

リフティングスキームによる四辺形メッシュの多重解像度表現

仲田 晋[†], 萩原 一郎[†]

3次元形状をメッシュモデルで詳細に表現するためには大規模なメッシュデータを必要とし、表示の際の計算量はその詳細度に応じて膨大なものとなる。このような詳細なメッシュからなるモデルを高速に表示するために、必要に応じてメッシュを簡略化して表示する多重解像度表現が広く用いられている。我々は四辺形メッシュからなる3次元形状モデルに対し、新しい多重解像度表現の手法を提案する。これは高速ウェーブレット変換アルゴリズムの1つであるリフティングスキームとメッシュの詳細化手法である細分割法に基づいた手法であり、以下、これをリフティングサブディビジョンと呼ぶ。本手法では四辺形メッシュの簡略化を高速に行うことができ、かつ可逆性を持つ。また、データ量がつねに一定、すなわち逆変換に必要な情報のための領域を新たに用意する必要がないという特徴を持つ。

Multiresolution Representations of Quadrilateral Meshes Using Lifting Scheme

SUSUMU NAKATA[†], and ICHIRO HAGIWARA[†]

To represent highly detailed three-dimensional geometric shape in computer graphics, large number of meshes and vertices are used and much computational costs are required for its rendering. Mesh simplification methods are widely used to reduce the time for rendering of highly detailed mesh models. In this paper we propose a new multiresolution representation method for three-dimensional models represented in quadrilateral meshes. This method, Lifting Subdivision, is based on the lifting scheme, which is one of the fast wavelet transform algorithms, and on the subdivision method, which generates interpolating surfaces from mesh models. The lifting subdivision method has the properties that the meshes are simplified with low computational costs and that the transformation is invertible. Moreover, the number of data, including the information for inverse transforms, is always the same, thus any other memory space is not required in the process of mesh simplification.

1. はじめに

コンピュータグラフィクスで扱われる3次元形状は、アニメーションやCAD/CAMモデリング、医療情報、地理情報等あらゆる分野で用いられ、その多くは物体の表面をメッシュで近似的に表現したモデルを用いる。一方、近年では形状をより正確に、詳細に表現するためにデータが大規模化し、レンダリングに膨大な演算量を必要とするという問題がある。そのため、現在ではレンダリングにともなう演算量を削減するために必要に応じてメッシュを簡略化する多重解像度表現が用

いられている。

詳細なメッシュの多重解像度表現を実現する手法としては、プログレッシブメッシュに代表される連続的な多重解像度表現^{(8),(9)}とウェーブレット変換に基づく段階的な多重解像度表現^{(4),(6),(11),(14),(15),(17),(18)}が代表的である。

プログレッシブメッシュは三角形メッシュで表現された形状のエッジを1つずつ消去することにより簡略化を行う連続的なメッシュ簡略化手法である。また、エッジを消去するための変換は可逆性を持ち、簡略化されたメッシュからオリジナルのメッシュを完全に再構成することが可能である。しかし、この手法は再構成は高速に行うことができるが、簡略化の演算ではエッジごとに簡略化後の新しい頂点を選択するための基準となる関数を評価する必要があり、特にメッシュが大規模となる場合は多くの演算量を必要とする。

一方、三角形メッシュをある単純なベースメッシュの

[†] 東京工業大学大学院理工学研究科

Graduate School of Science and Engineering, Tokyo Institute of Technology

現在、立命館大学理工学部情報学科

Presently with Department of Computer Science, Ritsumeikan University

ウェーブレット逆変換で表現することにより段階的な多重解像度表現を行う手法が提案されている^{6),14),15),18)}。ここでウェーブレットは、三角形メッシュの細分割法(subdivision scheme)であるパタフライスキーム²⁾等を用いて定義され、適切なベースメッシュを用いることにより任意位相のメッシュに適用可能である。さらにウェーブレットの性質を表す代表的な指標であるパニシングモーメント¹⁹⁾をコントロールするために、球上で定義されたウェーブレットをリフティングスキーム^{20),21)}により修正する手法が提案された¹⁷⁾。これは特に3次元形状の多重解像度解析だけでなく、球上の関数のウェーブレット変換にも適する手法である。ただし、これらの手法ではウェーブレットを三角形メッシュの細分割法で定義しているため、多重解像度表現されるモデルは細分割接続性¹⁵⁾(subdivision connectivity)で表現されていなければならない。そのため、オリジナルのメッシュから細分割接続性を持つメッシュへ変換する手法も提案されている⁴⁾。ただし、この変換は不可逆である。

また、詳細メッシュ上の点をベースメッシュ上の関数で近似することで形状を表現する手法^{7),13)}では、ベースメッシュ上の関数に対してウェーブレット変換を行うことにより多重解像度表現が可能となる。これは3次元形状のパラメータ化により特にデータ圧縮、テクスチャマッピング、モーフィング等の応用に有効な手法である。

ウェーブレットは変換のためのフィルタ係数がつねに一定である第1世代ウェーブレット(first generation wavelet)と、フィルタ係数が位置によって異なる第2世代ウェーブレット(second generation wavelet)に大別されるが、上記ウェーブレット変換に基づく多重解像度解析は三角形メッシュに対応させるために第2世代ウェーブレットを用いる必要がある。

我々はこれに対し、インデックスで順序付けされた四辺形メッシュを対象とする多重解像度表現手法を提案する。本手法の特徴は、その高速性と解像度変化にともなうデータの格納法にある。

まず、本手法はウェーブレットと細分割法に基づく方法である。ただし、ここでは四辺形メッシュのインデックスによる規則的なメッシュによる第1世代ウェーブレットを用いる。すなわち変換のフィルタはつねに一定であり、第2世代ウェーブレットと比較して演算量の面で有利である。

次に細分割法は、細分割曲線を構成する手法である4-ポイントスキーム³⁾を用いる。4-ポイントスキームでの補間点は隣接する4点から求められ、これは四辺

形メッシュから細分割曲面(subdivision surface)を構成するKobbeltスキーム¹⁰⁾の基礎となっている。本手法ではこれをリフティングスキームに適用することによりウェーブレットを構成し、パニシングモーメントをコントロールする。これによりパニシングモーメントが4となるウェーブレットを構成することができる。

一方、リフティングスキームに可変の重み付けによる細分割法を適用してウェーブレットを構成する手法¹¹⁾もあるが、ここでは高速性のため4-ポイントスキームによる一定の重みを用いる。さらにリフティングスキームは、ツースケール関係を用いた高速ウェーブレット変換¹⁹⁾と比較して演算量が少ないという利点を持つ。

ここでメッシュの簡略化はウェーブレットに基づく既存の手法と同様、1回の簡略化で頂点数が1/4となる段階的な簡略化である。また本手法では3次元形状モデルとしてインデックスで順序付けされた四辺形メッシュを対象としているが、これはCTやMRI等による複数断面像から構成される形状モデルや、四辺形メッシュを与える3次元レーザスキャナから測定された形状モデルを想定したものであり、3次元CGツールキットであるTGS社Open Inventor²²⁾ではこのような四辺形メッシュの形状モデルもサポートしている。これらの方法で得られる四辺形メッシュでは、本手法によりそのメッシュの形状を活かした高速多重解像度解析が可能となる。ただし、得られるモデルの各メッシュの頂点が同一平面上にある保証はなく、レンダリングの際に不都合を生じるが、これはたとえば各メッシュを2つの三角形に分割することで回避できる。

2章では四辺形メッシュの多重解像度表現を補間型スケール関数により定義する。3章ではリフティングサブディビジョンによるメッシュの簡略化と復元の手法について述べる。4章では実際の四辺形メッシュに対するリフティングサブディビジョンによる多重解像度表現の例を示す。

2. 頂点座標の多重解像度解析

2.1 双直交ウェーブレットと多重解像度解析空間

ある3次元形状を表すモデルが $n_1 \times n_2$ 個の頂点座標

$$P_{ij} = (x_{ij}, y_{ij}, z_{ij}), \\ i = 0, \dots, n_1 - 1, \quad j = 0, \dots, n_2 - 1$$

により与えられているものとする。ここでの目的は、与えられた頂点座標集合の簡略化メッシュを作成することである。

簡単のため、点の座標 P_{ij} は整数全体 $(i, j \in \mathbb{Z})$ で与えられているものとし、これを初期座標とする。このとき、 x, y, z 座標の値についての離散ウェーブレット変換を考え、簡略化された座標集合

$$\{P_{ij}^{(l)}\}_{i,j \in \mathbb{Z}}, \quad l = -1, -2, \dots \quad (1)$$

を得る方法を以下で述べる。

まず、簡略化座標集合 (1) を定義するための双直交関数²⁰⁾ の組 $\{\varphi, \tilde{\varphi}, \psi, \tilde{\psi}\}$ を考える。ここで、 $\varphi(t) \in L_2(\mathbb{R})$ はスケーリング関数、 $\psi(t) \in L_2(\mathbb{R})$ は対応するマザーウェーブレット、 $\tilde{\varphi}(t) \in L_2(\mathbb{R})$ は共役スケーリング関数、 $\tilde{\psi}(t) \in L_2(\mathbb{R})$ は共役ウェーブレットを表す。

また、共役ウェーブレット $\tilde{\psi}(t)$ とウェーブレット $\psi(t)$ のパニシングモーメント¹⁹⁾ をそれぞれ N, \tilde{N} で表す。

さらに、これらの関数の 2 進ダイレーションとシフトを $l, k \in \mathbb{Z}$ について次のように定義する。

$$\varphi_{l,k}(t) := \varphi(2^l t - k), \quad \tilde{\varphi}_{l,k}(t) := \tilde{\varphi}(2^l t - k), \quad (2)$$

$$\psi_{l,k}(t) := \psi(2^l t - k), \quad \tilde{\psi}_{l,k}(t) := \tilde{\psi}(2^l t - k). \quad (3)$$

このとき、スケーリング関数 $\{\varphi_{l,k}\}$ は多重解像度解析空間を構成する^{1),19)}。以下、 l をレベルと呼ぶ。

本論文ではスケーリング関数 $\varphi(t)$ が次の 2 つの性質を満たすものと仮定する。

仮定 1 スケーリング関数 $\varphi(t)$ は以下を満たす。

- (1) $\varphi(t)$ は補間型
- (2) $\text{supp } \varphi(t)$ はコンパクト

ここで、補間型のスケーリング関数は次のように定義される²⁰⁾。

定義 2 スケーリング関数 $\varphi(t) \in L_2(\mathbb{R})$ がすべての $i \in \mathbb{Z}$ に対して

$$\varphi(i) = \begin{cases} 1, & i = 0, \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

を満たすとき、 $\varphi(t)$ を補間型という。

以下では初期座標 (1) を補間する関数を補間型スケーリング関数により構成し、その多重解像度解析により頂点座標の簡略化を定義する。

2.2 補間型スケーリング関数による多重解像度表現の定義

本節では補間型スケーリング関数を用いて与えられた初期座標 (1) の多重解像度表現を定義する。ただし、以下では頂点座標 P_{ij} の x 座標の集合 $\{x_{ij}\}_{i,j \in \mathbb{Z}}$ についての多重解像度表現のみを示すこととする。 y 座

標、 z 座標については同様の方法で行うため省略する。

まず、与えられた集合 $\{x_{ij}\}_{i,j \in \mathbb{Z}}$ と補間型スケーリング関数 $\varphi(t)$ を用い、関数 $f^{(0)}(s, t)$ を

$$f^{(0)}(s, t) := \sum_{i,j \in \mathbb{Z}} x_{ij} \varphi_{0,i}(s) \varphi_{0,j}(t) \quad (5)$$

で定義する。ここで $\varphi(t)$ は補間型であるため、

$$f^{(0)}(i, j) = x_{ij}, \quad \forall i, j \in \mathbb{Z} \quad (6)$$

を満たす。したがって関数 $f^{(0)}(s, t)$ は、与えられた頂点座標集合 $\{P_{ij}\}_{i,j \in \mathbb{Z}}$ の x 座標 $\{x_{ij}\}_{i,j \in \mathbb{Z}}$ を補間する関数と見なすことができる。式 (5) に示したように、補間型スケーリング関数を基底として関数表現するとき、与えられた座標集合 $\{x_{ij}\}_{i,j \in \mathbb{Z}}$ をそのまま係数として用いることができる。仮定 1 でスケーリング関数として補間型を仮定したのはこのためである。

一方、仮定 1 ではさらに台がコンパクトであることを仮定しているが、これは 3 章で述べるツースケール関係を表す係数列が有限個の要素で表現できるための必要条件となる。

以下、与えられた座標を初期レベル $l = 0$ での頂点座標値、すなわち

$$x_{ij}^{(0)} := x_{ij} \quad (7)$$

と定義し、これより低いレベル $l = -1, -2, \dots$ での頂点座標値 $x_{ij}^{(l)}$ の表現を考える。レベル $l (\leq -1)$ でのスケーリング関数 $\{\varphi_{l,k}(t) \mid k \in \mathbb{Z}\}$ を用いて関数 $f^{(0)}(s, t)$ を展開した関数を

$$f^{(l)}(s, t) := \sum_{i,j \in \mathbb{Z}} x_{ij}^{(l)} \varphi_{l,i}(s) \varphi_{l,j}(t) \quad (8)$$

で表す。ここで係数 $x_{ij}^{(l)}$ は

$$x_{ij}^{(l)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f^{(0)}(s, t) \tilde{\varphi}_{l,i}(s) \tilde{\varphi}_{l,j}(t) ds dt \quad (9)$$

で与えられる。レベル $l \leq -1$ に対して式 (8) で表される関数 $f^{(l)}$ は、レベル 0 の関数 $f^{(0)}$ の多重解像度解析空間内での表現となる。

次に、レベル $l (\leq -1)$ における頂点座標の定義について述べる。関数 $\varphi(t)$ は式 (4) で定義される補間型スケーリング関数であるため、レベル l におけるスケーリング関数 $\varphi_{l,k}(t)$ は次の性質を満たす。

$$\varphi_{l,k}(i) = \begin{cases} 1, & i = 2^{-l}k, \\ 0, & \text{otherwise,} \end{cases} \quad \forall k \in \mathbb{Z}. \quad (10)$$

これと式 (8) より

$$f^{(l)}(2^{-l}i, 2^{-l}j) = x_{ij}^{(l)}, \quad \forall i, j \in \mathbb{Z} \quad (11)$$

となる。したがって (s, t) が間隔 2^{-l} の格子点 $(2^{-l}i, 2^{-l}j)$ 上にあるとき、多重解像度解析で表現された関数 $f^{(l)}(s, t)$ の値は、スケーリング関数で展開した係数、すなわち式 (9) で与えられる $x_{ij}^{(l)}$ そのものである。

この性質を用い、与えられた頂点座標集合 $\{P_{ij}\}_{i,j \in \mathbb{Z}}$ の x 座標の多重解像度表現を次のように定義する。

定義 3 補間型スケーリング関数 $\varphi(t)$ と集合 $\{x_{ij}\}_{i,j \in \mathbb{Z}}$ に対し、式 (9) で定義される集合

$$\{x_{ij}^{(l)}\}_{i,j \in \mathbb{Z}}, \quad l = 0, -1, -2, \dots$$

を $\{x_{ij}\}_{i,j \in \mathbb{Z}}$ のレベル l での多重解像度表現と呼ぶ。

y 座標、 z 座標についても定義 3 と同様に表現することにより、レベル l での頂点座標を

$$P_{ij}^{(l)} := (x_{ij}^{(l)}, y_{ij}^{(l)}, z_{ij}^{(l)})$$

として定義する。

式 (11) で示したように、レベル $l (\leq -1)$ での頂点座標の格子間隔は、レベル $l+1$ での格子間隔の 2 倍

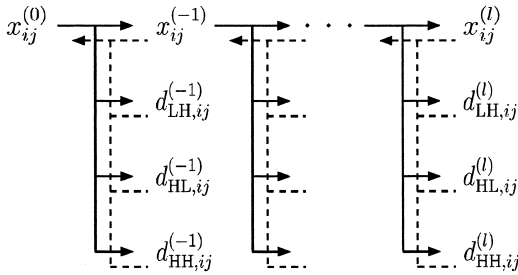


図 1 多重解像度解析 (実線: 順変換, 破線: 逆変換)
Fig. 1 Multiresolution analysis (solid line: forward transform, dashed line: inverse transform).

となる。

2.3 ツースケール関係と多重解像度表現

集合 $\{x_{ij}\}_{i,j \in \mathbb{Z}}$ の多重解像度表現 (定義 3) は式 (9) で与えられるが、これは双直交関数の組 $\{\varphi, \tilde{\varphi}, \psi, \tilde{\psi}\}$ のツースケール関係を用いて計算することができる。双直交関数の組 $\{\varphi, \tilde{\varphi}, \psi, \tilde{\psi}\}$ に対応するツースケール関係の係数列の組を $\{p_i, \tilde{p}_i, q_i, \tilde{q}_i\}_{i \in \mathbb{Z}}$ とすると、 $\{\tilde{p}_i, \tilde{q}_i\}$ により多重解像度表現の順変換が計算可能であり、一方、 $\{p_i, q_i\}$ によりその逆変換が計算可能となる¹⁶⁾。これを図 1 にまとめる。ここで、 $d_{LH,i,j}^{(l)}, d_{HL,i,j}^{(l)}, d_{HH,i,j}^{(l)}$ はツースケール関係により得られるウェーブレット係数であり、復元 (逆変換) のために必要な情報である。以下、このウェーブレット係数を詳細情報と呼ぶ。したがってウェーブレット多重解像度解析の計算にはツースケール関係の係数列 $\{p_i, \tilde{p}_i, q_i, \tilde{q}_i\}_{i \in \mathbb{Z}}$ を決定する必要があるが、これについては 3.2 節で示す。

また、変換後は変換前のデータを保存しておく必要がないため、図 2 に示すように変換前のデータに上書きすることが可能である。したがって変換後のデータ量は復元のためのウェーブレット係数も含めて一定である。

3. リフティングスキームによる多重解像度表現

3.1 リフティングスキームによる多重解像度解析

前章で述べたように、多重解像度解析の順変換と逆変換はツースケール列 $\{p_i, \tilde{p}_i, q_i, \tilde{q}_i\}_{i \in \mathbb{Z}}$ により計算することができる。ここではツースケール列による多重解像度解析の代わりに 1 次元リフティングスキーム^{5),20),21)} により順変換と逆変換を行う手法についてまとめる。

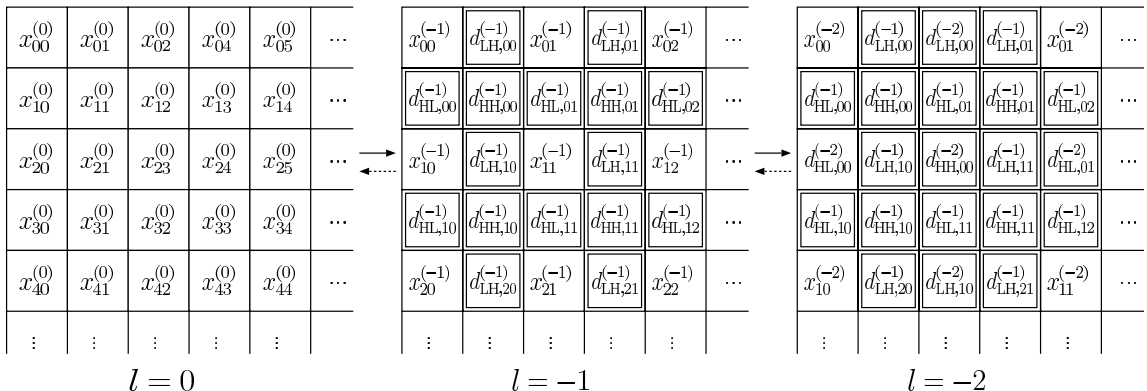


図 2 それぞれのレベルでの座標 (スケーリング係数) と詳細情報 (ウェーブレット係数) の配列
Fig. 2 Array of coordinates (scaling coefficients) and detail coefficients (wavelet coefficients) at each level.

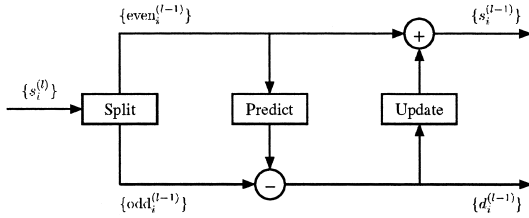


図 3 リフティングスキーム：順変換
Fig. 3 The lifting scheme: forward transform.

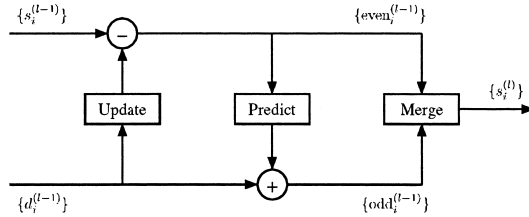


図 4 リフティングスキーム：逆変換
Fig. 4 The lifting scheme: inverse transform.

まず、1次元の離散信号 $\{s_i^{(0)}\}_{i \in \mathbb{Z}}$ に対する多重解像度解析を考える。レベル l での離散信号 $\{s_i^{(l)}\}_{i \in \mathbb{Z}}$ からレベル $l-1$ での離散信号 $\{s_i^{(l-1)}\}_{i \in \mathbb{Z}}$ とウェーブレット係数 $\{d_i^{(l-1)}\}_{i \in \mathbb{Z}}$ を求める順変換は、図 3 により構築される。ウェーブレット係数 $\{d_i^{(l-1)}\}_{i \in \mathbb{Z}}$ は原信号を復元するために必要な情報であり、以下、これを詳細信号と呼ぶ。また、Split は信号 $\{s_i^{(l)}\}_{i \in \mathbb{Z}}$ を偶数番目 $\{\text{even}_i^{(l-1)}\}_{i \in \mathbb{Z}}$ と奇数番目 $\{\text{odd}_i^{(l-1)}\}_{i \in \mathbb{Z}}$ の信号に分割する演算子であり、Predict は偶数番目の信号から奇数番目の信号を予測する演算子である。この予測値と奇数番目の信号 $\{\text{odd}_i^{(l-1)}\}_{i \in \mathbb{Z}}$ との差が詳細信号 $\{d_i^{(l-1)}\}_{i \in \mathbb{Z}}$ となる。Update はこの詳細信号を利用し、偶数番目の信号からレベル $l-1$ の信号 $\{s_i^{(l-1)}\}_{i \in \mathbb{Z}}$ を構成する演算子である。

またこのとき、逆変換は図 4 のように構成される^{20),21)}。ここで Merge は奇数番目の信号と偶数番目の信号を結合する演算子、すなわち

$$s_i^{(l)} = \text{Merge}(\{\text{even}_i^{(l-1)}\}_{i \in \mathbb{Z}}, \{\text{odd}_i^{(l-1)}\}_{i \in \mathbb{Z}}) = \begin{cases} \text{even}_{i/2}^{(l-1)}, & i : \text{even}, \\ \text{odd}_{(i-1)/2}^{(l-1)}, & i : \text{odd} \end{cases} \quad (12)$$

を表す。

スケーリング関数とウェーブレットのツースケール関係を表す係数 $\{p_i\}_{i \in \mathbb{Z}}$ と $\{q_i\}_{i \in \mathbb{Z}}$ は、リフティングスキームの Predict, Update それぞれの演算子により決定される。

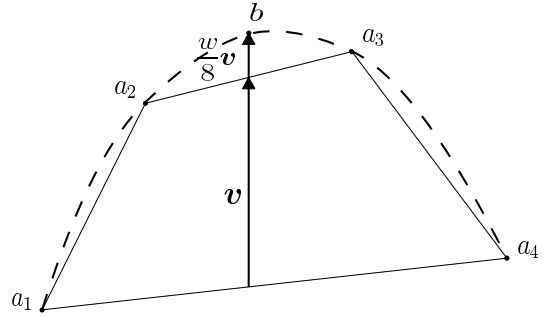


図 5 4-ポイントスキームによる補間
Fig. 5 Interpolation by 4-point scheme.

3.2 細分割法によるリフティングスキームの構築
本節では、図 3 および図 4 で示したリフティングスキームに対する Predict 演算子と Update 演算子の決定法について述べる。

ここではリフティングスキームの Predict 演算子を、細分割法^{3),10)}の1つである4-ポイントスキーム³⁾を用いて決定するものとする。4-ポイントスキームは隣接する4つの離散点の重み付き和により補間点を繰り返し求め、その極限として補間曲線を得る手法である。ここで離散信号の4点 a_1, a_2, a_3, a_4 からこの中心の点 b を4-ポイントスキームにより補間するとき、パラメータ w により定義される重み

$$\left[-\frac{w}{16}, \frac{8+w}{16}, \frac{8+w}{16}, -\frac{w}{16} \right]$$

により

$$b = -\frac{w}{16}a_1 + \frac{8+w}{16}a_2 + \frac{8+w}{16}a_3 - \frac{w}{16}a_4$$

で与えられる。この補間点は、図 5 に示すように a_1, a_4 の中点と a_2, a_3 の中点を結ぶベクトルを v としたとき、 a_2, a_3 の中点に $\frac{w}{8}v$ を加えた位置を表している。 $w = 0$ の場合は隣接する2点の線形補間となり、また $0 < w < 2(\sqrt{5}-1)$ の範囲で滑らかな補間曲面が得られることが知られている^{3),10)}。

特にパラメータ $w = 1$ の場合は4点を補間する3次多項式による補間に相当する¹⁰⁾。以下、 $w = 1$ での4-ポイントスキームを Predict 演算子として採用することとする。

このとき、奇数番目の信号 $\text{odd}_i^{(l-1)}$ の予測値は、偶数番目の4つの信号

$$e_i^{(l-1)} = \left[\text{even}_{i-1}^{(l-1)}, \text{even}_i^{(l-1)}, \text{even}_{i+1}^{(l-1)}, \text{even}_{i+2}^{(l-1)} \right] \quad (13)$$

と4-ポイントスキームの重み

$$\mathbf{h} := \left[-\frac{1}{16}, \frac{9}{16}, \frac{9}{16}, -\frac{1}{16} \right] \quad (14)$$

を用いて,

$$\text{odd}_i^{(l-1)} \approx (\mathbf{h}, \mathbf{e}_i^{(l-1)})$$

で与えられる. ここで記号 “ (\cdot, \cdot) ” は内積を, “ \approx ” は左辺を右辺で予測することを表す. すなわち, リフティングスキーム (図 3) の Predict 演算子は,

$$\text{Predict}(\{\text{even}_i^{(l-1)}\}_{i \in \mathbb{Z}}) = (\mathbf{h}, \mathbf{e}_i^{(l-1)}) \quad (15)$$

である.

したがって式 (15) および図 3 より, 詳細信号は

$$d_i^{(l-1)} = \text{odd}_i^{(l-1)} - (\mathbf{h}, \mathbf{e}_i^{(l-1)}), \quad i \in \mathbb{Z} \quad (16)$$

となる.

またこのとき, 共役ウェーブレット $\tilde{\psi}(t)$ のパニングメント N は 4 となる²⁰⁾.

一方 Update 演算子は, ここではウェーブレット $\psi(t)$ のパニングメント \tilde{N} が共役ウェーブレットのパニングメントと同様 4 となる条件により設計するものとする. この条件を満たすため,

$$\mathbf{d}_i^{(l-1)} := \left[d_{i-2}^{(l-1)}, d_{i-1}^{(l-1)}, d_i^{(l-1)}, d_{i+1}^{(l-1)} \right] \quad (17)$$

と重みベクトル

$$\mathbf{g} := \left[-\frac{1}{32}, \frac{9}{32}, \frac{9}{32}, -\frac{1}{32} \right] \quad (18)$$

を用いて, Update 演算子を以下のように定義する.

$$\text{Update}(\{\mathbf{d}_i^{(l-1)}\}_{i \in \mathbb{Z}}) := (\mathbf{g}, \mathbf{d}_i^{(l-1)}). \quad (19)$$

このとき $\tilde{N} = 4$ であり²⁰⁾, これにより, レベル $l-1$ の信号は次のように計算される.

$$s_i^{(l-1)} = \text{even}_i^{(l)} + (\mathbf{g}, \mathbf{d}_i^{(l-1)}), \quad i \in \mathbb{Z}. \quad (20)$$

以上, Predict 演算子と Update 演算子をそれぞれ式 (14), (18) で定義した重みベクトル \mathbf{h}, \mathbf{g} により決定した. この Predict 演算子と Update 演算子によるリフティングスキームの順変換 (図 3) をレベル l での頂点座標集合 $\{x_{ij}^{(l)}\}$ のインデックス i, j それぞれに対して順に適用することにより, レベル $l-1$ での頂点座標集合 $\{x_{ij}^{(l-1)}\}$ と詳細情報 $\{d_{\text{LH},ij}^{(l-1)}\}, \{d_{\text{HL},ij}^{(l-1)}\}, \{d_{\text{HH},ij}^{(l-1)}\}$ が得られる.

ここではリフティングスキームを細分割法の 4-ポイントスキームを用いて定義したが, 以下, このリフティングスキームによる解像度変化をリフティングサブディビジョンと呼ぶ.

ここで, 式 (16), (20) で定義された Predict, Up-

表 1 ツースケール関係の係数列

Table 1 Two-scale sequences

i	p_i	\tilde{p}_i	q_i	\tilde{q}_i
-6		$-1/2^9$		
-5			$-1/2^9$	
-4		$18/2^9$		
-3	$-1/16$	$-16/2^9$	$18/2^9$	
-2		$-63/2^9$	$16/2^9$	$1/16$
-1	$9/16$	$144/2^9$	$-63/2^9$	
0	1	$348/2^9$	$-144/2^9$	$-9/16$
1	$9/16$	$144/2^9$	$348/2^9$	1
2		$-63/2^9$	$-144/2^9$	$-9/16$
3	$-1/16$	$-16/2^9$	$-63/2^9$	
4		$18/2^9$	$16/2^9$	$1/16$
5			$18/2^9$	
6		$-1/2^9$		
7			$-1/2^9$	

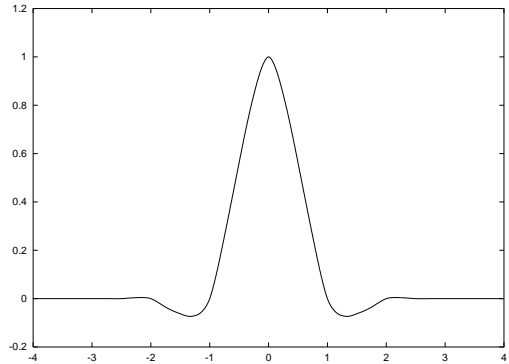


図 6 スケーリング関数

Fig. 6 Scaling function.

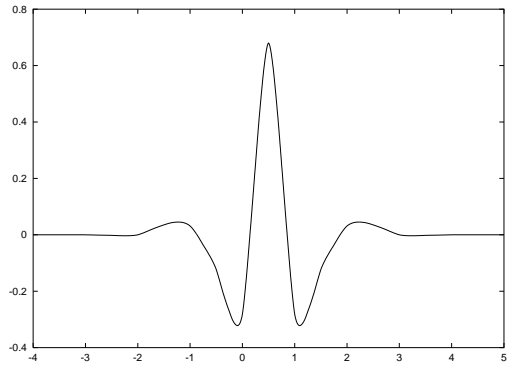


図 7 ウェーブレット

Fig. 7 Wavelet.

date 演算子によるリフティングスキームを用いた場合の, 双直交関数 $\{\varphi, \tilde{\varphi}, \psi, \tilde{\psi}\}$ のツースケール関係について示す. ツースケール関係の係数列 $p_i, \tilde{p}_i, q_i, \tilde{q}_i$ の値を表 1 に示す. ただし, 表に表記されていない値はすべて 0 である. またこのツースケール列により決定されるスケーリング関数 $\varphi(t)$ およびウェー

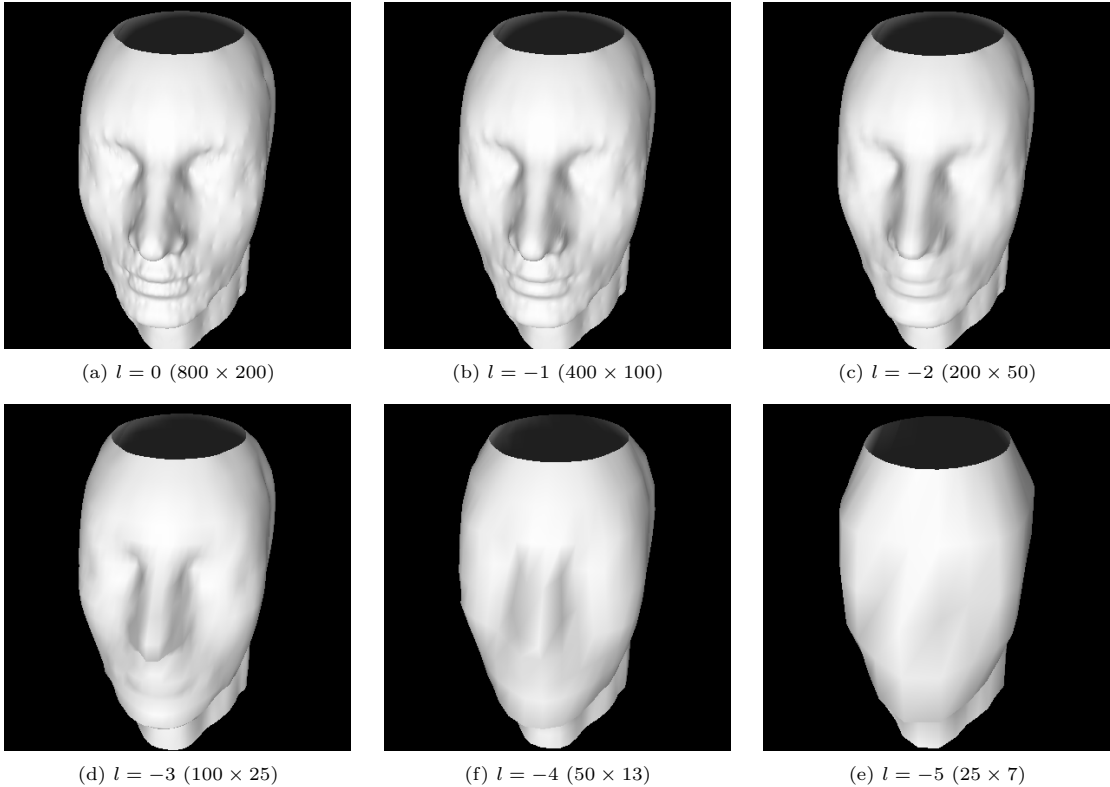


図 8 多重解像度表現 (Example 1)

Fig. 8 Multiresolution representations (Example 1).

ウェーブレット $\psi(t)$ を図 6, 図 7 にそれぞれ示す. したがってツースケール関係の係数列 (表 1) と高速ウェーブレット変換を用いれば, リフティングスキームと同様, レベル $l = -1, -2, \dots$ についての多重解像度表現 $\{x_{ij}^{(l)}\}_{i,j \in \mathbb{Z}}$ が得られ, さらに逆変換により再構成することができる.

以上では式 (1) に示したようにインデックス i, j は整数全体をとるものとしていたが, 実際には与えられるメッシュの頂点数は $n_1 \times n_2$ の有限個である. したがってリフティングスキームによる多重解像度解析を行う場合, メッシュの境界付近の演算では定義されたメッシュの外側の値が必要となる. このような問題を回避するため, 定義された領域の外側について値を拡張する周期拡張または対称拡張¹⁹⁾を行うことにより, メッシュの境界に対してもインデックス i, j が整数全体をとる場合と同様に多重解像度表現を行うことができる.

3.3 リフティングサブディビジョンの演算量

本節では, ツースケール関係による多重解像度解析とリフティングスキームによる多重解像度解析の計算

量について比較する. まず, レベル l での頂点座標値 $x_{ij}^{(l)}$ から, レベル $l-1$ での頂点座標値 $x_{ij}^{(l-1)}$ と詳細情報 $d_{LH,ij}^{(l-1)}, d_{HL,ij}^{(l-1)}, d_{HH,ij}^{(l-1)}$ を得るために要する演算量について考察する. ただし, レベル $l-1$ で求める頂点と詳細情報の数は $n_1 \times n_2$ とし, このとき, レベル l での頂点数は $2n_1 \times 2n_2$ である.

まず, ツースケール関係を用いた場合, 表 1 よりスケーリング係数を求めるためのツースケール列 $\{\tilde{p}_i\}$ の非零要素数は 11 であり, またウェーブレット係数を求めるための係数列 $\{\tilde{q}_i\}$ の非零要素数は 5 である. したがって 1 次元の n_1 個のスケーリング係数とウェーブレット係数を求めるのに要する積和演算数はそれぞれ $11n_1$ と $5n_1$ であり, 2 次元の多重解像度解析では合計で $32n_1n_2$ 回の積和演算となる.

一方, リフティングスキームを用いた場合, 式 (14), (18) より, 1 次元の n_1 個のスケーリング係数とウェーブレット係数を求めるのに要する積和演算数はともに $4n_1$ であり, これより 2 次元多重解像度解析では合計で $16n_1n_2$ となる.

したがってここで用いる双直交ウェーブレットでは,

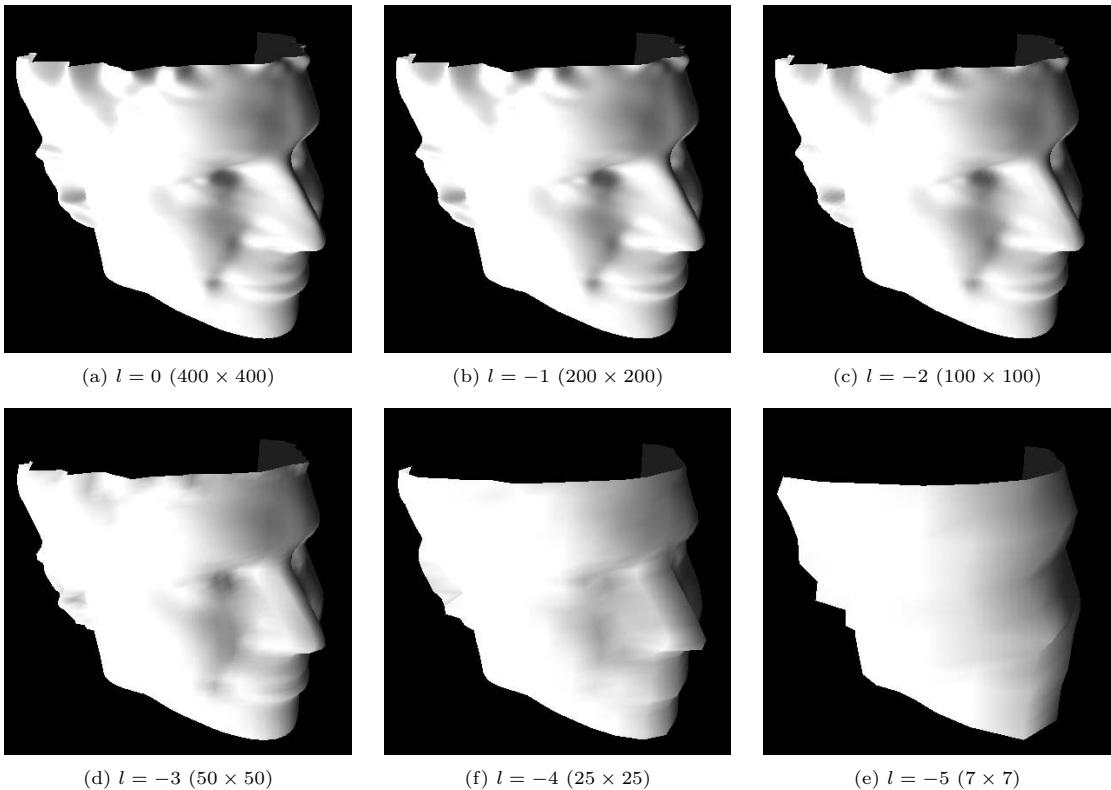


図 9 多重解像度表現 (Example 2)
Fig. 9 Multiresolution representations (Example 2).

リフティングスキームによる多重解像度解析は従来のツースケール関係を用いた場合と比べて半分の演算量で計算可能となる。また、逆変換の演算量はいずれも順変換と同じであり、リフティングサブディビジョンによる復元の演算量はツースケール関係による多重解像度表現の半分である。

4. 多重解像度表現の例

4.1 3次元形状モデルと多重解像度表現

本節ではリフティングスキームを四辺形メッシュで表された3次元形状モデルに適用し、多重解像度表現を行った例を示す。ここで用いた2つの形状モデルを図8(a), 図9(a)に示す。以下この2つの形状モデルをそれぞれExample 1, Example 2として、形状モデルの構成について示す。

Example 1: 図8(a)はCTやMRIにより得られる複数の断面像から構成された3次元形状を想定し、陰関数表現された形状モデル¹²⁾から複数断面の輪郭線上の点を抽出、さらに各断面の点を格子状に接続することにより四辺形メッシュを得たものである。ここでは頭部の表面形状を

$$F(x, y, z) = 0 \quad (21)$$

を満たす (x, y, z) の集合として表現したモデル¹²⁾を利用し、断面の数は200、それぞれの断面での輪郭線上の点を800とした。また、インデックスは断面の輪郭線方向を i とし、断面方向を j として表している。すなわちここでの四辺形メッシュの頂点数は 800×200 である。

このメッシュを与えられた初期メッシュ、すなわちレベル $l = 0$ での頂点座標集合(1)とし、レベル $l = -1, \dots, -5$ に対する簡略化された形状を求めている。これらの簡略化メッシュをレンダリングした結果とその頂点数を図8(b)~(f)に示す。ただし、このモデルは円筒状の形状であるため、メッシュの境界に対する拡張方式としてインデックス i については周期拡張、インデックス j については対称拡張を用いた。

Example 2: ここでは3次元レーザスキャナにより取得された形状モデルを想定する。図9(a)はTGS社Open Inventor²²⁾付属の四辺形メッシュ形状モデルをスプライン補間し、 400×400 の頂点座標を得たものである。

表 2 計算時間 [msec]
Table 2 CPU-time [msec].

Example 1 (Lifting Scheme)						
l	$0 \rightleftharpoons -1$	$-1 \rightleftharpoons -2$	$-2 \rightleftharpoons -3$	$-3 \rightleftharpoons -4$	$-4 \rightleftharpoons -5$	
Simplification	71.6	18.2	4.0	1.1	0.3	
Reconstruction	71.2	18.1	4.8	1.1	0.3	
Example 1 (Two-Scale Sequence)						
l	$0 \rightleftharpoons -1$	$-1 \rightleftharpoons -2$	$-2 \rightleftharpoons -3$	$-3 \rightleftharpoons -4$	$-4 \rightleftharpoons -5$	
Simplification	141.9	36.4	8.8	2.3	0.6	
Reconstruction	123.1	31.9	7.9	2.2	0.3	
Example 2 (Lifting Scheme)						
l	$0 \rightleftharpoons -1$	$-1 \rightleftharpoons -2$	$-2 \rightleftharpoons -3$	$-3 \rightleftharpoons -4$	$-4 \rightleftharpoons -5$	
Simplification	73.4	19.5	5.0	1.1	0.3	
Reconstruction	73.0	19.2	5.1	1.3	0.3	
Example 2 (Two-Scale Sequence)						
l	$0 \rightleftharpoons -1$	$-1 \rightleftharpoons -2$	$-2 \rightleftharpoons -3$	$-3 \rightleftharpoons -4$	$-4 \rightleftharpoons -5$	
Simplification	141.5	36.5	9.4	2.3	0.6	
Reconstruction	122.4	32.1	8.6	2.2	0.5	

この四辺形メッシュモデルに対し, Example 1. と同様, レベル $l = -1, \dots, -5$ に対する簡略化された形状を図 9 (b) ~ (f) に示す. ただし, このモデルはインデックス i, j ともに境界を持つモデルとなっているため, 境界に対する拡張方式はいずれも対称拡張を用いた.

次に, 多重解像度表現における解像度変化に要する計算時間について示す. 本論文で示したウェーブレット多重解像度解析のアルゴリズムはリフティングスキームによる方法 (図 3, 図 4, (15), (19)) とツースケール関係による方法 (表 1) があるが, これらは 3.3 節で示したように演算量が異なる. この演算量を比較するため, Example 1, Example 2 のそれぞれについての解像度変化に要する計算時間を表 2 に示す. ただし, ここで利用した計算機の環境は Pentium4 1.6 GHz, 512 MB RAM, Linux である.

4.2 多重解像度表現の結果と考察

図 8 に示した多重解像度表現の例について, メモリと演算量についての考察を行う. 図 8 (a) および図 9 (a) で示した $l = 0$ での形状は x, y, z 座標がそれぞれ $800 \times 200, 400 \times 400$ の配列で表されている. さらに $l = -1, -2, \dots$ についても逆変換のためのウェーブレット係数も含めてデータ量は一定であり, また $l = 0$ での形状を表す配列に上書きできるため, ウェーブレット変換の計算や逆変換のための情報として新たな記憶領域を用いる必要がない. すなわち, オリジナルの形状 (a) を表現する配列のみで解像度変化が可能であることが確認される.

次に解像度変化の計算量について考察する. まずリフティングスキームを用いた場合, メッシュの簡略化と復元に必要な計算量は同じであり, ここで用いた 16

万頂点からなるメッシュデータに対しても 0.1 秒以内での高速な解像度変化が可能となることが確認された. また, 計算時間は 3.3 節で示したようにほぼ頂点数に比例していることが確認された. さらにツースケール関係を用いた解像度変化と比較した場合, リフティングスキームを用いた場合の計算時間はおよそ 1/2 であり, これも 3.3 節で示した演算量に則している. 以上より, リフティングスキームによる多重解像度表現の計算量での有効性が確認された.

また, ここではリフティングスキームの Predict 演算子 (15) として用いた 4-ポイントスキームのパラメータを $w = 1$ としたが, w を変化させることにより異なる簡略化メッシュが得られる. たとえば $w = 0$ としたとき, 共役ウェーブレットのパニシングモーメントは 2 となるため²⁰⁾, $w = 1$ の場合 ($N = 4$) と比べ低周波の成分をウェーブレット係数に含むことになる. したがって, 形状にもよるが, 簡略化モデルを表すスケーリング係数は元の形状をよく近似しないということが考えられる. さらに, パニシングモーメントの値を増加させるために隣接する 6 点を補間する 5 次多項式や 8 点を補間する 7 次多項式による細分割を Predict 演算子として用いることも考えられる. ただし形状によっては高次多項式での補間が高精度とは限らず, また補間点の増加により演算量も増えるという問題がある.

以上より, 本論文で用いた 4-ポイントスキームでのパラメータ ($w = 1$) はパニシングモーメントの観点では適切な値と考えられ, また 4 点を用いた Predict 演算子によるリフティングスキームは比較的大規模なメッシュに対しても高速な演算が可能となる.

5. おわりに

本論文では四辺形メッシュからなる3次元形状モデルに対する多重解像度表現を、補間型スケーリング関数によるウェーブレット多重解像度解析のスケーリング係数で定義し、その多重解像度解析手法としてリフティングスキームと細分割法の組合せであるリフティングサブディビジョンを提案した。

リフティングスキームはウェーブレット変換におけるスケーリング関数とウェーブレットおよびそれらの共役を2つの演算子(Predict, Update)により構成する手法であるが、ここではそのPredict演算子として細分割法の1つである4ポイントスキームを適用し、ウェーブレット変換による3次元形状モデルの多重解像度表現手法を構成した。

またここではウェーブレット係数を4ポイントスキームにより決定する手法を用いた。これにより補間型のスケーリング関数と、バニシングモーメントが4となる双直交ウェーブレットの構築が可能となる。

本手法の特徴はリフティングスキームによるウェーブレット多重解像度解析の高速化とメモリ効率にある。すなわち、本手法は同様の双直交ウェーブレットを表すツースケール関係により多重解像度解析を行う場合と比べ半分の演算量であり、高速な解像度変化を可能とする。さらに変換後のスケーリング係数またはウェーブレット係数は変化前の係数の配列へ上書きすることが可能であり、したがって解像度変化にともなう頂点座標値とウェーブレット係数を初期メッシュを表す配列に格納することにより新たな記憶領域を用意する必要がない。

参 考 文 献

- 1) 芦野隆一, 山本鎮男: ウェーブレット解析, 共立出版 (1997).
- 2) 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).
- 3) Dyn, N., Gregory, J.A. and Levin, D.: A 4-Point Interpolatory Subdivision Scheme for Curve Design, *Computer Aided Geometric Design*, Vol.4, pp.257-268 (1987).
- 4) Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M. and Stuetzle, W.: Multiresolution Analysis of Arbitrary Meshes, *Computer Graphics*, Vol.29, pp.173-182 (1995).
- 5) Fernández, G., Periaswamy, S. and Sweldens, W.: LIFTPACK: A software package for

- wavelet transforms using lifting, *Wavelet Applications in Signal and Image Processing IV, Proc. SPIE 2825*, Unser, M., Aldroubi, A. and Laine, A.F.(Eds.), pp.396-408 (1996).
- 6) Finkelstein, A. and Salesin, D.H.: Multiresolution Curves, *Computer Graphics*, Vol.28, pp.261-268 (1994).
- 7) Guskov, I., Vidimce, K., Sweldens, W. and Schröder, P.: Normal Meshes, *Proc. Computer Graphics, SIGGRAPH 2000*, ACM SIGGRAPH, pp.95-102 (2000).
- 8) Hoppe, H.: Progressive Meshes, *Proc. Computer Graphics, SIGGRAPH '96*, ACM SIGGRAPH, pp.99-108 (1996).
- 9) Hoppe, H.: New Quadric Metric for Simplifying Meshes with Appearance Attributes, *IEEE Visualization*, pp.59-66 (1999).
- 10) Kobbelt, L.: Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology, *Computer Graphics Forum 15, Eurographics '96 Issue*, pp.409-420 (1996).
- 11) Kobbelt, L. and Schröder, P.: A multiresolution framework for variational subdivision, *ACM Trans. Graphics*, Vol.17, No.4, pp.209-237 (1998).
- 12) Kojekine, N., Savchenko, V., Berzin, D. and Hagiwara, I.: Software Tools for Compactly Supported Radial Basis Functions, *Computer Graphics and Imaging, Proc. IASTED CGIM 2001*, pp.234-239 (2001).
- 13) Lee, A.W.F., Sweldens, W., Schröder, P., Cowsar, L. and Dobkin, D.: MAPS: Multiresolution Adaptive Parameterization of Surfaces, *Computer Graphics*, Vol.32, pp.95-104 (1998).
- 14) Lounsbery, M.: Multiresolution Analysis for Surfaces of Arbitrary Topological Type, Ph.D. Thesis, University of Washington (1994).
- 15) 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).
- 16) 新島耕一: ウェーブレット画像解析, 科学技術出版 (2000).
- 17) Schröder, P. and Sweldens, W.: Spherical wavelets: Efficiently representing functions on the sphere, *Computer Graphics*, Vol.29, pp.161-172 (1995).
- 18) Stollnitz, E.J., DeRose, T.D. and Salesin, D.H.: Wavelets for Computer Graphics: A Primer, Part 2, *IEEE Computer Graphics and Applications*, Vol.15, No.4, pp.75-85 (1995).
- 19) Strang, G. and Nguyen, T.: *Wavelets and Filter Banks*, Wellesley-Cambridge Press (1996).
- 20) Sweldens, W.: The Lifting Scheme: A Custom-

Design Construction of Biorthogonal Wavelets, *J. Appl. Comput. Harmonic Analysis*, Vol.3, pp.186–199 (1996).

21) Sweldens, W.: The Lifting Scheme: A Construction of Second Generation Wavelets, *SIAM J. Math. Appl.*, Vol.29, No.2, pp.511–546 (1998).

22) Wernecke, J.: *The Inventor Mentor*, Addison Wesley, Reading, Massachusetts, USA (1996).

(平成 13 年 12 月 20 日受付)

(平成 15 年 2 月 4 日採録)



仲田 晋 (正会員)

平成 10 年筑波大学大学院理工学研究科修士課程修了, 平成 13 年同大学大学院工学研究科博士課程修了, 博士(工学). 同年東京工業大学大学院理工学研究科非常勤研究員, 現在

に至る. 専門分野, 数値解析. 日本応用数理学协会会员.



萩原 一郎 (正会員)

昭和 47 年京都大学大学院工学研究科数理工学専攻修士課程修了. 同年日産自動車入社. 総合研究所, 技術システム勤務. 平成 8 年東京工業大学工学部機械科学科教授, 平成 12

年東京工業大学理工学研究科機械物理工学専攻教授, 現在に至る. 工学博士. 専門分野, 機械工学・数値解析. 日本機械学会, 米国自動車技術会, 日本自動車技術会, 日本応用数理学会各論文賞等. 中国哈爾濱工業大学顧問教授, 上海交通大学名誉教授, 清華大学安全重点研究所外国人評議委員, 日本機械学会フェロー, 亜州智能建築学会名誉会員. 日本音響学会, 自動車技術会各会員, 日本応用数理学会理事, 日本機械学会, 日本計算工学会各評議員等.