

Non-iterative 行列演算による 3 次元空間における 曲面の最適埋蔵法

今藤 紀子[†] 落合 豊行[†]

本論文では、3次元曲面モデリングのための新たな手法を提案する。3次元曲面モデリングのための手法は、Catmull-Clark のスキームなどによる細分割曲面を発展させ利用したものが現在の主流となっている。これらの手法は、与えられた多角形ネットからそれに制御された滑らかな曲面を、イタレーティブに曲面を細分割した極限として定義している。これに対し本論文で述べる手法では、曲面を構成する頂点間の隣接関係から得られる変換行列の性質を利用することにより、イタレーティブな演算をせずに、1回の行列演算で極限となる滑らかな曲面を得ることができる。また、多角形ネットの入力も不要である。本論文においてこの手法を3次元空間における曲面の最適埋蔵法として提案するとともにイタレーティブな演算による手法との比較、評価を行い、その有効性を実証する。

A Non-iterative Matrix Method for Nicely Embedding of Smooth Surface into 3-space

NORIKO IMAFUJI[†] and MITSUYUKI OCHIAI[†]

This paper describes a non-iterative matrix method for obtaining a smooth surface. Since a variety of methods for 3-dimensional surface modeling such as extended subdivision approaches have been proposed, the basic common idea of these methods is that surfaces are defined as the limit of subdividing the given polygon net iteratively. The method in this paper, on the contrary, can obtain a smooth surface by non-iterative matrix computation without input of any complicated initial polygon nets. In this paper, We will propose this new method called *non-iterative method for nicely embedding of smooth surface into 3-space*, and show its effectiveness by evaluating of a comparison with an iterative method.

1. はじめに

3次元曲面モデリングの手法は、これまでに数多く提案されてきた。細分割曲面は、多角形ネットを初期状態として与え、これを繰り返し分割し、その極限として滑らかな曲面を得るというものである。近年では Catmull-Clark のスキームによる手法¹⁾ がほぼスタンダードとなり、このスキームを発展させた手法が多く提案されている^{5),8),10)}。3次元曲面モデリングの別のアプローチとして平滑化操作⁶⁾ が知られているが、これは、与えられた曲面からノイズを取り除いて、より滑らかな曲面を得るための手法で、信号処理によるものなどがある⁹⁾。この手法は先にあげた細分割曲面の手法と異なり、操作を通して曲面を構成する頂点間の隣接関係を変えない、つまり、新たな頂点を加えてメッシュの構造自体を変えることはない。最近では、細分

割操作と平滑化操作を組み合わせで行うというのが主流となり^{2)~4)}、より複雑で高度な曲面を得ることが可能になってきている。

しかし、これらの手法では、曲面は、何らかの処理を繰り返し行った後の極限として得られるので、曲面が複雑であればあるほど繰返しの演算に処理時間がかかるとともに、打ち切りの誤差が生じてしまう。また、初期情報として多角形ネットの入力が必要であるが、複雑な曲面になればなるほど、多角形ネットの定義自体の作業が容易ではないという欠点を持つ。

本論文では、3次元空間における曲面の最適埋蔵法と呼ばれる曲面定義のための新しい手法を提案する。最適埋蔵法の基本となる考え方は、曲面を構成する頂点を固定点と動点に分け、固定点を3次元空間内の適当な位置に配置し、動点のみを頂点間の隣接関係によって最適な場所へと移動させ滑らかな曲面を得るというものである。ただし最適とは、任意の動点とその頂点に隣接しているすべての頂点の重心に位置していることをいう。直感的には隣接する頂点間にある種の

[†] 奈良女子大学大学院人間文化研究科
Graduate School of Human Culture, Nara Women's
University

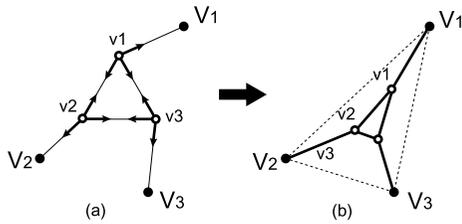


図 1 頂点移動に関する模式図
Fig.1 Transfer of vertices.

張力があり、頂点どうしがその張力により引き付け合い、すべての張力のバランスがとれているような状態であると考えればよい。図 1 は、固定点、動点をそれぞれ 3 つずつとり最適埋蔵法によって動点がどのように移動するかを示している。図 1 (a) において矢印で表されているように各頂点間に力がかかっていると考えると、各動点は図 1 (b) のようにすべての隣接し合う頂点間の引き付け合う力のバランスがとれているような位置に移動する。

この最適埋蔵法では、この張力がつねに「引き付け合う」方向に働いているため、得られる曲面はつねに固定点のみにより張られる凸包の内側に凹んだものとなる。以下にこの最適埋蔵法の特徴と利点をあげる。

- 平面展開グラフの利用；曲面を平面上に切り開いて展開したものを考え、それをグラフと見なす。曲面の分割、つまりメッシュ構造は、すべてこの平面展開グラフ上で定義される。切り開いた部分における貼り合わせの問題は、そのグラフ内で同一の頂点番号を与えるのみで、解決される。最適埋蔵法では、初期状態として必要となる情報は、固定点の座標とこの平面展開グラフの隣接関係のみであり、既存の手法と異なり、頂点の座標情報を与えて多角形ネットを定義する必要がない。
- 演算の繰返しの回避；平面展開グラフにおける頂点間の隣接関係から得られる隣接行列をもとに変換行列を作成するが、この変換行列の性質を巧く利用することによって、1 回の行列演算で極限となる曲面を構成するすべての頂点の配置が得られる。具体的には、固定点の配置行列、変換行列の一部分などから成る近似的な連立一次方程式を LU 分解で解くことにより行う。ただし、近似による誤差は十分に小さく、コンピュータ画面への描画に何ら影響を与えない。この近似的連立一次方程式において、固定点の配置行列は定数項に現れるため、一度係数行列の LU 分解が求められれば、異なる定数項に関してもその結果を利用することができるので、固定点の位置をいつでも自由に変

え、それにもなって変形した曲面をすぐに得ることが可能である。

以下、2 章では最適埋蔵法について述べ、3 章では、この手法を利用して得られた曲面の例をあげ、4 章では、イタレーティブな演算による手法との比較結果より、最適埋蔵法の有効性を示し、5 章で結論を述べる。

2. 最適埋蔵法

本章では、最適埋蔵法について述べる。まず、2.1 節で最適埋蔵法を行うために必要な変換行列を定義したのち、2.2 節で近似的連立一次方程式による最適埋蔵法の非イタレーティブな行列演算による手法を示す。

2.1 変換行列

G を求める曲面の平面展開グラフとし、 n 個の頂点を持つとする。 G 内の頂点の集合を

$$V = \{v_1, v_2, \dots, v_n\}$$

とし、 V のうち、固定点、動点を

$$V_O = \{v_1, v_2, \dots, v_r\},$$

$$V_I = V - V_O = \{v_{r+1}, v_{r+2}, \dots, v_n\}$$

とそれぞれ表すことにする。 $(x_{v_i}^k, y_{v_i}^k, z_{v_i}^k)$ を行列演算を k 回行った後の v_i の座標とすると、 k 回目の行列演算後の V の各頂点の座標は、

$$C_V^k = \begin{pmatrix} X^k \\ Y^k \\ Z^k \end{pmatrix} = \begin{pmatrix} x_{v_1}^k, x_{v_2}^k, \dots, x_{v_n}^k \\ y_{v_1}^k, y_{v_2}^k, \dots, y_{v_n}^k \\ z_{v_1}^k, z_{v_2}^k, \dots, z_{v_n}^k \end{pmatrix}$$

で表される。このとき、 C_V^k から C_V^{k+1} への変換は、 n 次の正方行列 F を用いて、

$$C_V^{k+1} = FC_V^k \quad (1)$$

で表され、この F を G の変換行列と呼ぶ。変換行列 F の i, j 成分、 f_{ij} は、は以下で定義される。

$$f_{ij} = \begin{cases} \delta_{ij} & (1 \leq i \leq r) \\ a_{ij}/d(v_i) & (r+1 \leq i \leq n) \end{cases}$$

ただし、 δ_{ij} は、 $i = j$ のとき 1、 $i \neq j$ のとき 0 を示す。また、 a_{ij} は、グラフ G の隣接行列の i, j 成分、 $d(v_i)$ は、頂点 v_i の次数を表している。この変換行列 F による行列演算は、固定点の初期状態における位置を保ったまま、動点のみがそれぞれ最適な位置へと必ず収束するということが分かっている⁷⁾。

4 章では、次節で示す non-iterative な行列演算との処理速度の比較として、上記の変換行列によるイタレーティブな行列演算を用いることにする。

2.2 Non-iterative な行列演算

変換行列 F は、その定義から次のような 4 つのブロックから成ると考えられる。

$$F = \begin{pmatrix} E & O \\ B & A \end{pmatrix}$$

$$\begin{cases} E; & r \text{ 次}の単位行列 \\ O; & r \times (n - r) \text{ の零行列} \\ B; & (n - r) \times r \text{ 行列} \\ A; & (n - r) \text{ 次}の正方行列 \end{cases}$$

ここで、 $C_{V_O}^0 = (X_{V_O}^0, Y_{V_O}^0, Z_{V_O}^0)$ を固定点の集合 V_O の初期状態における各頂点の座標、また、 $C_{V_I}^\infty = (X_{V_I}^\infty, Y_{V_I}^\infty, Z_{V_I}^\infty)$ を動点の集合 V_I の各頂点が収束した位置の座標を表すとする。固定点の座標は繰返しの演算を通して不変であるので、この $C_{V_I}^\infty$ のみが求まればよい。 $C_{V_I}^\infty$ は、次のような近似的な連立一次方程式の解として求めることができる。

$$(E - A)C_{V_I}^\infty \simeq BC_{V_O}^0 \tag{2}$$

ただし式 (2) における E は、 A と同じサイズの単位行列を表しており、変換行列内の E とは異なる。式 (2) は、以下のようにして容易に導かれる。

C_V^k から C_V^{k+1} への変換は、式 (1) により行われることから、

$$\begin{aligned} C_V^{k+1} &= \begin{pmatrix} C_{V_O}^{k+1} \\ C_{V_I}^{k+1} \end{pmatrix} \\ &= \begin{pmatrix} E & O \\ B & A \end{pmatrix} \begin{pmatrix} C_{V_O}^k \\ C_{V_I}^k \end{pmatrix} \end{aligned} \tag{3}$$

V_O の各頂点は演算を通して不変なので、

$$C_{V_O}^{k+1} = C_{V_O}^k = \dots = C_{V_O}^1 = C_{V_O}^0 \tag{4}$$

式 (3), (4) より、

$$C_{V_I}^{k+1} = BC_{V_O}^0 + AC_{V_I}^k \tag{5}$$

$k \rightarrow \infty$ のとき $C_{V_I}^\infty$ の各頂点は必ず適当な位置に収束することが分かっているので、

$$k \rightarrow \infty, C_{V_I}^{k+1} \simeq C_{V_I}^k \simeq C_{V_I}^\infty$$

としてよい。よって式 (5) は、

$$C_{V_I}^\infty \simeq BC_{V_O}^0 + AC_{V_I}^\infty \tag{6}$$

となり、この式から式 (2) が得られた。

3. 最適埋蔵法の実行

前章で説明した最適埋蔵法による曲面は、曲面を構

成する頂点間の隣接関係と、いくつかの固定点の位置のみにより定義される。そのため、望ましい曲面を得るためには、その平面展開グラフをいかに巧く与えるかが、すべての処理を通して最も重要な処理となる。本章では、はじめに実行例として図 2 のような曲面をとりあげ、この曲面を得るための平面展開グラフについて述べ、最適埋蔵法の実行について説明する。その後、最適埋蔵法の一般的な曲面への適用について述べる。

3.1 平面展開グラフ

図 2 の曲面は、図 3 に示されてあるように八面体の 8 つの頂点を円にしたものである。また、各面においては、もともと正三角形の頂点であった部分が切り取られ、円弧の部分を含めると六角形に近い形になっている。そこで、平面展開グラフは、以下のように考えるとよい。

曲面は、8 個の六角形によって構成されると考える。六角形の内部の分割の仕方は、何通りも考えられるが、図 4 のような分割が対称性において最適である。また、当然のことながら、分割数を増やせば増やすほど滑らかな曲面が得られる。この内部が分割された六角形は、平面展開グラフの一部となる。次に、これら 8 個の六角形の貼り合わせを考えなければならないが、貼り合わせるといった概念は、貼り合わせる部分の頂点に同じ番号を付けていくことで実現されるので、平面展開グラフを完成させるには、その貼り合わせの部分

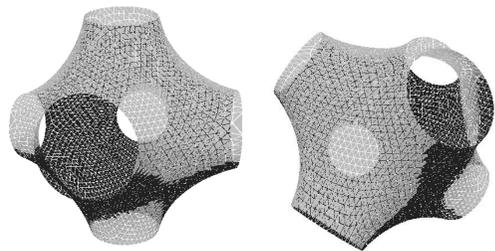


図 2 実行例
Fig. 2 An example.

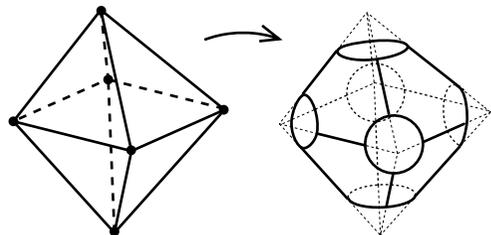


図 3 八面体の頂点を円にする
Fig. 3 Suppose each vertex of the octahedron is a circle.

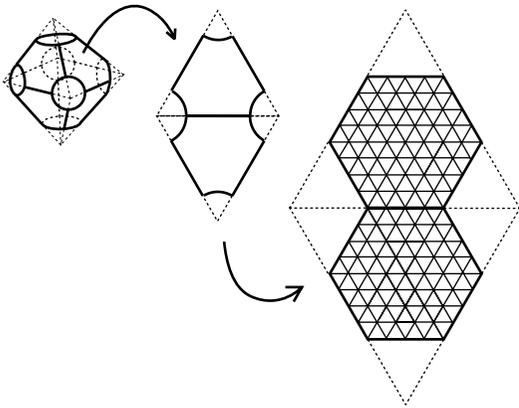


図 4 六角形内部の分割
Fig. 4 The division of hexagons.

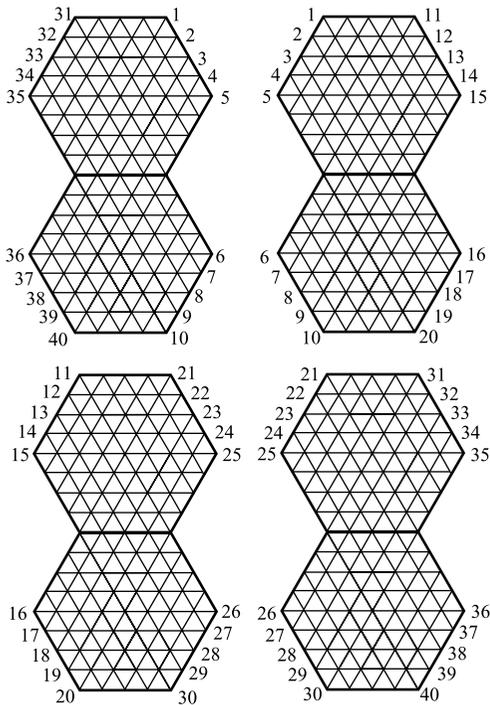


図 5 平面展開グラフ；番号は、各頂点の番号を示している（貼り付け部以外の頂点番号は省略）
Fig. 5 Planar unfolded graph; The numbers in the figure indicate the numbers of vertices.

に注意し、すべての頂点に番号を付けていけばよいだけである。ただし、この時点での番号付けは一時的なものであり、後で演算を行う際に、頂点番号の入れ替えが必要となる。図 5 は、図 2 の曲面の平面展開グラフを示している。

平面展開グラフが構成できた後、演算を実行するには、このグラフの隣接行列から変換行列を求めること

と固定点を指定することが必要である。まず、隣接行列については、頂点は六角形内部に規則的に配置されているので、容易に得ることができる。次に、固定点についてであるが、平面展開グラフの各成分を貼り合わせるにより生じる 6 つのループ上にある頂点を固定点として採用する。よって、平面展開グラフにおける六角形一辺上の分割数を m としたとき、固定点の数は、 $24m$ となる。ただし、この固定点のとり方は何通りも考えられ、固定点のとり方によって、動点はそれぞれ最適な場所へと移動する。固定点を多くとれば連立一次方程式における係数行列の大きさが小さくなるので、LU 分解などにおける計算量は減るが、それにもなって固定点の配置指定の手間がかかる。よって、この両側面を考慮し、この実行例の場合、この固定点のとり方が最もふさわしいものであるとした。

3.2 実行

まず、演算を簡単にするため頂点の番号を、以下のように入れ替え、固定点の初期位置を指定した後、演算を行う。

m を前述のように、六角形一辺の分割数で、 $r = 24m$ とするとき、6 つのループ上の頂点を

$$L_i = \{v_{4m(i-1)+1}, v_{4m(i-1)+2}, \dots, v_{4mi}\} \quad (1 \leq i \leq 6)$$

とする。その他の頂点、つまり動点に関しては上から順に再び番号をふる。よって、 n を平面展開グラフの頂点数とすると、

$$\begin{aligned} V_O &= \{v_1, v_2, \dots, v_r\} \\ &= \{v_1, \dots, v_{4m}, v_{4m+1}, \dots, v_{8m}, \\ &\quad v_{8m+1}, \dots, v_{24m}\} \\ V_I &= \{v_{r+1}, v_{r+2}, \dots, v_n\} \end{aligned}$$

となり、今後の演算においては、頂点はこの番号で扱われる。

固定点の初期位置は、6 つのループをそれぞれ 3 次元空間内のどの位置に配置するので考える。初期位置の変更は、前述のように容易に行えるので、初期位置を変えることによる曲面の微妙な調整が、いつでも可能である。図 2 は、 $m = 6$ 、6 つのループがおかれた平面上の円の中心は、 $L_1, L_2, L_3, L_4, L_5, L_6$ それぞれ $z = 3$ 上の点 $(0, 0, 3)$ 、 $z = -3$ 上の点 $(0, 0, -3)$ 、 $x = 3$ 上の点 $(3, 0, 0)$ 、 $y = 3$ 上の点 $(0, 3, 0)$ 、 $x = -3$ 上の点 $(-3, 0, 0)$ 、 $y = -3$ 上の点 $(0, -3, 0)$ として得られた実行結果である。

3.3 一般曲面への適用

最適埋蔵法を利用するための主な手順をまとめると以下ようになる。

- 曲面の想定；最適埋蔵法では、先に述べたように

固定点の配置と頂点間の隣接関係のみから曲面が定義される．一般的な曲面で利用する際には，特に固定点をどこにとるかが重要となり，まずはじめに曲面を想定すると同時に固定点をどのようにとるかある程度決めておく必要がある．

- 平面展開グラフの作成；平面展開グラフは，曲面をいくつかの多角形から成ると仮定し（必要である場合に）切り開いて何枚かの多角形に分割し，平面に展開する．その後，それら多角形の内部を分割しグラフとする．このとき分割の仕方は，たとえば図 4 のように対角線に平行な線で分割し格子状にしたものなど，ある程度規則性を持ったものである方が，後で隣接行列を作成しやすい．また，このグラフの頂点は分割により現れた頂点や，曲面境界上の頂点である．これらの分割によって各頂点の隣接関係が定まり，どの頂点が固定点であるか，動点であるかがすべてこの平面展開グラフ上で決定される．
- 演算の実行；平面展開グラフから得られる隣接行列から変換行列を求め，固定点の配置を設定し演算を実行する．

4. 最適埋蔵法の評価

本章では，2章で述べた最適埋蔵法の非イタレーティブな行列演算による手法（これを以下，手法 1 とする）とイタレーティブな行列演算による手法（これを以下，手法 2 とする）とを処理速度を 2 種類の方法で比較し，最適埋蔵法を評価する．具体的には手法 1 では，式 (2) を LU 分解を用いて解く方法で実行し，手法 2 は，式 (1) の演算を繰り返し行う方法をとっている．また，両手法とも描画に関しては，平面展開グラフで与えたメッシュ構造をそのまま反映させている．ただし，以下の処理時間はすべて，Pentium III 600 MHz，128 MB の Windows 2000 マシンで計測されたものである．

4.1 比較 1—単純描画

まず，図 2 の曲面を描画し，その処理時間を計測した．描画までの処理をまとめると以下ようになる．

- Step1 隣接行列の作成
- Step2 変換行列の作成
- Step3 固定点の配置設定
- Step4 それぞれの手法での行列演算の実行
- Step5 描画

Step4 以外の操作は 2 つの手法に共通したものであるので，Step4 にかかる処理時間を平面展開グラフの六角形内部の分割数を 3 段階， $m = 4, 8, 12$ に分け

表 1 処理時間比較 1—単純描画

Table 1 Comparison (1)—computation time for drawing.

m	4	8	12
頂点数	428	1628	3596
ポリゴン数	768	3072	6912
表面積	82.389091	81.877693	81.536362
手法 1 (秒)	0.46	56.73	739.20
手法 2 (秒)	4.06	186.32	1927.01
(演算回数)	(185)	(639)	(1358)

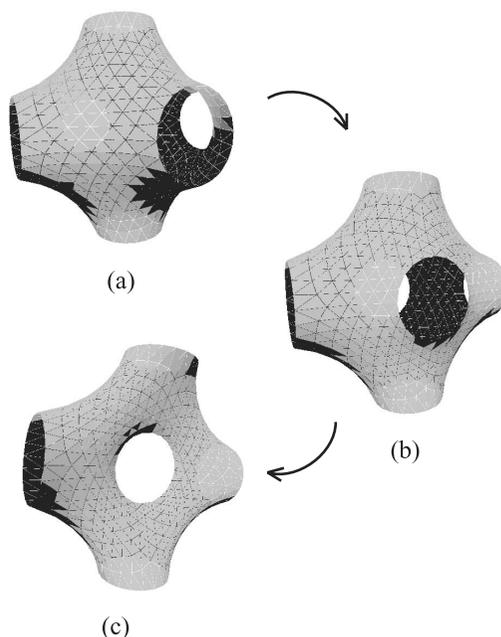


図 6 変形操作

Fig. 6 Process of modification.

て計測した．表 1 の上部は，分割数ごとの構成情報を示している．手法 2 における反復演算は，求める曲面の表面積と極限となる曲面の表面積との差が 10^{-6} 以下になると収束したと見なし演算を終了する．計測結果を表 1 に示す．

4.2 比較 2—変形処理

図 2 の曲面を変形させるときの処理時間の比較を行う．変形は，図 2 が得られた後，6 つの固定ループのうち L_1, L_2 をそれぞれ z 軸の負の方向に -0.3 ，正の方向に $+0.3$ ずつ移動させ重なった状態，つまり 10 度目の変形で終了する（図 6 (a) 初期状態，(b) 5 回変形後，(c) 終了状態）．変形処理をまとめると以下ようになる．

- Step1 隣接行列，変換行列の作成
手法 1 のとき LU 分解を行う
- Step2 固定点の配置設定
- Step3 それぞれの手法での行列演算の実行

表 2 処理時間比較 2—変形操作
Table 2 Comparison (2)—computation time for modification.

m	4		8	
	手法 1	手法 2	手法 1	手法 2
処理 1	0.49	0.07	57.30	1.00
処理 2	0.16	32.17	3.26	1614.92
処理 3	0.58	0.59	0.76	0.76
Total	1.23	32.83	61.29	1616.68

Step4 描画

Step5 固定点の配置を再設定し Step3 に戻る

計測は、隣接行列、変換行列作成などの前処理（処理 1）、固定点の配置、行列演算から座標データの取得のための処理（処理 2）、描画のための処理（処理 3）に分けて行い総処理時間を求める。分割数 $m = 4, 8$ において計測を行った結果を表 2 に示す。処理 2、処理 3 に関しては、10 回の変形にかかった総時間を示している。また、ここでの手法 2 における反復演算は、 k 回目と $k-1$ 回目の演算で得られた曲面の表面積の差が 10^{-6} 以下となった時点で収束したと見なす。

4.3 評価

表 1 より、両手法の処理時間の差は、頂点の増加にともなって若干小さくなっていくことが予想されるが、頂点数約 3600 個の場合においても手法 1 が手法 2 の約 2.6 倍の速さで曲面が求められているのが分かる。さらに変形のための処理時間は、一度係数行列の LU 分解を求めればその結果を用いた一度の演算で曲面が求められるため、表 2 で確認できるように手法 1 の方が明らかに高速であることが分かる。また、両手法での処理時間の差は頂点の増加にともなって次第に大きくなっていくことが予想される。

以上より、最適埋蔵法によって、イタレーティブな演算の極限として曲面を得るよりも高速に極限となる曲面が得られることが示された。また、変形における処理速度の差は頂点数が大きいものほど顕著であり、曲面を連続的に変形させて描画する際には非常に有効な手法であるといえる。

5. まとめ

本論文では、曲面を平面展開グラフとして扱い、構成する頂点間の隣接関係から変換行列を作成し、その変換行列の性質を巧く利用した非イタレーティブな行列演算により曲面を定義する、3 次元空間における曲面の最適埋蔵法を提案した。この手法では、イタレーティブな演算をせずに LU 分解を用いた近似的な連立一次方程式の解として極限となる曲面を構成する頂点

の配置が得られるので、一度その連立一次方程式の係数行列部分の LU 分解が求めると、異なる定数項に關してもそれを用いて解を得ることができ、固定点の位置を変えることによる連続的な曲面の変形が高速に行えることが確認された。

応用事例として、連続的な曲面の変形を要する 3 次元動画系生成などがあげられ、この手法を用いるることによって、計算量の削減が期待できる。

謝辞 LU 分解に関する貴重な助言をいただいた査読者に感謝申し上げます。

参考文献

- 1) Catmull, E. and Clark, J.: Recursively generated b-spline surfaces on arbitrary topological meshes, *Computer Aided Design*, Vol.10, pp.350–355 (1978).
- 2) Desbrun, M., Meyer, M., Schroder, P. and Barr, A.H.: Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow, *SIGGRAPH '99 Proceedings*, pp.317–324 (1999).
- 3) Guskov, I., Sweldens, W. and Schroder, P.: Multiresolution Signal Processing for Meshes, *SIGGRAPH '99 Proceedings*, pp.325–334 (1999).
- 4) Kobbelt, L., Campagna, S., Vorsats, J. and Seidel, H.P.: Interactive Multi-Resolution Modeling on Arbitrary Meshes, *SIGGRAPH '98 Proceedings*, pp.105–114 (1998).
- 5) Levin, A.: Interpolating nets of curves by subdivision surfaces, *SIGGRAPH '99 Proceedings*, pp.57–64 (1999).
- 6) Moreton, H.P. and Sequin, C.H.: Functional optimization for fair surface design, *SIGGRAPH '92 Proceedings*, pp.167–176 (1992).
- 7) 落合豊行, 山下 靖, 山田修司: 結び目理論研究支援システムの設計, 日本応用数理学会論文誌, Vol.4, No.4, pp.337–348 (1994).
- 8) Stam, J.: Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter, *SIGGRAPH '98 Proceedings*, pp.395–404 (1998).
- 9) Taubin, G.: A Signal Processing Approach to Fair Surface Design, *SIGGRAPH '95 Proceedings*, pp.351–358 (1995).
- 10) Zorin, D., Biermann, H. and Levin, A.: Piecewise Smooth Subdivision Surfaces with Normal Control, *SIGGRAPH '00 Proceedings*, pp.113–120 (2000).

(平成 12 年 8 月 24 日受付)

(平成 12 年 10 月 13 日再受付)

(平成 12 年 11 月 15 日採録)



今藤 紀子

1996年奈良女子大学理学部卒業．
1998年同大学大学院理学研究科（情報科学専攻）修士課程修了．2001年同大学大学院人間文化研究科（複合領域科学専攻）博士後期課程修了

見込．



落合 豊行

1969年横浜市立大学文理学部卒業．1971年東京大学大学院理学研究科修士課程修了．1973年日本ユニパック退社．1974年東京工業大学理学部助手．1980年大阪大学理学部講師．1984年同大学理学部助教授．1988年奈良女子大学理学部教授．1999年同大学大学院人間文化研究科教授．理学博士．

