

Wrinkly 曲面：手続き的補間によるパラメトリック曲面の形成

藤本 忠博[†] 大野 義夫^{††}

本論文では、制御点からパラメトリック曲面を形成する補間方法として、任意の手続きの適用を考える「手続き的補間 (procedural interpolation)」の概念を提唱し、その一例として「wrinkly 曲面 (wrinkly surface)」と呼ぶ曲面形式を提案する。一般に、従来のスプライン曲面は、幾何学的連続性 (C^n , G^n 連続性) を基準とした制御点の補間手続きにより形成される。一方、本論文で提案する wrinkly 曲面は、その補間手続きとして「反復交差変換 (Iterated Shuffle Transformation)」というパラメータ変換を適用する。この補間手続きは制御点を区分的に補間するものであり、wrinkly 曲面は空間方向の局所性とともにスケール方向の自己相似性 (「空間/スケール方向の局所的類似性」) を持つパラメトリック曲面となる。Wrinkly 曲面は従来のスプライン曲面とフラクタルとを融合したものであると見なすことができ、従来のスプライン曲面に比べて、特に自然物などの表現に適していると考えられる。本論文では、wrinkly 曲面が、空間方向での制御点の局所的な制御により、従来の大域的なフラクタル手法では生成が困難である形状を容易につくり出すことを可能とすることを述べ、実例を通してその有効性を示す。

Wrinkly Surfaces: Generating Parametric Surfaces by Procedural Interpolation

TADAHIRO FUJIMOTO[†] and YOSHIO OHNO^{††}

In this paper, we propose an idea of the “procedural interpolation”, which generates a parametric surface by specifying a procedure for the interpolation of control points. We also present a “wrinkly surface” as an example of the procedural interpolation. Usual interpolation gives a spline surface which satisfies some geometric continuity such as C^n or G^n . On the other hand, a wrinkly surface is generated by applying “Iterated Shuffle Transformation” to the interpolation. Such interpolation process generates a wrinkly surface that is a parametric surface which has locality in the space direction and self-similarity in the scale direction, that is, the “local resemblance in space/scale direction”. A wrinkly surface can be considered as a unification of a spline surface and a fractal surface. It is more suitable for the representation of natural objects than the spline surface. By moving control points in the space direction, the shape of the surface can be changed locally. We can thus obtain various surface models easily that are difficult to create by the global fractal method. Some examples show this advantage of a wrinkly surface.

1. 序 論

ある特定の特徴的な形状を容易に生成する方法に「手続き的モデリング (procedural modeling)」と呼ばれるアプローチがある^{2),3)}。これは、関数やルール等の手続きに基づいて目的とする形状を生成する方法である。代表的なものとして、フラクタル手法^{4)~11)}、パーティクルシステム^{12)~16)}、L-システム^{17)~20)}などがあげられ、このほかにも、一般のさまざまな形状ある

いはテクスチャを生み出すための方法が提案されている^{2),3),21)~25)}。これらの手法の最大の利点は、ある種の複雑な形状を簡単な記述によって効率的に生成可能とすることであり、パラメータの変更によりバラエティに富む形状の変更が容易に行えることも、‘database amplification’¹⁷⁾と呼ばれる長所となっている。

一方、3次元空間上に物体形状を作成する方法として、「スプライン曲面 (spline surface)」*は最も有効な方法の一つである。スプライン曲面は、3次元空間上に置かれた制御点からなる制御メッシュを滑らかに補

[†] 慶應義塾大学大学院理工学研究科

Graduate School of Science and Technology, Keio University

^{††} 慶應義塾大学理工学部

Faculty of Science and Technology, Keio University

* ここでは、Bézier 曲面や B-spline 曲面など、制御点の滑らかな補間により生成される曲面を総称して「スプライン曲面」と呼ぶこととする。なお、本論文での「滑らかさ」とは C^n , G^n 等の連続性を意味する。

間することで、パラメータ (u, v) によるパラメトリック形式として形成される。制御点の位置の移動により曲面の形状を自由に制御することが可能であり、この制御点を仲介した操作性は、陽/陰関数式による形状表現に比べて、ユーザにとってより直観的かつ自由度の高い形状モデリングを可能とする¹⁾。

本論文では、上記の「手続き的モデリング」と「制御点補間」とを融合した、「手続き的補間 (procedural interpolation)」の概念を提唱し、その具体的な例として「wrinkly 曲面 (wrinkly surface)」と呼ぶ曲面形式を提案する。手続き的補間は、従来までの曲面の連続性 (C^n , G^n 連続性) のみを基準とした補間だけではなく、制御点群から曲面を形成する方法としてさまざまな一般の手続きの適用を考えるものである。この「手続き的補間」は、複雑な形状を効率良く生み出す「手続き的モデリング」の豊かな表現力、「制御点補間」の柔軟な操作性、という両者の利点をあわせ持つものである。

Wrinkly 曲面の補間に適用する「反復交差変換」²⁶⁾ という手続きは、曲面に対して「スケール方向の自己相似性」というフラクタル的な効果を生み出す。これは、従来のスプライン補間に「コントロール集合」の考えを取り入れたものにとらえられる。Wrinkly 曲面は、従来のスプライン曲面と同様に、3次元空間上の制御メッシュ上に (u, v) に関するパラメトリック曲面として形成される。各制御点は (x, y, z) の位置座標値の他に、変位のための高さ値や色、透過率などの属性値を持つ。これらの属性値に対して「反復交差変換」を適用し、色などの属性値も含めた ‘displacement mapping’ を行うことで、「空間/スケール方向の局所的類似性」²⁶⁾ を有する曲面形状を生成する。図6にみられるように、wrinkly 曲面はスケール方向の視野移動 (拡大/縮小) に関して自己相似的な特性を持ち、空間方向の視点移動に関しては位置による局所性を持つ。この効果は通常の人域的なフラクタル手法では生み出すことができない。なお、displacement mapping の変位による凹凸が wrinkly 曲面の本質であることから、その名前に「しわになりやすい」という意味を持つ “wrinkly” という言葉を用いた。

本論文では、以下、2章で関連するこれまでの研究の紹介、3章で手続き的補間の提唱、4章で反復交差変換の説明、5章で wrinkly 曲面の提案、6章で実例の提示、7章で結論を述べる。

2. これまでの研究

Wrinkly 曲面は、フラクタルな凹凸の形成により山脈地形を生成する中点変位法^{2)~4),7),10),11)}、および、

縮小変換の繰返しにより自己相似的な形状を生成する IFS 法^{5),6),9)}とから基本的なアイデアを得ている。

スプライン曲面とフラクタルを融合するという試みとしては、Szeliski と Terzopoulos による研究¹¹⁾がある。文献11)では互いに相補的な性質を持つ決定論的スプラインと確率論的フラクタルとを融合した制約的フラクタル (constrained fractal) を提案している。この手法では、変形エネルギー関数の関数値が最小となる曲面式として定義される連続性制御スプライン (controlled-continuity spline) をもとに、その関数値を最小化する反復処理の過程で適切なノイズを加えることでフラクタル形状を生成する。結果として、初期データとして与えた離散的な高さ情報からフラクタル的な山脈地形が生成され、さらに、連続性制御スプラインの特性により、空間方向について局所的に異なる連続性の制御が実現されている。

本論文で提案する wrinkly 曲面は、スプライン曲面とフラクタルの融合を目指すという点では文献11)と同様であるが、その具体的な目的、実現方法および生み出される効果はまったく異なるものである。

3. 手続き的補間

制御点の補間による2次元曲面の生成という操作は、離散的な制御点の集合から連続的な曲面形状をつくり出すことである。制御点の集合は、通常、それらの接続情報をとめない制御メッシュとされる。制御メッシュ情報は $M = (C, P)$ と表すことができる。ここで、 C は制御点の接続状態に関する情報であり、 $P = \{\mathbf{p}_i\}$ は制御点 \mathbf{p}_i 自身の情報である。 \mathbf{p}_i は、具体的には、その制御点の位置座標値や属性値などのデータである。生成される曲面がパラメトリック曲面である場合には、 C を各制御点 \mathbf{p}_i のパラメータ位置の情報とすると、 $C = \{(u_i, v_i)\}$ となる。そして、補間により生成される曲面式を

$$\mathbf{p} = \mathbf{f}(u, v) \quad (1)$$

とすると、補間の働きは M から \mathbf{f} を形成することであり、以下のように示すことができる。

$$H_{Intp} : M \rightarrow \mathbf{f} \quad (2)$$

従来の連続性を基準とした補間では、 C の情報に基づいて P の値を連続的に滑らかに補間する。本論文で提案する「手続き的補間」は、この補間方法 H_{Intp} として任意の手続きを考えようというものである。

ここで、話を分かりやすくするため、1次元の値を補間することを考える。パラメトリック形式の場合を考えると、 $C = \{u_i\}$ および $P = \{x_i\}$ であり、制御点としてパラメータ u_i に対応してデータ値 x_i が与えら

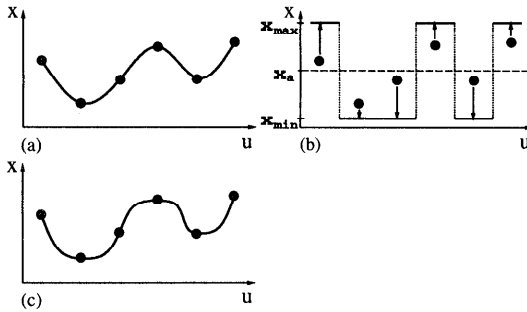


図1 手続き的補間の例: 黒丸は制御点, 太線が関数 $x = f(u)$ を表す. (a) 3次スプライン補間. (b) 二値化. x_i が閾値 x_a より大きい小さいかにより, $f(u)$ の値は定数値 x_{\min} あるいは x_{\max} のいずれかをとり. (c) (a) に対するパラメータ変換

Fig. 1 Examples of procedural interpolation.

れる. そして, 補間により以下の関数式が求められる.

$$x = f(u) \quad (3)$$

図1は異なる手続きによる補間の例である. 図1(a)は3次スプラインによる補間であり, 各制御点が連続的に滑らかに補間されている典型的な例である. 一方, 図1(b)はある閾値による二値化であるが, これも制御点の補間の1例と見なすことができ, 生成される関数式(3)は不連続なものとなる.

一方, 図1(c)は図1(a)のパラメータ u 軸方向に変化を加えたものである. このとき, 図1(c)のパラメータを u , 図1(a)のパラメータを u' とした場合に, それらの間に以下の関係があるものと考えることができる.

$$u' = \mathcal{F}_p(u) \quad (4)$$

すると, 図1(a)の関数式が式(3)である場合には, 図1(c)の関数式は以下のように表すことができる.

$$x = f(\mathcal{F}_p(u)) \quad (5)$$

補間により生成される関数式がパラメトリック形式である場合, 上記の式(4)で示されるようなパラメータ変換を補間の過程に取り入れることは, 任意の手続き的補間を考えるうえで有効な手段の1つである. そして, 図1(a)と図1(c)の間にみられるような連続的なパラメータ変換と同様に, 不連続的な変換というものも考えられる.

以上の議論は2次元あるいはそれ以上の次元に対して容易に拡張できる. 本論文で提案する wrinkly 曲面はパラメトリック形式で表現される曲面であり, 4章で述べる不連続的なパラメータ変換により特徴的な形状を生み出すものである.

4. 反復交差変換

Wrinkly 曲面を生成する手続き的補間には, 著者ら

が文献26)で提案した「反復交差変換 (Iterated Shuffle Transformation: IST)」をパラメータ変換として適用する. 本章では, この IST について説明する.

4.1 反復交差変換の定義

IST は2つの座標軸 u_s と u_d の間に,

$$u_d = \mathcal{F}_{IST}(u_s) \quad (6)$$

で示される対応関係を構築する変換であり, 変換式 \mathcal{F}_{IST} は不連続的な関数となる. その変換手続きを以下に示す (図2).

(1) 初期設定

- 1) 座標軸上の区間 $[a, b]$ 上に N 個のベース区間 $A_i = [a_i, a_{i+1})$ を設定する ($i = 0, \dots, N-1$, $a_0 = a$, $a_N = b$).
- 2) 各ベース区間 A_i に対して交差区間数 M_i を設定する.
- 3) 各ベース区間 A_i に関するリスト L_i^0 をつくり, それぞれのベース区間自身を小区間として格納する. $L_i^0 \leftarrow \{A_i\}$

(2) 反復処理

反復回数 K を決め, 以下の処理を $k = 1, \dots, K$ について繰り返す.

- 1) 各ベース区間に関するリスト L_i^{k-1} 中に格納されている各小区間を $2M_i + 1$ 個に分割する.
- 2) 1) で分割したものを自分自身と両脇 M_i 個の計 $2M_i + 1$ 個のベース区間に振り分け, それらを新たなリスト L_i^k 中に格納する.

上記の手続きを図示したものが図2である. ここで, 図2中のはじめの d^0 軸を u_s 座標軸, K 回の反復後の d^K 軸を u_d 座標軸として, $u_s - u_d$ 間の座標値の対応関係により \mathcal{F}_{IST} が定義される.

上記の手続きにより, 座標軸上の小区間は反復ごとに細かくなり, もとの座標値が周囲に拡散していく. また, \mathcal{F}_{IST} は, u_s と u_d の座標値を 1:1 対応の関係で対応付ける*. そして, 反復が増すごとに \mathcal{F}_{IST} の不連続度が増していく.

IST を適用する座標軸のタイプとして, 軸の両端が開放されている「並進軸」と両端が閉じている「巡回軸」の2種類が考えられる**. 図3のように, これらは端点での変換に違いがみられる.

また, \mathcal{F}_{IST} の具体的な計算は, 以下のように行う.

- 1) 小区間の分割は, それぞれ振り分けられるベース

* 小区間の「つなぎ目」の値は後方の小区間に含まれるものとする.

** 本来, 「並進」および「巡回」は群論における用語であるが, 座標軸の性質を表すものとしてこれらの用語を用いる.

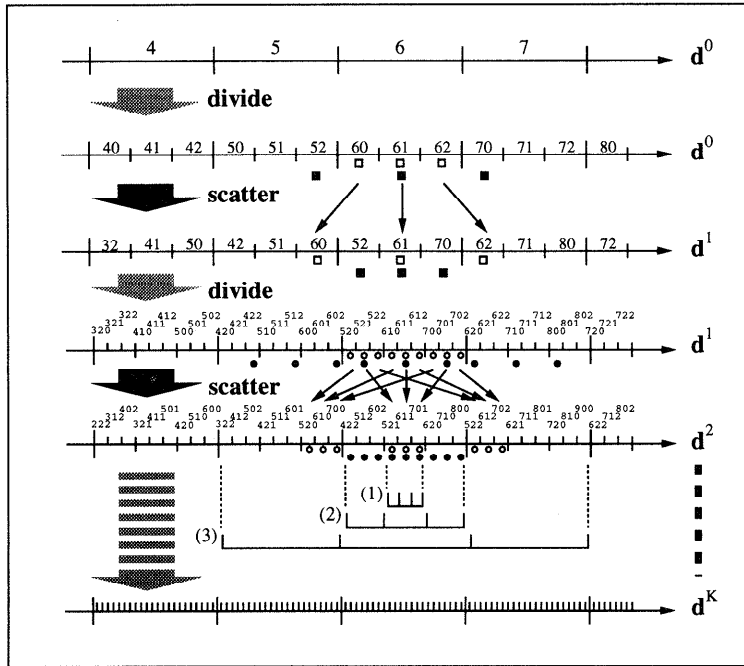


図2 反復交差変換

Fig. 2 Iterated Shuffle Transformation.

K 回の反復処理により、座標軸が $d^0 \rightarrow d^1 \rightarrow d^2 \rightarrow \dots \rightarrow d^K$ のように変換される。この例では、すべてのベース区間で交差区間数は 1 である。最上図の軸上の 4 ~ 7 は各ベース区間に付けた番号であり、左右にさらに続くものとする。また、小区間を表すための番号付けは、 $6 \rightarrow \{60, 61, 62\} \rightarrow \{600, 601, 602\}, \{610, 611, 612\}, \{620, 621, 622\} \rightarrow \dots$ という具合に、分割に従って下位の方向に n 進数で表記していくものとする (この例では $n = 2 \cdot 1 + 1 = 3$ である)。図中の白黒の四角および丸印は分割された各小区間が変換によりどこに移るかを表す。

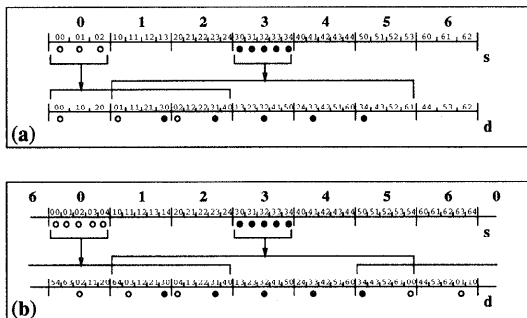


図3 並進軸と巡回軸：すべてのベース区間で交差区間数を 2 とした場合。(a) 並進軸。(b) 巡回軸

Fig. 3 Translation coordinate axis and cyclic coordinate axis.

区間の長さ に比例した比率で行う。

2) K 回反復後の u_s 値と u_d 値の対応は、各ベース区間 A_i ごとに、座標区間 $[a_i, a_{i+1})$ とリスト L_i^K に含まれる小区間のそれぞれの座標区間とを線形に対応させる。

4.2 カントール集合との関連

図 4 (a) はカントールの 3 進集合を表す。カントール集合は、座標区間 $[0, 1]$ に対して 2 つの縮小写像 $w_0(x) = \frac{1}{3}x$, $w_2(x) = \frac{1}{3}x + \frac{2}{3}$ を繰り返し適用することで、そのアトラクタとして得られるものである。ここで、さらにもう 1 つの縮小写像 $w_1(x) = \frac{1}{3}x + \frac{1}{3}$

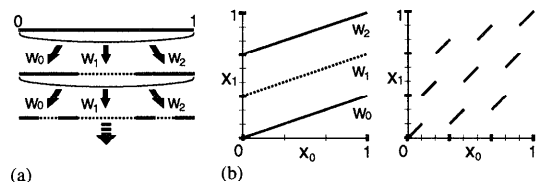


図4 カントール集合：(a) カントール集合をつくる縮小写像。(b) 1 回目の反復による座標変換。1 回目の反復で区間 $[0, 1]$ 中の座標 x_0 が x_1 に移る。左図：カントール集合。右図：反復交差変換 (並進軸の区間 $[0, 1]$ 上の 3 つのベース区間 $[0, \frac{1}{3}]$, $[\frac{1}{3}, \frac{2}{3}]$, $[\frac{2}{3}, 1]$ に対して交差区間数をすべて 2 とした場合を考える)

Fig. 4 Cantor set.

を導入すると、3 つの縮小写像 w_0, w_1, w_2 は軸上を隙間なく埋める。

IST はこのカントール集合の生成手続きをヒントにしたものであるが、以下の 3 点に関して異なる。

- a) 縮小写像の適用ではなく、小区間を細分割することで座標軸上での座標値の混ぜ合せを行っており、もとの座標軸と反復後の座標軸の間で 1:1 の座標値の対応付けが可能となる (図 4 (b))。
- b) カントール集合では 3 つの縮小写像を区間 $[0, 1]$

☆ この 3 点の他に、IST では任意の交差区間数が選択可能であるという違いがあるが、これはカントール集合の縮小写像の数を任意にする (任意の n 進集合を許す) ことに相当する。

全体に大域的に適用しているが、ISTでは、図2のように、区分的、すなわち、座標値の混ぜ合わせを局所的に行っている \star 。

- c) ISTは必ずしも反復 ∞ 回のアトラクタとして定義されるわけではなく、任意の反復回数の下で定義される。

4.3 空間/スケール方向に関する特性

ISTは座標軸上に「空間/スケール方向の局所的類似性」の性質をつくり出す。

「空間方向」とは「縦、横、高さ」のような「視点」の移動に関する方向性であり、「スケール方向」とは「拡大、縮小」という「視野」の移動に関する方向性である。本論文では、この2つの方向性は形状をはかるうえで重要な尺度であると考えられる。

ISTによって構築される座標軸上の小区間の不連続的な並び方は、空間方向とスケール方向の双方に関するわずかな移動の下で類似した並び方を維持する(図2)。この性質が「空間/スケール方向の局所的類似性」であり、空間方向の連続性とスケール方向の自己相似性とを融合した概念であると考えられる²⁶⁾。通常のフラクタルはスケール方向の移動の下での自己相似性のみを実現し、その自己相似性は空間方向に関して大域的であるのに対して、ISTは空間方向に関する局所性を持った自己相似性を実現する(4.2節の項目b)に関連)。ただし、この自己相似性にはスケール変化の下での不変性といった従来どおりの数学的な厳密さはなく、その意味では厳密なフラクタルではない。よって本論文では「類似性」という言葉を用いている。

図2の d^2 軸上の小区間番号の並び方について、図2中に示した(1), (2), (3)の範囲ごとに、3桁の小区間番号の最上桁の数字に注目すると、各範囲内で3つに区切られたそれぞれの区間は、互いにその数字が1ずつずれたものとなっていることが分かる $\star\star$ 。たとえば、(3)の範囲では、最上桁が $\{\{3, 4, 5\}, \{4, 5, 6\}, \{5, 6, 7\}\} \rightarrow \{\{4, 5, 6\}, \{5, 6, 7\}, \{6, 7, 8\}\} \rightarrow \{\{5, 6, 7\}, \{6, 7, 8\}, \{7, 8, 9\}\}$ であり、 \rightarrow の前後で各数字が1だけ増えている。この関係は任意の反復回数 k における d^k 軸上で成り立つ $\star\star\star$ 。これは、ISTが「空間/スケール方向の局所的類似性」の構造を生み出していることを具体的に示している。

\star ただし、図4(b)右図ではコントロール集合との比較のため交差区間数を2として大域的にしている。

$\star\star$ 最上桁以外の数字についても、各範囲ごとに規則性(ただし、最上桁の規則性とは異なる)が保たれている。

$\star\star\star$ 図2はすべてのベース区間の交差区間数を1としたケースであるが、一般にすべてのベース区間の交差区間数を共通とした場合、任意の交差区間数について同様の性質が成り立つ。

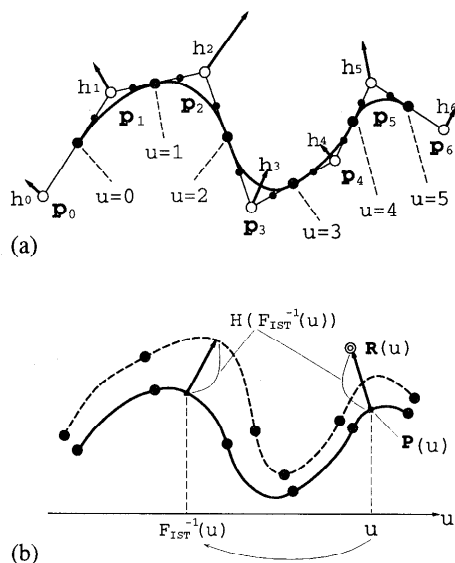


図5 Wrinkly 曲線: (a) 白丸は wrinkly 制御点 p_i の位置、白丸から出る矢印は高さ値 h_i を表す。大小の黒丸は Bézier 制御点であり、これによる3次 Bézier 曲線(図中の太曲線)群が基準線となる。 $u=i$ が各 Bézier 曲線の接続部分でのパラメータ値となる。(b) 太実線が基準線 $P(u)$ 、太点線が高さ関数 $H(u)$ を表す。点 $R(u)$ が wrinkly 曲線上の点となる。

Fig. 5 Wrinkly curve.

5. Wrinkly 曲面

本章では、スプライン曲面とフラクタルとを融合した、空間/スケール方向の局所的類似性の特性を有する「wrinkly 曲面 (wrinkly surface)」の提案を行う。

5.1 Wrinkly 曲面の定義

2次元の曲面を扱う前に、はじめに1次元の曲線(wrinkly 曲線と呼ぶ)について説明する(図5)。

Wrinkly 曲線は、従来のスプライン曲線と同様に、制御点からなる制御ポリゴン上に形成される。2次元空間上で考えた場合、各制御点は (x, y) の位置座標値と変位のための高さ値を持つ。ここで、後の説明を分かりやすくするため、この制御点を「wrinkly 制御点」と呼ぶこととする。

まず、wrinkly 曲線のベースとなる基準線を規定する。これは、個々の wrinkly 制御点に対応する3次 Bézier 曲線をつくり、それらを G^1 連続になるように接続することで決定される。具体的には、図5(a)のように、各 wrinkly 制御点 p_i を結ぶ線分上に適切な Bézier 制御点をとることで Bézier 曲線をつくる $\star\star$ 。

$\star\star$ Wrinkly 曲線にとって、この基準線の生成や後述する高さ関数等の定義に必ずしも Bézier 補間を用いる必要はないのであるが、使いやすさという理由で Bézier 補間を採用した。

また、これにともない、wrinkly 曲線のパラメータ化を行う。任意のパラメータ化が可能ではあるが、本論文では、簡単のため、一様なパラメータ化を行う。すなわち、図 5 (a) のように、wrinkly 制御点 \mathbf{p}_i に対応する Bézier 曲線が $u \in [i-1, i)$ のパラメータ範囲をとるように wrinkly 曲線全体をパラメータ化する。これにより、パラメータ u に関して区間 $[i-1, i)$ ごとに区分的に基準曲線上の基準点 \mathbf{p} の位置が決まる。

$$\mathbf{p} = \mathbf{P}(u) = \{\mathbf{P}_i(u)\} \quad (7)$$

また、上記の基準曲線と同様の方法で、各 wrinkly 制御点の高さ値 h_i をもとに、パラメータ u での高さ値 h を規定する区分的な高さ関数 H を定義する。

$$h = H(u) = \{H_i(u)\} \quad (8)$$

続いて、この高さ関数に基づく displacement mapping を行う。この際に、上記で設定したパラメータ軸 u に対して IST を適用する。パラメータ軸上で各 wrinkly 制御点 \mathbf{p}_i に対応する区間 $[i-1, i)$ を IST のベース区間 A_i とする。そして、すべてのベース区間に共通の交差区間数、すなわち、wrinkly 曲線全体に対して 1 つの交差区間数 M を与える*。これは、各 wrinkly 制御点の曲線に対する局所的な影響範囲に関する尺度という意味で、従来のスプライン曲線の次数に相当するものととらえることができる。また、IST の反復回数は 5.2 節に示す方法で決定する。パラメータ値 u に対する wrinkly 曲線上の点 \mathbf{r} は、次のように求められる。まず、基準曲線上の基準点 \mathbf{p} を式 (7) から求める。そして、変位する高さとして以下の高さ値を求める。

$$h = H(\mathcal{F}_{IST}^{-1}(u)) \quad (9)$$

ここで、基準点 \mathbf{p} から基準曲線の法線方向に式 (9) の高さ h 分だけ移動した点を wrinkly 曲線上の点として定義する。すなわち、wrinkly 曲線上の点 \mathbf{r} は、

$$\begin{aligned} \mathbf{r} &= \mathbf{R}(u) \\ &= \mathbf{P}(u) + H(\mathcal{F}_{IST}^{-1}(u))\mathbf{N}(u) \end{aligned} \quad (10)$$

と求められる (図 5 (b))。ただし、 $\mathbf{N}(u)$ は基準曲線上のパラメータ u における単位法線ベクトルを表す。

以上の wrinkly 曲線は、容易に 2 次元の wrinkly 曲面に拡張可能である。この拡張は、通常のスプライン曲線をスプライン曲面に拡張するのと同様の方法で行われ、3 次元空間上の wrinkly 制御点からなる制御メッシュ上に wrinkly 曲面が形成される。この場合、2 つのパラメータ u, v により曲面が決定され、 u, v 軸のそれぞれに対して別々に IST が適用される。Wrinkly

曲面上の点 \mathbf{r} は以下の式で求められる。

$$\begin{aligned} \mathbf{r} &= \mathbf{R}(u, v) \\ &= \mathbf{P}(u, v) + H(\mathcal{F}_u^{-1}(u), \mathcal{F}_v^{-1}(v))\mathbf{N}(u, v) \end{aligned} \quad (11)$$

ここで、 $\mathbf{P}(u, v)$ は基準曲面上の点、 $\mathbf{N}(u, v)$ は基準曲面上の単位法線ベクトルを表す。また、 $\mathcal{F}_u, \mathcal{F}_v$ はそれぞれ u, v 軸に適用する IST の変換式とする。Wrinkly 曲面の例を図 6 (p.10 参照) に示す。

以上から分かるように、wrinkly 曲面の変位の与え方は、中点変位法のものとは異なり、基準曲面上の各点が属性として持つ高さ値を IST の変換に基づいて面方向に移動させることで変位の大きさを決定する (図 5 (b))。

以上の説明では高さ値に対する IST の適用についてのみ述べたが、図 6 のように、IST によるパラメータ変換で曲面上を移動させることができるのは高さ値だけではなく、色あるいは透過率などあらゆる属性値に対して IST を適用することが可能である。

5.2 サンプリング処理

5.1 節で定義した wrinkly 曲面をスクリーン上に表示するために、wrinkly 曲面をポリゴン群に変換し Z バッファ法等による通常のレンダリングを行う。このためには、wrinkly 曲面に対して適切なサンプリングを行い、ポリゴンの頂点データを取り出すことが重要である。生成されるポリゴン群は以下の条件を満たさなければならない。

- スクリーン上に投影される大きさが適切であること。
- Wrinkly 曲面の持つ「空間/スケール方向の局所的類似性」を適切な形で維持していること。
- 視点の位置や視点からの距離によって曲面全体の見た目が変わってしまうという不都合を生じないこと。

そこで、以下の方法でサンプリングを行う。

まず、条件 a) を満たすように IST の反復回数 K を決定する。図 2 に示すように、反復ごとにパラメータ軸上での小区間幅は小さくなっていくが、このパラメータ幅に対応する実空間上での距離 (各小区間により異なる) をスクリーン上に投影した場合に、すべての小区間のパラメータ幅がスクリーン上で 1 ピクセル以下になるまで反復を行うこととし、その反復回数を K とする。

ここで、仮にパラメータ空間上で上記の K 回反復後の各小区間ごとに 1 点ずつのサンプリングを行ったとすると、曲面上のほとんどの位置でサンプリングが必要以上に細かくなってしまふ。そこで、以下に示すような「適応的サンプリング」を行う。これは、各

* 一般の IST では各ベース区間 A_i が異なる交差区間数 M_i を持つことを許す。

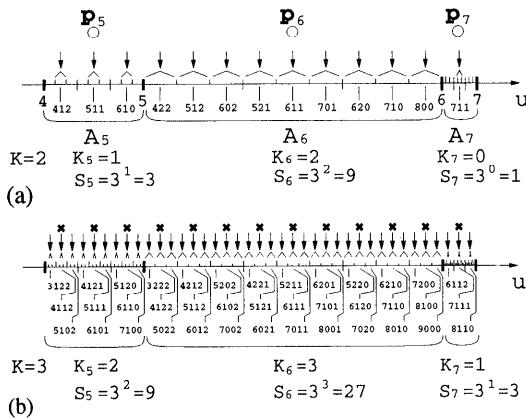


図7 Wrinkly 曲線のサンプリング: 交差区間数 $M=1$ とする。 A_i は制御点 p_i に対するパラメータ区間 $[i-1, i)$, K_i は区間 A_i での反復レベルを表す。軸下の3桁 ((a)) および4桁 ((b)) の数字は図2と同様の小区間番号を表す(ただし, サンプリングする小区間の番号のみ表記している)。サンプリングを矢印の小区間内で行う。なお, この図ではパラメータ区間長を実空間での距離に合わせたものとしている。
(a) 反復回数 $K=2$ の場合。(b) 反復回数 $K=3$ の場合。 \times 印のところでサンプリングされる小区間の番号は(a)でサンプリングされている小区間の番号と類似している(特に最上桁は等しい)ため, 反復回数 K が変わった場合にも曲線(2次元の場合には曲面)の見た目は不自然な変わり方をせず, さらに, \times 印の両脇でサンプリングされる小区間とともに1階層深い「空間/スケール方向の局所的類似性」の構造を形成する。

Fig. 7 Sampling of wrinkly curve.

wrinkly 制御点に対応するパラメータ区間ごとに最適なサンプリングを行うというものであり, 上記の条件 a), b), c) のすべてを満足させる方法である。

再び1次元の wrinkly 曲線の場合について説明する(図7)。まず, 各パラメータ区間 $A_i = [i-1, i)$ ごとに最適な反復レベル K_i を決定する。上記の反復回数 K の決定と同様の方法で, 各パラメータ区間 A_i ごとにそれぞれの小区間に対応する実空間上での距離をスクリーン上に投影した幅がすべて1ピクセル以下となる反復回数として K_i を求める。当然, K_i の最大値が K に等しくなる \star 。

$$K = \max_i \{K_i\} \quad (12)$$

反復レベル K_i が小さな場合にはパラメータ区間 A_i がスクリーン上に与える距離が小さく, K_i が大きな場合にはその距離は大きい。よって, K_i が大きくなるに従ってサンプリング数を増やす必要がある。ISTの反復回数そのものは K であるから, 交差区間数を M とすると, すべてのパラメータ区間 A_i が i によ

らず以下に示す S 個の小区間を持つ。

$$S = (2M+1)^K \quad (13)$$

パラメータ区間 A_i に関して, この S 個のうちから次に示す S_i 個の小区間でサンプリングを行う。

$$S_i = (2M+1)^{K_i} \quad (14)$$

そして, サンプリングを行う具体的な小区間は, 図7に示すように, $\frac{S/S_i+1}{2}$ 番目の小区間から始めて S/S_i 個おきにとっていく。この方法により, 図7に示すように, パラメータ区間 A_i ごとにスクリーン上での距離に応じた数のサンプリングが行われる(条件 a))。さらに, サンプリングされる小区間は, 各パラメータ区間 A_i ごとに「空間/スケール方向の局所的類似性」を適切な形で維持している。すなわち, 図7でみられるように, サンプリングされる小区間の区間番号の並び方が4.3節で述べた構造を維持している(条件 b))。また, 視点の位置の変更等によりスクリーンと曲面との位置関係が変わる場合には, ISTの反復回数 K が変更される。このとき, 各パラメータ区間 A_i に対する反復レベル K_i も変更され, サンプリングする小区間の個数 S_i も変わる。しかし, 上記のサンプリング方法に従えば, 図7(a)と(b)に示すように, サンプリングされる小区間の番号が反復回数 K の値ごとにまったく無関係に変わってしまうといった不都合を生じることなく, 「空間/スケール方向の局所的類似性」の規則性を維持しながらサンプリングが行われる(条件 c)) $\star\star$ 。

小区間内での具体的なサンプリング点の位置は, サンプリング対象となる小区間内である限りは任意の点でよい。小区間の中央の点を選んだ場合には規則的なサンプリングとなり(図8(a)), ランダムな位置を選ぶことで「ゆらぎ」を与えることも可能である(図8(b))。また, 反復回数 K を余分に増やすことで, 「空間/スケール方向の局所的類似性」の構造そのものに対して「ゆらぎ」を与えることができる(図8(c))。

Wrinkly 曲面のサンプリングは, 以上に示した方法を2次元に拡張して行われる。パラメータ空間 (u, v) 上での適応的サンプリングで得られた頂点を結ぶことで, 上記の条件 a), b), c) を満たす適切なポリゴン群が求められる。

5.3 Wrinkly 曲面の特性

Wrinkly 曲面は IST による不連続的な変位を本質とするため, 曲面が実空間上で通過する正確な点を(wrinkly)制御点によって規定することは難しい。し

$\star\star$ 図7では交差区間数 $M=1$, 反復回数 $K=2,3$ の場合についてのみ示しているが, 一般に任意の M の下でいかなる K の値に対しても上記が成立する。

\star 実際には, K を決定する過程で K_i が求められる。

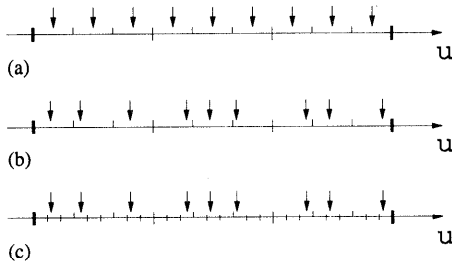


図8 サンプリング点の位置とゆらぎ: 図7の区間 A_6 の部分を抜き出したものとする。視点と曲面との位置関係から反復回数 $K = 2$, 反復レベル $K_6 = 2$ が求められたとする。このとき, $S_6 = 3^2 = 9$ 個のサンプリングをそれぞれ矢印の位置で行う。(a), (b) では $K = 2$ のまま, (c) では反復を1回余分に行い $K = 3$ とする。サンプリングされる小区間の番号の最上桁に注目すると, (a), (b) では $K = 2$ であるため図7(a)より $\{\{4, 5, 6\}, \{5, 6, 7\}, \{6, 7, 8\}\}$ という規則的な番号の並び方が維持される。一方, (b) と (c) ではサンプリングの位置は同じ(サンプリング対象となる各小区間でランダムな位置)であるが, (c) では $K = 3$ であるため図7(b)より $\{\{4, 4, 6\}, \{6, 6, 6\}, \{7, 7, 9\}\}$ という並び方となり「空間/スケール方向の局所的類似性」の規則的な構造が崩される(構造そのものに対する「ゆらぎ」)。

Fig. 8 Positions of sampled points and randomness.

かし, 確実に通過する点が各制御点に対して1つだけ存在する。それは, パラメータ区間 $[i-1, i)$ 上で中央のパラメータ値 $u = i - \frac{1}{2}$ に対応する点である。この点はISTの反復の下でつねに中央の小区間に位置し不動である(図2)。このパラメータ値に対応するwrinkly曲面上の点 q_i を「固定点 (fixed point)」と呼ぶことにする。

Wrinkly曲面は従来のスプライン曲面と同様に制御点により形成され, 空間方向に関する局所性を持つが, 制御点と曲面形状との関係が従来のスプライン曲面とはまったく異なる。スプライン曲面では, 曲面上の各点が影響を受ける近傍の制御点の個数はスプラインの次数により決まっており, 曲面上のどの点でも一定である。一方, wrinkly曲面では, 5.2節の適応的サンプリングに従うとすると, その個数は3次元空間上での制御点の配置および視点の位置に依存することになる。簡単のため, パラメータ区間 $[i-1, i)$ が制御点 p_i からのみ影響を受けると考えると*, 図7(a)の場合, 中央の区間 A_6 では $p_4 \sim p_8$ の5個の制御点から影響を受けるのに対して, 区間 A_5 では $p_4 \sim p_6$ の3個の影響のみであり, さらに区間 A_7 では p_7 のみから影響を受けることが, サンプリングされる小区間の番号から判断できる。すなわち, 曲面に対する制御

点の影響に関して, 制御点の配置が実空間上で疎である部分はより広い周囲からの影響を受け, 密である部分は影響を受ける範囲が狭い, ということがいえる。つまり, 制御点が密になるほど空間方向の局所性が増すということである。

上記のことから, wrinkly曲面を用いて形状モデリングを行う場合には, 従来のスプライン曲面によるやり方とはまったく異なる考え方が必要であることが分かる。スプライン曲面では, はじめに少ない制御点でおおまかな滑らかな形状をつくり, 次第に制御点を増やすことで複雑な形状を形作っていく。一方, wrinkly曲面では, 制御点の数が疎であるうちは広範囲からの影響のため一般に変位が激しく, 制御点が密になるにつれて各制御点の周囲への影響が狭められ, 各制御点の固定点を中心にして変位が固定化されていく(図9(a2), (d2))。

Wrinkly曲面は, 以下の2つの理由から厳密なフラクタルではない。1つの理由として, wrinkly曲面の「空間/スケール方向の局所的類似性」は曲面の面方向に関して構築される性質であり, 高さ方向については考慮されていない, ということがあげられる。5.1節で述べているように, 変位の高さ値を(色や透過率等と同様な)ISTによって曲面上を移動する属性値の1つとしてのみ扱っており(式(11)), 高さ値に対してスケール変化の下での自己相似性を維持するような操作を行っていないため, 中点変位法でつくられるような3次元的なフラクタル形状とはならない**。もう1つの理由は, 4.3節で述べたとおり, wrinkly曲面のスケール方向の自己相似性は区分的に空間方向にずれを持つため, 不変性という厳密な性質を持つものではなく, 数学的に厳密な定義どおりのフラクタルではないということである。

図6は制御メッシュ(a)上に形成されたwrinkly曲面である。各制御点は位置座標値と高さ値の他に色値を情報として持つ。曲面の左端, 中央, 右端のそれぞれに関して, スケール方向の拡大の下で類似の構造が維持されていることが分かる。図中では拡大とともに変位の相対的な高さが大きくなっており, このwrinkly曲面は厳密な意味で3次元的なフラクタルではない。これは, 上記の1つ目の理由による。もちろん, 高さ値に対して適切な縮小変換を作用させることで3次元的にフラクタル化させることも可能である。一方, 色については拡大の下での類似構造がはっきりと確認で

* 実際には, Bézier補間のため p_i の前後の制御点からも影響を受ける。

** 厳密な3次元フラクタルをつくるためには, 面方向と高さ方向の間でスケール変化の下での自己相似関係が成り立つような縮小変換を行う必要がある。

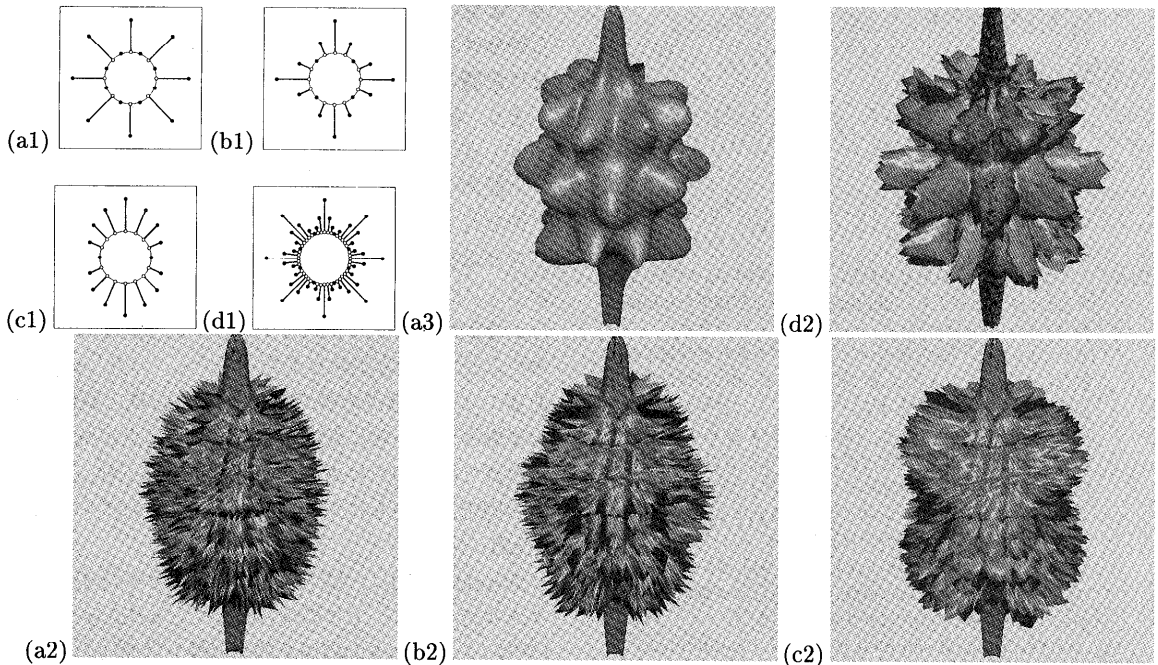


図9 高さ値の与え方による凹凸変化の例：(a1), (b1), (c1) および (d1) の白丸は制御点位置, 黒丸は高さ値分だけ法線方向に移動した点を表す. また, サンプリングには「ゆらぎ」を与えている.

Fig. 9 Examples of bumps based on height values.

きる. また, 異なるスケールレベルごとにみた場合, 曲面の左端, 中央, 右端という空間方向の視点移動の下で, 空間方向の位置ごとの局所性がすべてのスケールレベルを通じて同様に維持されていることが分かる. 以上のように, 図6はwrinkly曲面の持つ「空間/スケール方向の局所的類似性」を明確に示している. この特徴は通常の大域的なフラクタル手法ではつくり出すことができず, 形状モデリングへの応用を考えた場合, 特に自然物等のモデリングに対して効果を発揮することが期待できる.

なお, 図6は5.2節の適応的サンプリングに基づき視点位置によりサンプリング数を変更して作成している. 5.2節の条件a), b), c)が満たされ, いわゆる, LOD (Level Of Detail) が実現されている.

6. 実 例

以下にwrinkly曲面で作成した形状モデルの実例を示す. なお, 以下の例では, すべてのwrinkly曲面の交差区間数を $M=1$ としている. また, レンダリングにはZバッファ法を用いた.

高さ値の与え方による凹凸変化の例 (図9)

図9の(a1), (b1), (c1)は(a2), (b2), (c2)のもとになる円柱形の制御メッシュを中央部分で水平に切っ

た断面図である. それぞれ異なる高さ値を与えた結果として異なる形状ができていることが分かる. (a1)では高さ値が円周方向で大小交互に与えられているため「トゲトゲ」した曲面((a2))ができ, (c1)では高さ値が徐々に変化するため「トゲトゲ」が緩やか((c2))である.

また, (d2)は(a2)の制御メッシュに制御点を追加した結果である(断面図は(d1)). 5.3節に述べたように, (d2)は制御点が密であるため変位が固定化される.

なお, (a3)は(a2)と同一の制御メッシュをスプライン補間したものである.

自然の風景 (図10)

図10(b), (c)はそれぞれスプライン曲面とwrinkly曲面であり, 「草原」, 「川」, 「空」を形作る3つの曲面から構成されている. とともに同一の制御メッシュに基づく(「草原」に関するメッシュを(a)に示す). この例はwrinkly曲面の空間方向の位置による局所性を明示している. なお, 「川」と「空」については透過率に関してもISTを適用した.

(e), (f)は「川」を抜き出したもの(制御メッシュは(d))であるが, それぞれ u, v 軸の一方にスプライン補間, 他方にISTを適用したものである. 両者を

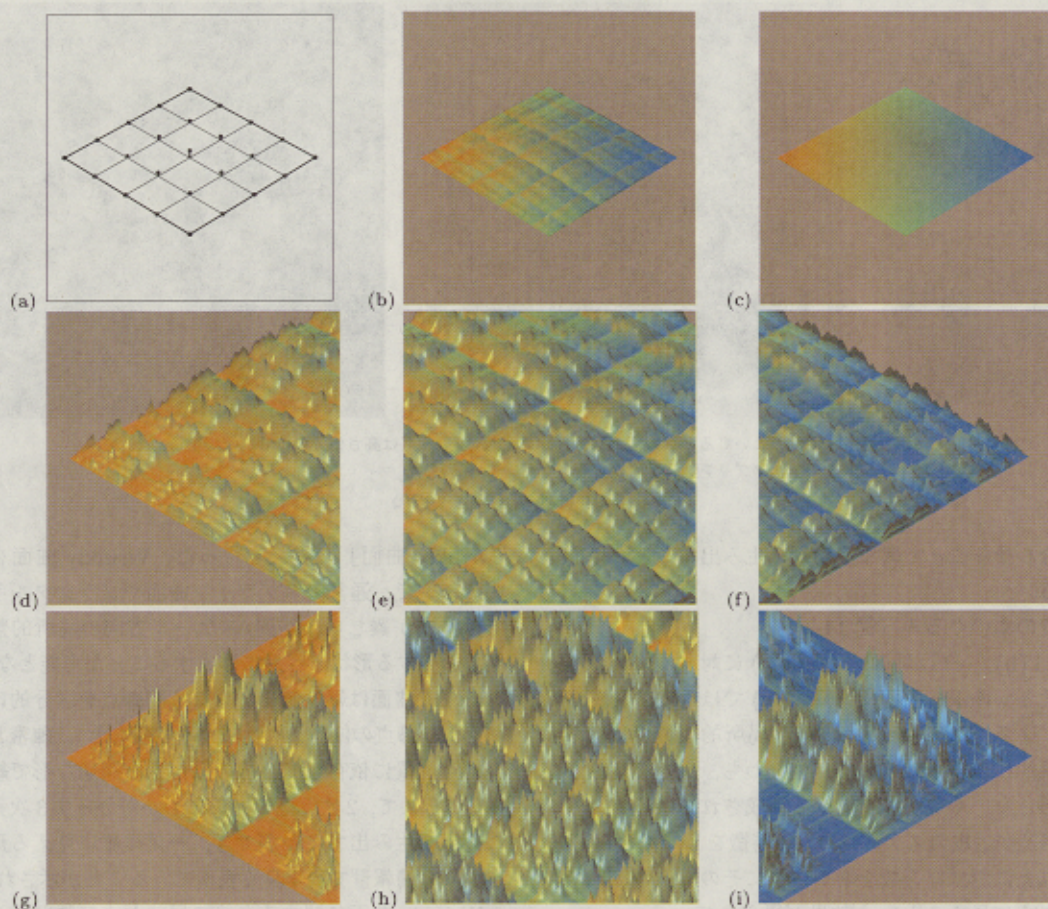


図6 Wrinkly 曲面の例：制御メッシュ(a)に対する wrinkly 曲面 (b) とスプライン曲面 (c), (d), (e), (f) は (b) の左端, 中央, 右端を 3 倍に拡大したもの. (g), (h), (i) はそれらをさらに 3 倍に拡大したもの. 制御メッシュ(a)中の黒点は制御点位置 (メッシュの格子点) から高さ値分だけ持ち上げた点である.

Fig. 6 An example of wrinkly surface.



図10 「自然の風景」の例：制御メッシュ(a), (d)上のトゲは高さ値分の持ち上げである. また, サンプルには「ゆらぎ」を与えている.

Fig. 10 Images of a natural scene.

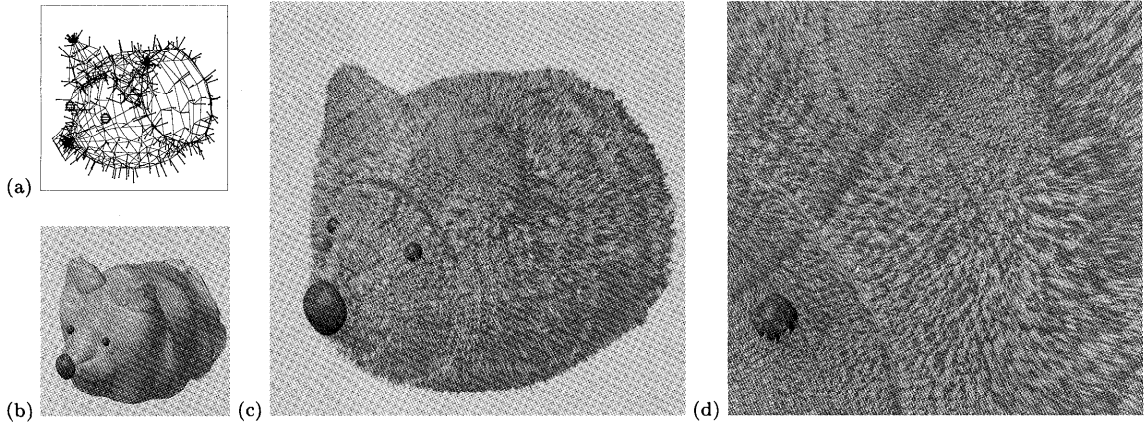


図 11 「毛皮のぬいぐるみ」の例：制御メッシュ(a)上のトゲは高さ値分の持ち上げである。また、サンプリングには「ゆらぎ」を与えている。

Fig. 11 Images of a stuffed toy.

組み合わせることで異なる効果を生み出すことが可能となる。

毛皮のぬいぐるみ (図 11)

図 11 (b), (c) は制御メッシュ(a)に対するスプライン曲面と wrinkly 曲面である。(c) では毛皮の質感が IST による空間/スケール方向の局所的類似性によって効果的に表現されている。すなわち、大きな変位がより小さないくつかの変位から構成されているという階層構造が毛皮のスケール方向の特徴をうまくとらえ、必要以上にポリゴン数を増やさずにその構造が続いていくように見えるという効果が得られている。

また、(d) は (c) の一部を拡大した図である。適切なサンプリングにより、上記の質感を損なうことなく、適切な LOD が実現されている*。

ここで、(d) 中の毛皮には「切れ目」が生じていることが分かる。これは IST のベース区間のつなぎ目 (パラメータ値 $u = i$) の部分にあたる。図 2 から分かるように、この部分では不連続の度合 (座標値の差) が大きくなるため、wrinkly 曲面上でも段差として目立ってしまう。これは中点変位法における初期形状のための 'crease' の問題²⁾に類似している。サンプリングに「ゆらぎ」を与えることである程度は緩和することは可能であるが、さらに根本的な解決を考えるべき問題である。

7. 結論および今後の展望と課題

本論文では、制御点の補間を任意の手続きにより行う「手続き的補間」の概念を提唱し、その 1 例として

「wrinkly 曲面」の提案を行った。Wrinkly 曲面を用いることで、通常のスプライン曲面やフラクタル手法では作成が難しい「空間/スケール方向の局所的類似性」を有する形状を容易に作成することが可能となる。Wrinkly 曲面は属性値をとまなう制御点を区分的に補間し、制御点の位置、属性値 (高さ値など)、疎密および視点位置に依存して上記の類似性を適切な形で維持する。そして、2次元面であるにもかかわらず3次元的な効果を生み出すことができ、一見複雑に見える形状を少ない情報量で効率良く表現できることが示された。

Wrinkly 曲面に関する今後の展望および課題としては、以下の項目があげられる。

- 図 11 (d) にみられるような「切れ目」の回避。
- 不規則メッシュ上での wrinkly 曲面の形成。
- FFD (Free Form Deformation) との融合 (変位の任意方向化 (法線方向以外) も含む)。
- ポリゴン化せず直接レンダリングを行う方法の提案。

さらに、今後、wrinkly 曲面以外のさまざまな手続き的補間に関する研究を行う必要がある。

謝辞 本研究の一部は文部省科学研究費補助金 (基盤 C 一般 09680346) の援助を受けている。

参考文献

- 1) Farin, G.E.: *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 2nd ed., Academic Press, Boston, MA (1990).
- 2) Watt, A. and Watt, M.: *Advanced Animation and Rendering Techniques: Theory and Practice*, ACM Press, New York (1992).
- 3) Ebert, D.S. (Ed.): *Texturing and Modeling: A*

* 毛皮の 3 次元テクスチャを扱った文献²⁴⁾では、拡大時の質感の維持が困難であることが問題とされている。

- Procedural Approach*, Academic Press, Boston, MA (1994).
- 4) Peitgen, H.O. and Saupe, D. (Eds.): *The Science of Fractal Images*, Springer-Verlag, New York (1988).
 - 5) Barnsley, M.F.: *Fractals Everywhere*, 2nd ed., Academic Press, Boston, MA (1993).
 - 6) Demko, S.: Construction of Fractal Objects with Iterated Function Systems, *Computer Graphics (SIGGRAPH '85)*, Vol.19, No.3, pp.271-278 (1985).
 - 7) Miller, G.S.P.: The Definition and Rendering of Terrain Maps, *Computer Graphics (SIGGRAPH '86)*, Vol.20, No.4, pp.39-48 (1986).
 - 8) Oppenheimer, P.E.: Real Time Design and Animation of Fractal Plants and Trees, *Computer Graphics (SIGGRAPH '86)*, Vol.20, No.4, pp.55-64 (1986).
 - 9) Barnsley, M.F., Jacquin, A., Malassenet, F., Reuter, L. and Sloan, A.D.: Harnessing Chaos for Image Synthesis, *Computer Graphics (SIGGRAPH '88)*, Vol.22, No.4, pp.131-140 (1988).
 - 10) Musgrave, F.K., Kolb, C.E. and Mace, R.S.: The Synthesis and Rendering of Eroded Fractal Terrains, *Computer Graphics (SIGGRAPH '89)*, Vol.23, No.3, pp.41-50 (1989).
 - 11) Szeliski, R. and Terzopoulos, D. From Splines to Fractals, *Computer Graphics (SIGGRAPH '89)*, Vol.23, No.3, pp.51-60 (1989).
 - 12) Reeves, W.T.: Particle Systems—A Technique for Modeling a Class of Fuzzy Objects, *Computer Graphics (SIGGRAPH '83)*, Vol.17, No.3, pp.359-376 (1983).
 - 13) Reeves, W.T. and Blau, R.: Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems, *Computer Graphics (SIGGRAPH '85)*, Vol.19, No.3, pp.313-322 (1985).
 - 14) Peachey, D.R.: Modeling Waves and Surf, *Computer Graphics (SIGGRAPH '86)*, Vol.20, No.4, pp.65-74 (1986).
 - 15) Sims, K.: Particle Animation and Rendering Using Data Parallel Computation, *Computer Graphics (SIGGRAPH '90)*, Vol.24, No.4, pp.405-413 (1990).
 - 16) Fleischer, K.W., Laidlaw, D.H., Currin, B.L., and Barr, A.H.: Cellular Texture Generation, *Computer Graphics (SIGGRAPH '95)*, pp.239-248 (Aug. 1995).
 - 17) Smith, A.R.: Plants, Fractals and Formal Languages, *Computer Graphics (SIGGRAPH '84)*, Vol.18, No.3, pp.1-10 (1984).
 - 18) Prusinkiewicz, P., Lindenmayer, A. and Hanan, J.: Developmental Models of Herbaceous Plants for Computer Imagery Purposes, *Computer Graphics (SIGGRAPH '88)*, Vol.22, No.4, pp.141-150 (1988).
 - 19) Prusinkiewicz, P., Hammel, M.S. and Mjolsness, E.: Animation of Plant Development, *Computer Graphics (SIGGRAPH '93)*, pp.351-360 (Aug. 1993).
 - 20) Prusinkiewicz, P., James, M. and Mèch, R.: Synthetic Topiary, *Computer Graphics (SIGGRAPH '94)*, pp.351-358 (July 1994).
 - 21) Perlin, K.: An Image Synthesizer, *Computer Graphics (SIGGRAPH '85)*, Vol.19, No.3, pp.287-296 (1985).
 - 22) Perlin, K. and Hoffert, E.M.: Hypertexture, *Computer Graphics (SIGGRAPH '89)*, Vol.23, No.3, pp.253-262 (1989).
 - 23) Lewis, J.P.: Algorithms for Solid Noise Synthesis, *Computer Graphics (SIGGRAPH '89)*, Vol.23, No.3, pp.263-270 (1989).
 - 24) Kajiya, J.T.: Rendering Fur with Three Dimensional Textures, *Computer Graphics (SIGGRAPH '89)*, Vol.23, No.3, pp.271-280 (1989).
 - 25) Worley, S.: A Cellular Texture Basis Function, *Computer Graphics (SIGGRAPH '96)*, pp.291-294 (Aug. 1996).
 - 26) 藤本, 大野: 形状と質感の融合のためのモデリング法, 情報処理学会グラフィクスとCAD研究報告, No.84, pp.13-18 (1997).

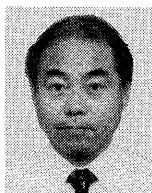
(平成 9 年 9 月 17 日受付)

(平成 10 年 5 月 8 日採録)



藤本 忠博 (学生会員)

昭和 41 年生。平成 4 年慶應義塾大学大学院理工学研究科計算機科学専攻修士課程修了。同年 (株) 三菱総合研究所入社。平成 7 年慶應義塾大学大学院理工学研究科博士課程入学。現在在学中。コンピュータグラフィクス、特に形状モデル理論に興味を持つ。



大野 義夫 (正会員)

昭和 45 年慶應義塾大学大学院理工学研究科修了。同年同大学情報科学研究科助手。昭和 62 年助教授。平成 7 年より同大学理工学部計算機科学専攻教授。コンピュータグラフィクス、DTP 等に関連したアルゴリズムに興味を持つ。工学博士。