

軸による自由形状変形の制御

柿本 正憲 村上 公一
富士通研究所

自由形状変形 (FFD) を対話的に行なうために、物体モデルの中心軸を利用した変形方式を考案した。モデルの方向性を考慮して主軸を与え、FFD座標系を設定する。そしてその主軸を曲げてBézier曲線セグメントとすると、モデルも主軸に沿って滑らかに変形する。また、最初から曲線となっている主軸を与える一般化FFD座標系を設定することによって、曲がったモデルをまっすぐに復元することが可能である。さらに、主軸と垂直な平面内での回転及びスケーリング変換によってひねりやふくらみの変形を行なうことができる。本方式を用いたリアルタイムの会話型モデラを試作してその有効性を確認し、仮想生物モデルのデザインに適用した。

Control of Free Form Deformation using an axis

Masanori Kakimoto Koichi Murakami
Fujitsu Laboratories

1015 Kamikodanaka, Nakahara-ku, Kawasaki 211, Japan

A method to control Free Form Deformation (FFD) of 3D geometric models by using the axis of the model is described. An FFD coordinate system is defined by placing an axis in consideration of the direction of the model. When the axis is deformed into a Bézier curve, the model is smoothly deformed along the curve. Reconstruction of the deformed model is possible by defining the initial axis as a curve, which forms a generalized FFD coordinate system. Rotation or swell deformation can be performed by matrix transformation in a plane perpendicular to the axis. Effectiveness of the method was verified by implementation in an interactive modeler. The method was applied to the design of virtual creature geometric models.

1 はじめに

人工的な構造物に比べて、生物の形状を作るためには、変形を容易に扱うことができることが重要なポイントとなる。通常、物体の変形を扱うには様々な方法がある。それらは、複数モデルの補間によるいわゆるメタモルフォーゼの方法、形状をパラメトリックに定義してそのパラメータを変化させる方法、有限要素法のようなシミュレーションによる方法、自由形状変形による方法などである。

メタモルフォーゼによる方法は、現実には起こりえないような不思議な効果を出すには良く、扱いも簡単である。しかし、変形の自由度が少ない上、生物の姿勢や動作に伴うような自然な変形はできない。パラメータを用いる方法は、物体モデルを定義するのがプログラムを書くのと同じような作業となり時間がかかる上、変形の自由度も限られてくる。シミュレーションによる方法はリアリステイックな変形を行なわせることができる反面、処理時間がかかる、モデル化が難しく少し複雑な物体になると適用できないなどの問題点がある。また、モデルとしての生物は柔らかく自律的に動くので、剛体に適用できるような単純な物理法則は当てはめられない。自由形状変形による方法は、適用できる物体も任意で変形の自由度も大きいのが、変形を決めるために空間内に配置する制御点が多くなると扱いが困難になる。

生物のモデリングを行なうためにはシミュレーションよりもデザイナーの操作による方法が適しているため、自由形状変形による方式が最も有望である。筆者らは、インタラクティブな操作でリアルタイムの自由形状変形を実現するために、軸を用いる自由形状変形手法を考案した。

2 自由形状変形 (FFD)

自由形状変形 (FFD: Free Form Deformation) は、パラメトリックな曲線や曲面を決定する制御点の適用範囲を有限空間内の任意の点に拡張することにより、三次元形状を滑らかに変形させる手法で、1986年に Sederberg らによって考案された^{[1][2]}。

三次元物体の自由形状変形では、変形させたい物体を囲む $(l+1)(m+1)(n+1)$ 個の格子点を用意する。ここで、格子点が形成する平行六面体のローカル座標系を考える。これをFFD座標系と呼ぶ。FFD座標系は、原点 X_0 を平行六面体の一頂点に置き、 X_0 から発する三つ

の辺に沿ってベクトル S, T, U を決め、これをローカル座標軸とする。格子点は S に沿って $(l+1)$ 個、 T に沿って $(m+1)$ 個、 U に沿って $(n+1)$ 個、それぞれ等間隔に並ぶものとする。この座標系で (s, t, u) となる点の世界座標を X とすると、

$$X = X_0 + sS + tT + uU \quad (2.1)$$

となる。逆に X が与えられたとき、FFD座標は、

$$s = T \times U \cdot (X - X_0) / T \times U \cdot S \quad (2.2)$$

$$t = U \times S \cdot (X - X_0) / U \times S \cdot T \quad (2.3)$$

$$u = S \times T \cdot (X - X_0) / S \times T \cdot U \quad (2.4)$$

によって与えられる。平行六面体の内部は、FFD座標系で、 $0 < s < 1, 0 < t < 1, 0 < u < 1$ で表現される。また、任意の格子点 P_{ijk} は、FFD座標系で $(i/l, j/m, k/n)$ 、世界座標系では、

$$P_{ijk} = X_0 + (i/l)S + (j/m)T + (k/n)U \quad (2.5)$$

で与えられる。ここで、 $0 \leq i \leq l, 0 \leq j \leq m, 0 \leq k \leq n$ である。

自由形状変形は、格子点 P_{ijk} を変位させることによって行なう。変形後は、FFD座標系の点 (s, t, u) は、Bernstein多項式を評価する次の式によって世界座標系に変換される。

$$X = \sum_{i=0}^l B_{i,l}(s) \left[\sum_{j=0}^m B_{j,m}(t) \left\{ \sum_{k=0}^n B_{k,n}(u) P_{ijk} \right\} \right] \quad (2.6)$$

ここで、 $B_{i,n}(t)$ はBernstein基底関数

$$B_{i,n}(t) = \binom{n}{i} (1-t)^{n-i} t^i \quad (2.7)$$

である^[3]。自由形状変形によって、FFD座標系を形成していた平行六面体の各辺はBézier曲線へ、各面はBézier曲面パッチへと変形される。

式(2.6)は幾何学的には次のような意味を持つ。まず全格子点を U 方向につながった $(l+1)(m+1)$ 本の制御ポリゴンとみなし、 $(l+1)(m+1)$ 本の n 次Bézier曲線を考え、それらをすべてパラメータ u の値で評価して曲線上に $(l+1)(m+1)$ 個の点を求める。さらに、求めた点を T 方向につながった $(l+1)$ 本の制御ポリゴンとみなして $(l+1)$ 本の m 次Bézier曲線を考え、それらをすべてパラメータ t の値で評価して $(l+1)$ 個の点を求める。さらにその $(l+1)$ 個の点によって決定される l 次のBézier曲線を考え、パラメータ s の値で評価する。こうして求められた点を変形後の (s, t, u) の世界座標となる。

二次元平面上での自由形状変形の例を図1に示す。

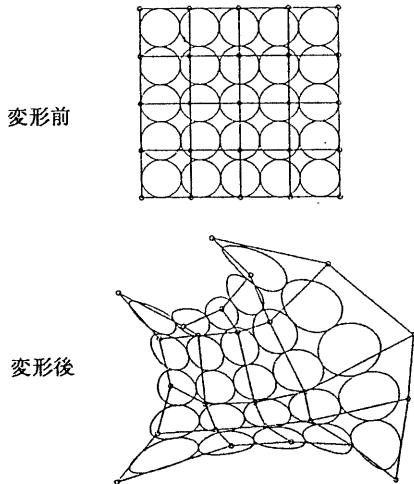


図1 二次元平面における4次のFFDの例

自由形状変形の最大の特徴は、変形パラメタがモデルとは独立であるという点にある。適用範囲が広く、モデルの形状に依存しないため、既存の三次元物体を利用できる。また、変形の自由度が非常に大きい。さらに、自由形状変形は一次元のBézier曲線の三次元空間への拡張と考えられるので、幾何学的にはBézier曲線の持つ扱いやすい性質を継承している。例えば、連続性のための条件や、変動減少性から来る制御の安定さ、凸閉包性などはBézier曲線と同様である。

3 軸を利用した自由形状変形

3-1 自由形状変形の問題点

自由形状変形を実際の三次元物体のモデリングに適用しようとすると、変形の自由度の大きさが逆に扱いにくくなる。三次元の格子状に配置された制御点は数が多く、また、思うような位置に動かすのはかなり難しい。

モデリングを行なう対象にもよるが、実用的な変形モデリングでは、曲げる・縮める・延ばす・ひねる・凹凸を作るなど操作の種類は限られたものであり、三次元の格子点を設けるほどの自由度は必要としない。

そこで筆者らは、変形の自由度を実用的な範囲に制限することによって操作をやすくする軸変形モデリングの手法を考案した。

3-2 FFD座標の計算

軸変形モデリングでも一般の自由形状変形と基本的には同様であるが、便宜上直方体のFFD座標系(t, u, v)を考える。 t は、主軸方向 T の座標値、 u, v は主軸に垂直な平面方向 U, V の座標値である。変形の主軸は、例えば直方体の対称軸となるような線分を選ぶ。このようにFFD座標を設定すると、原点 $X_0(0, 0, 0)$ は主軸の一端となる。ここで、主軸のもう一端の座標が $(1, 0, 0)$ となるように T の大きさを正規化する。 U, V の大きさは、便宜上世界座標の単位で1になるように設定する。ここで、

$$\mathbf{X} = \mathbf{X}_0 + t\mathbf{T} + u\mathbf{U} + v\mathbf{V} \quad (3.1)$$

となる。逆に、世界座標 \mathbf{X} が与えられたとき、FFD座標(t, u, v)は直方体を形成するので(2-2)~(2-4)よりも簡単に、

$$t = (\mathbf{X} - \mathbf{X}_0) \cdot \mathbf{T} / (\mathbf{T} \cdot \mathbf{T}) \quad (3.2)$$

$$u = (\mathbf{X} - \mathbf{X}_0) \cdot \mathbf{U} \quad (3.3)$$

$$v = (\mathbf{X} - \mathbf{X}_0) \cdot \mathbf{V} \quad (3.4)$$

によって与えられる。

図2に軸変形モデリングにおけるFFD座標系を示す。主軸は、端点を含めた、等間隔に配置された四つの制御点 $P_i (i=0, 1, 2, 3)$ によって区切られている。ここで、 $P_0 = X_0$ である。図2では、主軸に沿った変形が3次のBézier曲線の拡張として行なわれる。

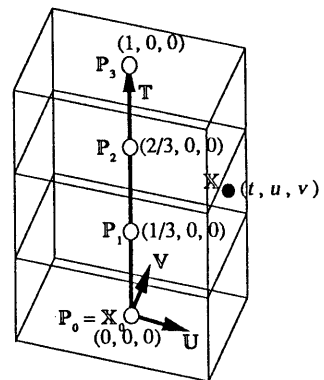


図2 FFD座標系

3-3 フロア座標系の設定

自然な曲げ変形を行なうには、初期状態の主軸 $P_0 \sim P_3$ を曲げたときに、各制御点における T, U, V の方向も同時に曲がる必要がある。そこで、各制御点 P_i に対して個別に T_i, U_i, V_i を設定してやらなければならない。このとき、 P_i を原点とし、法線ベクトルが T_i で、

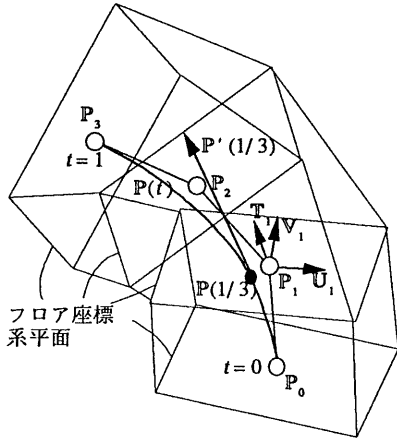


図3 フロア座標系

U_i, V_i を基本ベクトルとするローカル平面座標系を考えると、変形前はこの平面が直方体の建物の各フロアのように見えるので、これをフロア座標系*i*と呼ぶことにする(図3)。

フロア座標系平面は主軸と垂直になると考えるのが自然である。そこで、フロア座標系平面*i*の法線ベクトル T_i として、主軸の方向ベクトルの $t = i/n$ での値を用いる。図2に示すようなFFD座標系で、主軸上の制御点 P_i を移動させると主軸はBézier曲線

$$P(t) = \sum_{i=0}^n B_{i,n}(t) P_i \quad (3.5)$$

となる。ここで、主軸の方向ベクトルは、 $P(t)$ の一次微分によって得られる。Bézier曲線の場合には、次式のようにいわゆるHodographを計算することによっても求まる。

$$P'(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t) n(P_{i+1} - P_i) \quad (3.6)$$

フロア座標系*i*における主軸方向ベクトル T_i は、式(3.6)で $t = i/n$ として、

$$T_i = P'(i/n) / |P'(i/n)| \quad (3.7)$$

によって求める。

次に、 U_i, V_i であるが、一つの考え方として、 $P(t)$ の二次微分をとってこれを例えば、

$$U_i = P''(i/n) / |P''(i/n)| \quad (3.8)$$

などとすることができる。しかし、この方法では、二次微分が0になる変曲点では方向が定まらない。また、変曲点付近で $P''(t)$ の方向が反転した場合、ある*i*

について U_i と U_{i+1} が反転するために、制御点 P_i と P_{i+1} の間で急激なねじれが生じる。

二番目の方法として、単純に、例えば U_i を、 T_i に垂直なベクトルのうち*x*成分が常に0となるものとするというように求める方法がある。この方法でも、 T_i が*x*軸に平行となる場合には特殊処理が必要となる。実際のシステムの実現に当たっては、この二番目の方法によって U_i を計算した。

V_i は、 T_i と U_i の外積をとることによってその方向を求めることができる。

なお、軸に沿って不自然なねじれを生じないローカル座標を設定するアルゴリズムは文献[3]で示されている。

3-4 軸変形モデリング

軸変形モデリングは、変形の対象とするモデルに対して、一定方向の主軸を重ね合わせ、その主軸を折り曲げることによってモデルを変形する手法である。このため、ちょうどBézier曲線をデザインするような感覚で三次元物体の変形ができる。

一般的な自由形状変形における式(2.6)に対応する軸変形モデリングの変換式は、同次座標系を用いて、

$$X = \sum_{i=0}^n B_{i,n}(t) ((u \ v \ 1) F_i) \quad (3.9)$$

と表現される。ここで、 (t, u, v) は注目する点のFFD座標である。また、 F_i は 4×3 の座標変換行列で、フロア座標系*i*の上にある点 $[u \ v]$ を、世界座標に変換する。

幾何学的には、三次元の自由形状変形のうち、主軸に垂直な二つの次元(フロア座標平面)については行列による幾何変換を用い、残りの一次元(主軸方向)についてだけBernstein多項式によるBézier曲線の評価を行なう。

フロア座標系*i*は原点が P_i で基本ベクトルが U_i 及び V_i であることを考慮すると、式(3.9)は次のように書ける。

$$X = \sum_{i=0}^n B_{i,n}(t) ((u \ v) \begin{bmatrix} U_i \\ V_i \end{bmatrix} + P_i) \quad (3.10)$$

さらに、それぞれのフロア座標系における回転及びスケール変換を挿入すると、

$$X = \sum_{i=0}^n B_{i,n}(t) ((u \ v) M_i \begin{bmatrix} U_i \\ V_i \end{bmatrix} + P_i) \quad (3.11)$$

となる。ここで、 M_i は、フロア座標系 i で原点 P_i を中心として回転とスケーリングを行なう 2×2 の行列である。すなわち、

$$M_i = \begin{pmatrix} s_i \cos r_i & s_i \sin r_i \\ -s_i \sin r_i & s_i \cos r_i \end{pmatrix} \quad (3.12)$$

となる。

式(3.11)(3.12)で表現される変形の幾何学的な意味は次のようになる。モデル全体を曲げる・延ばす・縮めるという操作は、Bézier曲線となる主軸を決定する制御点 P_i の位置を動かすことによって実現できる。モデルの一部分を主軸の周りにひねる操作は、 r_i を変えることによって実現できる。モデルの一部分を太らせたり細くしたりする変形操作は s_i の値を変えることによって実現できる。

3-5 一般化FFD座標系

3-2では、変形前のFFD座標系として、直方体の形状を設定したために、世界座標からFFD座標への変換は、式(3.2)~(3.4)によって簡単に求めることができる。このようなFFD座標系は、もともとまっすぐのものを曲げていくような変形には適してしる。しかし、実際のモデリングでは、もともと湾曲していたり、曲がりくねっていたりするモデルを扱いたい場合もある。このようなモデルに対しては、初期状態がまっすぐの主軸を設定しても思ったような変形の制御を行なうことはできない(図8-a、図8-b参照)。

そこで、もともと曲がったモデルにはそれに沿った主軸を最初に設定できる一般化FFD座標系を考案し、次に示すようなアルゴリズムで点の座標値 (t, u, v) を求める。

世界座標の点 X の一般化FFD座標 (t, u, v) を求める問題は、式(3.10)の逆変換と考えられる。式(3.10)を展開すると、

$$X = \sum_{i=0}^n B_{i,n}(t)uU_i + \sum_{i=0}^n B_{i,n}(t)vV_i + \sum_{i=0}^n B_{i,n}(t)P_i \\ = uU(t) + vV(t) + P(t) \quad (3.13)$$

となる。ここで、 $U(t)$ 及び $V(t)$ はそれぞれ U_i 及び V_i を制御点とするBézier曲線である。 $P(t)$ を移項すると、

$$X - P(t) = uU(t) + vV(t) \quad (3.14)$$

が成り立つ。この式の両辺のベクトルと、ベクトル $U(t) \times V(t)$ との内積をとると、

$$(X - P(t)) \cdot (U(t) \times V(t)) \\ = (uU(t) + vV(t)) \cdot (U(t) \times V(t)) \\ = uU(t) \cdot (U(t) \times V(t)) + vV(t) \cdot (U(t) \times V(t)) \\ = 0 \quad (3.15)$$

となって、 u と v を消去することができ、 t についての方程式

$$f(t) = (X - P(t)) \cdot (U(t) \times V(t)) = 0 \quad (3.16)$$

が得られる。ここで主軸 $P(t)$ が例えば3次のBézier曲線であるとする、 $U(t)$ も $V(t)$ も3次のBézier曲線となるから、評価式 $f(t)$ は t についての27次式となる。これを解析的に解くのは困難だから、Newton法によって解くことにより、 t の値を求める。

u の値は、式(3.14)の両辺と $P(t) \times V(t)$ との内積をとることによって v を消去すれば、

$$u = (P(t) \times V(t)) \cdot X / (P(t) \times V(t)) \cdot U(t) \quad (3.17)$$

として求められる。同様に、

$$v = (P(t) \times U(t)) \cdot X / (P(t) \times U(t)) \cdot V(t) \quad (3.18)$$

となる。

なお、方程式(3.16)は一般には複数解を持つ。その場合は座標値は決定できない。しかし、曲線 $P(t)$ の曲がり具合が緩やかな場合は解は一意に求まり、座標値を決定することができる。

4 リアルタイム軸変形モデリングの実現

前章で説明した軸変形の手法を実際のシステムとしてインプリメントし、実時間で操作できる軸変形モデラを実現した。

4-1 システムの概要

軸変形モデラは、会話型の変形専用のモデラで、既存の三次元物体モデルを入力とし、マウスを使って制御ポリゴンの形状を調整することによってモデルを変形して出力する。変形機能のほか、実時間の視点変更・三面図表示機能・テクスチャマッピング・中割りアニメーションなどの基本的機能を備えている。

ハードウェアはシリコングラフィックスのIRIS-4D/VGXを使用している。

4-2 変形機能

変形のアプレーションでは、まず物体モデルに合わせて制御ポリゴンを発生させる。図4は、3次のFFDを

行なう場合の制御ポリゴンを物体モデルとともに表示している様子である。

図4, 5, 6, 8で用いた物体モデルは、仮想生物システム⁴⁾で用いたもので、11×23のポリゴンメッシュで表現されており、表面にはテクスチャマッピングを施してある。

4-2-1 主軸の曲げ

直方体状の制御ポリゴンの中心を貫く主軸は、3次の変形の場合四つの制御点をつないだものとなる。この制御点のうちの一つをカーソルでピックしてドラッグするとその制御点が動き、制御ポリゴンの形状が3-4で示したアルゴリズムに従って変更される。図5に曲げ変形を行なっている様子を示す。

主軸方向の延ばし変形や縮み変形も、制御点の操作によって同様に行なう。

4-2-2 ひねり変形

各制御点を通り主軸に垂直な平面、すなわちフロア座標系の平面が、空間内で長方形として示されている。この長方形を単にフロアと呼んでいる。あるフロアの頂点をピックしてドラッグすることによって、そのフロアを制御点の周りに回転させることができる。同時にモデルにはひねり変形が与えられる。このオペレーションは、式(3.12)で角度 r_i を変化させることに相当する。図6にひねり変形の例を三面図と透視図で示す。

4-2-3 太さ変形

フロアの辺をピックしてドラッグすることによってフロアの大きさを変えることができる。同時に物体の一部分を主軸周りに太くしたり細くしたりすることができる。このオペレーションは、式(3.12)でスケール s_i を変化させることに相当する。図7-a及び図7-bに太さ変形の例を示す。

4-3 一般化FFD座標系による変形モデルの復元

図8-aで示すような、もともと変形された物体モデルが入力として与えられた場合、自動生成した制御ポリゴン(図8-aのもの)を初期状態とすると、モデルをまっすぐ伸びるように復元しようとしても、図8-bのように不自然な歪みが生じてしまう。

そこでモデルの変形を行なう前に、制御ポリゴンを

モデルの形状に合わせて変形してやる。オペレーション方法は4-2-1で示した方法と同様である。位置合わせが終わったら(図8-c)そこで初めて、3-5で述べた方法を用いて各頂点の一般化FFD座標 (t, u, v) を計算し、変形可能な状態にする。その後変形を行なって形状を復元した結果が図8-dである。

4-4 アニメーション機能

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。FFDモデラでは、前述のような変形モデリングの機能を用いて作った物体モデルを適宜保存しておけば、あとでそれらをキーフレーム物体として編集して簡単にアニメーションを作ることができる。

4-5 応答性

ほとんどの操作において実時間の応答が得られた。マウスを動かしてドラッグを行なうと物体の変形も同時に追従する。最も時間のかかる操作は、モデルの全頂点に対する一般化FFD座標の計算であるが、11×23ポリゴンからなる物体に6次の一般化FFD座標計算を施したときでも1秒以内であり、ほぼリアルタイムといつてよい。

5 まとめ

軸を用いた自由形状変形を提案し、いろいろな変形操作を実現できることを示した。また、提案手法をインプリメントした実時間のモデラを試作し、インタラクティブな操作による軸変形モデリングの有効性を確認した。

謝辞

FFDの考案者でありその概念について御指導下さったブリガムヤング大学のシーダバーグ教授に感謝いたします。また、仮想生物のオリジナルのデザインを行なっていただいた日本電子専門学校の野地先生、日頃から議論等を通じて御指導を下さる、大阪学院大学の大村教授、富士通研究所の森田部長に感謝いたします。

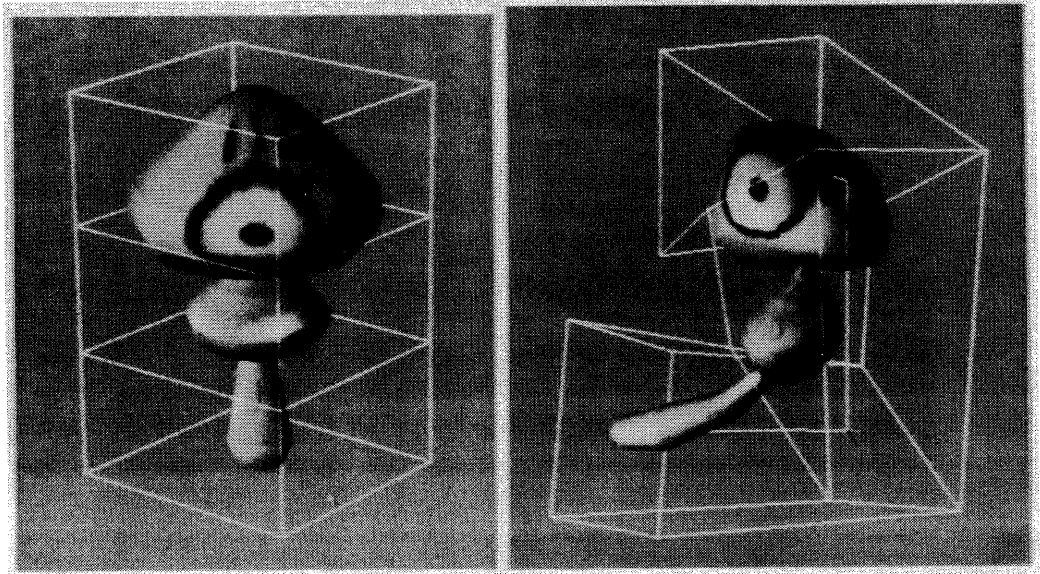


図4 仮想生物モデルと3次のFFD座標系

図5 主軸の曲げによる変形

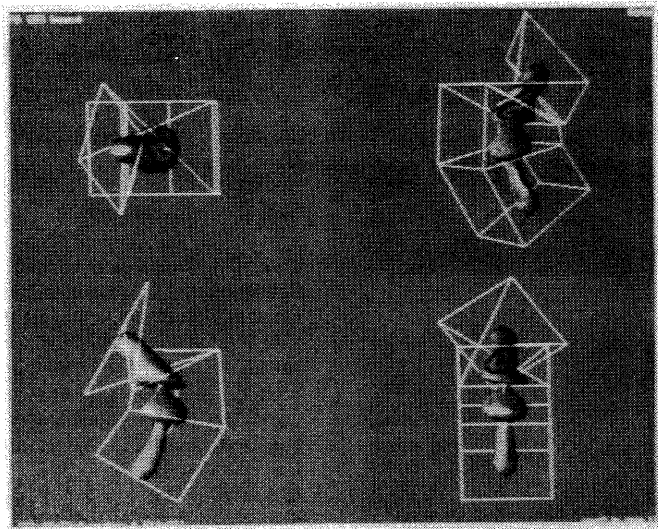


図6 ひねり変形

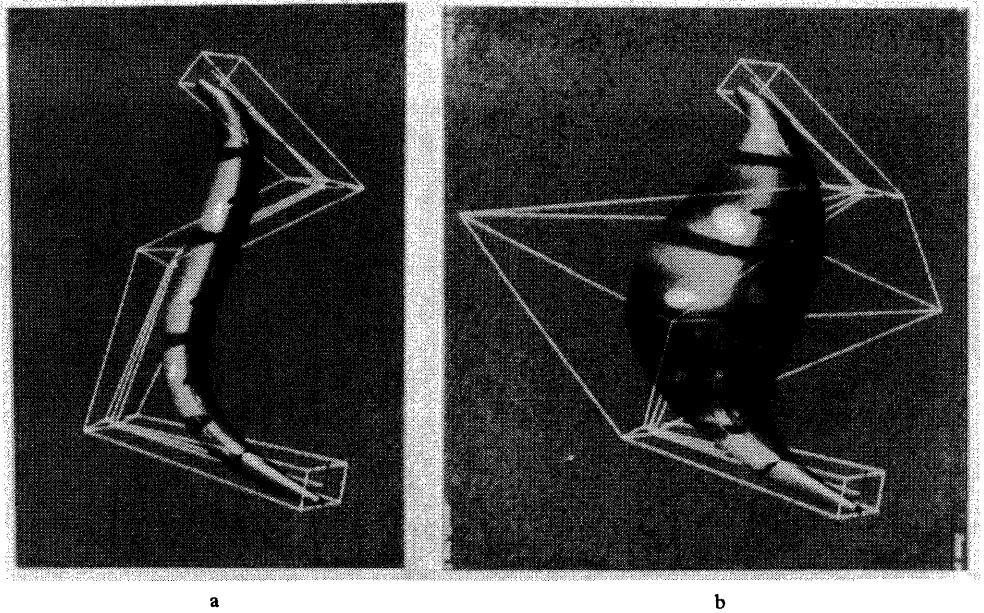


図7 太さ変形

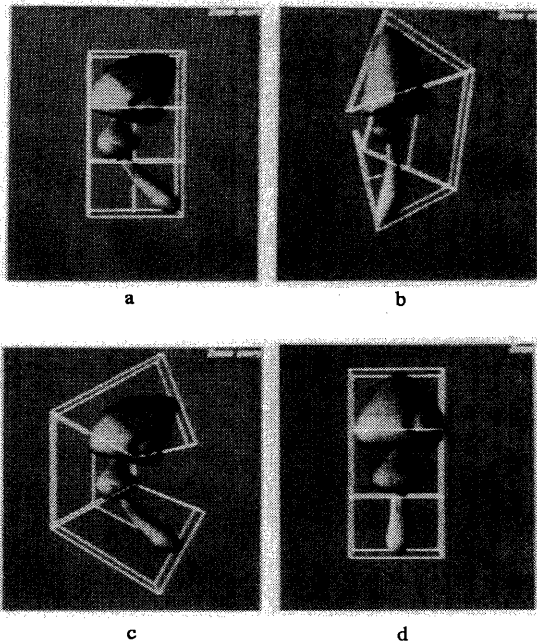


図8 一般化FFD座標系(c)を用いた変形モデル(a)の復元

参考文献

- [1] Thomas W. Sederberg and Scott R. Parry, Free Form Deformation of Solid Geometric Models, *Computer Graphics*, vol. 20, No. 4, August 1986, pp.151-160.
(邦訳: "ソリッド幾何モデルの自由形状変形", *PIXEL*, No.95, (1990年8月), pp.137-146.)
- [2] Scott R. Parry, Free-form deformations in a constructive solid geometry modeling system, Ph.D. Dissertation, Department of Civil Engineering, Brigham Young University, April 1986.
- [3] 柿本ほか, 一般化シリンダを用いた枝分れ物体のモデリングとマッピングの手法, 情報処理学会グラフィックスとCAD研究会資料, 89-CG-39-7, 1989年7月.
- [4] 林ほか, 仮想生物の行動生成モデル, 情報処理学会グラフィックスとCAD集中研究会資料 (掲載予定), 1991年8月.
- [5] 山口富士夫, コンピュータディスプレイによる形状処理工学(III), 日刊工業新聞社, 1982年.