

楕円球メッシュを用いた3次元モーフィング

湯本麻子 鈴木香緒里 佐々木繁

asako@flab.fujitsu.co.jp

富士通研究所 メディア処理研究部

〒211 川崎市中原区上小田中1015

近年、映画やCMにおいて、3次元CGの動き表現の1つであるモーフィングが広く使用されている。このモーフィングの処理対象を2次元画像から3次元形状に拡張した3次元モーフィングは、異形状間の頂点座標補間として実現できるが、各形状の頂点数や面数が異なる時は、頂点同士の対応づけを行うのが困難である。筆者らはこの課題を解決するため、同じ楕円球メッシュをモーフィングさせたい各形状に近似し、その近似形状同士を線型補間する方法を提案する。本論文では、楕円球メッシュを使って近似形状を作成する方法と、それを利用して実際にモーフィングを行った例を示し、その有効性を考察する。

3 DIMENSIONAL MORPHING USING OVAL-SPHERICAL MESH

Asako YUMOTO, Kaori SUZUKI, and Sigeru SASAKI

Media Information Laboratory, Fujitsu Laboratories Ltd.

1015, Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa. 211 Japan

Recently many movies, commercial-films, etc. use morphing animation for expressing motion in 3D computer graphics. Though 3D morphing, to extend its treatment from 2D images to 3D shapes, is realised by linear interpolation of vertex points between 2 shapes, it has difficulties in making correspondences, when the number of facets or vertices is different. We suggest the idea of using approximate shapes instead of original "real" shapes, each made by deforming the same oval-spherical mesh. In this paper, we describe how to make approximate shapes, present the 3D morphing example using them, and estimate its effectiveness.

1. はじめに

最近の映画やCMでは、物体が変形するアニメーションが盛んに用いられている。

「ターミネータ2」での変形(変身)シーンが話題を呼んだことは、特に記憶に新しい。これらの変形アニメーションを実現する手法の1つであるモーフィングは、既存の2つのオブジェクトを融合するCGにおける動き表現の1つとして研究されており[1]、パソコンレベルまで広がってきたアニメーション制作において、効率の良い映像制作のオーサリングを提供するものと期待されている。

従来のモーフィングは、2次元の画像融合が主流であった。これは、既存の2つの画像の色情報を補間して新しい融合画像を作成し、形状の変形を疑似的に表現する方法である[2,3]。しかし、この方法では視線を動かして自由な方向から融合されていく様子を見ることができないという難点がある。これに対し、3次元に拡張したモデル形状自体を融合変形させる3次元モーフィングは、より自由度の高いアニメーションを実現することができる。

3次元モーフィングは、異なる形状同士の融合形状を得る手法であるが、ポリウムレンダリングを利用して実現する方法が知られている。この方法は、形状変化を体積変化と見なして融合形状を取得するが、ソリッドモデルを使用するためにデータ量が多すぎ、パソコンレベルのシステムでは実現が困難である。また、弾性網を使い、バルーンモデル[4]における形状フィッティングをモーフィング前の形状からモーフィング後の形状へのフィッティングと看做すエネルギー最小化を利用した手法[5]も提案されているが、平衡状態までの計算量が莫大なため処理時間がかかりすぎること、モーフィング前後の形状の特徴点同士の対応が難しいこと、などの課題がある。従来、計算が複雑なため非現実的とされた3次元モーフィングにとって、計算が簡単であることは重要なポイントであり、かつ、形状変化において

アニメータの意図を正しく反映させるため、各形状の特徴点同士がどう対応するか、簡単に細かく指定できなければならない。

我々は、これらの課題を考慮して、最も簡単な融合方法である、各ポリゴン頂点座標同士の線型補間を利用することにした。しかし、異なる形状同士は、面数や頂点数が異なる場合が多く、これらをいかに簡単に対応づけするかが問題となる。そこで我々は、この面数や頂点数が異なる形状間での3次元モーフィングを、楕円球メッシュを変形した近似形状を利用して実現する手法を提案する。

本論文では、近似形状の作成を中心にこの手法の詳細を述べ、実際に適用した実現例を示し、この手法に対する考察を述べる。

2. 3次元モーフィングの概要

頂点座標を線型補間して3次元モーフィングを実現する際に、最も問題になる点は、モーフィング前後の形状(元形状と呼ぶ)の面数や頂点数が異なる場合、どれがどれとモーフィングするのか、頂点同士を1つずつ対応づける必要があるということである。この対応づけは、頂点数を揃えるだけでなく、頂点や面の位置関係を保存するように設定しないと、モーフィング途中で形状がバラバラに飛び散ってしまったり、穴が空いたりして円滑な変形結果を得ることができない。よってこの対応づけは非常に困難な作業であり、このアプローチにおける3次元モーフィング実現の課題であった。

上記課題を克服するため、我々は近似形状という概念を新たに導入し、それぞれ元形状に近似変形させた同じ楕円球メッシュ形状同士を元形状の代わりにモーフィングさせる手法を考案した。これは、図1の3次元モーフィングのフローチャートのように実現する。

図1に示すように、実際の補間計算は、モーフィ

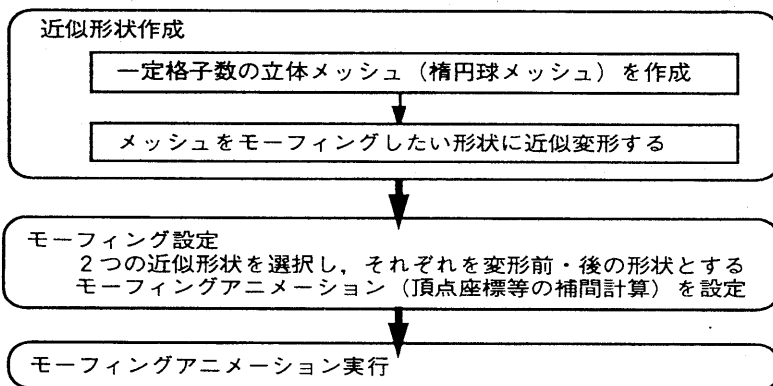


図1 3次元モーフィングのフローチャート

ングさせたい元形状に対してではなく、近似形状に対して行う。すなわちこの手法は、異なる元形状同士を同じ面や頂点構成のものとして扱えるよう対応づけの努力をするのではなく、あらかじめ用意した頂点や面の位置関係の等しい形状をそれぞれ元形状に近似変形させてその代用とし、近似形状同士をモーフィングする。

実際には、元形状は複雑なため、複数の近似形状(パーツ)に分け、各パーツ毎に上記図1の処理を行うことが多い。これを特にパーツモーフィングと呼び、図2にこのパーツ分けの例を示す。

パーツモーフィングでは、元形状同士でパーツ数をそろえる必要がある。これは、髪は髪、顔は顔

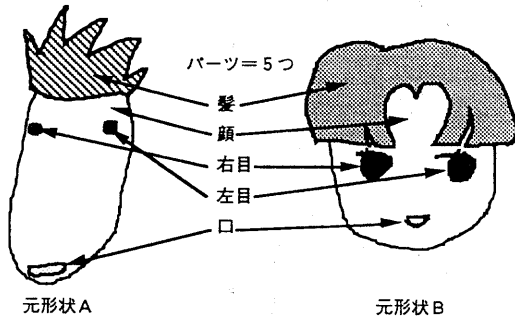


図2 パーツモーフィングのパーツ分け

..というように元形状のパーツ同士をペアにし、それぞれで近似形状を作り、モーフィングアニメを実行するためである。

パーツモーフィングは、パーツ分割しない場合と異なり、パーツの形状変形と同時に、終了後正しい位置にくるようパーツの移動を行う(詳細は4.で述べる)。

近似形状を用いて3次元モーフィングを実現するこの手法には、対応づけの困難はないが、いかに簡単に元形状に似た近似形状を作成するかが重要なポイントとなる。次の章では、この近似形状作成方法について述べる。

3. 近似形状作成

本システムの3次元モーフィングでは、立体メッシュを近似変形した近似形状を使う。立体メッシュを利用するのは、パーツモーフィングを行う際に、パーツが移動(詳細は4.参照)した後にパーツの継ぎ目などで穴が空くのを防ぐためである。今回は立体メッシュの中でも比較的データ保持の簡単な楕円球メッシュを用いる。

近似メッシュの作成、近似変形処理

は、エディタを介したユーザとの対話的な編集で行われ、自動変形機能と手動補正機能を用意することで、負担の省力化を計る。

3.1 楕円球メッシュ作成

近似形状用に、図3のような元形状を包括する楕円球メッシュを自動作成する。自動変形時に正しい変形先を取得するため、近似形状は元形状の外接球として算出作成する。

3.2 楕円球自動変形

作成した楕円球メッシュは、断面である行スライス毎の自動圧縮変形と、手動変形による補正変形を経て、元形状の近似形状となる。

自動変形は、各行スライスの全格子点を、内包される元形状と交差または接触するまで基準となる点(参照点と呼ぶ)の方向に圧縮変形するものである。具体的には、以下の手順で行スライス毎に処理する。

- 1) 行スライスの全格子点の圧縮移動方向を決定する参照点を、あらかじめ指定する。本システムのデフォルトは、その行スライスの全格子点の位置の平均である。
- 2) 各格子点と参照点を結ぶ直線(探索直線)を求める。探索直線は圧縮移動方向を表わし、格子点の移動先はその直線上とする。
- 3) 探索直線と元形状との交点を算出する。交点を複数算出した場合は、格子点の現在位置に最も近い交点を有効とする。
- 4) 交点算出と同時に、交点の属性(色、法線)を算出し、格子点の頂点データとして設定する。算出方法は元形状の属性データ保持状況によって以下の3通りがあり、a→cの順に優先する。法線は、現在の楕円球メッシュの格子点位置座標から算出しても良い。

a. 頂点データがある場合

→ 交点を含む小面の全頂点の属性の平均値。

b. 小面データがある場合

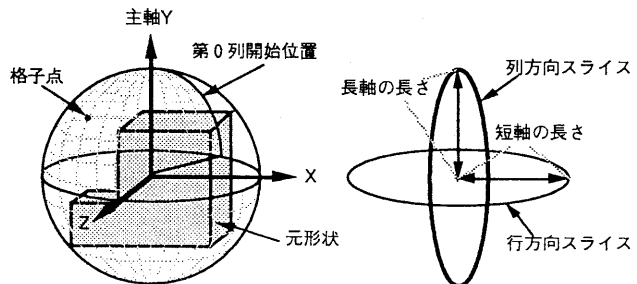


図3 使用する楕円球メッシュの形状

- 交点を含む小面の属性値。
- c. 頂点、小面データがない場合
- 交点を含むオブジェクトの属性。法線はなし。

図4に参照点を利用した自動変形の例を示す。この例では、格子点Aの探索直線と格子点Bの探索直線が等しくなっているが、必ずしも等しい必要はない。探索直線と元形状の交点はM、Nであり、現在

位置に近いものが有効なため、格子点AはMに、BはNに移動する。

元形状の形から、圧縮方向が複数あった方が良くと判断した場合は、参照点を複数使用する。このとき、各行スライスの格子点がどの参照点と対応するかグループ分け（円弧分割）する。

図5は、行スライスを2つに円弧分割する例を示す。左図は円弧分割せず参照点が1つの場合で、全ての格子点は中心の参照点に向けて圧縮変形する。

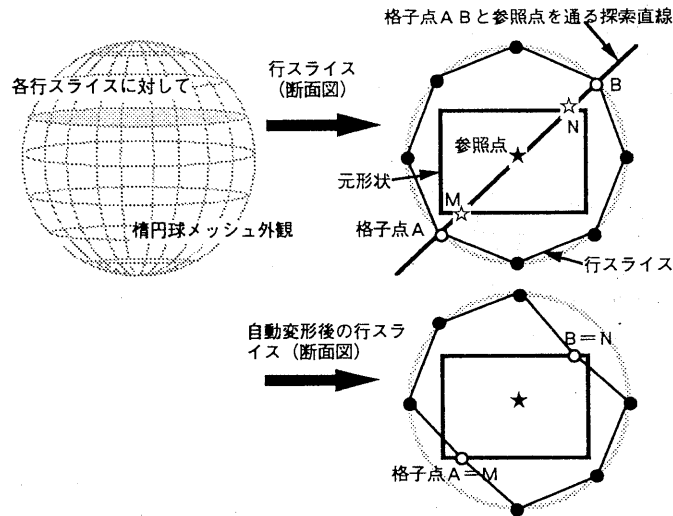


図4 参照点を利用した自動変形例

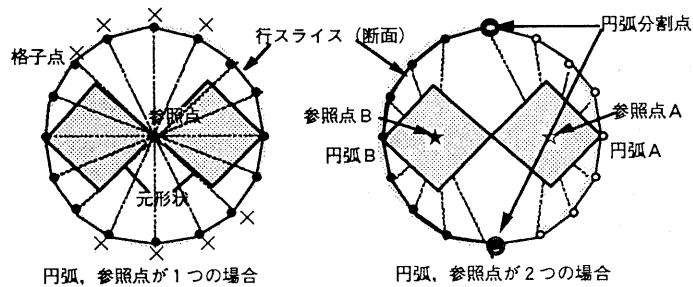


図5 円弧分割の例

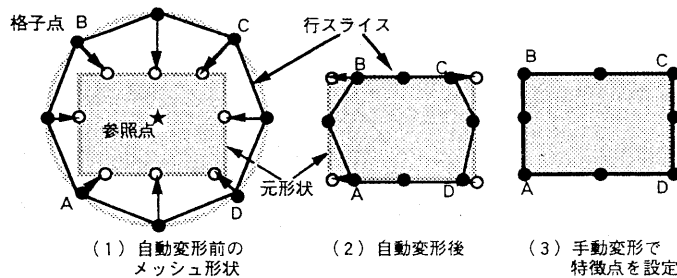


図6 手動補正例 (特徴点の通過指定)

この場合、×のついた8点が元形状と交差せず、移動先を確保できない。右図は、円弧分割して参照点を2つとした例である。全ての格子点に対して移動先が算出できる。

3. 3 楕円球手動補正

必ず通らせない特徴点がある場合(図6)や、自動変形で移動先を算出できなかった場合(図7)などは、楕円球メッシュの格子点位置の手動変形補正を行う。

図6は、自動変形後の楕円球メッシュ(行スライス断面)に対して、元形状矩形の特徴点を通るように手動補正をした例を示す。

図7は、明らかに移動先が自動算出できない例として、参照点がきちんと元形状内部に入っていない場合を示す。このほかに、探索直線が明らかに元形

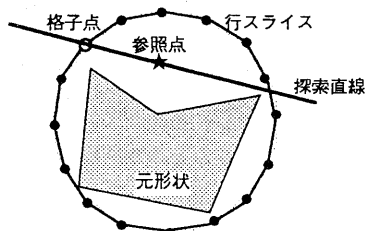


図7 移動先の算出できない例

状と交差しない場合(元形状の面上に含まれる場合も含む)も移動先を算出できない。

図8は、現在手動対応している凹形状の例で、元形状がおわん型の場合である。

これらで示されるように、ユーザの希望を正確に反映させ、より正確な近似変形を行うためには、手動補正機能が不可欠である。

4. モーフィングアニメーションの実行

モーフィング前後のメッシュ近似形状同士に対

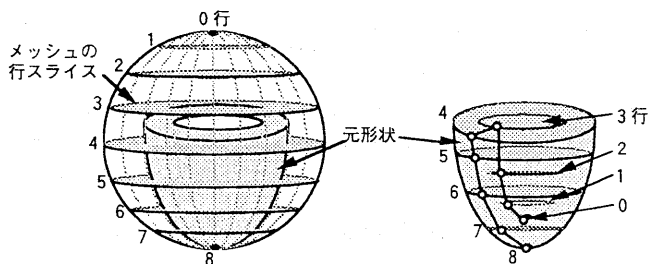


図8 凹形状の実現例

し、モーフィングアニメーションを設定する。

モーフィングアニメーションは、モーフィング前後の近似形状間で、タイムステップ毎に以下の補間を行う。

- ・同じメッシュ位置の格子点座標(=形状補間)
- ・同じメッシュ位置の格子点の属性(=色、法線補間)
- ・近似形状パーツ同士の座標系(=位置補間、パーツモーフィング時)

各近似形状パーツはその形状に合った(例えば口なら鼻の下にある、など)位置を持つため、位置をそのままにして形状だけ変形すると、パーツ毎にばらばらの位置になってしまい、最終的に求める形状にならない。よって、パーツモーフィングを行う際には、パーツ同士の固有座標系の補間が必須である。

図9で座標系補間の有効性を示す。この図は、人間の頭部を髪・顔・目・口の5つのパーツに分け、長細い顔から丸い顔へ3次元モーフィングする例を表わす。パーツの変形と同時に座標系補間を行わな

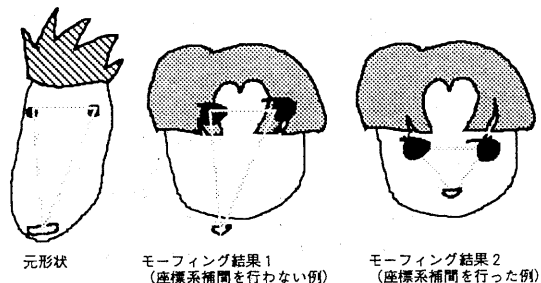


図9 パーツモーフィング時の座標系補間

い結果1(中央)は、目と口の位置が長い顔の時のままなので、顔からはみ出て正しい結果にならないが、座標系補間を行った結果2(右)は、正しい目と口の位置で丸い顔にモーフィングする。

5. 実現と評価

5. 1 実現システム

上述の3次元モーフィングのシステムを、ハード(ホスト=Sun Sparc Station-10+CGアクセラレータ=AG-600)、ソフト(タイムリアスティック3次元CGソフトウェア彩飛)上に試作した。

近似形状作成やモーフィングアニメの設定などの編集部分は、ソフト

上のエディタ機能として実現する。

試作システムにおいて、ViewPoint社の人間、あひる、恐竜(トリケラトプス)の各頭部データを元形状とし、近似形状を作成して3次元モーフィングを行ったところ、約11コマ/秒の円滑な形状変化を確認した。

各元形状・近似形状と、パーツ数、ポリゴン数を図10に示す。

また、図11にあひるから恐竜へのモーフィングの様子を示す。

5. 2 評価

当手法を用いた試作システムの実現により、以下のことが分かった。

(1) 頂点属性補間型の3次元モーフィングの実現

従来では困難だった面数、頂点数の異なる形状同士の頂点对応づけを避け、頂点補間型の3次元モーフィングを実現した。これは、対応づけが可能なように元形状同士の面構成を揃える困難なアプローチを捨て、元形状に近似した面構成の等しい形状を代用するという、新しい発想を基本としている。

(2) 近似形状の質

現在の試作システムにおける近似形状作成機能では、以下の点で元形状と完全に似た近似形状を作成するには至っておらず、これは今後の課題である。しかし現状でも、モーフィング開始・終了時に元形状と近似形状とをすり替え、変形途中の映像制作に利用するには有効なアプローチと考える。

・凹形状の対処

凹形状の近似は、現状ではパーツ分割や手動変形に頼っているが、FFDなどの機能を使用したり、想定される形状に対して多くの種類の近似メッシュを用意することなどで、対処が可能と考える。これらの機能の実現で、ユーザのオーサリングが更に簡単になると期待できる。

・近似形状の格子点における頂点属性の確保

近似形状作成時に変形移動先となる元形状との交点が求められない格子点は、その頂点属性を近傍交点の属性を補間することで確保するが、色がぼやけるなどの問題がある。

・テクスチャの対処

元形状に貼られたテクスチャを近似形状で再現するためには、頂点属性の一種としてテクスチャ座標を取得する必要がある。

(3) 近似形状作成機能の応用

当手法で使用する近似形状編集は、複数の形状ができる。このため応用として、複雑な形状を効率

よく描画するために、視点から遠くにある時だけ置き換えて代用する、より簡単な近似形状を作成するためのポリゴンリダクションへの利用なども期待できる。

(4) パーツシナリオ

複雑な形状に対して行うパーツモーフィングを拡張し、パーツ毎に別のシナリオを設定することで、各部分が少しずつ変形していくようなアニメーションの実現も考えられる。

6. おわりに

計算量のすくない3次元モーフィングを実現するため、ポリゴンでの頂点属性補間を使った3次元モーフィングの実現を目指し、その方式でネックとなる面数、頂点数の異なる形状間のポリゴン対応づけの課題を避けるため、楕円球メッシュを変形した近似形状を利用する方法を考案した。

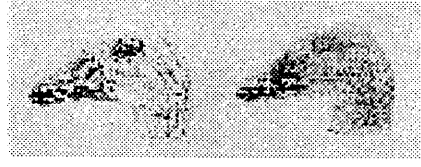
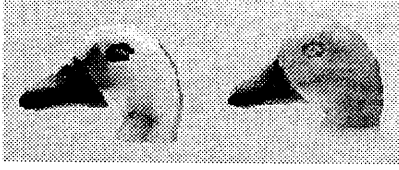
元形状に対する近似の正確性をさらに改善していく必要はあるが、小さなシステムにも対応できるデータ量の少ないポリゴン変形型の3次元モーフィングの1手法として有効であると考えられる。

謝辞

当システムの開発にあたり、多大な協力と貴重なアドバイスをいただいた、(株)富士通静岡エンジニアリングの増井隆弘氏と研究所の皆様へ感謝します。

参考文献

- [1] 中島、町田：“映像情報としてのモーフィング”，テレビジョン学会，画像処理コンピュータビジョン研究会
- [2] 西田、藤井、中前：“Bezier Clippingを用いたモーフィング”，情報処理学会全国大会講演論文集，Vol.46，No.2，pp2.463-2.464，1993
- [3] スコット・アンダーソン：“モーフィング入門”，海文堂，1994
- [4] 土屋、松尾、岩田：“アクティブバルーンモデルと対称性仮説を用いた3次元再構成”，信学会，Vol.J76-DII，No.9，pp.1967-1976，1993
- [5] 中島、黒田、高橋：“エネルギー最小化に基づく3次元モーフィング”，情報処理学会全国大会講演論文集，Vol.50，No.2，pp2.399-2.400，1994



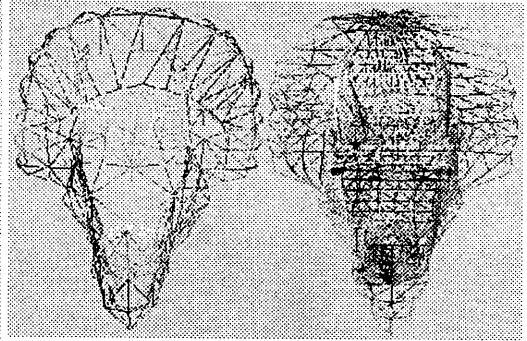
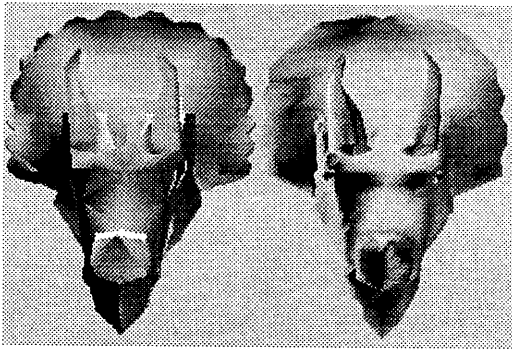
(1) あひるの頭部

左=ViewPointData, 約250ポリゴン, 4 パーツ

右=近似形状Data, 約2000ポリゴン, 6 パーツ

※近似形状は (1) ~ (3) まで全て同じポリゴン数・パーツ数である。

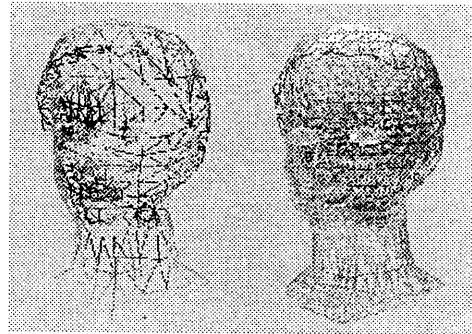
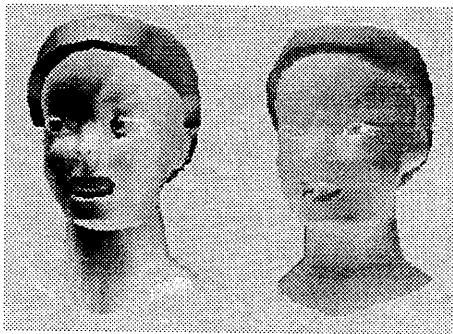
※パーツ数は左右で1パーツと数えた場合。



(2) 恐竜 (トリケラトプス) の頭部

左=ViewPointData, 約300ポリゴン, 3 パーツ

右=近似形状Data, 約2500ポリゴン, 6 パーツ



(3) 人間の頭部

左=ViewPointData, 約500ポリゴン, 5 パーツ

右=ViewPointData, 約2500ポリゴン, 6 パーツ

図10 3Dモーフィング実験に使用したモデル

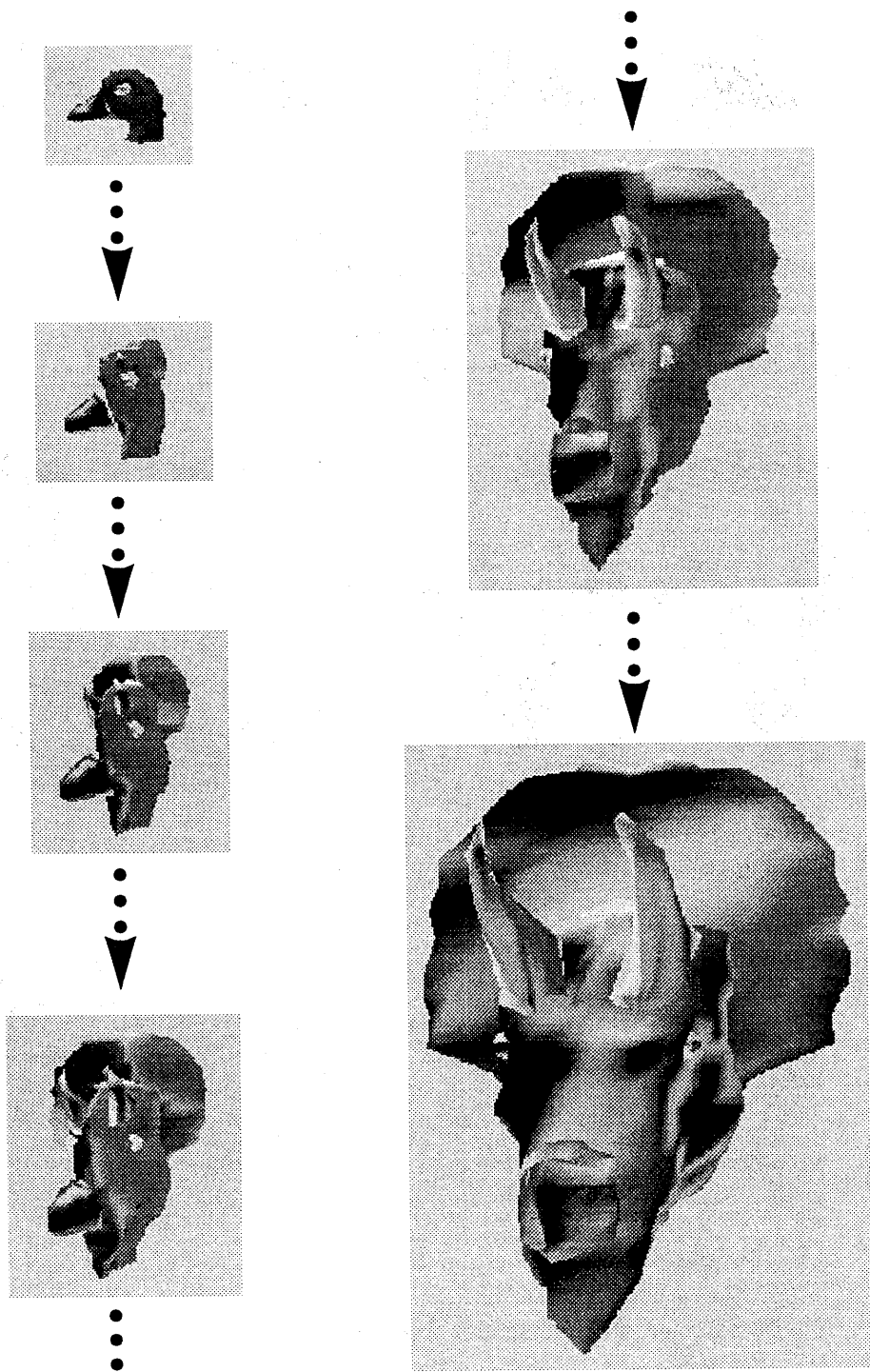


図11 あひるから恐竜へのモーフィング例