

ラティス構造に基づく軽量で高品質な Web3D データ表現

脇田 玲[†] 矢島 誠^{††} 原田 毅 士^{††}
鳥谷 浩 志^{††} 千代倉 弘明[†]

VRML に代表されるポリゴンメッシュに基づく Web3D データは、データサイズが膨大になり多くの転送時間を要する。また、形状の表現能力が低いために使用用途が大きく限定されてしまう。これらの問題を解決するために、本論文では軽量で高品質な 3D データの転送を可能にするラティス構造に基づく形状表現を提案する。ラティス構造では自由曲面データ扱うことができるため、少ないデータサイズで形状を転送することができる。また、受信側では高品質な曲面形状をレンダリングすることが可能である。自由曲面データは、頂点間の接続関係が自身と同一である単純なポリゴンとその付加情報のセットとしてデータを管理するラティス構造に変換される。このデータ表現はポリゴンとして扱えるために VRML などの既存のデータフォーマットに対して容易に自由曲面情報を付加できる。さらに、ラティス構造はポリゴンモデルの操作を通して、直感的に曲面モデルを変形することができるため、モデリング手法としても有用である。

A Compact and Qualified Web3D Representation Based on Lattice Structure

AKIRA WAKITA,[†] MAKOTO YAJIMA,^{††} TSUYOSHI HARADA,^{††}
HIROSHI TORIYA^{††} and HIROAKI CHIYOKURA[†]

The data size of Web3D representation based on polygon meshes is so large that transferring practical data fast is a hard problem. This paper proposes lattice structure, a new framework for compact 3D representation with high quality surface shape. By utilizing a free-form surface technique, qualified surfaces are transferred with limited amount of data size and rendered. Lattice structure can be regarded as both polygon meshes and free-form surfaces. Therefore, it can be easily integrated to existing Web3D data formats, for example VRML. Lattice structure also enables modeling free-form surface shapes intuitively with polygon modeling-like operations.

1. はじめに

インターネット上での 3D データ活用（以後、Web3D）への需要が高まっている。VRML (Virtual Reality Modeling Language) は ISO の国際標準として認証された Web3D フォーマットである^{*}。しかし、形状表現にポリゴンメッシュを採用しているため、高品質な形状は膨大なデータサイズになり、データ転送にも多くの時間を要する。また、データを受信する計算機の処理能力が低い場合は、インタラクティブなデータ処理が困難である。さらに、NURBS に代表される自由曲面データではないために、CAD/CAM を

利用する製造業などでの利用には向いていない。安価な CG システムやミッドレンジ CAD の普及により、様々な分野からの 3D データが溢れている一方で、これらのデータをインターネット上で効果的に利用する術はまだ存在しない。

このような現状の中で、ポリゴンメッシュの問題点を階層性の付加^{1),2)}により解決しようとする研究がさかんである。これらの手法により、インターネット上でポリゴンメッシュを効率的に扱うことが可能になった。しかし、既存の CAD/CG システムは自由曲面の加工を中心に作成されており、大規模なポリゴンメッシュを柔軟に扱えるものは稀である。そのため、データ転送後の再加工は困難であり用途が大きく限定されてしまう。

一方、細分割曲面 (Subdivision Surfaces)^{3)~7)} と

[†] 慶應義塾大学大学院政策・メディア研究科
Graduate School of Media and Governance, Keio University

^{††} ラティス・テクノロジー株式会社
Lattice Technology, Inc.

^{*} VRML97, ISO/IEC 14772-1:1997

呼ばれる形状表現が近年注目を集めている。この手法は、データサイズの小さい初期形状から複雑な曲面形状を構築することが可能であるため、Web3Dに適した形状表現である。また、頂点間の接続関係（以後、位相）が任意である形状に対して適用可能であるために汎用性が高い。一方で、最終的な生成形状を確認するためには複数のフィルタリングが必要である。また、既存のCAD/CGシステムが認識可能な自由曲面情報を保持していないという問題もある。

我々の研究目的はWeb3Dに適した軽量で高品質な3Dデータを作成することである。この目的を達成するためには、自由曲面データを扱えることが望ましい。そこで我々は、自由曲面データを効果的に扱うラティス構造という新しいフレームワークを提案する。ラティス構造はGregory パッチ⁸⁾によって構築される自由曲面データ、およびその自由曲面データと同一位相を保持する単純なポリゴンデータから構成される。このポリゴンデータ（以後、初期多角形）は自由曲面データを構築するための位相情報、頂点情報および重み付け情報を保持しており、高速に曲面形状を生成することが可能である。形状データは初期多角形として転送される。初期多角形はデータサイズが非常に小さいために、ポリゴンメッシュを転送する場合と比較すると、大幅な転送時間の短縮が実現される。転送先では初期多角形から曲面データを生成することにより、高品質なレンダリングが実現される。曲面データは分割数の調節により詳細度制御が可能であり、用途に応じた表示精度を得ることができる。また、転送先ではCAD/CAMシステムでのデータ再利用も可能である。一方、初期多角形の操作を通して自由曲面を直感的に操作することが可能であるため、変形手法としても有用である。

自由曲面データは丸め操作（Rounding Operations）^{10)~12)}によって生成される。従来の手法は操作の前後で形状の位相に変化が生じるため、生成形状の予測や操作後の変形が困難であった。本手法は、丸め操作適応前後で1対1に対応する位相を保持することが可能である。また、本手法は丸め形状から丸める前の形状を再構築することが可能である。さらに、形状の頂点と稜線に重み付けをすることにより、柔軟な曲面生成を実現した。

Web3Dに最適な形状表現を実現するためには、インターネット上で容易に利用できるデータフォーマットであることが求められる。我々は以前、ラティス構造をVRMLの言語仕様に基づいて記述する手法について提案した¹³⁾。その後、VRMLはXML

（eXtensible Markup Language）¹⁴⁾によって記述されるX3D（eXtensible 3D）としての仕様変更が進められている。そこで、我々はX3Dの言語仕様の一部としてラティス構造を再定義し、標準化委員会に提案した¹⁵⁾。本論文では、ラティス構造をXMLにより記述するためのDTD（Document Type Definition）についても紹介する。

本論文では、まず2章でWeb3Dに最適な形状表現について述べる。3章では、曲面モデルの生成手法について解説する。初期多角形の位相要素に重み付けを行うことにより、柔軟性のある曲面生成を行う手法と、曲面モデルから初期多角形を再構築する手法についても述べる。続く4章では、ラティス構造をVRMLデータに埋め込む方法およびXMLを用いたデータフォーマットについて紹介する。5章では応用例として、モデリング手法としての利用、CAD/CAMシステムとのデータ交換および詳細度制御手法としての有用性について述べる。6章で検証および考察を行う。最終章では、結論および今後の展望を述べる。

2. Web3Dのための形状表現

2.1 ポリゴンメッシュ

我々の研究目的はインターネット上での3Dデータ利用（Web3D）に最適な軽量で高品質な形状表現を作成することである。VRMLは形状表現にポリゴンメッシュを採用していた。ポリゴンメッシュは三角形や四角形の平面の集まりによって形状を構成する最も一般的な形状表現である。直感的かつ容易に形状を操作することが可能であり、レンダリングも高速であるため、多くのCGシステムで採用されている。一方、高品質な形状を表現するためには数万ものポリゴン数が必要になる。そのため、データサイズが膨大になり、データ転送時間の増大やハードディスク容量の圧迫など様々な問題を引き起こす。さらに、データを受信する計算機の処理能力が低い場合は、モデルを構成する膨大なポリゴンをインタラクティブに処理することができず、ユーザに大きなストレスを与えることになる。VRMLがWeb3Dデータフォーマットの標準として提案されながらも普及しなかった最も大きな原因はここにある。

このような現状の中で、ポリゴンメッシュに様々な属性を付加することで、これらの問題を解決しようとする研究がさかんである。Hoppe¹⁾は、エネルギー関数の最小化により複雑なメッシュ形状を単純化し、その単純化の過程を階層的に管理することによって、メッシュの精度を自由に調節する手法を提案した。こ

の手法により、高精度なポリゴンメッシュが転送された場合でも、受信側の計算機の処理能力に応じて形状の精度を調節し、インタラクティブに形状を表示することが可能になる。Abadjev ら²⁾は Hoppe の手法を拡張し、幾何情報に加えてテクスチャ情報も階層的に管理することで、3D データの効果的なストリーミングを実現するフレームワークを提案した。

これらの手法を用いることにより、インタラクティブな表示制御は可能であるが、データサイズを縮小させることはできない。ポリゴンメッシュのデータを削減するためにはポリゴン数を削減せざるをえず、その結果形状に歪みが生じ、使用用途が著しく限定されてしまう。Web3D の形状表現としてポリゴンメッシュを採用するのは実用的ではない。

2.2 細分割曲面

一方、細分割曲面 (Subdivision Surfaces) を用いることで問題解決を図るアプローチが近年注目を集めている。細分割曲面とは単純なポリゴン形状に、分割と重み付けの操作を繰り返し適用することで、滑らかな曲面形状を生成する手法のことである。Doo ら³⁾および Catmull ら⁴⁾によって基礎理論が構築され、Loop⁵⁾が三角形メッシュを対象とした手法を提案したことにより、広く用いられるようになった。Zorin ら⁶⁾は Loop の細分割手法を局所的に適用することを可能にした。これにより、精度が不要な部分は荒く分割することが可能になり、細分割後の最終形状でもデータサイズをコンパクトにすることが可能になった。Lee ら⁷⁾は複雑な表面を持つ形状を、Loop 細分割曲面の初期メッシュとスカラー値の集合に分解して保存することにより大幅にデータを削減する手法を提案した。細分割曲面は、任意の位相を持つポリゴン形状に適用することが可能であるため汎用性が高い。また、データサイズの小さい初期形状から複雑な曲面形状を構築することが可能であるため、インターネット利用に適した形状表現である。一方で、最終的な生成形状を確認するためには複数のフィルタリングが必要である。また、既存の CAD/CG システムが認識可能な NURBS などの曲面制御点情報を保持していないという問題がある。

2.3 ラティス構造

以上の考察より、既存の形状表現を Web3D にそのまま利用するのは適策ではない。Web3D に特化した形状表現を作り上げるには以下の条件を満たす必要がある。

- 軽量かつ高品質であること
- データの用途を限定しないこと

これらの条件を満たす最も効果的な解決策は、自由

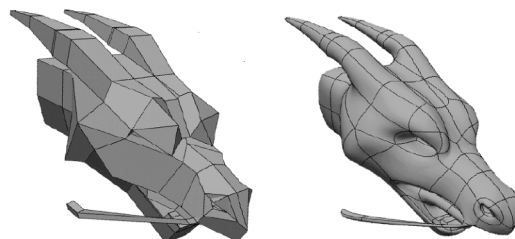


図 1 ラティス構造. 左が初期多角形で右がラティス曲面. 同一の位相を保持している

Fig. 1 Lattice structure. The left one is the base mesh and the right one is the corresponding lattice surface. Topologies of these models have one to one correspondence.

曲面に基づく形状表現を使用することである。自由曲面は複雑な曲面形状をわずかな数の制御点で表現することができる。これにより、複雑な形状をポリゴンメッシュと比較して非常に少ないデータサイズで表現することが可能になり、前者の条件が満たされる。また、自由曲面データは多くの CG システムや CAD/CAM システムで利用されているため、データ転送した後に様々な用途に再利用することが可能になる。製造業をはじめとする様々な分野における Web3D 利用が可能になり、後者の条件を満たすことができる。

我々は Web3D 用途において自由曲面データを効果的に表現することができる新しいフレームワークを提案する。本手法において、形状データは自由曲面とポリゴンの両方の特性を持つものとして扱われる。データ転送の際には、自由曲面形状の特徴を反映した単純なポリゴン (以後、初期多角形) として扱うことが可能である。そのためデータサイズはコンパクトになり、高速なデータ転送が実現される。データを受信した計算機上では、転送されてきた初期多角形から自由曲面 (以後、ラティス曲面) を構築し、高品質な自由曲面形状を表示することができる。このように、1 つの形状データを用途に応じて自由曲面とポリゴンの 2 つの形式で扱うことができる枠組みをラティス構造と呼ぶ。図 1 にラティス構造を示す。右がラティス曲面であり、左がそれに対応する初期多角形である。初期多角形はラティス曲面と同一位相であり、頂点、稜線、面、重み付け情報によって構成されている。初期多角形はラティス曲面より再構築することが可能であり、相互に変換することが可能である。

ラティス構造では、初期多角形を変形することで内部の曲面形状を変形することができるため、変形アニメーションにおいて有効である。また、本手法では双 3 次 Gregory パッチ⁸⁾に基づく自由曲面データを扱

う。このことは、データ転送後に CAD/CAM システムへ曲面制御点データを入力し、再利用可能であることを意味する。細分割曲面でも曲面形状を表現することができるが、曲面制御点を明示的に保持していないため、細分割後の密なメッシュデータを入力する必要がある。さらに、ラティス構造では、曲面の分割数を変更することで、用途に応じた詳細度制御をすることができる。

3. ラティス曲面生成と初期多角形再構成

本章では、初期多角形からラティス曲面を生成する手法および、生成されたラティス曲面から初期多角形を再構成する手法について述べる。まず、ラティス曲面を構成する双3次 Gregory パッチの曲面表現式とその特徴について解説する。次に、双3次 Gregory パッチの特徴を利用することにより、初期多角形と位相的に等しいラティス曲面を生成する曲面生成手法について解説する。

ラティス曲面は、細分割曲面のようにポリゴンメッシュとして得られるものではなく、自由曲面の制御点データとして取得できる点が大きな特徴である。ラティス曲面を構築する曲面生成手法は初期多角形の線形変換として定義できるため、逆変換を求めることによりラティス曲面から初期多角形を再構築することが可能である。

3.1 Gregory パッチ

ラティス曲面において、自由曲面データは双3次 Gregory パッチ⁸⁾により表現される。双3次 Gregory パッチは、一般 Coons 曲面のツイストベクトルを境界曲線の横断ベクトルとして表した Gregory による拡張⁹⁾を双3次 Bézier パッチに適用したものである。図2に双3次 Gregory パッチを示す。

図2に示すように、双3次 Gregory パッチは20個の制御点 P_{ijk} ($i = 0, \dots, 3; j = 0, \dots, 3; k = 0, 1$) により構成される。双3次 Gregory パッチの曲面表

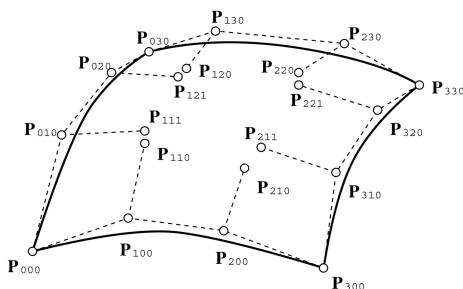


図2 双3次 Gregory パッチ
Fig. 2 Gregory patch.

現式は以下ようになる。

$$S(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 B_i^3(u) B_j^3(v) Q_{ij}(u, v) \quad (1)$$

$$(0 \leq u \leq 1, 0 \leq v \leq 1)$$

ここで $B_i^3(u)$ と $B_j^3(v)$ は Bernstein 基底関数である。また、制御点 P_{ijk} ($i = 0, \dots, 3; j = 0, \dots, 3; k = 0, 1$) と Q_{ij} ($i = 0, \dots, 3; j = 0, \dots, 3$) には次のような関係がある。

$i \neq 1, 2$ または $j \neq 1, 2$ のとき

$$Q_{ij}(u, v) = P_{ij0} \quad (2)$$

$i = 1, 2$ または $j = 1, 2$ のとき

$$Q_{11}(u, v) = \frac{uP_{110} + vP_{111}}{u + v},$$

$$Q_{12}(u, v) = \frac{uP_{120} + (1-v)P_{121}}{u + (1-v)},$$

$$Q_{21}(u, v) = \frac{(1-u)P_{210} + vP_{211}}{(1-u) + v},$$

$$Q_{22}(u, v) = \frac{(1-u)P_{220} + (1-v)P_{221}}{(1-u) + (1-v)} \quad (3)$$

ただし $0 \leq u \leq 1, 0 \leq v \leq 1$ である。

境界曲線は3次 Bézier 曲線によって定義され、双3次 Bézier パッチの内部制御点を二重にした構造を持っている。また、流れベクトル関数 (Cross Boundary Derivative) を u, v のパラメータごと独立に定義することができる。そのため、輪郭曲線列さえ確定すれば、曲面間の G^1 連続を容易に確保することが可能である。双3次 Gregory パッチは連続性に優れた曲面表現であり、NURBS にも変換可能¹⁶⁾である。この特徴を利用することにより、ラティス構造で表現される形状データを CAD/CAM システムで利用することが可能になる。

3.2 ラティス曲面生成

ラティス曲面は丸め操作 (Rounding Operations)^{10)~12)}によって生成される。丸め操作とは、単純なポリゴン形状を入力として滑らかな曲面形状を作成するモデリング手法のことである。従来の丸め手法は位相が著しく変化してしまい、生成された曲面形状と丸め前の格子形状の対応付けが困難であった。そのため、格子形状と丸め形状の連携が必要である直感的な変形操作や、格子形状の再構築などが困難であった。本手法の丸め操作は格子形状と位相的に等しい丸め形状を作成する。また、丸め形状から格子形状を再構築する (以後、逆丸め) ことが可能である。

本丸め手法は以下の3つの手順により構成される。

STEP1 初期多角形の各頂点に対応するラティス曲面の頂点を求める。

STEP2 初期多角形の各稜線に対応するラティス曲面の曲線を求める。

STEP3 曲線によって囲まれた領域に曲面を内挿する。

以下、各手順の詳細について述べる。

STEP1: ラティス曲面の頂点作成

初期多角形の頂点に対応する、ラティス曲面の頂点を作成するには、面取り操作を用いる。まず、図3(a)に示すように初期多角形の各頂点を V_i ($i = 0 \dots n$) とする。初期多角形の各面について、面に属するすべての稜線の中点と、面に属するすべての頂点の平均座標値に位置する点(以後、平均点)を計算し、これらを接続する(図3(b))。ここで、初期多角形の頂点 V_0 に着目する。 V_0 に接続しているすべての面の平均点を求め、それらを接続する。これにより、初期多角形の頂点 V_0 に対して新しい面 F_0 が定義される(図3(c)(d))。同様に、初期多角形に属するすべての頂点 V_i に対して、これらと1対1に対応する面 F_i ($i = 0 \dots n$) が定義される。最後に F_i の平均点 P_i ($i = 0 \dots n$) を求め、それをラティス曲面上の頂点とする。 P_i は V_i と1対1に対応するため、 V_i の接続関係をそのまま P_i に適用することで、初期多角形と同一位相を構築することができる。この面取り操作は、稜線の中点と面の平均点を利用しているという点で Doo-Sabin 細分割手法の第1分割に似ている。そのため、生成されるラティス曲面は Doo-Sabin 細分割手法の収束曲面に近い形状になる。

STEP2: ラティス曲面の曲線作成

初期多角形の線分に対応するラティス曲面の曲線分を作成する方法について述べる。ラティス曲面の曲線データは3次 Bézier 曲線として表現する。これはラティス曲面の曲面表現式に双3次 Gregory パッチを用いているためである。図4(a)に示すように、初期多角形の線分 V_0V_1 に対応する曲線を求めるとする。3次 Bézier 曲線は4つの制御点によって構成されており、両端の制御点は曲線の終点および始点と等しい(図4(b))。よって、図4(a)に示すように、曲線の始点 C_0 と終点 C_3 は STEP1 で算出済みであることになる。3次 Bézier 曲線を定義するには内側の2つの制御点 C_1, C_2 を求めればよい。まず、図4(c)に示すように、STEP1 により作成された F_0 に属する稜線の中点 Q と P_0 ($= C_0$) によって生成されるベクトルを定義する。このベクトルを3次 Bézier 曲線の始点における接線ベクトルとする。同様に、終点における接線ベクトルは R と C_3 によって定義されるとする。制御点 C_1, C_2 は以下の式を満たす値として定

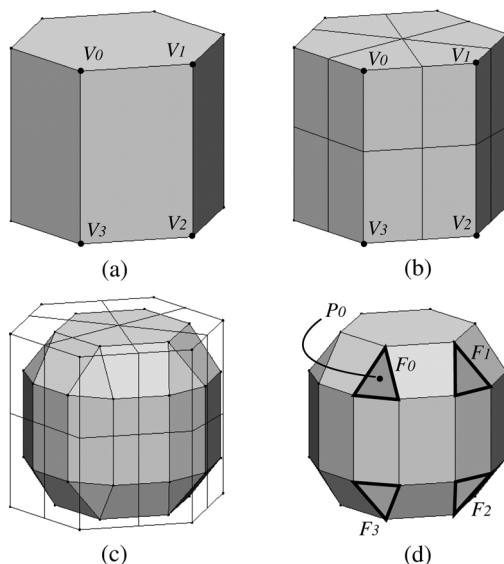


図3 ラティス曲面の頂点計算

Fig. 3 Calculation of vertices of lattice surface.

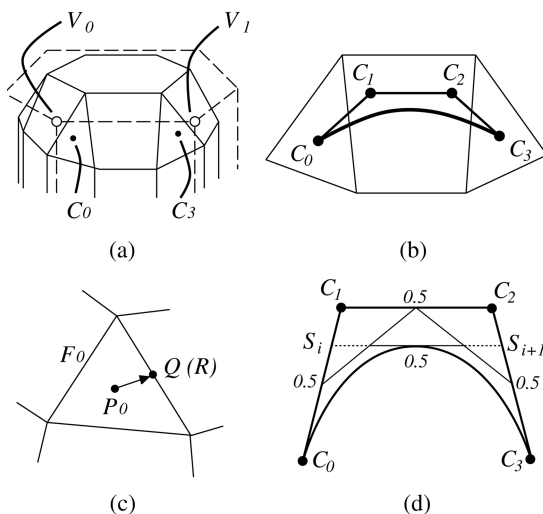


図4 ラティス曲面の曲線計算

Fig. 4 Calculation of curves of lattice surface.

義する。

$$C_0C_1 = \frac{4}{3}C_0Q \quad C_3C_2 = \frac{4}{3}C_3R \quad (4)$$

ここで、 C_0C_1 を C_0Q の $4/3$ 倍のベクトルとして定義する理由は、ラティス曲面が初期多角形の内部に収まるようにするためである。3次 Bézier 曲線が左右対称であると仮定すると、図4(d)に示すように、制御点も左右対称に配置される。パラメータ値 0.5 における曲線上の点を計算するには、制御ポリゴンの各稜線をパラメータ値 0.5 で分割し、新たに生成された頂点を

接続するという操作を再帰的に行えばよい．曲線上の点のがの稜線を左右に延長し，元の制御ポリゴンと交わる点を S_i, S_{i+1} とすると，曲線の最小の凸包を構成する $C_0 S_i S_{i+1} C_3$ を得る．この凸包が初期多角形であると仮定すると， $C_0 C_1 = \frac{4}{3} C_0 S_i$ ， $C_3 C_2 = \frac{4}{3} C_3 S_{i+1}$ になる． S_i, S_{i+1} をそれぞれ Q, R と見なすと，初期多角形に収まるラティス曲面を生成するには， $C_0 C_1$ を $C_0 Q$ の $4/3$ 倍のベクトルとして定義するのが望ましいことが分かる．初期多角形に極端な凹凸や非多様体に分類される部位が存在する場合は，Bézier 曲線どうしが交差したり破断することがありうる．また，3 次 Bézier 曲線が左右対称であるという仮定の妥当性も問題になる．しかし，初期多角形を用意するのがユーザであるという一般的な条件下で使用する限り，これらの問題は致命的なことではない．

STEP3：曲面の内挿

曲線によって囲まれた領域に曲面を内挿する方法について述べる．STEP2 までの手順により，すべての領域は 3 次 Bézier 曲線により囲まれている．この領域に Chiyokura ら⁸⁾ の手法を用いて双 3 次 Gregory パッチを内挿する．双 3 次 Gregory パッチの 20 個の制御点の中で輪郭に位置する 12 個の制御点は，STEP2 において求めた境界曲線の制御点に等しい．そのため，ここで求めるべきものは内部の 8 つの制御点である．ここで曲面間の連続性を確保するために，彼らの手法では，基礎パッチ法¹⁸⁾ と呼ばれる手法を用いている．この手法は，接続する曲面間の境界曲線上に仮想的なパッチを想定し，そのパッチとそれぞれの曲面を接続する手法である．この手法により，曲面間が G^1 連続を保つように曲面を内挿することができる．以上の操作により，初期多角形と位相的に等しい自由曲面形状を生成することができる．

3.3 重み付け

初期多角形の要素に重み付けをすることにより，ラティス曲面の形状を局所的に変形することが可能である．重みを付けることができるのは，初期多角形の頂点と稜線である．重み値を大きくすれば，生成されるラティス曲面の形状は初期多角形の形状に近づく．

稜線に重みを付けるには，図 4(b) における 3 次 Bézier 曲線 $C_0 C_1 C_2 C_3$ とそれに対応する初期多角形の線分 $V_0 V_1$ に着目する．曲線の内部の制御点 C_1, C_2 を，それぞれ V_0, V_1 に近づけることによって稜線の重み付けがなされる．頂点に重みを付けるには，3 次 Bézier 曲線の両端の制御点 C_0, C_3 を，それぞれ V_0, V_1 に近づける．稜線，頂点ともに重みを最大に付けると，3 次 Bézier 曲線 $C_0 C_1 C_2 C_3$ は線分 $V_0 V_1$

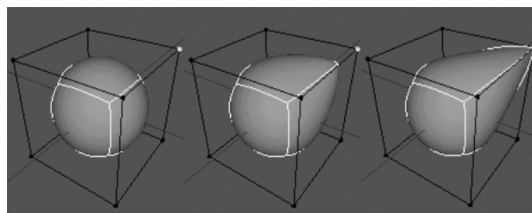


図 5 重み付け
Fig. 5 Weighting.

と等しくなり，生成される曲面形状も初期多角形と等しくなる．図 5 に重み付けを適用した例を示す．

さらに，本手法では始点丸めベクトルと終点丸めベクトルという概念を用いる．これは，生成される 3 次 Bézier 曲線の始点と終点における接平面ベクトルのことである．これらの要素を自由に調節することにより，より柔軟な曲面形状生成が実現される．

3.4 逆丸め

3.2 節より，ラティス曲面を生成するには，境界曲線を構成する 3 次 Bézier 曲線の制御点を求めればよいことが分かる．また，3 次 Bézier 曲線の制御点は初期多角形の線形変換によって取得することができる．そのため，ラティス曲面の制御点を P_0 ，初期多角形の頂点を V_i とすると，以下の関係が成り立つ．

$$P_0 = \sum k_i V_i \quad (5)$$

ただし， k_i は実数である．これはラティス曲面の制御点が初期多角形の頂点の線形変換によって取得できることを表している．線形変換には一般に逆変換が存在するため，初期多角形の頂点 V_0 は

$$V_0 = \sum l_j P_j \quad (6)$$

として求めることができる．ただし， l_j は実数である．これらの関係は，ラティス曲面から初期多角形が再構築可能であることを示す．ラティス曲面の各頂点に対応する初期多角形の各頂点を求めた後に，ラティス曲面の各曲線を直線に変換することで初期多角形が再構築される．

逆丸め操作により初期多角形を再構築することの意義は，形状の変形操作における有効性にある．初期多角形は曲面形状の特徴を反映した単純な多角形情報であるために，変形操作の際の制御格子として用いることができる．5.1 節で詳細を述べるが，初期多角形を変形媒体とすることで，変形操作の際に曲面間の連続性を容易に確保することができる．

4. Web3D データフォーマット

4.1 Web3D データフォーマットの必要性

Web3D に最適な形状表現を実現するためには、インターネット上で容易に利用できるデータフォーマットであることが求められる。現在、国際標準として用いられている Web3D データフォーマットは VRML97[☆]である。産業としての大幅な普及には至らなかったものの、ポリゴンメッシュによる明解な幾何形状表現とテクスチャマッピングなどの多彩な材質表現を実現しているため、研究機関などでは広く用いられている。我々はラティス構造の有用性を示す第 1 段階として、VRML97 に初期多角形情報を埋め込む手法について提案した¹³⁾。これにより、従来ポリゴンメッシュしか扱うことのできなかった VRML97 に自由曲面を扱う枠組みを付加することが可能になった。

また、近年注目を集めているインターネット文書作成言語として XML (eXtensible Markup Language) があげられる。拡張性を持つ次世代のマークアップ言語として大きな期待を集めており、W3C¹⁴⁾ が中心になり、詳細な仕様決定や様々なツールの作製が進められている。インターネット上での 3D データ利用を促進するためには、この XML によりデータを表現することが必須になる。そのような現状のなかで、VRML97 の標準化委員会である Web3D コンソーシアムでは、この XML による拡張性を持たせた VRML の仕様を現在作成中である。現在は X3D (eXtensible 3D) という仮称で仕様が作成されているが、2002 年を目標に VRML2002 として ISO スタンドアードとして提案する予定である。X3D は XML の特徴を活かし、様々な拡張形式をサポートすることができる。我々はラティス構造を X3D の拡張形式として Web3D コンソーシアムに提案した¹⁵⁾。

以下の本章では、VRML97 にラティス構造を適用する手法と、X3D の拡張形式としてラティス構造を適応させた例について述べる。

4.2 VRML97

図 6 に、初期多角形の情報を VRML に組み込んだ例を示す。以後、ラティス構造により表現される VRML を XVL (eXtended VRML with Lattice) と呼ぶ。ポリゴンとして扱うことにより、ラティス構造をあらゆる VRML ブラウザで読み込むことが可能になる。もしブラウザが XVL をサポートするものであれば、初期多角形からラティス曲面を構築し、滑らか

```

1:  #VRML V2.0 utf8
2:  ##### 重み付け情報ノード
3:  PROTO XVL_EDGE [
4:      field SFFloat round_val 0
5:      field SFVec3f round_str 0 0 0
6:      field SFVec3f round_end 0 0 0
7:  ]
8:  { Text { string [ "weight of edge " ] } }
9:  ##### 丸めフラグノード
10:  PROTO XVL_STATUS [
11:      field SFString status "XVL_LATTICE"
12:  ]
13:  { Text { string [ "status of shape " ] } }
14:  Group{
15:      children [
16:          ##### 初期多角形情報
17:          Group {
18:              children [
19:                  Shape {
20:                      geometry IndexedFaceSet {
21:                          #####
22:                          ## 初期多角形の幾何, 位相情報
23:                          #####
24:                      }},
25:                      ##### ラティス曲面の重み付け情報
26:                      Switch {
27:                          choice [
28:                              XVL_STATUS {
29:                                  status "XVL_GREGORY"
30:                              }
31:                              XVL_EDGE {
32:                                  round_val 0.5
33:                                  round_str 0 0 0
34:                                  round_end 0 0 0
35:                              }
36:                              IndexedLineSet {
37:                                  coordIndex [ 24 103 ]
38:                              }
39:                          ]
40:                          whichChoice -1
41:                      }
42:              ]

```

図 6 VRML にラティス構造を入れ込む方法

Fig. 6 Method of migrating lattice structure to VRML.

な曲面形状が描画される。図 6 にあるように、XVL は Group ノードと Switch ノードの 2 つのパートから構成される。Group ノードには初期多角形の幾何情報が保存される。Switch ノードにはラティス曲面を構築するために必要な属性情報が保存される。Switch ノードの whichChoice フィールドが -1 になっているために、XVL を認識できないブラウザはこの情報を無視することができる。この場合は初期多角形の形状が描画されることになる。XVL を認識する場合は、choice フィールドから必要な情報を取得し、ラティス曲面を生成する。XVL ファイルをラティス曲面として表示するか、もしくは初期多角形として表示するか

☆ ISO/IEC 14772-1:1997

は XVLSTATUS ノードによって決定される。status フィールドが XVLGREGORY であればラティス曲面を、XVLLATTICE であれば初期多角形を描画する。図 6 の場合はラティス曲面を描画する。曲面を生成するには、頂点と稜線に重み付けをすることで様々な形状を生成することができる。重み付け情報である重み値 (round_val), 始点丸めベクトル (round_str), 終点丸めベクトル (round_end) は XVLEDGE ノードおよび XVLVERTEX ノードに保存される。図 6 の場合はある 1 つの稜線について重み付けがされることになる。稜線は IndexedLineSet ノードにより記述される。座標値は初期多角形情報が記述されている Group ノードに用いたものを使用し、稜線を構成する 2 つの頂点インデックスのペアを記述することで稜線情報を記述する。図 6 の場合は、初期多角形で用いられている 24 番目と 103 番目の頂点により構成される稜線に重み付けをすることになる。

4.3 X3D (VRML2002)

XML には DTD (Document Type Definition) というタグ仕様の宣言ファイルが必要になる。XML パーザは DTD を参照しながらファイルの構文チェックを行う。我々はラティス構造を実現するデータ構造を定義し、それを DTD としてまとめた (図 7)。

CAD/CAM 分野での利用も前提にするためには、データ構造はソリッドモデル¹⁷⁾を扱うことができなくてはならない。我々は位相要素として、シェル (XvlShell), 面集合 (Faces), 稜線集合 (Edges), 頂点集合 (Vertices), 面 (Face), 稜線 (Edge), 頂点 (Vertex) を扱う。幾何要素としては、座標値 (XYZ), テクスチャ座標値 (UV) を扱う。材質要素としては、テクスチャ (Texture), マテリアル (Material) を扱う。さらに、レンダリングはワイヤフレーム表示とシェーディング表示を制御する。ワイヤフレーム表示では、ポリライン (CS.POLYLN), 曲線 (CS.BEZIER) の表示が可能である。シェーディング表示では、初期多角形表示 (SS.LATTICE), ラティス曲面表示 (SS.GREGORY), ポリゴンメッシュ表示 (SS.POLYGON) が可能である。これらの表示モードの情報は XvlShellTypes タグで制御される。シェル要素 (XvlShell) は自身の属性値 (アトリビュート) として、レンダリングモード (Type), 自身の ID (DEF), 他のシェルへのポインタ (USE) を保持している。面要素 (Face) は属性値として、頂点座標のインデックス (ix), テクスチャ座標のインデックス (tx) を保持している。稜線要素 (Edge) は属性値として、頂点座標のインデックス (ix), 始点と終点の丸めベクトル

```

1:  <!ENTITY % XvlShellTypes " CS_POLYLN | CS_BEZIER
2:    | SS_POLYGON | SS_LATTICE | SS_GREGORY " >
3:  <!ELEMENT   XvlShell   (XYZ, UV, Faces*, Edges*,
                           Vertices*)>
4:  <!ATTLIST   XvlShell
5:    Type      ( %XvlShellTypes; ) #REQUIRED
6:    DEF       ID          \#IMPLIED
7:    USE       IDREF       \#IMPLIED
8:  >
9:  <!ELEMENT   Faces      (Material? , Texture? ,
                           Face+) >
10: <!ELEMENT   Edges      (Material? , Edge+) >
11: <!ELEMENT   Vertices   (Material? , Vertex+) >
12: <!ELEMENT   Face       (Material? , Texture?
                           %XvlG1T1FaceChild;) >
13: <!ATTLIST   Face
14:   ix        %MInt32;   #REQUIRED
15:   tx        %MInt32;   #IMPLIED
16: >
17: <!ELEMENT   Edge       (Material?
                           %XvlG1T1EdgeChild; ) >
18: <!ATTLIST   Edge
19:   ix        %MInt32;   #REQUIRED
20:   vec       %MInt32;   #IMPLIED
21:   WRnd      %SFloat;   #IMPLIED
22: >
23: <!ELEMENT   Vertex     (Material?) >
24: <!ATTLIST   Vertex
25:   ix        %MInt32;   #REQUIRED
26:   VRnd      %SFloat;   #IMPLIED
27: >
28: <!ELEMENT   Texture     (EMPTY) >
29: <!ATTLIST   Texture
30:   url       CDATA      \#IMPLIED
31:   repeatS   %SBool;    "true"
32:   repeatT   %SBool;    "true"
33:   bin       IDREF      \#IMPLIED
34:   DEF       ID          \#IMPLIED
35:   USE       IDREF      \#IMPLIED
36: >
37: <!ELEMENT   XYZ        \#PCDATA >
38: <!ELEMENT   UV         \#PCDATA >

```

図 7 X3D のためのラティス構造 DTD
Fig. 7 DTD of lattice structure for X3D.

(vec), 重み値 (WRnd) を保持している。頂点要素 (Vertex) は属性値として、頂点座標のインデックス (ix), 重み値 (VRnd) を保持している。テクスチャ要素 (Texture) は属性値として、テクスチャ画像ファイルの URL (url), テクスチャの反復とクランプの情報 (repeatS, repeatT) を保持している。

5. 応用事例

5.1 ラティスモデリング

制御点を直接操作する従来の自由曲面の変形操作は、操作性が悪く連続性を確保しながら作業を進めるのに熟練を要する。一方で、ラティス構造を用いたモデリ

ング（ラティスマデリング）では，初期多角形が変形制御格子として機能するため，操作が容易である．さらに，双 3 次 Gregory パッチ⁸⁾ の特性により生成される形状は G^1 連続が確保される．これは，細分割曲面において，初期メッシュを操作することで収束曲面を変形する操作に類似している．この操作はユーザにとって非常に直感的な作業であるため，変形手法として有効である．細分割曲面においては最終的に得られる情報がポリゴンメッシュの頂点座標値であるが，ラティスマデリングの場合は自由曲面の制御点座標値を得ることができる．このことは，ポリゴンモデリングの操作感によって自由曲面モデル生成を実現することを意味する．ラティス曲面は NURBS にも変換できるため，他の曲面モデリングシステムにデータをエクスポートすることも可能である．さらに重み付けを適用することで，様々な種類の形状を作製することが可能

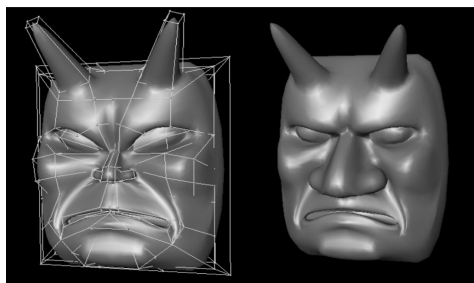


図 8 ラティスマデリング．初期多角形の操作により内部のラティス曲面を変形した例

Fig. 8 An example of lattice modeling. The internal surface shape is deformed through lattice deformation.

になる．図 8 に本モデリング手法による例を示す．

5.2 詳細度制御

Web3D ではインタラクティブに形状を操作できることが必須である．VRML のようなポリゴンメッシュを用いたデータの場合，メッシュを構成するポリゴン数が多いとインタラクティブな操作は非常に困難であり，詳細度（Level of Details）制御を行う必要性がでてくる．その際，形状の特徴を損なわずにポリゴン数を削減するために，頂点を削除する順番と削除後の頂点座標値の移動ベクトルを指定するための様々な工夫が必要になる．一方，自由曲面を用いれば，ポリゴン分割する際の分割数を調節することで表示データの精度を制御することができる．また，曲面データの各輪郭線に沿ってポリゴンが作成されるために，形状が保持する特徴線が失われることはない．本手法では分割数の指定に加えて最も荒い精度として初期多角形を表示している．これは大まかな形状把握や，モデルが視点座標値から離れた場所にある場合の表示要素として利用する場合に効果的である．図 9 に自由曲面の分割数の制御を通じた形状の詳細度制御の例を示す．

ところで，CAD/CAM システムにおいては精度が非常に高いデータが要求される．詳細度制御によって，高精度なデータを生成すれば，データをこれらの用途に使用することも可能である．図 10 にラティス構造によって表現されたデータをもとに CAM システムで生成した実モデルを示す．

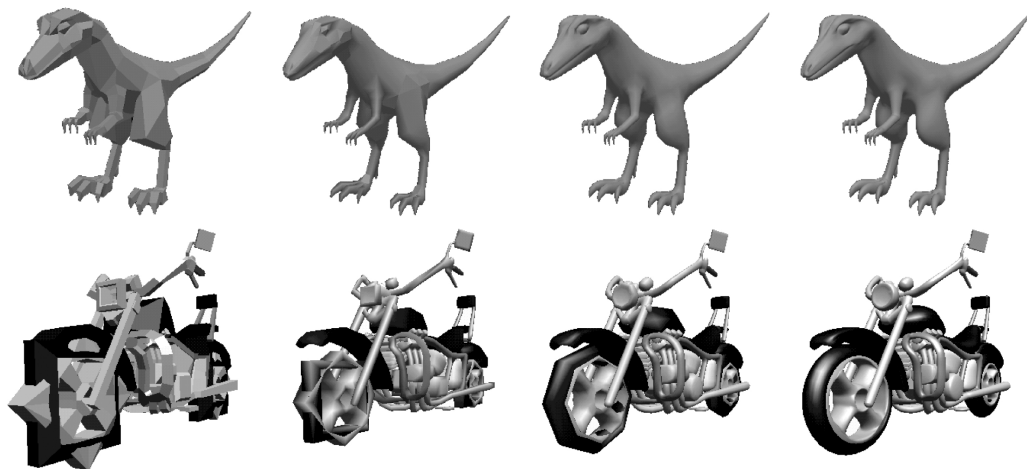


図 9 分割数の制御を通じた形状の詳細度制御．左から初期多角形，分割数 1，分割数 2，分割数 8 のモデルを表している．

Fig. 9 Level of details controls with division number. From left to right, base mesh, division number 1, division number 2 and division number 8.

表 1 データサイズ比較と曲面生成時間
Table 1 Data size and calculation times spent in surface generations.

	VRML(div.8)	VRML(div.4) (KByte)	XVL	Mesh num.	Lattice num.	Surface gen. (second)
high-heels	395	128	19	6340	49	0.010
oni	1393	453	48	20992	168	0.030
raptor	4087	1355	150	65920	532	0.080
woman	14403	4866	570	223488	1724	0.361
bike	17641	5660	733	286848	2207	0.370



図 10 CAM システムによって生成した実モデル

Fig.10 A physical model generated with CAM system.

6. 結果と考察

表 1 はポリゴンメッシュとラティス構造との比較を示すものである。正確なデータ比較のために、データフォーマットには VRML を採用した。これは、4.2 節で述べたように、ラティス構造は VRML により表現することができ (XVL 形式)、ポリゴンメッシュも VRML の IndexedFaceSet ノードで表現することができるためである。high-heels, oni, raptor, woman, bike の 5 つのデータについて、ラティス構造の初期多角形 (XVL)、8 分割の三角形メッシュ (VRML(div.8))、4 分割の三角形メッシュ (VRML(div.4)) のデータサイズを測定した。三角形メッシュは本手法により作成されたラティス曲面を分割して作成したものであるが、一般的な三角形メッシュとして扱えるものである。既存の三角形メッシュとの比較のために 8 分割した際の面数 (Mesh num.) を同時に示した。また、初期多角形の面数 (Lattice num.) についても示す。データサイズに着目すると、XVL は分割数 4 の VRML データと比較しても、10% 強のファイルサイズで表現可能なことが分かる。これはデータ転送が高速に行えることを意味するため、インターネット利用においての有用性を示すことになる。

ところで、細分割曲面を正式にサポートする汎用的な Web3D ファイルフォーマットは存在しないが、本手法との比較を示す必要がある。もし、形状に特別な属性 (重み付けなど) が存在しないと仮定すれば、両者

の初期多角形はプレーンなポリゴンメッシュと見なすことができるため、面数は等しくなる。また、VRML の IndexedFaceSet ノードを用いると、両者は同一なデータとして記述することができるため、データサイズも同一のものになる。データ転送後の用途が表示のみの場合、曲面制御点の生成とポリゴンメッシュの生成の 2 つのプロセスが必要な本手法よりも、直接ポリゴンメッシュを生成できる細分割曲面の方が有効である。転送後に CAD/CG システムを用いて形状を再加工する必要がある場合は、自由曲面の制御点を明示的に保持することができる本手法を採用するのが望ましい。

曲面生成時間 (Surface gen.) の測定には、PentiumIII 933 MHz, 256 MB Memory の PC を用いた。ラティス曲面の生成は非常に高速で、インタラクティブなレスポンスで実行できることが分かる。これは、フレームごとでの曲面生成が可能であり、曲面を用いた変形アニメーションがインタラクティブに実行できることを示す。しかし、曲面分割数が多い場合には生成される三角形メッシュ数 (Mesh num.) が膨大になり、一般的な PC での処理が困難になる。特に bike を 8 分割した際には、30 万弱の三角形メッシュ数になるため、インタラクティブな表示のためには、ポリゴン削減などの最適化手法との併用が必要である。

7. 結論と展望

本論文では、軽量で高品質な曲面形状を表現できる Web3D 表現を提案した。自由曲面を効果的に管理するラティス構造により、高速なデータ転送と高品質なレンダリングイメージの取得が可能であることを示した。ラティス構造を VRML および X3D に組み込む具体的な方法を考案することで、Web3D データとしての実用性を示した。Gregory パッチの制御点情報を保持することにより、CAD/CAM 分野でのインターネット利用に有用なことを示した。さらに、ラティス構造の変形手法としての有用性を示した。

今後の展望としては、以下の 2 点があげられる。まず第 1 に、複雑な曲面形状の表現に用いられることの

あるトリム曲面への対応である。トリム曲面を扱えるようにラティス構造を拡張し、VRMLとX3Dによる具体的な記述方法を提案することにより、複雑な機械部品などを扱うエンジニアリング分野でのWeb3D展開が期待できる。第2に、ポリゴンメッシュなどの既存データを本論文で提案したデータ表現に変換する手法の確立である。これは、ポリゴンメッシュを細分割曲面に変換するサーフィスフィッティングの手法により解決することができる。以上の対応により、より多様なデータをインターネット上で利用する機会を提供できるように努力していきたい。

謝辞 本論文を執筆するにあたり、多くの適切な助言をいただきました慶應義塾大学環境情報学部金井崇氏に心より感謝申し上げます。

参 考 文 献

- 1) Hoppe, H.: Progressive Meshes, *Computer Graphics (Proc. SIGGRAPH 96)*, pp.99–108 (1996).
- 2) Abadjev, A., Rosario, M. del., Lebedev, A., Migdal, A. and Paskhaver, V.: MetaStream, *VRML'99* (1999).
- 3) Doo, D. and Sabin, M.: Behaviour of Recursive Division Surfaces Near Extraordinary Points, *Computer Aided Design*, pp.356–360 (1978).
- 4) Catmull, E. and Clark, J.: Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes, *Computer Aided Design*, pp.350–355 (1978).
- 5) Loop, C.: Smooth Subdivision Surfaces Based on Triangles, Master's thesis, University of Utah, Department of Mathematics (1987).
- 6) Zorin, D., Schroder, P. and Sweldens, W.: Interactive Multiresolution Mesh Editing, *Computer Graphics (Proc. SIGGRAPH 97)*, pp.259–268, ACM Press, New York (1997).
- 7) Lee, A., Moreton, H. and Hoppe, H.: Displaced Subdivision Surfaces, *Computer Graphics (Proc. SIGGRAPH 2000)*, pp.85–94, ACM Press, New York (2000).
- 8) Chiyokura, H. and Kimura, F.: Design of Solids with Free-form Surfaces, *Computer Graphics (Proc. SIGGRAPH 83)*, pp.289–298, ACM Press, New York (1983).
- 9) Gregory, J.A.: Smooth Interpolation without Twist Constraints, *Computer Aided Geometric Design*, pp.71–87, Academic Press, New York (1974).
- 10) Harada, T., Toriya, H. and Chiyokura, H.: An Enhanced Rounding Operation between Curved Surfaces in Solid Modeling, *CG International 90*, pp.563–588 (1990).
- 11) Loop, C.: Smooth Spline Surfaces over Irregular Meshes, *Computer Graphics (Proc. SIGGRAPH 94)*, pp.303–310, ACM Press, New York (1994).
- 12) Chiyokura, H.: An Extended Rounding Operation for Modeling Solids with Free-form Surfaces, *IEEE Computer Graphics and Applications* (1987).
- 13) Wakita, A., Yajima, M., Harada, T., Toriya, H. and Chiyokura, H.: XVI: A Compact and Qualified 3D Representation with Lattice Mesh and Surface for the Internet, *Web3D-VRML2000*, pp.45–51, ACM Press, New York (2000).
- 14) World Wide Web Consortium: Extensible Markup Language (XML). <http://www.w3.org/XML/> (2000).
- 15) Lattice Technology, Inc.: Lattice Extensions for X3D. <http://www.web3d.org/TaskGroups/x3d/lattice/LatticeProposal.html> (2000).
- 16) Takamura, T., Ohta, M., Toriya, H. and Chiyokura, H.: A Method to Convert a Gregory Patch and a Rational Boundary Gregory Patch to a Rational Bezier Patch and its Application, *CG International 90*, pp.543–562, Springer-Verlag, Tokyo (1990).
- 17) 鳥谷浩志, 千代倉弘明: 3次元CADの基礎と応用, 共立出版 (1991).
- 18) 千代倉弘明: ソリッドモデリング, 工業調査会 (1985).

(平成12年10月6日受付)

(平成13年3月9日採録)

脇田 玲 (学生会員)



昭和11年慶應義塾大学大学院政策・メディア研究科修士課程修了。同年同大学院博士課程に進学。形状モデリングおよびWeb3Dの基礎研究, それらを応用したメディアアート作品の製作に従事。平成11年日経アーキテクチュアデジタルデザインコンペ最優秀賞, 平成12年マルチメディアグランプリ2000ネットワーク部門情報デザイン賞受賞。

**矢島 誠**

平成 3 年東京理科大学理学部物理学
学科卒業。平成 8 年東京大学大学院
理学系研究科物理学専攻博士課程修
了。同年 (株) アドバンスに入社。平
成 9 年ラティス・テクノロジー (株)
に入社。現在同社インターネットグラフィックス事業
部ソリューションエンジニア。

**原田 毅士**

昭和 61 年東京大学工学部精密機
械工学科卒業。昭和 63 年東京大学
大学院精密機械工学専攻修士課程修
了。同年 (株) リコー入社。ソリッ
ドモデラ DESIGNBASE の研究開
発に従事。平成 10 年ラティス・テクノロジー (株) 入
社。Web3D 用カーネルの開発に従事。現在同社イン
ターネットグラフィックス事業部チーフエンジニア。

**鳥谷 浩志 (正会員)**

昭和 58 年東京大学理学部情報科
学科卒業。同年 (株) リコー入社。
ソフトウェア研究所でソリッドモデ
ラ DESIGNBASE の研究開発に従
事。平成 10 年ラティス・テクノロ
ジー (株) 入社。現在、同社代表取締役社長。

**千代倉弘明 (正会員)**

昭和 54 年慶應義塾大学工学部数
理工学科卒業。昭和 59 年東京大学
大学院精密機械工学専攻博士課程修
了。同年 (株) リコー入社。ソリッ
ドモデラ DESIGNBASE の研究開
発に従事。平成 2 年慶應義塾大学環境情報学部助教授。
平成 9 年より同大学同学部教授。