

## 二次元パラメータ空間上での線形内挿による 顔の三次元アニメーション

新井 清志      栗原 恒弥

(株)日立製作所 中央研究所

本報告では、二次元のパラメータ空間上で制御点を移動させ、その移動量を線形内挿して表情を生成する手法を提案する。顔の表面上の点の移動量を円柱座標で表現すると、その半径成分は他の二つの成分に比べてはるかに小さい。そこで、半径成分を持たない二次元のパラメータ空間を用いて顔表面の変形を表現する。本手法を用いて顔の三次元アニメーションを制作する実験を行った結果、複雑な表情を容易に生成できた。

### THREE-DIMENSIONAL FACIAL ANIMATION USING LINEAR INTERPOLATION IN THE TWO-DIMENSIONAL PARAMETER SPACE

Kiyoshi Arai      Tsuneya Kurihara

Central Research Laboratory, Hitachi, Ltd.

1-280 Higashi-Koigakubo, Kokubunji, Tokyo 185, Japan

We propose a new method of generating facial expressions. We move control points in the 2-D parameter space, and interpolate the movement of the control points linearly. When the movement of a point on a facial surface are expressed in cylindrical coordinates, its radius component is much less than the other two components. Therefore we express the deformation of a facial surface using the 2-D parameter space which does not have the radius component. We made a 3-D facial animation using this method, and the experimental result shows that complex facial expressions can be generated easily.

## 1 はじめに

コンピュータグラフィックスにおいて、顔の表情の生成は重要な研究課題の一つである。顔のモデルを変形して表情を生成する手法としては、キーフレームのライブラリの中から選び出したデータを補間する手法[1]や、限られたパラメータの値によって顔の変形を記述する手法[2][3]や、筋肉の物理的特性をシミュレートする手法[4][5]などがある。しかし、これらの手法では、複雑な表情を生成する際に煩雑な操作が必要となる。

本報告では、顔のモデルを円柱面にマッピングし、得られた二次元のパラメータ空間上で制御点を移動させ、その移動量を線形内挿して表情を生成する手法を提案する。本手法によれば、複雑な表情を容易に生成することができる。

## 2 顔の変形に関する従来技術と技術課題

### 2.1 変形の表現方法

一般に、顔の三次元モデルは、顔の表面上の点の三次元座標と接続情報とからなる。顔のモデルの変形は、これらの点の三次元的な移動によって表現される。二次元的な移動量の計算や対話的な指定は簡単である。しかし、三次元的な移動量を扱うことは難しい。したがって、三次元的な移動量をできるだけ簡単な形式で表現することが必要である。

顔の表面上の点の座標を円柱座標で表現し、表情の変化に伴うこれらの点の移動量を半径成分、角度成分、高さ成分に分解した場合、移動量の半径成分の大きさは他の二つの成分の大きさに比べてはるかに小さい。本報告で提案する手法では、この性質に着目し、表情の変化に伴う顔のモデルの変形を円柱座標の角度成分と高さ成分のみの変形とみなす。このとき、顔の表面上の点の移動量を内挿する必要があるが、角度成分と高さ成分からなる距離に基づいて行えばよい。そこで、三次元の顔のモデルを二次元のパラメータ空間と対応づけ、顔の表面上の点の移動量の内挿はこの空間上で行う。

### 2.2 変形の制御方法

顔の三次元モデルを変形するとき、顔の表面上のすべての点の移動量を求める必要がある。その際、これらの移動量をすべて指定する作業は煩雑なので、まず表情を決定する特徴的な点の移動量を決定することが多い。このような点を制御点と呼ぶ。残りの点の移動量は制御点の移動量を用いて計算できるため、制御点を用いれば少数の点の移動量を指定するだけで済む。しかし、制御点の移動量を簡単に指定する方法と、制御点の周囲の点の移動量を合理的に計算する方法を確立しなければ、複雑な変形を容易に制御することはできない。

本報告で提案する手法では、制御点を二次元のパラメータ空間上で定義することによって、制御点の操作を容易にする。

### 2.3 変形の計算方法

制御点の移動量を用いて周囲の点の移動量を求めるとき、制御点の移動量が周囲の点に及ぼす影響力は、制御点からの距離の関数で与えるのが一般的である。しかし、関数の決め方の自由度が大きいため、最適な関数を決定することが難しく、多くの試行錯誤が必要である。また、制御点の分布が一様でないなどの理由により、制御点からの方向によって移動量の影響力を変える必要が生じた場合、制御点からの距離だけの関数ではこれに対応できない。

本報告で提案する手法では、制御点の分布に偏りがあっても計算が破綻しないようにするため、制御点の移動量を二次元のパラメータ空間上で線形内挿する。最適な線形内挿を行うため、制御点を頂点とするドローネ網[6]を用いる。ドローネ網は、二次元平面上に不規則に分布している点を接続して、二次元平面を最適な三角形領域に分割するものである。二次元平面上の任意の点での物理量は、その点が含まれている三角形領域の三頂点での物理量の線形内挿によって求められる。

### 3 パラメータ空間と顔のモデルとの対応

#### 3.1 パラメータ空間と顔の基本モデルとの対応

顔のモデルの形状は、人相と表情とによって決定される。本報告で提案する手法では、様々な人相の顔のモデルに対応する共通のパラメータ空間を定義する。そして、このパラメータ空間上の点の移動によって表情を表現し、表情のデータを人相に依存しないものにする。

まず、顔の表面の三次元形状を三角形パッチで表しているモデルを一つ用意し、これを顔の基本モデル $M_f$ と名付ける。図1に示すように、 $M_f$ 上の点 $P_f$ の位置を半径成分 $r$ 、角度成分 $\theta$ 、高さ成分 $h$ から成る円柱座標で表す。 $P_f$ の角度成分 $\theta$ 、高さ成分 $h$ をそれぞれ $\Theta$ 成分、 $H$ 成分とする二次元空間をパラメータ空間 $S$ と名付け、 $S$ 上の点 $P_s$ を、 $P_f$ に対応する点とする。 $M_f$ の形状は、円柱座標により高さと角度の一価関数で表現できるものに限定する。これにより、 $P_f$ の半径成分は角度成分と高さ成分から一意に求められ、 $P_s$ に対応する $P_f$ は一意に定まる。 $M_f$ 上の点 $P_f(x_f, y_f, z_f)$ からこれに対応する $S$ 上の点 $P_s(\theta, h)$ を求める式を、次のようになる。

$$(\theta, h) = (\tan^{-1}(z_f / x_f), y_f) \quad (1)$$

逆に、 $P_s(\theta, h)$ から $P_f(x_f, y_f, z_f)$ を求める式は、次のようになる。

$$(x_f, y_f, z_f) = (r \cos \theta, h, r \sin \theta) \quad (2)$$

$$r = R_f(\theta, h) \quad (3)$$

ここで、 $R_f$ は $P_f$ の角度成分 $\theta$ と高さ成分 $h$ から半径成分 $r$ を求める関数である。 $M_f$ の形状を、円柱座標により高さと角度の一価関数で表現できるものに限定したので、 $r$ は、 $\theta$ と $h$ から一意に求められる。

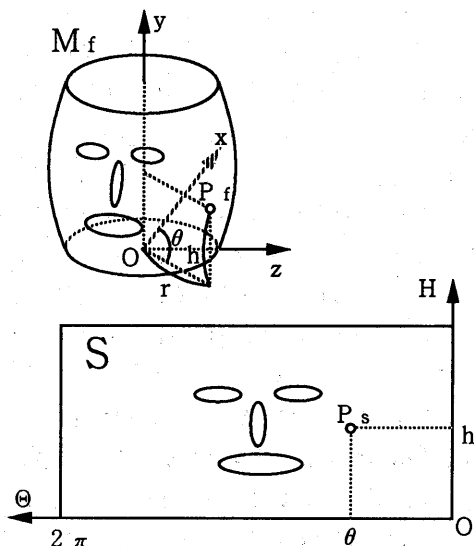


図1 顔の基本モデル $M_f$ とパラメータ空間 $S$

#### 3.2 パラメータ空間と顔の個別モデルとの対応

顔の基本モデル $M_f$ と異なる人相の顔のモデルを、顔の個別モデル $M_i$ と名付ける。 $S$ 上の点と $M_i$ 上の点との対応は、 $M_f$ 上の点と $M_i$ 上の点との対応を定義することによって間接的に定義する。このとき、 $M_f$ 上の三角形パッチが、 $M_i$ 上の三角形の領域と一対一に対応している必要がある。そこで、 $M_i$ は次のようにして作成する。

(1)  $M_f$ の形状を変形して $M_i$ を作成する場合は、 $M_f$ を構成する各三角形パッチの頂点の座標を変更するだけで $M_f$ の形状を変形し、 $M_f$ の三角形パッチのトポロジーを変えずに $M_i$ を作成する。

(2)  $M_f$ とは無関係に新たなモデルとして $M_i$ を用意する場合は、 $M_f$ と $M_i$ の三角形パッチのトポロジーを一致させるため、 $M_f$ 上の三角形パッチの頂点に対応する $M_i$ 上の点を求める。具体的には、目や口の輪郭など、対応の明確な点を対話的に指定し、この対応に基づいて残りの点の対応を決定する。

対応が決定した後、 $M_f$ 上の三角形パッチの頂点に対応する $M_i$ 上の点を、あらためて $M_i$ 上の三角形パッチの頂点と考え、 $M_i$ 上の三角形パッチを再構成する。

図2に示すように、 $M_f$ 上の点 $P_{f1}(x_{f1}, y_{f1}, z_{f1})$ ,  $P_{f2}(x_{f2}, y_{f2}, z_{f2})$ ,  $P_{f3}(x_{f3}, y_{f3}, z_{f3})$ を頂点とする三角形パッチ $T_f$ が $M_f$ 上の点 $P_{i1}(x_{i1}, y_{i1}, z_{i1})$ ,  $P_{i2}(x_{i2}, y_{i2}, z_{i2})$ ,  $P_{i3}(x_{i3}, y_{i3}, z_{i3})$ を頂点とする三角形パッチ $T_i$ に対応しているとする。本報告で提案する手法では、 $T_f$ と $T_i$ との対応から、 $T_f$ 内の点と $T_i$ 内の点との対応も定義する。この定義によって、 $S$ 上のすべての点が、 $M_f$ の表面上の点だけでなく $M_i$ の表面上の点にも対応することになる。この対応関係を用いれば、 $M_f$ 上および $M_i$ 上の点の移動量を同一の手法で計算できる。 $T_f$ 内の点 $P_f(x_f, y_f, z_f)$ に対応する $T_i$ 内の点 $P_i(x_i, y_i, z_i)$ の位置は線形内挿によって求める。具体的には、式(4)を満たす係数 $a_1, a_2, a_3$ を式(5)に代入することによって、 $P_i$ を求める。

$$\begin{aligned} (x_f, y_f, z_f) \\ = (a_1 x_{f1} + a_2 x_{f2} + a_3 x_{f3}, \\ a_1 y_{f1} + a_2 y_{f2} + a_3 y_{f3}, \\ a_1 z_{f1} + a_2 z_{f2} + a_3 z_{f3}) \end{aligned} \quad (4)$$

$$\begin{aligned} (x_i, y_i, z_i) \\ = (a_1 x_{i1} + a_2 x_{i2} + a_3 x_{i3}, \\ a_1 y_{i1} + a_2 y_{i2} + a_3 y_{i3}, \\ a_1 z_{i1} + a_2 z_{i2} + a_3 z_{i3}) \end{aligned} \quad (5)$$

式(2)～(5)を用いて、 $S$ 上の点 $P_s(\theta, h)$ に対応する $P_i(x_i, y_i, z_i)$ を求める。

以上により、 $S$ 上の点と $M_f$ 上および $M_i$ 上の点は一対一に対応する。

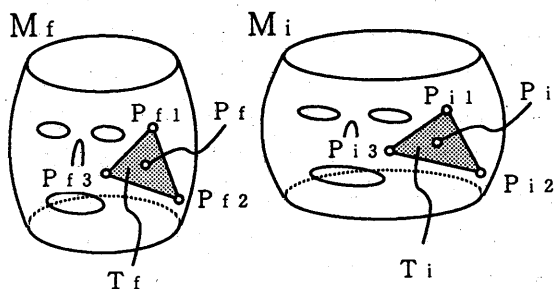


図2 顔の基本モデル $M_f$ と顔の個別モデル $M_i$

#### 4 パラメータ空間上での線形内挿による顔の変形

##### 4.1 パラメータ空間上の制御点の決定と移動

顔のモデルの変形操作は、二次元のパラメータ空間 $S$ 上の点の移動によって行う。 $S$ 上の点の移動量は顔のモデル上の点の角度方向と高さ方向の移動量に対応するものであり、半径方向の移動量は $S$ 上での点の移動では表現できない。しかし、2.1で述べたように、表情の変化に伴う顔の表面上の点の移動量の半径成分は、角度成分、高さ成分に比べてはるかに小さいので、半径方向の移動量を与える必要はない。

顔の基本モデル $M_f$ および顔の個別モデル $M_i$ の変形を行うには、 $M_f$ および $M_i$ の三角形パッチの頂点对应する $S$ 上のすべての点の移動量を求める必要があるが、これらをすべて対話的に指定する作業は煩雑である。そこで、一部の点の移動量だけを対話的に指定し、その移動量から残りの点の移動量を求める。移動量を指定する $S$ 上の点を制御点と名付ける。図3に示すように、制御点は目、口、鼻の輪郭や眉、あごなど、表情を決定する際に重要と思われる部分に配置する。

制御点を配置した後、これにパラメータ空間 $S$ 上での移動量を対話的に与える。 $S$ 上の制御点 $C(\theta, h)$ を $S$ 上の点 $C'(\theta', h')$ に移動させたとき、 $S$ 上での $C$ の移動量 $\Delta C(\Delta\theta, \Delta h)$ を求める式は次のようになる。

$$(\Delta\theta, \Delta h) = (\theta' - \theta, h' - h) \quad (6)$$

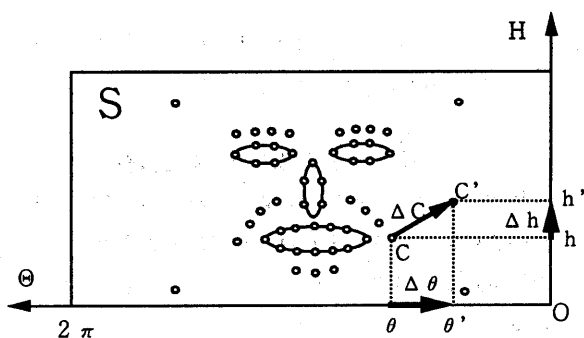


図3 制御点の配置と移動

#### 4.2 ドローネ網を用いた移動量の線形内挿

制御点の移動量を指定した後、残りの点の移動量をパラメータ空間S上での線形内挿によって求める。まず、S上で制御点を頂点とするドローネ網を描き、S上の任意の点の移動量がどの制御点の移動量から求められるのかを決定する。ドローネ網は、二次元平面上に不規則に分布している点を接続して、二次元平面を三角形領域に分割するものである。このとき、個々の三角形ができるだけ正三角形に近くなるように分割される。二次元平面上の任意の点での物理量は、その点が含まれている三角形領域の三頂点での物理量の線形内挿を行うことによって求められる。

図4に示すように、パラメータ空間S上の点 $P_s$  ( $\theta$ ,  $h$ )が制御点 $C_1(\theta_1, h_1)$ 、 $C_2(\theta_2, h_2)$ 、 $C_3(\theta_3, h_3)$ を頂点とする三角形領域に含まれているとき、 $P_s$ の移動量 $\Delta P_s(\Delta \theta, \Delta h)$ は、制御点 $C_1$ 、 $C_2$ 、 $C_3$ の移動量 $\Delta C_1(\Delta \theta_1, \Delta h_1)$ 、 $\Delta C_2(\Delta \theta_2, \Delta h_2)$ 、 $\Delta C_3(\Delta \theta_3, \Delta h_3)$ の線形内挿によって求める。具体的には、式(7)を満たす係数 $a_1$ 、 $a_2$ を式(8)に代入することによって、 $\Delta P_s$ を求める。

$$\begin{aligned} (\theta, h) &= (a_1 \theta_1 + a_2 \theta_2 + (1 - a_1 - a_2) \theta_3, \\ &\quad a_1 h_1 + a_2 h_2 + (1 - a_1 - a_2) h_3) \quad (7) \\ (\Delta \theta, \Delta h) &= (a_1 \Delta \theta_1 + a_2 \Delta \theta_2 + (1 - a_1 - a_2) \Delta \theta_3, \\ &\quad a_1 \Delta h_1 + a_2 \Delta h_2 + (1 - a_1 - a_2) \Delta h_3) \quad (8) \end{aligned}$$

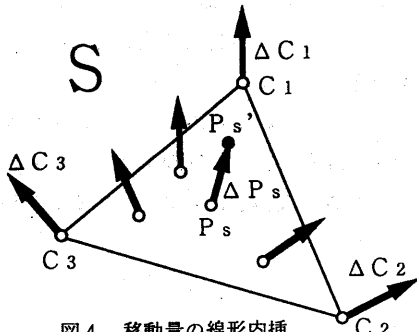


図4 移動量の線形内挿

$P_s$ の移動後の点 $P_s'(\theta', h')$ を求める式は、次のようになる。

$$(\theta', h') = (\theta + \Delta \theta, h + \Delta h) \quad (9)$$

#### 4.3 顔のモデル上の点の移動

3.1および3.2で述べたように、パラメータ空間S上の点と顔の基本モデル $M_f$ 上、個別モデル $M_i$ 上の点是一对一に対応している。このため、S上での変形によって $M_f$ 上、 $M_i$ 上での変形が定義できる。

図5に示すように、S上の点 $P_s$ に対応する $M_f$ 上、 $M_i$ 上の点をそれぞれ $P_f(r_f, \theta_f, h_f)$ 、 $P_i(r_i, \theta_i, h_i)$ とし、 $P_s$ の移動後の点 $P_s'$ に対応する $M_f$ 上、 $M_i$ 上の点をそれぞれ $P_f'(r_f', \theta_f', h_f')$ 、 $P_i'(r_i', \theta_i', h_i')$ とする。ここで、これらの点の位置は円柱座標で表している。

2.1および4.1で述べたように、表情の変化に伴う顔の表面上の点の移動量は、ほとんど角度成分と高さ成分だけからなる。そこで、顔のモデル上の点は、移動前の半径成分を保存したまま角度方向と高さ方向に移動するものとする。 $P_f$ および $P_i$ の移動後の点をそれぞれ $P_f''(r_f'', \theta_f'', h_f'')$ および $P_i''(r_i'', \theta_i'', h_i'')$ とすると、これらは次の式を用いて求める。

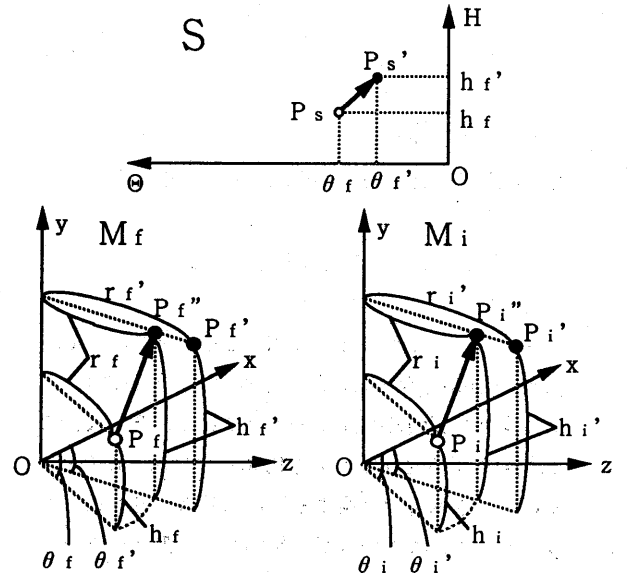


図5 顔のモデル上の点の移動

$$\begin{aligned} (r_i'', \theta_i'', h_i'') \\ = (r_i, \theta_i', h_i') \end{aligned} \quad (10)$$

$$\begin{aligned} (r_i'', \theta_i'', h_i'') \\ = (r_i, \theta_i', h_i') \end{aligned} \quad (11)$$

$P_s$ が $P_s'$ に移動することによって、 $P_f$ は $P_f''$ に移動し、 $P_i$ は $P_i''$ に移動する。

#### 4.4 顔の変形データの合成

パラメータ空間上の制御点の移動量を指定して作成した各々の変形データは、任意の割合で合成することができる。複数の変形データを合成したのも、一つの変形データとして取り扱うことができる。ただし、各々の変形データは、共通の制御点を用いて作成するものとする。

$n$ 個の変形データ $D_1, D_2, \dots, D_n$ を、それぞれ共通の $m$ 個の制御点 $C_1, C_2, \dots, C_m$ を用いて作成し、このとき変形データ $D_i$ における制御点 $C_j$ の移動量が $\Delta C_{ij}$ であったとする。 $D_1, D_2, \dots, D_n$ にそれぞれ重み $w_1, w_2, \dots, w_n$ を与えて合成することによって変形データ $D_x$ を作成するとき、 $D_x$ における制御点 $C_j$ の移動量 $\Delta C_j$ は、次の式によって求める。

$$\Delta C_j = \sum_{i=1}^n w_i \Delta C_{ij} \quad (12)$$

式(12)を用いることによって、複数の表情を合成して新たな表情を生成することができる。

#### 5 FACSによる顔の表情の生成

FACS(Facial Action Coding System)[7]は、AU(Action Unit)と呼ばれる顔の動きの単位を組み合わせることで様々な表情を記述する概念であり、顔の表情の生成[4]や顔の動画像の符号化[8]などに広く使われている。本報告で提案する手法にFACSを適用するときには、各々のAUに対応する変形データをバ

ラメータ空間上の制御点の移動で表現し、ライブラリを作成する。AUを組み合わせる表情は、各々のAUに対応する変形データを、4.4で述べた手法を用いて合成することによって得られる。ただし、合成された変形データが、顔のモデルの目や口の部分などを不自然に変形させてしまうことを防ぐため、上まぶた上の制御点列と下まぶた上の制御点列は互いに交わらず、上唇上の制御点列と下唇上の制御点列も互いに交わらないという制限を設ける。

AUを組み合わせることで作成した表情も、一つの表情としてライブラリに登録できる。これにより、顔のモデルの変形データは階層構造を持ち、必要に応じて様々なレベルの変形データを作成して顔の表情を生成することができる。

なお、目玉や歯などの動きは、パラメータ空間上の線形内挿では表現できないので、これらの部分の動きは個別に定義する。

#### 6 実験結果

本報告で提案した手法を用いて、顔の三次元アニメーションの作成する実験を行った。実験に用いた顔のモデルを図6に示す。顔の基本モデルは、マネキンの頭部をレーザースキャナで測定し、文献[9]の手法でデータを削減したものを基にして作成したものであり、2950個の三角形パッチからなる。顔の個別モデルは、顔の基本モデルの頂点の座標を変更することによって作成した。これらのモデルに対応するパラメータ空間を図7に示す。パラメータ空間上の制御点は、顔の表面上の主要な部分に99個、パラメータ空間の輪郭線上に6個、合計105個配置した。制御点の移動量の線形内挿は、これらの制御点を頂点とするドロネー網に基づいて行った。アニメーションを作成する際に生成した主な表情を図8に示す。基本モデル、個別モデルのいずれの場合にも、リアルな表情を容易に生成することができた。

図6 実験に用いた  
顔のモデル

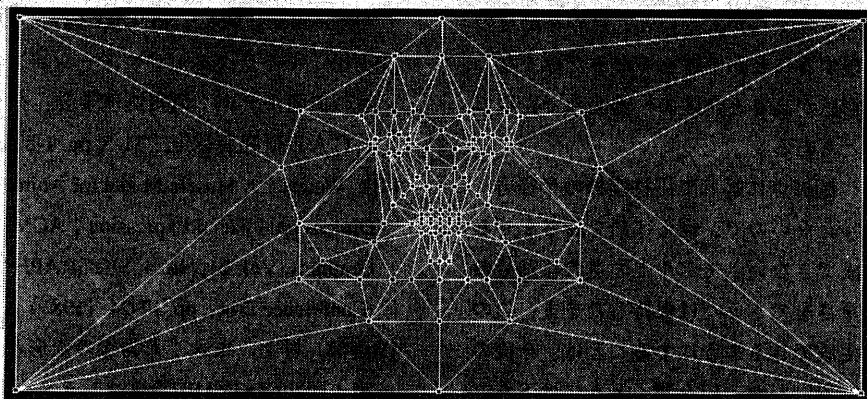
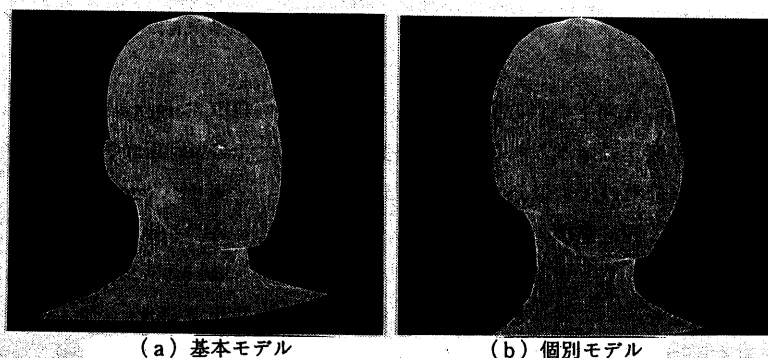


図7 実験に用いた顔のモデルに対応するパラメータ空間上の制御点と、制御点を頂点とするドローネ網

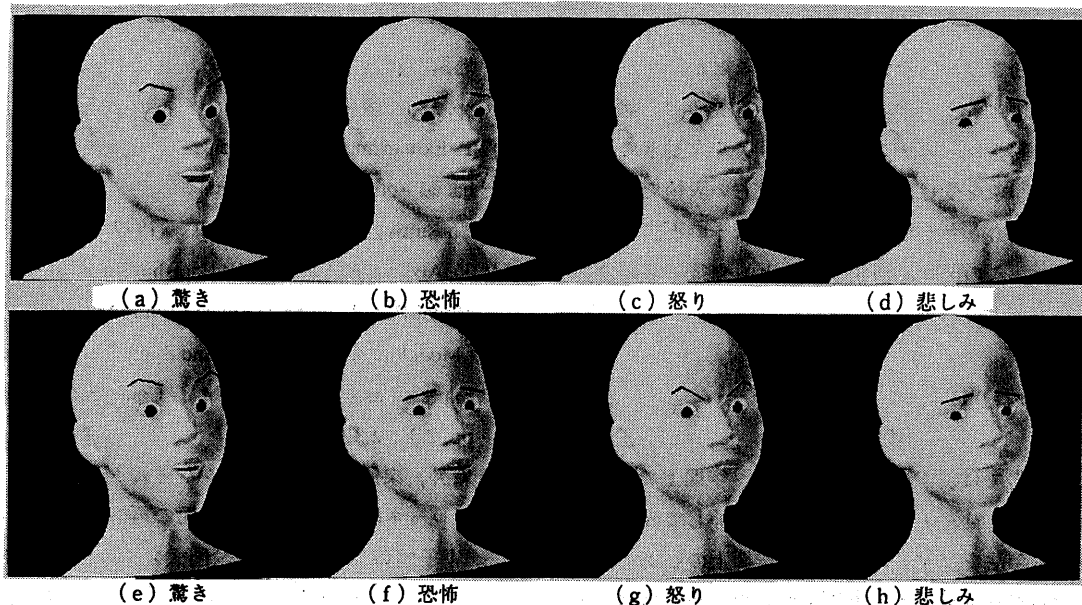


図8 基本モデルの表情生成例 ((a) ~ (d)) および個別モデルの表情生成例 ((e) ~ (h))  
(同一の表情は同一の表情データにより生成)

## 7 考察

本手法では、顔の表面上の点の座標を円柱座標で表現し、表情の変化に伴うこれらの点の移動量を角度成分と高さ成分だけで近似した。この近似が有効であるためには、円柱座標の座標軸が顔のモデルに対して傾いていないことが必要である。すなわち、顔のモデルが正面を向いているときに、円柱座標の座標軸はモデルの底面に対して垂直である必要がある。この方向からのずれが大きいと、点の移動量の半径成分を無視することができなくなるだけでなく、モデルの表面形状が高さと角度の一価関数で表現できなくなることもある。

本手法では、制御点に対して円柱座標の半径方向の移動量は与えないので、例えば頬を膨らませる表情などは生成できない。モデル上の点の移動量の表現形式の簡単さをできるだけ保持したまま、このような表情を生成できるようにすることは、今後の課題の一つである。なお、口を開くときには下あごの部分は回転運動をするため、半径方向の動きも生じるが、口の開け方が極端に大きくない限り、二次元的な動きで自然な近似ができることがわかった。

実験で制作したアニメーションを見ると、顔の表面の動きはスムーズであり、リアルな動きであるという印象を受けた。しかし、激しい表情をしているときにもしわが現れず、これに若干の不自然さを感じた。この問題を解決することも今後の課題である。

## 8 おわりに

本報告では、二次元のパラメータ空間上で制御点を移動させ、その移動量を線形内挿して表情を生成する手法を提案した。二次元空間上の制御点に二次元の移動量を与える方法を用いた結果、複雑な表情を容易に生成することができた。今後は、半径方向の移動量の扱い方をさらに工夫したり、しわの表現手法などについて検討する予定である。

## 参考文献

- [1]P. Bergeron and P. LaChapelle :  
Controlling Facial Expression and Body Movements in the Computer Generated Shot 'Tony de Peltrie' ;  
ACM SIGGRAPH '89 Course Notes #22, State of the art in Facial Animation, pp. 202-220 (1989) .
- [2]F. I. Parke : Parameterized Models for Facial Animation ; IEEE Computer Graphics and Applications, Vol.2, No.9, pp. 61-68 (1982) .
- [3]小松：キャラクタ・アニメーションのための顔の曲面モデル；情報処理学会 グラフィクスとCAD研究会報告，31-5, pp. 1-8 (1988) .
- [4]K. Waters : A Muscle Model for Animating Three-Dimensional Facial Expression ; ACM Computer Graphics, Vol. 21, No. 4, SIGGRAPH '87 Conference Proc., pp. 17-24 (1987) .
- [5]渡部，村上，末永：人物像の変形と生成；第4回NICOGRAPH論文コンテスト論文集，pp. 147-154 (1988) .
- [6]伊理 監修、腰塚 編集：計算幾何学と地図情報処理；bit別冊，共立出版，第4章，pp. 126-148 (1986) .
- [7]P. Ekman and W. V. Friesen :  
Facial Action Coding System ;  
Consulting Psychologists Press (1977) .
- [8]K. Aizawa, Y. Yamada, H. Harashima and T. Saito :  
Model-Based Synthesis Image Coding System - Modeling A Person's Face and Synthesis of Facial Expressions - ; GLOBECOM '87 Conference Record, pp. 2.3.1-2.3.5 (1987) .
- [9]新井，栗原：ラブラシアンフィルタとドローネ網を用いた物体表面の再構成；1990年電子情報通信学会春季全国大会，D-601 (1990) .