

Bezier Clippingを用いた2次元画像の自由形状変形法

西田 友是*・藤井 利久**

A Method of Free-Form Deformation for 2-D Images
using Bezier Clipping

Tomoyuki NISHITA and Toshihisa FUJII

ABSTRACT

Recently, computer graphics comes to be used in various fields such as CAD/CAM, scientific visualization, entertainments, and its importance rises more. In the field of entertainment, the technique of image deformation is recognized as an attractive technique, and widely used for TV commercials or movies.

The deformation technique is referred as "morphing", which is an image processing technique typically used as an animation tool for the metamorphosis from one to another. Deforming the image is performed by overlapping mesh with the source image to transform a two-dimensional image to another and by moving the lattice points of the mesh.

In the proposal method, the lattice points are handled as the control points of Bezier patches or B-spline patches, then the method guarantees the continuity of first derivative of the deformed image. We calculate the image deformation as an inverse mapping. The one of authors has already proposed the Bezier Clipping method which is used for the intersection test between Bezier patches and ray, the inverse mapping is performed by the Bezier Clipping method. By using the Bezier Clipping method, the proposed methods provide an animation technique with smooth deformation.

1 はじめに

近年、コンピュータグラフィックスは、様々な分野で用いられるようになり、その重要性はますます高くなっている。CAD/CAMシステムや科学シミュレーションの他に、芸術やアニメーションなどのエンターテイメント分野へと応用範囲も広い。エンターテイメント分野において、2次元画像の自由変形の技術が魅力的な技法

として注目を浴び、最近映画やテレビのコマーシャルなどに用いられるようになってきた。この処理はメタモルフォシスまたはモーフィング(メタモルフォシスを短縮した造語:morphing)と呼ばれ、実写映像を2次元の画像処理で変形・合成する技術であり、手軽に使える大きな効果がある。

一般に、2次元画像の自由変形を行なうには元の画像

*情報処理工学科

**電子・電気工学科

にメッシュをオーバーラップし、そのメッシュの格子点を移動することにより画像を変形する方法が用いられる。この処理において問題となるのは、変形画像に不連続な部分が生じることである。そこで、提案法においては、メッシュをBezier曲面またはB-spline曲面の格子点として処理し、変形後も1次微分まで連続な図形となる変形を行なう。著者はすでに、Bezier曲面と直線との交点を効率よく求める方法としてBezier Clipping法を提案しており^{1),2)}、この方法を適用し、変形を滑らかに精度よく行なうアニメーション技法について提案する。

2 形状変形の従来法

2次元画像の自由変形には、2つの全く異なるオブジェクトがスムーズに変形するものと、ある一つのオブジェクトがその形態の性質は変化させずに変形するものとに分けられる。前者は、最近の映画（ウィロウ、ターミネータII等）やテレビ、コマーシャルに見られるように、虎から魔女へ、人間からミイラへ変化するものなどがある。また、後者は、旗をはためかせるものや、雲や炎の変形（ひもを結ばせるもの）などがある。

メタモルフォシス（変形）には3次元物体を変形するものと2次元画像を変形するものがあり、次の3つの方法がある。

1) キーフレーム法

2次元多角形の変形によるアニメーションには、キーフレーム法がよく利用される。この方法は、線画または輪郭（折れ線表現）中を塗りつぶした図形で表現されるものに適用される。この方法では、ある図形から他の図形へ変形する場合、2つの図形間で対応する頂点を指定しておき、これらの頂点間を線形補間することにより、中間の図形を得るものである。この方法では、変形の中間で不自然（例えば輪郭が交差する）になることもある。これに対処する方法として、物理則に基づく変形法が、最近Sederbergらによって開発された(shape blending)³⁾。

この方法は基本的には線画であるためリアル感に欠ける。

2) FFD法（自由形状変形法）

3次元モデルを変形する方法であり、代表的なものに、Sederbergら⁴⁾はFFD法（Free Form Deformation）を開発した。この方法はBernstein多項式を利用しており、3次元形状のみでなく2次元図形の変形にも適用できる。変形しようとする図形を直交格子で囲みその格子点を移動させることにより、図形を変形する方法である。図1は2次元の場合の例である。この格子点はBezier曲面の制御点に対応している。格子内の点の座標はパラメータ(u,v)で表され、(n+1) × (m+1)の格子の場合、格子点P_{ij}の座標(x_{ij}, y_{ij})は次式で表される。

$$P_{ij} = O + (i/n)U + (j/m)V \quad (1)$$

(i=0,1,...,n, j=0,1,...,m)

ここで、Oは格子の原点で、U、Vは基準ベクトルで、一般に、それぞれX軸、Y軸に平行である。格子内のある点P(x,y)はパラメータ(u,v)を用いて、次式によって表される。

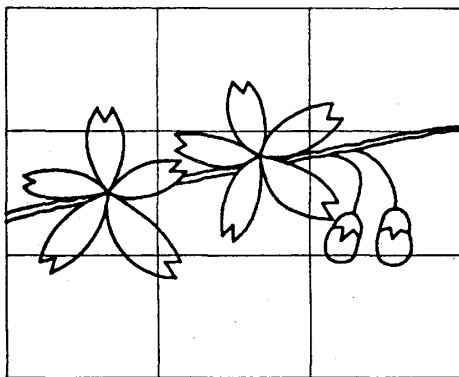
$$P = O + uU + vV \quad (0 \leq u \leq 1, 0 \leq v \leq 1) \quad (2)$$

移動した後の格子点をP'_{ij}とすると、変形後の点P'は次式によって求められる。

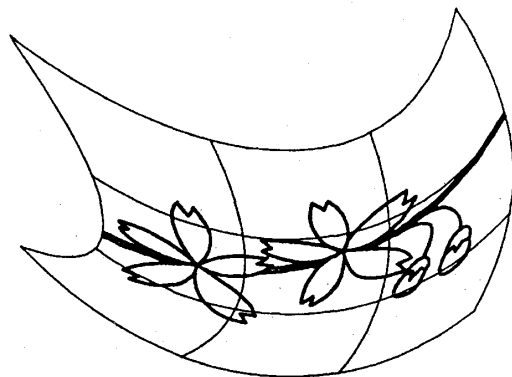
$$P'(u,v) = \sum_{i=0}^n \sum_{j=0}^m P'_{ij} B_i^n(u) B_j^m(v) \quad (3)$$

ここで、P'_{ij}はBezier曲面の制御点に相当している。BはBernsteinの多項式であり、次式で定義される。

$$B_i^n(u) = \binom{n}{i} (1-u)^i u^{n-i} \quad (4)$$



(a)



(b)

Fig.1 Free form deformation

この方法では、まず式(2)を用いてある点 $P(x,y)$ に対応する (u,v) を求め(1次式であるから簡単)、この (u,v) を用いて式(3)から変換後の (x',y') 座標が求まる。この方法では、一般に図形は多角形(または微小線分)で近似されて表示される。

3) モーフィング

FFD法は基本図形要素により構成されるモデル(線分、多角形、曲線など)に適用されるが、この方法は画像(実写画像またはCG画像)に適用される。2次元画像の自由変形法は、変形させたい画像に $n \times m$ のメッシュをオーバーラップさせ、格子点をマウスで移動し変形すると、格子の変形にともない画像も変形される。画像はソース(図2上図)とディステーション(図2下図)の2枚を用意し、それぞれの画像上で格子点を移動させる。なお、FFD法では元の図形に重ねるメッシュは直交格子であるが(図1参照)、この方法では対象図形の複雑さに合わせて不規則に移動させてある(図2参照)。変形した図形を表示するには、 (x,y) から (u,v) を求め、いわゆる逆写像が必要である。すなわち、変換後のスクリーン上のある点 (x,y) から、その点に対応する (u,v) 座標を求め、次に変換前の (u,v) 座標に対応する点での色を割り当てる。FFD法のようにメッシュとしてBezier曲面を利用する場合、この逆写像を行なうにはかなりの高次式を解く必要から、格子を多角形近似する方法などが用いられる。

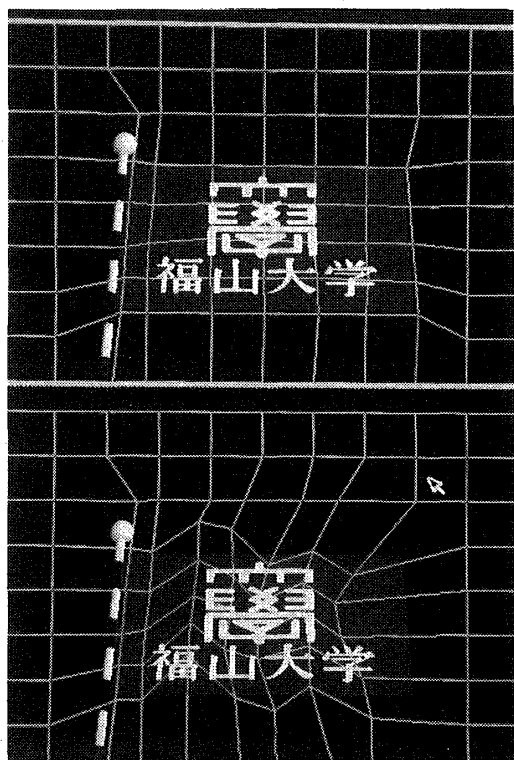


Fig.2 Specification method of deformation

3 提案法の基本的考え方

提案法は、同一画像が変形するものと、ある画像から他の画像へ変形するものの2つの方法に適用できる。画像はCGで製作したものあるいは写真画像を考える。すなわち、前者は画像AからA'、後者はAからBへの変形である。メッシュは元画像の複雑さ、変形の激しい部分に同期して格子点を配置しておく。AからA'への変形の場合、例えば背景中(同一色)に顔などの画像があり、この顔を変形することを考える。変形が明確に現れるのは輪郭部である。画像の輪郭に沿って格子点が配置されるようにマウスを使って会話的に移動する。次に、変形したい付近の格子点を移動させる。

一方、画像Aから画像Bへ変換する場合(例えば、大人の顔から子供の顔)には、AのみでなくBの画像上の格子点も輪郭に沿うよう移動させる。

スムーズな変形操作を得るため、格子点はBezier曲面の制御点とみなす。

以下に、画像A(ソース)から画像B(ディステーション)に変形させる場合の手順を示す。ただし、画像Aのスクリーンに座標系を (x,y) 、画像Bの座標系を (x',y') とする。

- 1) 画像AおよびBを表示させる。
- 2) 両画像上それぞれにメッシュ($n \times m$)を重ねる。
- 3) 画像AおよびB上のメッシュの格子点をそれぞれマウスを用いて移動させる。
- 4) 両者のメッシュの格子点からBezier曲面を求める。
- 5) Bのメッシュの格子点を制御点とするBezier曲面を考え、これらを走査し各点において次の処理を行なう。
 - 5.1) 走査線上の点 $P'(x',y')$ に対応する (u,v) 座標を求める。
 - 5.2) Aのメッシュから (u,v) に対応する画像Aでの (x,y) 座標を求め、その点Pでの色を抽出し、計算点P'の色と合成する。

処理3)において、格子点が画像Aの輪郭に沿うように配置し、それらの輪郭に対応する点がBの画像の輪郭に沿うようにしておくと、AからBへの変形が行なわれる。

処理4)において、基本的には、メッシュとしては3次のBezier曲面の制御点を用いる(したがって最低 4×4 の格子)。複雑な形状のものについては、格子点を増加するが、Bezier曲面どうしのスムーズな接続は容易ではないので、格子点はB-spline曲面の制御点として与え、内部処理としてBezier曲面に変換⁵⁾した後、次章で述べ

る逆マッピング (処理5.1) を行なう。

色はAとBとの同一 (u, v) 座標の色を合成 (混色) する。アニメーションの際は、AとBとの格子点位置を時間間隔で補間し、4)、5) の処理が繰り返される。

本稿では、4)、5) の処理について主に議論する。

4 逆マッピング

本稿では、1次微分まで連続 (すなわち C^1 接続) な変形画像を得るため3次Bezier曲面を考える。なお、k次微分まで連続にするには $(n+2)$ 次の曲面にすればよい。この方法は、例えば一部のみ局所的に変形した場合、変更しない部分と変更した部分の境界において1次微分まで連続性が保持される。なお、境界部のみでなく変形部分の何れの場所においても同様に連続性は確保されている。

スクリーン座標の (x, y) 空間からパラメータ空間への逆写像を考える。対象画像上にいくつかの隣接するBezier曲面をオーバーラップするが、ここでは1つのBezier曲面について説明する。3次Bezier曲面内のある点Pはパラメータ (u, v) を用いて、次式によって表現される (式(1)参照)。

$$x(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 x_{ij} B_i^3(u) B_j^3(v)$$

$$y(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 y_{ij} B_i^3(u) B_j^3(v) \quad (5)$$

ここで、 (x_{ij}, y_{ij}) はBezier曲面の制御点 (すなわち格子点) の座標で、 B は前述のBernsteinの多項式である。

Bezier曲面は、一般に (u, v) を与えると (x, y) が求まるが、ここでは (x, y) から逆に (u, v) を計算する必要がある。 (x, y) 座標から (u, v) 座標を得ることは、レイトレーシング法と等価である。すなわち、点 (x, y) を通過するレイがBezier曲面と交差する点の (u, v) 座標を求めることに等しい。

4.1 逆写像の分類

逆写像は、曲面を含む3次元モデルの隠面消去およびテクスチャマッピングに必要な技術である。すなわち、3次元モデルの場合、投影面上のある点に対応する曲面上のパラメータを求め、これからその点の (x, y, z) 座標を計算し、可視かどうかが判定される。したがって、2次元の場合も、曲面の隠面消去の観点から分類すると、逆変換には次の方法が考えられる。

1) 双1次曲面近似による逆写像

曲面を多角形近似した隠面消去に相当する考え方である。すなわち、曲面を細分割し、いくつかの4辺形の集

合で近似する。この場合、細分割されたパッチは双1次曲面 (境界は直線) である。したがって、 (u, v) 座標は2次式で解くことができる⁶⁾。しかしながら、この方法では、細分した曲面 (ねじれた多角形) どちらの境界は C^0 接続であるため、画像が不連続になる (すなわち折れ曲がる)。分割数を多くすれば、不連続性は減少するが、この場合記憶容量が増加し、またどの4辺形に計算点が含まれるかのチェックに時間を要す。

2) レイトレーシング法による逆写像

曲面に対するレイトレーシングには、大別して細分法と数値解法がある。前者の方法にはWhittedの方法⁷⁾がある。この方法の場合、精度を上げるにはかなりの分割回数が必要になる。例えばサイズが 512×512 の画像に1つの曲面がある場合、 (u, v) を求めるには1点につき20回以上の分割が必要である。後者の場合、Kajiya⁸⁾の方法がある。この方法においては、双3次曲面の場合18次式を解く必要がある。その後いくつかの方法が開発されたが、効率のよい方法として、近年著者ら¹⁾の方法が開発された。この方法は、有理Bezier曲面に対するレイトレーシングであり、1次式を用いた繰り返し法によって (u, v) を求めることができる (この方法をBezier Clippingと呼ぶ)。

レイトレーシング法は精度がよい反面、計算時間を要す欠点がある。

3) スキャンライン法による逆写像

曲面に対するスキャンライン法として、多角形への分割法がある。代表的な方法として、Laneら⁹⁾の方法がある。この方法では、各走査線に対してダイナミックに、走査線と交差する曲面を多角形近似する方法である。しかしこの方法では、近似多角形間に隙間を生じる欠点がある。この欠点を解消するため、西田らの方法¹⁰⁾が提案された。この方法では、多角形でなく曲線の境界を持つサブパッチに分割された。なお、このサブパッチは平坦とみなされるまで分割された。サブパッチの境界と走査線との交点が求められ、交点間の画素における (u, v) 値は線形補間によって求められた。

この方法は、レイトレーシング法より処理時間が少ない利点を持つ。

4.2 等パラメータ線との交点を用いたマッピング

本稿では、スキャンライン法に属する方法を用いる。走査線上の総ての画素について逆写像を行なうのではなく、数点おきにパラメータ座標を求め、それらの間は補間する方式とした。

図3に示すように、走査線上に存在するサブパッチを抽出する。次に、サブパッチの境界曲線と走査線との交点を求める。走査線を $y=y_s$ とすると、式(5)の y 座標

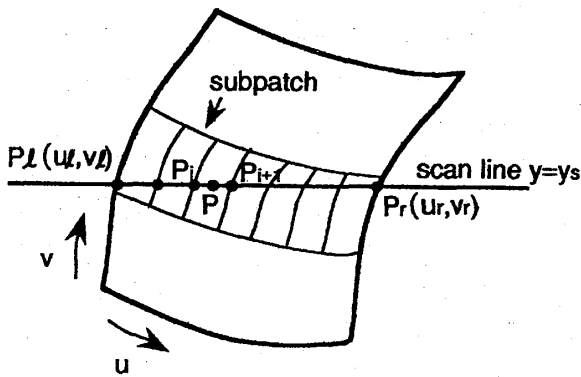


Fig.3 Inverse mapping

が走査線上に存在する条件から、サブパッチは次式によって求められる。

$$\sum_{i=0}^3 \sum_{j=0}^3 y_{ij} B_i^3(u) B_j^3(v) = y_s$$

すなわち、

$$\sum_{i=0}^3 \sum_{j=0}^3 d_{ij} B_i^3(u) B_j^3(v) = 0 \quad (6)$$

ここで、 $d_{ij} = y_{ij} - y_s$ であり、走査線と制御点との距離に相当する。式(6)を満たす u, v の区間はBezier Clipping法^{1), 2)}によって求めることができるので、この区間以外を切り捨てる。サブパッチの境界線は3次Bezier曲線であるから、これらの曲線と走査線との交点を求める。この際、切捨てたことによって生じた境界線とは必ず交差しないので、他の境界線(一般に2曲線)のみと交差判定すればよい。

サブパッチの交点を x_l, x_r 、それらの交点でのパラメータを $(u_l, v_l), (u_r, v_r)$ とすると、この $u_r - u_l$ と $v_r - v_l$ との大きい方の成分を等分割する。例えば、 u 成分が大きい場合、 du 間隔で $u = u_l (= u_{l-1} + du)$ の等パラメータ曲線を求め、この曲線と走査線との交点を求める。すなわち、この交点での v を求め、これから x 座標を式(5)から求める。これを v_i とし、その交点を x_i とすると、 x_i と x_{i+1} との間の点 x における (u, v) は次式の線形補間によって求められる。

$$\begin{aligned} u &= u_i + t(u_{i+1} - u_i) \\ v &= v_i + t(v_{i+1} - v_i) \end{aligned} \quad (7)$$

ただし、

$$t = (x - x_i) / (x_{i+1} - x_i), \quad 0 < t < 1$$

なお、 du はサンプリング間隔が2,3画素程度になるように $(x_r - x_l)$ から算出する。等パラメータ曲線は3次Bezier曲線であるから、走査線と等パラメータ曲線との交点での v は次式を解くことにより求まる。

$$\sum_{j=0}^3 d_j B_j^3(v) = 0 \quad (8)$$

ここで、 $d_j = y_j - y_s$ であり、 y_j は等パラメータ曲線の制御点の y 座標である。

式(8)を満たす v はBezier Clipping法によって求められる。ここで、サブパッチの境界線はもとの曲面より小さくなっているから、境界曲線は直線に近くっており、交点計算の繰り返し数は少なくてもよい。

この方法によって18次式の問題が3次の問題に低減できる。なお、等パラメータ曲線と走査線との交点では精度よくパラメータ値が求まるが、他の区間では線形補間されるから、精度はサンプリング間隔に依存する。

5 アニメーション

スムーズなアニメーションを行なうには、最初の画像と変形後の画像との間を数ステップで変形させる。同一図形への形状変形の場合(AからA'の場合)、時刻(各ステップ)により、形のみが変形されるが、別の形状への変換の場合(AからBへの変形)、形状の変化のみでなく色の変化もともなり。

形の変形については、初期値(画像A)の格子点の位置と最終位置(画像A'またはB)との間をキーフレーム法のように線形補間する。また、色は同一パラメータ値に対応する色を補間(混色)する。

変形するための格子点の変形軌跡は線形としたが、変形する速度はスクリーンのどの点でも一定とは限らない。すなわち、画像中の1部分から順に変形していくことを可能とした。例えば顔の場合、上から順に変形すると効果的である。格子点は u, v に方向づけられているから、 u 方向(あるいは v 方向)に順に変形を進ませることができる。したがって、速度は u, v の関数(1次または3次のBezier関数)として与えられるようにした。

一般に、画像中の対象となる画像のみを変形し、他の部分すなわち背景画像は変化しない。そのため変形したい領域はマスク処理して抽出しておき、変形後に背景画像と合成¹⁰⁾する。

したがって、本システムでは、変形前の画像、変形後の画像に加えて背景画像が入力として必要である。

変形の結果をアニメーションで確認することは実用上重要である。本システムでは、変形していく過程の各画像を保存することができる。ファイルに保存された画像はグラフィックワークステーションを用いて高速表示することにより、リアルタイムのアニメーションが実現できる。

6 適用例

図4は同一オブジェクトの変形例であり、旗が変形したものである。(a)は原画像(福山大学の校旗)で、(b)は変形後背景と合成したものである。図5は、異なるオブジェクトへの変形例で、自動車からハンディタイプの掃除機への変形を示す。(a)(c)が原画像で、(b)は一部が変形している。なお、原画像は、スキャンライン法による曲面表示プログラム¹¹⁾を用いて作成したものである。

(d),(e),(f)は虎から馬への変形を示す。(e)は変形途中(前から変形していく)の画像である。これらの図はアニメーションの1コマである。

本システムでは、変形の計算にはワークステーション NEC EWS4800/260を使用した。また、計算後のリアルタイムでのアニメーションにはグラフィックワークステーション IRIS 4D/20Gを用いた。

計算時間は、図4が20秒、図5(b)が18秒、図5(e)が22秒である(画像サイズ500×400)。

7 おわりに

本稿では、2次元画像の自由変形にBezier Clippingを適用する方法を提案し、滑らかな変形を行なうことが可能であることを示した。すなわち、局所的にもあるいは大局的にも変形でき、その導関数の連続性を保持して適用できる。

Bezier Clippingは、(1)隠線消去、(2)隠面消去、(3)影・反射・屈折をとまなり陰影表示、(4)種々の光源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に適用できることを既に示したが²⁾、本稿によってさらにその有効性が実証された。

今後の課題としては、画像処理技術を使い自動的に画像の輪郭を抽出し、その輪郭に沿ったメッシュの自動生成を可能とし、人手を要しないシステムとすることである。また、画像合成時におけるエリアシングの解決も望まれる。

謝辞

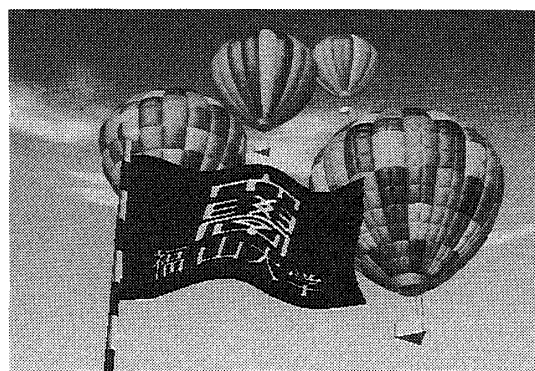
本研究を行なうにあたり、ワークステーションを借用させて頂いたNSS(日本ソフトウェアサービス株式会社)および広島大学工学部中前研究室に感謝致します。また、本システムの基礎実験用プログラムを作成した卒業生の池田正人君(現 日新システムズ)に感謝する。

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).
- 2) 西田: Bezier Clippingおよびその計算機支援形状設計への応用, 福山大学工学部紀要 第13号, pp.45-58, (1991).
- 3) T.Sederberg, E.Greenwood, A Physically Based Approach to 2-D Shape Blending, Computer Graphics, Vol.26, No.2, pp.25-34, (1992).
- 4) T.Sederberg, S.Parry, Free-Form Deformation of Solid Geometric Models, Computer Graphics, Vol.20, No.4, pp.151-160, (1986).
- 5) G.Farin著, 木村, 山口訳: CAGDのための曲線・曲面理論, 共立出版, pp.175-178, (1991).
- 6) A.Glassner, An Introduction to Ray Tracing, Academic Press, p.59-64.
- 7) T. Whitted, An Improved Illumination Model for Shaded Display, CACM, Vol.23, No.6, pp.96-102, (1980).
- 8) J. Kajiya, Ray tracing Parametric Patches, Computer Graphics, Vol.16, No.3, pp.245-254, (1982).
- 9) J. Lane, L.Carpenter, T.Whitted, Scan line Methods for Displaying Parametrically Defined Surfaces, CACM, Vol.23, No.1, pp.23-34, (1980).
- 10) E.Nakamae, K.Harada, T.Ishizaki, T.Nishita, Montage: The Overlaying of The Computer Generated Image onto a Background Photograph, Computer Graphics, Vol.20, No.3, pp.207-214, (1986).
- 11) T.Nishita, K.Kaneda, E.Nakamae, A Scanline Algorithm for Displaying Trimmed Surfaces by using Bezier Clipping, The Visual Computer, Vol.7, No.5, pp.269-279, (1991).

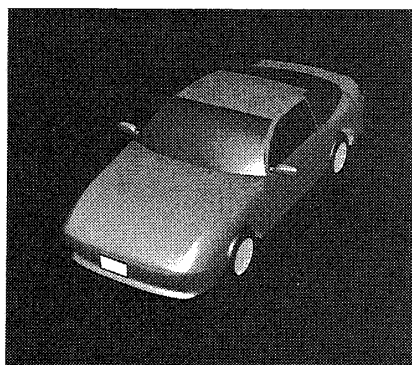


(a)

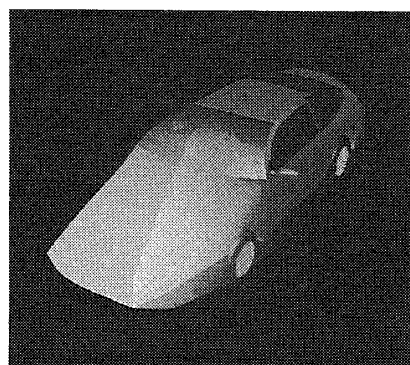


(b)

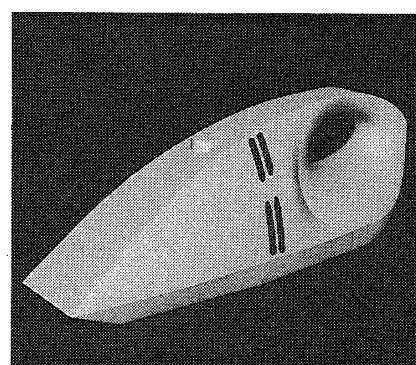
Fig.4 Examples



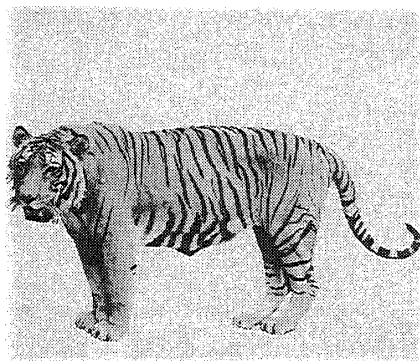
(a)



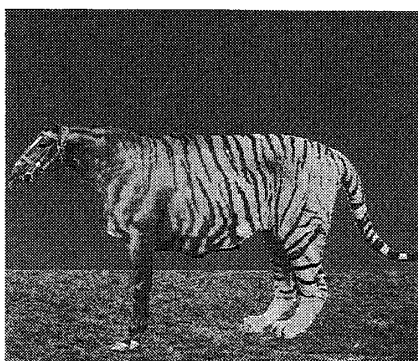
(b)



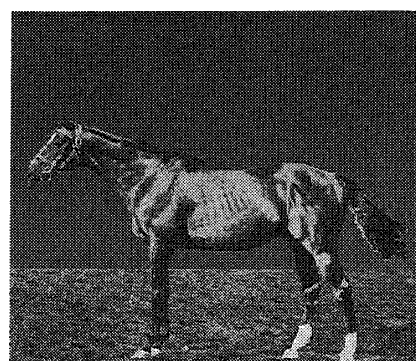
(c)



(d)



(e)



(f)

Fig.5 Examples