

多重解像度局所平滑化を用いた細分割曲面設計

高橋 成 雄[†]

細分割曲面は、任意位相の曲面を設計する効果的な手法を提供する。細分割曲面のような多面体曲面を設計する際には、その膨大な数の頂点の幾何位置を制御する必要から、局所平滑化フィルタリングが広く用いられるようになってきた。本論文では、多重解像度局所平滑化を用いた細分割曲面の新しい設計手法を示す。提案手法では、異なる幾何制約を異なる分割レベルの細分割曲面に同時に与えることができる。提案手法のさらなる利点として、重み値をそれぞれの分割レベルに割り当てることにより、同じ制約からでも異なる滑らかさの細分割曲面が生成できる点があげられる。いくつかの設計例を通じて、提案手法の表現能力についても示す。

Designing Subdivision Surfaces via Multiresolution-based Local Filtering

SHIGEO TAKAHASHI[†]

Subdivision surfaces provide a promising approach to designing surfaces of arbitrary topological type. In designing such polygonal surfaces, local smooth filtering has been widely used to control a large number of vertex positions. This article presents a new method of designing subdivision surfaces via multiresolution-based local smooth filtering. In this method, different geometric constraints can be imposed on the subdivision surface at different refinement levels. The method can also control the smoothness of subdivision surfaces by assigning a weight value to each refinement level even when the same set of constraints are given. Several design examples are included to demonstrate the capability of this method.

1. はじめに

従来、複雑な形状を持つ曲面の設計は、B-スプラインなどのパラメータ曲面表現を用いて行い、さらにそこから離散的にサンプル点をとって多面体曲面 (polygonal surface) 表現に変換し、ディスプレイ上に表示していた。ところが、パソコンに代表される近年の計算機性能の向上により、膨大な頂点数を持つ多面体曲面を直接操作することができるようになった。また、多面体曲面は任意位相の曲面を柔軟に表現することができ、パラメータ曲面で必要だったパッチ接続の連続性を考慮する必要がないのも利点である。レーザ・スキャナなどの計測装置から得られる密なサンプル点を表現する際にも、多面体曲面は広く用いられており、形状表現の標準的な道具として広く研究がなされている。膨大な頂点数のデータを持つ多面体曲面を、適宜簡単化して階層表現を導入する手法も提案されている¹⁾。

多面体曲面の中で、細分割曲面 (subdivision surface)^{2)~5)} は形状設計の分野において広く用いられてきており、最近の計算機性能の向上がその発展をさらに加速させている。細分割曲面は、初期多面体形状の面を再帰的に細分割して徐々にその面を細かくしていくが、その際生じる多面体形状の系列を、曲面の階層表現としてとらえる手法も提案されている。たとえば、Lounsbery ら⁶⁾ は、信号処理の道具であるウェーブレット (wavelet) を導入し、細分割曲面の多重解像度表現を構築した。さらに、この細分割曲面の多重解像度表現を利用してその形状を設計する手法も提案されており、中でも Zorin ら⁷⁾ は、Taubin⁸⁾ の局所平滑化フィルタを用いて細分割曲面を階層的に表現し、加えて曲面の細分割を部分ごとに適応的に制御できる設計手法を提案している。

本論文では、Lounsbery⁶⁾ らの階層表現を用いて、細分割曲面を設計する新たな手法を提案する。Zorin らの手法を含め、従来手法は複数の解像度レベルの形状を同時に制御できなかったのに比べ、我々の手法は、異なる解像度レベルに異なる幾何制約を付加して、同時に形状を制御することができる点が優れている。こ

[†] 群馬大学総合情報処理センター
Computer Center, Gunma University

ここで、異なる解像度レベルに付加できる幾何制約のモデルを、本論文では多重解像度制約と呼ぶことにする。我々の手法のさらなる利点として、解像度レベルごとに重み値を割り当てその重み値を制御することで、同じ制約から異なる滑らかさの形状を生成できることがあげられる。形状の滑らかさは、それぞれの解像度において Taubin の局所平滑化フィルタ⁸⁾を用いて制御する。Taubin の局所平滑化フィルタは、細分割曲面の頂点数に比例するオーダーで計算ができるため、従来の頂点数の 2 乗の線形方程式を解くグローバルな形状最適化手法⁹⁾に比べ、飛躍的に計算時間を短縮することができる。

以下、2 章においてウェーブレットを用いた細分割曲面の多重解像度解析について、3 章では本手法で用いる細分割規則について、4 章では多重解像度表現に必要なフィルタの計算方法について、5 章では幾何制約の計算方法について、順次説明していく。さらに、6 章で単独解像度レベルの局所平滑化について述べたのち、7 章で多重解像度レベルの局所平滑化について詳しく述べる。最後に、本手法で作成した形状設計例を 8 章で示し、9 章でまとめと今後の課題について触れる。

2. 細分割曲面上の多重解像度解析

細分割曲面は、初期制御多面体 $\Theta^{(0)}$ の面を、 $\Theta^{(1)}$, $\Theta^{(2)}$, ... へと再帰的に細分割していったときの、最終的に収束する曲面 $\Theta^{(\infty)}$ として定義される。Lounsbery ら⁶⁾は、この細分割の回数を解像度レベルととらえ、ウェーブレットを用いた多重解像度表現を実現している。

まず、細分割曲面 $\Theta^{(k)}$ の頂点数を $m^{(k)}$ とし、それと同じ数の基底関数を並べたベクトル $\varphi^{(k)} = (\varphi_0^{(k)}, \dots, \varphi_{m^{(k)}}^{(k)})$ を用意する。すると、細分割曲面 $\Theta^{(k)}$ の形状を、基底関数の線形和 $\varphi^{(k)\top} \cdot c^{(k)}$ として表現することができる。ここで、 $c^{(k)} = (c_0^{(k)}, \dots, c_{m^{(k)}}^{(k)})$ は、 $\varphi^{(k)}$ に対応する係数をベクトルとしてまとめたものである。多重解像度解析において、基底関数 $\varphi_\alpha^{(k)} (\alpha = 1, \dots, m^{(k)})$ はスケール関数と呼ばれる。

次に、 $\varphi^{(k+1)}$ と $\varphi^{(k)}$ が表す 2 つの関数空間の差を考え、その差を表す関数空間の基底関数を、次のようなベクトル $\psi^{(k)} = (\psi_0^{(k)}, \dots, \psi_{n^{(k)}}^{(k)})$ として用意する。このとき、 $\varphi^{(k)}$ の表す関数空間の次元は $m^{(k)}$ であるから、 $\psi^{(k)}$ の表す関数空間の次元 $n^{(k)}$ は、 $n^{(k)} = m^{(k+1)} - m^{(k)}$ の関係にあることに注意する。基底関数 $\psi_\alpha^{(k)} (\alpha = 1, \dots, n^{(k)})$ はウェーブレッ

トと呼ばれる。

細分割曲面 $\Theta^{(k+1)}$ が、1 つ分割の粗い細分割曲面 $\Theta^{(k)}$ とそれ以外の起伏に分解されることを考えると、

$$\varphi^{(k)}(x)^\top = \varphi^{(k+1)}(x)^\top P^{(k)} \quad (1)$$

$$\psi^{(k)}(x)^\top = \varphi^{(k+1)}(x)^\top Q^{(k)} \quad (2)$$

を満たす行列 $P^{(k)}$, $Q^{(k)}$ が存在することが分かる。行列 $P^{(k)}$, $Q^{(k)}$ は、合成フィルタと呼ばれる。ウェーブレットに基づく階層表現は、係数ベクトル $c^{(k)}$, $d^{(k)}$ に線形変換を施すことで得ることができる。まず、ウェーブレット分解は、

$$c^{(k)} = A^{(k)} c^{(k+1)}, \quad d^{(k)} = B^{(k)} c^{(k+1)} \quad (3)$$

と表現される。ここで、(3) の $A^{(k)}$, $B^{(k)}$ は、分解フィルタと呼ばれる。逆にウェーブレット再構成は、

$$c^{(k+1)} = P^{(k)} c^{(k)} + Q^{(k)} d^{(k)} \quad (4)$$

となり、式 (1), (2) と同じ行列 $P^{(k)}$, $Q^{(k)}$ を用いて表現できる。また、分解・合成フィルタの間には、次の関係が存在する。

$$\begin{pmatrix} A^{(k)} \\ B^{(k)} \end{pmatrix} = \begin{pmatrix} P^{(k)} & Q^{(k)} \end{pmatrix}^{-1} \quad (5)$$

3. 細分割規則

$c^{(k)}$ を、解像度レベル k の細分割曲面 $\Theta^{(k)}$ 上の頂点の 1 つの座標 (たとえば x, y, z 座標のいずれか) を要素とするベクトルとする。すると、細分割規則から

$$c^{(k+1)} = P^{(k)} c^{(k)} \quad (6)$$

の関係が得られる。ここで、 $P^{(k)}$ は $m^{(k+1)} \times m^{(k)}$ 行列であり、分解フィルタ $P^{(k)}$ のそのものになると考える。さらに、後の分解・合成フィルタの計算に備えて、スケール関数のベクトルを、 $\varphi^{(k+1)\top} = (\mathcal{O}^{(k+1)\top} \mathcal{N}^{(k+1)\top})$ のように分割しておく。ここで、ベクトル $\mathcal{O}^{(k+1)}(x)$ は、スケール関数 $\varphi_\alpha^{(k+1)}(x)$ ($\alpha = 1, \dots, m^{(k+1)}$) のうち、 $\Theta^{(k)}$ に元々存在した頂点に対応するものを要素にとり、 $\mathcal{N}^{(k+1)}(x)$ は、 $(k+1)$ 回目の細分割処理で新しく生じた頂点に対応するスケール関数を要素とする。これより、式 (1) を

$$\varphi^{(k)\top} = \begin{pmatrix} \mathcal{O}^{(k+1)\top} & \mathcal{N}^{(k+1)\top} \end{pmatrix} \begin{pmatrix} \mathcal{O}^{(k)} \\ \mathcal{N}^{(k)} \end{pmatrix} \quad (7)$$

のように書き直すことができる。ここで、 $\mathcal{O}^{(k)}$ と $\mathcal{N}^{(k)}$ は $P^{(k)}$ の一部分であり、それぞれ $\mathcal{O}^{(k+1)}(x)$

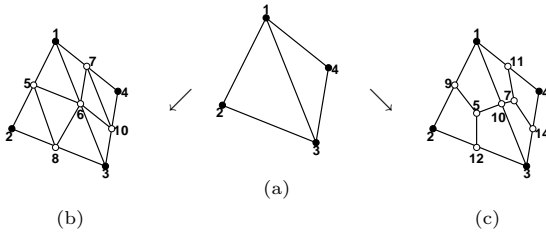


図 1 三角形分割と四角形分割規則：(a) 四面体，(b) 三角形分割規則を施したもの，(c) 四角形分割規則を施したもの

Fig. 1 Refinement rules for triangular and rectangular subdivision surfaces: (a) an original tetrahedron, (b) a triangular refinement, and (c) a rectangular refinement.

と $\mathcal{N}^{(k+1)}(x)$ に対応している。

本論文では、三角形細分割と四角形細分割規則を用いる。図 1 は、四面体が、三角形細分割と四角形細分割規則を用いて、どのように分割されるかを示したものである。これらは、もとの多面体の面を形状を変えずに、平面的に分割するものであり、それぞれ位相的には Loop⁴⁾ と Catmull-Clark²⁾ 細分割曲面に等しい。本手法では細分割を行うごとに新しく生じた頂点の位置を局所平滑化を用いて決めるため、平面的な細分割規則を用いても最終的に十分滑らかな曲面を生成することができる。また、本手法では解像度レベルごとの重みづけを行うため、平面的な細分割を用いると、形状の滑らかさの変更を非常に効果的に行うことができる(図 6 参照)。Loop 細分割規則や Catmull-Clark 細分割など、滑らかな細分割規則を用いることも考えられるが、これらに対応するスケール関数の形が複雑になったり、細分割の際に形状の縮小が起こり、逆に本手法に適用するには細分割規則の修正を考慮する必要がある。

4. 分解・合成フィルタの計算

Lounsbery ら⁶⁾ は、分解・合成フィルタを、スケール関数やウェーブレットの形を明示的に求めることなく、細分割規則から得られる線形連立方程式を解くことにより求めている。そのためには、細分割曲面上に定義される、スケール関数どうしの内積を要素に持つ行列 $I^{(k)}$ を計算する必要がある。ここで細分割曲面上の内積は、その多面体の面の形や大きさに依存しないように定義される。

行列 $I^{(k)}$ は、 (i, j) 成分が $\langle \varphi_i^{(k)}, \varphi_j^{(k)} \rangle$ となるような、 $m^{(k)} \times m^{(k)}$ 行列になる。ここで、 $\langle \varphi_i^{(k)}, \varphi_j^{(k)} \rangle$ は $\varphi_i^{(k)}$ と $\varphi_j^{(k)}$ の内積である。細分割規則から、次のような $I^{(k)}$ と $I^{(k+1)}$ の間の関係が導かれる。

$$I^{(k)} = P^{(k)\top} I^{(k+1)} P^{(k)} \tag{8}$$

細分割は局所的に行われるので、ある一定以上の解像度レベルにおいて、式 (8) は 0 でないスケール関数の内積に関する連立線形方程式となり、結果としてすべての解像度レベルにおいて $I^{(k)}$ を計算することができる。

$I^{(k)}$ が得られると、細分割曲面の多重解像度解析のための分解・合成フィルタは、以下のように計算できる。まず、 $m^{(k)} \times n^{(k)}$ 行列 $\alpha^{(k)} = (\alpha_{i,j}^{(k)})$ を考える。また、 $\varphi_\alpha^{(k+1)}$ ($\alpha = 1, \dots, n^{(k)}$) を、解像度レベル $k+1$ の細分割曲面 $\Theta^{(k+1)}$ において、新しく生成された頂点に対応するスケール関数であるとする。ここで、ウェーブレット $\psi_\alpha^{(k)}$ ($\alpha = 1, \dots, n^{(k)}$) がスケール関数 $\varphi_\alpha^{(k)}$ の最小二乗射影になるように決めると、 $\forall \gamma \in \rho_\alpha^{(k)}$ に関して以下が満たされる。

$$\sum_{\beta \in \rho_\alpha^{(k)}} \alpha_{\beta,\alpha}^{(k)} \langle \varphi_\beta^{(k)}, \varphi_\gamma^{(k)} \rangle = \langle \varphi_\alpha^{(k+1)}, \varphi_\gamma^{(k)} \rangle \tag{9}$$

この式 (9) を用いて、 $\alpha^{(k)}$ を計算する。ここで、 $\rho_\alpha^{(k)}$ は、文献 6) で説明されているように $\psi_\alpha^{(k)}$ の局所台を表していて、あらかじめ決めておくことができる。また、 $\langle \varphi_\beta^{(k)}, \varphi_\gamma^{(k)} \rangle, \langle \varphi_\alpha^{(k+1)}, \varphi_\gamma^{(k)} \rangle$ は、それぞれ $I^{(k)}, I^{(k+1)}$ の要素として求めることができる。

分解・合成フィルタは、 $O^{(k)}, N^{(k)}$ と $\alpha^{(k)}$ から求めることができる。特に、本手法においては $O^{(k)} = 1$ であるので、それぞれのフィルタは、以下のように簡単に書くことができる。

$$\begin{pmatrix} P^{(k)} & Q^{(k)} \end{pmatrix} = \begin{pmatrix} 1 & -\alpha^{(k)} \\ N^{(k)} & 1 - N^{(k)}\alpha^{(k)} \end{pmatrix} \tag{10}$$

$$\begin{pmatrix} A^{(k)} \\ B^{(k)} \end{pmatrix} = \begin{pmatrix} 1 - \alpha^{(k)}N^{(k)} & \alpha^{(k)} \\ -N^{(k)} & 1 \end{pmatrix} \tag{11}$$

具体的なフィルタの計算例は、文献 6) を参照されたい。

5. 幾何制約による形状制御

この章では、ある解像度レベル k の細分割曲面 $\Theta^{(k)}$ に与えられた制約を、どのように定式化するかを説明する。本研究で用いる幾何制約は、文献 9) にあるように、有限次元制約と超限次元制約に分類される。有限次元制約は、点や微分値などのある離散パラメータ値における制約である。たとえば、曲面 $\Theta^{(k)}$ をあるパラメータ値 x_0 において点 θ_0 に固定する制約は、

$$\Theta^{(k)}(x_0) = \varphi^{(k)}(x_0)^T \cdot c^{(k)} = \theta_0 \quad (12)$$

と書くことができる．超次元制約は，曲線や曲面などの連続パラメータ領域における制約である．たとえば，パラメータ曲線 $l(t)$ に課せられた曲線制約 $L(l(t))$ は， $l(t)$ 上において曲面と制約の差の最小二乗をとることにより，次のように書くことができる．

$$\begin{aligned} & \left(\int_l \varphi^{(k)}(l) \cdot \varphi^{(k)}(l)^T dt \right) c^{(k)} \\ &= \int_l L \cdot \varphi^{(k)}(l) dt \end{aligned} \quad (13)$$

これより，どちらの制約も最終的には次のような $c^{(k)}$ に関する線形連立方程式になる．

$$M^{(k)} c^{(k)} = r^{(k)} \quad (14)$$

制約を課する際に，細分割曲面上にパラメータを導入する必要がある．本手法では，細分割曲面上のそれぞれの面に重心座標を導入する．たとえば，点 v_0 が三角形 $\Delta v_i^{(k)} v_j^{(k)} v_k^{(k)}$ に含まれていて， $v_0 = t_i v_i^{(k)} + t_j v_j^{(k)} + t_k v_k^{(k)}$ の式が成り立つとする．このとき，その点のパラメータ値は，

$$(0 \cdots 0 t_i 0 \cdots 0 t_j 0 \cdots 0 t_k 0 \cdots 0), \quad (15)$$

と書くことにする．ただし，式(15)において， t_i, t_j, t_k の値は i, j, k 番目の要素として現れることに注意する．一般的な多角形の重心座標については，文献10)を参照されたい．

6. 細分割曲面の平滑化

6.1 局所平滑化フィルタリング

本手法は，与えられた幾何制約を満たす滑らかな形状を求めるために，Taubinの局所平滑化フィルタ⁸⁾を用いる．Taubinの手法は，頂点数に比例する計算量で，滑らかな多面体曲面形状を求めることができる点が優れている．

Taubinの手法は，形状の収縮を回避しながら Gaussian フィルタを low-pass フィルタに拡張することで実現される．まず，インデックス i の頂点の (x, y, z) いずれかの座標を，以下の Gaussian フィルタで平滑化する．

$$v_i' = v_i + p \Delta v_i \quad (16)$$

ここで， p は， $0 < p < 1$ を満たすようなスケール因子である．また， Δv_i は隣接する頂点との差を平均した値 $\Delta v_i = \frac{1}{n} \sum_{j=1}^n (v_j - v_i)$ であり， $j (= 1, \dots, n)$ は隣接する頂点のインデックスを表している．Gaussian フィルタだけでは形状の収縮が生じるので，次のような負の Gaussian フィルタを用いて，形状を膨ら

ませる．

$$v_i'' = v_i' + q \Delta v_i' \quad (17)$$

ここで， q は負のスケール因子で， $q < -p$ を満たす．Taubinの論文⁸⁾では，スケール因子 p と q は， $\frac{1}{p} + \frac{1}{q} = 0.1$ を満たすように選ばれている．

6.2 線形制約の扱い

Taubinは，彼の論文⁸⁾で頂点の座標に課せられた線形連立方程式を扱う手法を示している．確かに彼の手法は，1つの解像度レベルの線形制約に対してはよく機能するが，その制約が多重解像度レベルにわたるものだと，あまり良い結果を示さない．そこで，我々は次のような手法で，線形制約を満たす滑らかな形状を計算する．ここでは，ある特定の解像度レベル k に限定して考え，さらに上付き添字 (k) を省略していく．

まず c_j が， j 回平滑化処理を施した後の，細分割曲面の頂点(の1つの座標)のベクトルを表すとする． $j+1$ 回目の平滑化処理により， c_j が \bar{c}_j に変更され，与えられた制約 $Mc = r$ から少し逸脱することになる．ベクトル \bar{c}_j が $Mc = r$ を満たすようにするため，微小量 \tilde{c}_j を加えて $c_{j+1} = \bar{c}_j + \tilde{c}_j$ とし，方程式 $Mc_{j+1} = r$ を満たすようにする．その際，その微小量の大きさの二乗 $\tilde{c}_j^T \tilde{c}_j$ を最小化する．これは，ラグランジュ乗数 y を用いて $\frac{1}{2} \tilde{c}_j^T \tilde{c}_j + (M\tilde{c}_j - (r - M\bar{c}_j))^T y$ を最小化することに置き換えることができ，最終的には

$$\tilde{c}_j = M^T (MM^T)^{-1} (r - M\bar{c}_j). \quad (18)$$

を解くことによって \tilde{c}_j が得られる．この処理を，十分滑らかな曲面が得られるまで続けるわけである．ここで，行列 MM^T さらに $M^T (MM^T)^{-1}$ は，上の局所平滑化処理を繰り返す前に計算できることに注意する．線形方程式の数を σ とすると，行列 M は粗行列であり，さらに MM^T は $\sigma \times \sigma$ 行列であるので，この行列計算は線形方程式の数にほぼ依存する．

7. 多重解像度局所平滑化

この章では，細分割曲面のそれぞれの分割レベルに与えられた制約を，局所平滑化を用いてどのように解くかについて説明する．ここで注意すべき点は，本提案手法が，我々が B-スプライン曲面の際¹¹⁾に実現した，グローバルな形状最適化⁹⁾に基づく解像度レベルごとの滑らかさの重み付けの効果を，非常に効率良く近似，そして模倣できる点である．

本手法の方針として，低解像度レベル 0 から高解像度レベル K まで各解像度レベルごとに，細分割曲面に形状の起伏を加えていきながら，それぞれの解像度レベルで局所平滑化を施すことによって，最終的な

$$\begin{array}{rccccccc}
 \mathbf{c}^{(K)} & = & \mathbf{P}^{(K-1)} \dots \mathbf{P}^{(0)} \widehat{\mathbf{c}}^{(0)} & + & \mathbf{P}^{(K-1)} \dots \mathbf{P}^{(1)} \widehat{\mathbf{c}}^{(1)} & + & \dots & + & \widehat{\mathbf{c}}^{(K)} \\
 & & \uparrow & & \uparrow & & & & \downarrow \\
 \vdots & & \vdots & & \vdots & & & & \vdots \\
 \mathbf{c}^{(1)} & = & \mathbf{P}^{(0)} \widehat{\mathbf{c}}^{(0)} & + & \widehat{\mathbf{c}}^{(1)} & + & \dots & + & \mathbf{A}^{(1)} \dots \mathbf{A}^{(K-1)} \widehat{\mathbf{c}}^{(K)} \\
 & & \uparrow & & \downarrow & & & & \downarrow \\
 \mathbf{c}^{(0)} & = & \widehat{\mathbf{c}}^{(0)} & + & \mathbf{A}^{(0)} \widehat{\mathbf{c}}^{(1)} & + & \dots & + & \mathbf{A}^{(0)} \dots \mathbf{A}^{(K-1)} \widehat{\mathbf{c}}^{(K)}
 \end{array}$$

図2 ベクトル $\widehat{\mathbf{c}}^{(k)}$ ($k = 0, \dots, K$) と $\mathbf{c}^{(k)}$ ($k = 0, \dots, K$) の関係
 Fig.2 Relationships between the vectors $\widehat{\mathbf{c}}^{(k)}$ ($k = 0, \dots, K$) and $\mathbf{c}^{(k)}$ ($k = 0, \dots, K$).

形状を決定していく。そこでまず、解像度レベル k で与えられる形状の起伏をベクトル $\widehat{\mathbf{c}}^{(k)}$ ($k = 0, \dots, K$) と書くことにする。このとき、解像度レベル k の細分割曲面の頂点座標を表す $\mathbf{c}^{(k)}$ とは、図2のような関係があることに注意する¹¹⁾。

最初に考えなければならないのは、それぞれのレベル k に重み値 $\kappa^{(k)}$ ($k = 0, \dots, K$) が与えられたとき、 $\widehat{\mathbf{c}}^{(k)}$ ($k = 0, \dots, K$) をどのように求めればよいかである。本手法では、付録 A.1 の考察に基づき、レベル k だけを考慮に入れて求めた形状の

$$\frac{\kappa^{(k)}}{\kappa^\Sigma} \tag{19}$$

の割合を $\widehat{\mathbf{c}}^{(k)}$ に割り当てることにする。ただし、

$$\kappa^\Sigma = \sum_{\lambda=0}^K \kappa^{(\lambda)} \tag{20}$$

である。以下、各解像度レベル k ($k = 0, \dots, K$) において、線形制約方程式 (14) が課せられているとして、解像度レベルごとに、どのように形状の起伏を決定していくかを具体的に見ていく。

まず、解像度レベル 0 においては、そのレベルに課せられている制約方程式

$$\mathbf{M}^{(0)} \widehat{\mathbf{c}}_f^{(0)} = \mathbf{r}^{(0)} \tag{21}$$

を満たすようなベクトル $\widehat{\mathbf{c}}_f^{(0)}$ を考える。実際に求める形状の起伏 $\widehat{\mathbf{c}}^{(0)}$ は、先に述べた重みの割当てより、

$$\widehat{\mathbf{c}}^{(0)} = \frac{\kappa^{(0)}}{\kappa^\Sigma} \widehat{\mathbf{c}}_f^{(0)} \tag{22}$$

として求める。ここで、式 (21) と (22) より、 $\widehat{\mathbf{c}}^{(0)}$ に関する制約方程式

$$\mathbf{M}^{(0)} \widehat{\mathbf{c}}^{(0)} = \frac{\kappa^{(0)}}{\kappa^\Sigma} \mathbf{r}^{(0)} \tag{23}$$

が得られるので、実際には式 (23) が満たされるように、 $\widehat{\mathbf{c}}^{(0)}$ に 6 章で述べた局所平滑化を施せばよい。その際、 $\widehat{\mathbf{c}}^{(0)}$ の初期値も重要になるが、今は何もなところから始めているので、単純に

$$\widehat{\mathbf{c}}^{(0)} = \mathbf{0} \tag{24}$$

を初期値に設定する。

次に、解像度レベル 1 について考える。最初に、このレベル 1 で課せられた制約を満たす形状の起伏 $\widehat{\mathbf{c}}_f^{(1)}$ を考える。これは、文献 12) で提案した手法を用いて、レベル 0 と 1 のそれぞれの制約方程式をまとめた

$$\begin{pmatrix} \mathbf{M}^{(0)} \mathbf{A}^{(0)} \\ \mathbf{M}^{(1)} \end{pmatrix} \widehat{\mathbf{c}}_f^{(1)} = \begin{pmatrix} \mathbf{r}^{(0)} \\ \mathbf{r}^{(1)} \end{pmatrix} \tag{25}$$

を満たすベクトル $\widehat{\mathbf{c}}_f^{(1)}$ を求めればよい。ここで、解像度レベル 1 だけでなくレベル 0 の方程式を考慮に入れるのは、解像度レベル 0 の制約を保持したまま、解像度レベル 1 の制約を満たす滑らかな形状を求める必要があるためである。

実際の形状の起伏 $\widehat{\mathbf{c}}^{(1)}$ は、求めた形状 $\widehat{\mathbf{c}}_f^{(1)}$ のうち解像度レベル 0 と 1 に相当する重みの割合を求め、さらにそこからすでに得ている $\widehat{\mathbf{c}}^{(0)}$ に対応する部分を取り除いて求める。こうすることにより、解像度レベル 1 に見合った細かさの起伏を、 $\widehat{\mathbf{c}}^{(1)}$ に含ませることができる。具体的には、 $\widehat{\mathbf{c}}_f^{(1)}$ に解像度レベル 0 と 1 に相当する重みの割合 $(\kappa^{(0)} + \kappa^{(1)})/\kappa^\Sigma$ をかけたものから、 $\widehat{\mathbf{c}}^{(0)}$ に相当する部分 $\mathbf{P}^{(0)} \widehat{\mathbf{c}}^{(0)}$ を取り去ったものを $\widehat{\mathbf{c}}^{(1)}$ とするので、

$$\widehat{\mathbf{c}}^{(1)} = \frac{\kappa^{(0)} + \kappa^{(1)}}{\kappa^\Sigma} \widehat{\mathbf{c}}_f^{(1)} - \mathbf{P}^{(0)} \widehat{\mathbf{c}}^{(0)} \tag{26}$$

となる。式 (25) と (26) より、 $\widehat{\mathbf{c}}^{(1)}$ は以下の方程式を満たす。

$$\begin{pmatrix} \mathbf{M}^{(0)} \mathbf{A}^{(0)} \\ \mathbf{M}^{(1)} \end{pmatrix} \widehat{\mathbf{c}}^{(1)} = \frac{\kappa^{(0)} + \kappa^{(1)}}{\kappa^\Sigma} \begin{pmatrix} \mathbf{r}^{(0)} \\ \mathbf{r}^{(1)} \end{pmatrix} - \begin{pmatrix} \mathbf{M}^{(0)} \mathbf{A}^{(0)} \\ \mathbf{M}^{(1)} \end{pmatrix} \mathbf{P}^{(0)} \widehat{\mathbf{c}}^{(0)} \tag{27}$$

解像度レベル 0 のときは異なり、 $\widehat{\mathbf{c}}^{(1)}$ の初期値は慎重に選ばないと、解像度レベル 1 に見合う細かさの起伏を $\widehat{\mathbf{c}}^{(1)}$ に含めることができない。初期値としては、解像度レベル 1 の起伏は含まれないが、レベル 0

の起伏は含まれているものが望ましい。このとき、解像度レベル 0 までで計算した起伏は、解像度レベル 1 では $P^{(0)}\hat{c}^{(0)}$ と書くことができるので、重みの割合を考慮すると、

$$\frac{\kappa^{(1)}}{\kappa^{(0)}}P^{(0)}\hat{c}^{(0)} \quad (28)$$

を $\hat{c}^{(1)}$ の初期値にすればよいことが分かる。

ここまで考えると、一般的な解像度レベル k についても、まとめることができる。まずベクトル $\hat{c}_f^{(k)}$ が、次のような方程式を満たすと考える。

$$\widehat{M}^{(k)}\hat{c}_f^{(k)} = \hat{r}^{(k)} \quad (29)$$

ただし、

$$\widehat{M}^{(k)} = \begin{pmatrix} M^{(0)}A^{(0)} \dots A^{(k-1)} \\ \vdots \\ M^{(k)} \end{pmatrix}, \quad \hat{r}^{(k)} = \begin{pmatrix} r^{(0)} \\ \vdots \\ r^{(k)} \end{pmatrix} \quad (30)$$

である。この方程式は、レベル 0 から k までの制約方程式をまとめたものである¹²⁾。次に、 $\hat{c}^{(k)}$ と $\hat{c}_f^{(k)}$ の関係は、重みの割合を考慮して、

$$\hat{c}^{(k)} = \frac{\kappa^{(0)} + \dots + \kappa^{(k)}}{\kappa^\Sigma} \hat{c}_f^{(k)} - \sum_{\lambda=0}^{k-1} P^{(k-1)} \dots P^{(\lambda)} \hat{c}^{(\lambda)} \quad (31)$$

となればよい。式 (31) は方程式 (29) を満たすベクトル $\hat{c}_f^{(k)}$ から、そのレベルまでの割合

$$\frac{\kappa^{(0)} + \dots + \kappa^{(k)}}{\kappa^\Sigma} \quad (32)$$

をかけて、そこから今まで求めた $\hat{c}^{(0)}, \dots, \hat{c}^{(k-1)}$ の分だけ取り除いたものを、 $\hat{c}^{(k)}$ として残すという意味を表している。ここで、式 (29) と (31) より、

$$\widehat{M}^{(k)}\hat{c}^{(k)} = \frac{\kappa^{(0)} + \dots + \kappa^{(k)}}{\kappa^\Sigma} \hat{r}^{(k)} - \widehat{M}^{(k)} \left(\sum_{\lambda=0}^{k-1} P^{(k-1)} \dots P^{(\lambda)} \hat{c}^{(\lambda)} \right) \quad (33)$$

のような $\hat{c}^{(k)}$ に関する方程式を得ることができる。実際の平滑化は、式 (33) が満たされるように、 $\hat{c}^{(k)}$ に対して行えばよい。この際、 $\hat{c}^{(k)}$ には、解像度レベル k に見合った細かさの起伏が含まれることになる。

平滑化を行う際の $\hat{c}^{(k)}$ の初期値についてもまとめる。

解像度レベル 1 の初期値の選び方と同じように、今まで求めた解像度レベル $k-1$ までの起伏 $\hat{c}^{(0)}, \dots, \hat{c}^{(k-1)}$ の和に、適当な割合をかけたものを初期値として採用する。今まで求めた起伏は、レベル k においては

$$\sum_{\lambda=0}^{k-1} P^{(k-1)} \dots P^{(\lambda)} \hat{c}^{(\lambda)} \quad (34)$$

と書くことができる。それぞれの解像度レベル λ には、 $\kappa^{(\lambda)}$ の重みが割り当てられているから、 $\hat{c}^{(k)}$ の実際の初期値には、

$$\frac{\kappa^{(k)}}{\kappa^{(0)} + \dots + \kappa^{(k-1)}} \sum_{\lambda=0}^{k-1} P^{(k-1)} \dots P^{(\lambda)} \hat{c}^{(\lambda)} \quad (35)$$

を与えればよい。この初期値には、解像度レベル $k-1$ までの起伏は含まれているが、レベル k の起伏は含まれていないことに注意する。

このようにして、解像度レベル 0 から K まで順々に局所平滑化を施して、 $\hat{c}^{(k)}$ ($k=0, \dots, K$) を求めればよい。実装したシステムにおいては、実際には

$$\kappa^{(k)} = \gamma^{k-K} \quad (36)$$

のように指数関数を用いて重み値を割り当てている。

8. 設 計 例

図 3 は、キャラクタの顔形状を設計した例である。ここで曲面に付加されている点は点制約を表す。この例では、大局から局所へ幾何制約を付加して形状を制御していて、レベル 0 で大まかな形状を指定し、レベル 1 で頬の膨らみを、レベル 2 で口の膨らみをレベル 3 でくぼみを作っている。図 4 は、同じ制御多面体から異なる幾何制約を付加して、おもちゃの豚の顔形状を設計した例である。ここで、曲面に付加されている線は曲線制約を表す。図 5 は、これら 2 つの形状にテクスチャを施したものである。

図 6 は、式 (36) の γ の値が異なるときの、トーラスの形状の滑らかさの変化を示している。本手法では、形状は各座標独立に計算されるので、図 7 に示されるように、曲面に座標ごとに違う滑らかさを与えることができる。図 8 は、任意位相の曲面の設計例である。

実装したシステムでは、初期制御多面体 $\Theta^{(0)}$ が入力されると、分解・合成フィルタを一度前処理として計算し、制約に依存する計算だけを実際の設計作業の際に行う。図 3 の例では、Pentium 350 MHz の CPU を持つ PC (Linux OS) を用いて、前処理で分解・合成フィルタを計算するのに 81 秒、付随するフィルタ

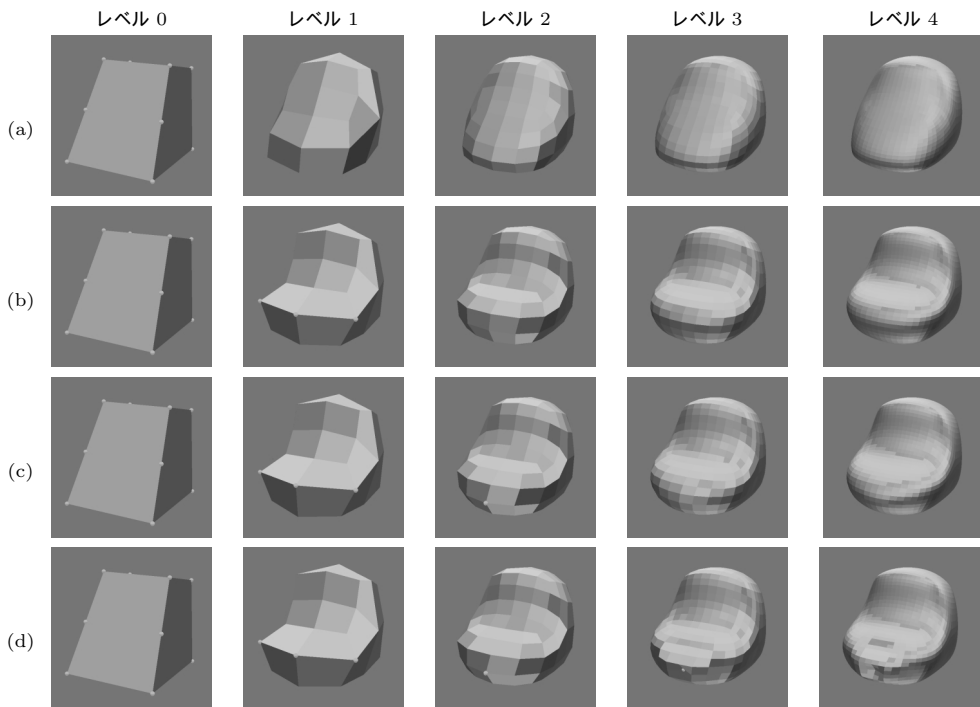


図 3 幾何制約を用いたキャラクターの顔形状の設計 : (a) レベル 0 まで, (b) レベル 1 まで, (c) レベル 2 まで, (d) レベル 3 まで
 Fig. 3 Designing the facial shape of a character using constraints up to (a) level 0, (b) level 1, (c) level 2, and (d) level 3.



図 4 おもちゃの豚の顔形状の設計
 Fig. 4 Designing the facial face of a toy pig.

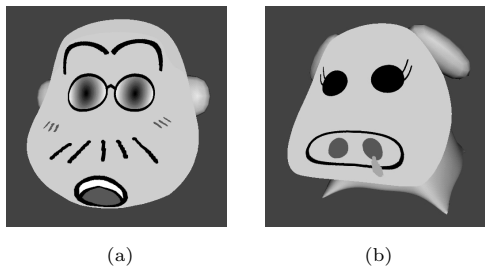


図 5 設計例 : (a) キャラクターの顔, (b) おもちゃの豚
 Fig. 5 Design examples: (a) facial shape of a character and (b) a toy pig.

を計算するのに 106 秒, さらに実際の設計の際にレベル 3 の形状なら 6 秒, レベル 4 の形状なら 61 秒かかる .

9. まとめと今後の課題

本論文では, 細分割曲面の異なる分割レベルに付加された幾何制約に対し, それらを満たす滑らかな形状を求める新しい手法を提案した . 提案手法は, 頂点数に比例する計算量の局所平滑化を用いていて, 低解像度から高解像度へとそれぞれのレベルで局所平滑化を施して滑らかな形状を求める . 本手法の特質として, それぞれの解像度レベルに異なる重みを割り当てることで, 細分割曲面の滑らかさを制御することができる点あげられる .

本手法のさらなる発展として, 曲面の細分割が部分ごとに異なるような, 適応的細分割を実現することがあげられる . また, 形状のある部分だけに局所平滑化

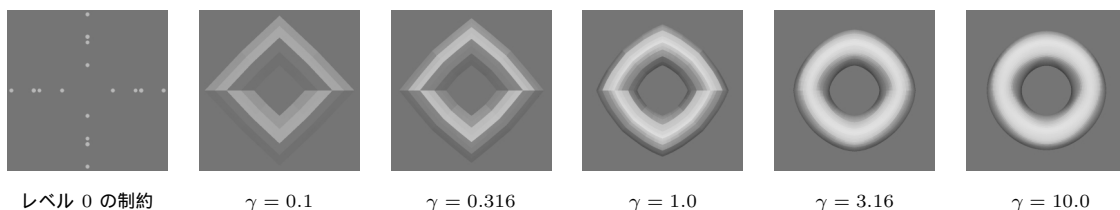


図 6 解像度レベル 0 の制約と、滑らかさの異なるレベル 4 のトーラスの形状. γ の値は式 (36) を参照
Fig. 6 Geometric constraints at level 0 and tori with different smoothness at level 4. See Eq. (36) for the definition of γ .

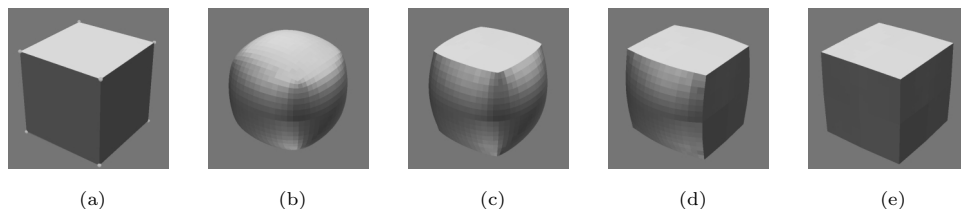


図 7 座標ごとの滑らかさの制御: (a) レベル 0 の幾何制約, (b) すべての座標に滑らかさを適用, (c) x , y -座標に滑らかさを適用, (d) x -座標に滑らかさを適用, (e) 同じく滑らかさを適用しない場合

Fig. 7 Controlling the smoothness of each coordinate: (a) constraints at level 0, (b) surfaces at level 4 where the local smoothness is applied to all the coordinates, (c) to the x - and y -coordinates, (d) to the x -coordinate only, and (e) the local smoothness is not applied.

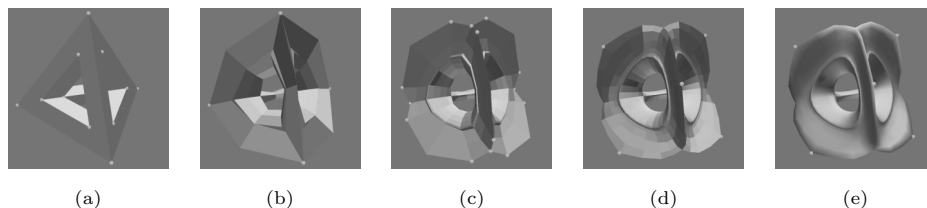


図 8 穴のある四面体: (a) レベル 0, (b) レベル 1, (c) レベル 2, (d) レベル 3 の形状と, (e) そのスムーズ・シェーディング

Fig. 8 A tetrahedron with holes: surfaces at (a) level 0, (b) level 1, (c) level 2, and (d) level 3. (e) its rendered image.

を施し, その他の部分は固定できるような枠組みも, 計算時間の短縮につながる. さらに, 制御多面体 $\Theta^{(0)}$ の位相構造が局所的に変化したとき, 分解・合成フィルタの再計算を効果的に行うことも考慮すべきである. レーザ・スキャナなどから取得した実際の形状サンプル・データを, 本システムに組み込むことも今後の課題である.

謝辞 本論文の執筆にあたり, ご協力いただいた群馬大学工学部情報工学科の愛島快行, 山野智久両氏に感謝いたします. 本研究の一部は, 文部省科学研究費奨励研究(A)(課題番号 10780165)および栢森情報科学振興財団の助成による.

参考文献

- 1) Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W.: Mesh Optimization, *Computer Graphics (Proceedings Siggraph '93)*, pp.19–26 (1993).
- 2) Catmull, E. and Clark, J.: Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes, *Computer-Aided Design*, Vol.10, No.6, pp.350–355 (1978).
- 3) Doo, D. and Sabin, M.: Behaviour of Recursive Division Surfaces near Extraordinary Points, *Computer-Aided Design*, Vol.10, No.6, pp.356–360 (1978).
- 4) Loop, C.: Smooth Subdivision Surfaces Based on Triangles, Master's Thesis, University of

Utah, Department of Mathematics (1987).

- 5) Dyn, N., Levin, D. and Gregory, J.A.: A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control, *ACM Trans. Graphics*, Vol.9, No.2, pp.160–169 (1990).
- 6) Lounsbery, M., DeRose, T.D. and Warren, J.: Multiresolution Analysis for Surfaces of Arbitrary Topological Type, *ACM Trans. Graphics*, Vol.16, No.1, pp.34–73 (1997).
- 7) Zorin, D., Schröder, P. and Sweldens, W.: Interactive Multiresolution Mesh Editing, *Computer Graphics (Proc. Siggraph '97)*, pp.259–268 (1997).
- 8) Taubin, G.: A Signal Processing Approach To Fair Surface Design, *Computer Graphics (Proc. Siggraph '95)*, pp.351–358 (1995).
- 9) Welch, W. and Witkin, A.: Variational Surface Modeling, *Computer Graphics (Proc. Siggraph '92)*, pp.157–166 (1992).
- 10) Loop, C.T. and DeRose, T.D.: A Multisided Generalization of Bézier Surfaces, *ACM Trans. Graphics*, Vol.8, No.3, pp.204–234 (1990).
- 11) Takahashi, S.: Variational Design of Curves and Surfaces using Multiresolution Constraints, *The Visual Computer*, Vol.14, No.5/6, pp.208–227 (1998).
- 12) Takahashi, S., Shinagawa, Y. and Kunii, T.L.: Curve and Surface Design using Multiresolution Constraints, *Computer Graphics International '97*, Vince, J. and Reeth, F.V. (Eds.), pp.121–130, IEEE Computer Society Press (1997).
- 13) Terzopoulos, D.: Regularization of Inverse Visual Problems Involving Discontinuities, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.8, No.4, pp.413–424 (1986).

付 録

A.1 薄板モデルからの近似

ここでは, Terzopoulos¹³⁾ の薄板モデルに基づき, 多重解像度制約¹¹⁾ を解く際に, 求められる各解像度レベルの起伏 $\hat{c}^{(k)}$ ($k = 0, \dots, K$) が, 解像度に割り当てられた重み値とどのような関係にあるかを考察する. まず, $\hat{c}^{(k)}$ をまとめて,

$$c_\sigma^\top = \left(\hat{c}^{(0)\top} \quad \hat{c}^{(1)\top} \quad \dots \quad \hat{c}^{(K)\top} \right) \quad (37)$$

と表す. このとき, $c^{(k)}$ と $\hat{c}^{(k)}$ との関係は, 図 2 に示されるとおりである. 各解像度レベル k において, 線形制約方程式 (14) が与えられているとすると, 図 2 の関係を用いて次のような c_σ に関する方程式を得ることができる¹¹⁾.

$$M_\sigma c_\sigma = r_\sigma \quad (38)$$

次に, 最小化すべき関数を考える. 各解像度レベル k ($k = 0, \dots, K$) において, $\hat{c}^{(k)}$ に関するエネルギー関数は $\hat{c}^{(k)\top} S^{(k)} \hat{c}^{(k)}$ となる¹¹⁾. ここで $S^{(k)}$ は, 上記関数が解像度レベル k の形状の起伏の曲げ・伸びを表すようにとられた正定値行列である⁹⁾. 解像度レベル k のエネルギー関数に, $1/\kappa^{(k)}$ の重みが割り振られたとき, c_σ に関するエネルギー関数は, 満たすべき制約にラグランジュ乗数をかけたものと合わせて,

$$\frac{1}{2} c_\sigma^\top S_\sigma c_\sigma + (M_\sigma c_\sigma - r_\sigma)^\top y \quad (39)$$

となる. ただし,

$$S_\sigma = \begin{pmatrix} (1/\kappa^{(0)})S^{(0)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & (1/\kappa^{(K)})S^{(K)} \end{pmatrix} \quad (40)$$

である. 結局 c_σ は, 以下の線形連立方程式を解いて求めることができる.

$$c_\sigma = K_\sigma J_\sigma^{-1} r_\sigma \quad (41)$$

ただし,

$$J_\sigma = M_\sigma S_\sigma^{-1} M_\sigma^\top, \quad K_\sigma = S_\sigma^{-1} M_\sigma^\top \quad (42)$$

である. また, 行列 S_σ の逆行列 S_σ^{-1} は,

$$S_\sigma^{-1} = \begin{pmatrix} \kappa^{(0)}S^{(0)-1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \kappa^{(K)}S^{(K)-1} \end{pmatrix} \quad (43)$$

と書くことができる.

ここで, 線形制約方程式が 1 つしかない場合を考える. このとき, 制約を表す行列 M_σ とベクトル r_σ は, それぞれベクトル m_σ とスカラー値 r になる. また, ベクトル m_σ の要素を, 各解像度レベルの形状を表す $c^{(k)}$ の次元にあわせて, $m_\sigma = (m_\sigma^{(0)} \dots m_\sigma^{(K)})$ と分割しておく. すると, 行列 $J_\sigma = M_\sigma S_\sigma^{-1} M_\sigma^\top$ は 1×1 行列, つまりスカラーになり

$$J_\sigma^{-1} = \frac{1}{\sum_{k=0}^K \kappa^{(k)} m_\sigma^{(k)\top} S^{(k)-1} m_\sigma^{(k)}} \quad (44)$$

となる. 式 (41) より,

$$\hat{c}^{(k)} = \frac{\kappa^{(k)} r S^{(k)-1} m_\sigma^{(k)}}{\sum_{k=0}^K \kappa^{(k)} m_\sigma^{(k)\top} S^{(k)-1} m_\sigma^{(k)}} \quad (45)$$

と書くことができる. ここで,

$$h^{(k)} = m_\sigma^{(k)\top} S^{(k)-1} m_\sigma^{(k)} (> 0) \quad (46)$$

とおく，

$$\hat{\mathbf{c}}^{(k)} = \frac{\kappa^{(k)}}{\sum_{k=0}^K h^{(k)} \kappa^{(k)}} r \mathbf{S}^{(k)-1} \mathbf{m}_\sigma^{(k)\top} \quad (47)$$

さらに， $h^{(k)}$ がすべて同じ大きさになるという仮定をおいて簡単化を行うと，

$$\hat{\mathbf{c}}^{(k)} = \frac{\kappa^{(k)}}{\kappa^\Sigma} r \mathbf{S}^{(k)-1} \mathbf{m}_\sigma^{(k)\top} \quad (48)$$

となる (κ^Σ の定義は，式 (20) を参照). これより，各解像度レベルで $\kappa^{(k)}$ ($k = 0, \dots, K$) の重みが割り当てられているとすると，そのレベル k の形状は，他のレベルを考慮に入れないで求めた値の

$$\frac{\kappa^{(k)}}{\kappa^\Sigma} \quad (49)$$

の割合で近似することにする．複数の制約が与えられたときには，実際には解像度間の重みの割当ての影響

が複雑になるが，式 (49) の近似を用いて局所平滑化で求めた結果は，薄板モデル下で得た結果を非常によく模倣する．

(平成 11 年 8 月 31 日受付)

(平成 11 年 11 月 4 日採録)



高橋 成雄 (正会員)

昭和 43 年生．平成 9 年東京大学理学系研究科情報科学専攻博士課程修了．博士 (理学)．同年群馬大学工学部情報工学科助手．平成 11 年より同大学総合情報処理センター助教授．コンピュータ・グラフィックス，幾何形状モデリング，地理情報システム等に興味を持つ．ACM，IEEE，電子情報通信学会各会員．