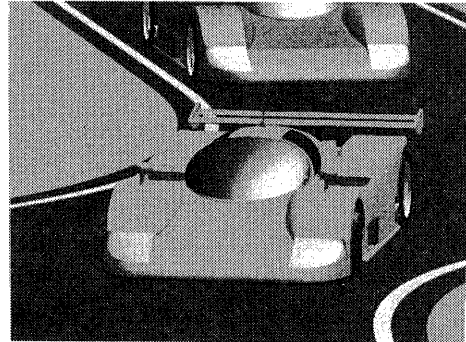
(a)



(b)



(c)



(d)

図 1 1 スキャンライン法による曲面の表示例

- 7) Kajiya, J., "Ray Tracing Parametric Patches", Computer Graphics, Vol. 126, No. 3, 1982, pp. 245-254.
- 8) Nishita, T., Sederberg, T., Kakimoto, M., "Ray Tracing Trimmed Rational Surface Patches", Computer Graphics, Vol. 24, No. 4, 1990, pp. 337-345.
- 9) Blinn, J.F., "A Scan line Algorithm for Displaying Parametrically Defined Surfaces.", Computer Graphics, Vol. 12, No. 3, 1978, pp. 348-354.
- 10) Whitted, T., "A Scan Line Algorithm for Computer Display of Curved Surfaces.", Computer Graphics, Vol. 12, No. 3, 1978, pp. 26
- 11) Lane, Carpenter, Whitted, and Blinn, "Scan Line Methods for Displaying Parametrically Defined Surfaces.", Comm. ACM, Vol. 23, 1980, pp. 23-34.
- 12) Clark, JH, "A Fast Scan-Line Algorithm for Rendering Parametric Surfaces.", Computer Graphics, Vol. 13, No. 2, 1979, pp. 174.
- 13) Nishita, T., Kaneda, K., Nakamae, E., "A Scanline Algorithm for Displaying Trimmed Surfaces by Using Bezier Clipping", The Visual Computer, Vol. 7, 1991, pp. 269-279.

表 1 曲面のレンダリング法の分類

1) Z-バッファ法	<ul style="list-style-type: none"> • Cutwell (1974) • Rockwood (1987) • Lion, Shantz, Pratt (1987) • Shantz, Chang (1988) • Rockwood, Heaton, Davis (1990) 	双 3 次曲面 N^* ラミナリク (有理 Bezier) 曲面 有理 双 3 次曲面 有理 NURBS 有理 NURBS	【曲面の分割】 【前進差分法】 【適応型前進差分法】 【適応型前進差分法】 【微小面への分割】
2) レイトラック法	2.1 分割法 <ul style="list-style-type: none"> • Whitted (1980) • Woodward (1989) • 西田, Sederberg, 柿本 (1990) 2.2 数値解析法 <ul style="list-style-type: none"> • Kajiya (1982) • Sederberg (1984) • Toth (1985) • Joy, Bhatnabholla (1986) • Sweeney, Bartels (1986) • Levner (1987) • Lischinski, Gonczarowski (1990) 	双 3 次曲面 Bezier 曲面 有理 Bezier 曲面 双 3 次曲面 シュタイナ 曲面 N^* ラミナリク (Bezier) 曲面 N^* ラミナリク 曲面 B-スプライン 曲面 B-スプライン (または 双 3 次) 曲面 Bezier 曲面	【曲面の分割、包含球】 【曲面の分割、Whittedの拡張】 【Bezier Clipping法】 【Lagurre法、18次式の数値解析】 【区間Newton法】 【疑似Newton法】 【曲面分割後 2 変数Newton法】 【Sweeneyの変形】 【Tothの変形】
3) スキャンライン法	3.1 数値解析法 <ul style="list-style-type: none"> • Blinn (1980) • Whitted (1980) • Schweitzer, Cobb (1982) 3.2 分割法 <ul style="list-style-type: none"> • Lane, Carpenter (1980) • Clark (1979) • Griffiths (1984) • Pueyo, Brunet (1987) • 西田, 金田, 中前 (1990) 	N^* ラミナリク 曲面 N^* ラミナリク 曲面 N^* ラミナリク (Bezier) 曲面 N^* ラミナリク 曲面 N^* ラミナリク 曲面 N^* ラミナリク (3 次スプライン) 曲面 双 3 次 曲面 Bezier 曲面	【Newton法】 【Newton法】 【Whittedの拡張, Newton法】 【走査線毎の多角形近似】 【多角形近似】 【メッシュ分割】 【メッシュ分割】 【Bezier Clipping法】 [] 内は手法に関するキーワードである