

反復関数系を用いたテクスチャリング

堀江 大輔, 蔡 東生

筑波大学 システム情報工学研究科

現在、コンピュータグラフィックスにおいて、テクスチャリングは不可欠なものとなっている。本研究は、反復関数系 (Iterated Function System : IFS) の持つフラクタル性、またその描画アルゴリズムである Chaos Game Algorithm とポイントレンダリングとの類似性に着目し、IFS をテクスチャ生成と描画に用いることを目指す。ベジエ曲面を (確率付き) IFS に変換し、その不変測度をテクスチャとして用いることにより、物体のテクスチャリングと描画の両方を IFS によって実現した。まずベジエ曲線、IFS について述べ、ベジエ曲線から IFS への変換手法を説明し、本手法のベジエ曲面への適用を示す。

Generating texture using Iterated Function System

Daisuke HORIE, DongSheng CAI

Graduate School of Systems and Information Engineering at University of Tsukuba

Recently, texturing is an indispensable tool in computer graphics. We notice fractality and similarity between point rendering and Chaos Game Algorithm in Iterated Function System (IFS). The goal of this research is the utilization of IFS for both generating texture and drawing geometry. We achieve this goal by transforming Bezier surface into the IFS attractor, and by using invariant measure of IFS as a texture. First, we describe about the Bezier curve and IFS, and explain how to transform Bezier curve into IFS. Finally, we show the application of this method into Bezier surface.

1. はじめに

現在、コンピュータグラフィックスにおいて、テクスチャリングは不可欠な技術となっている。テクスチャの利用例としては、通常のテクスチャマッピングやディスプレイマッピング、HyperTexture 等[1]が挙げられるが、これらに共通して言えることは、立方体や球などの物体に、色や不透明度と言ったパラメータを与える、3次元実数空間による関数である、ということである。

一方、物体のモデリング方法には、主にポリゴンモデリング、プリミティブを組み合わせる CSG 表現、NURBS 曲面によるモデリング等がある。本研究では、扱いが容易であることから、ベジエ曲面[2]によるモデリングについて見ていく。

ところで、フラクタル画像の生成に、(確率付き) 反復関数系 (Iterated Function System : IFS) [3] という手法がある。IFS は、アトラクタと呼ばれるモノクロ画像と、それに濃淡値を付加した不変測度を持つ。また、それらの

描画手法に、点のプロットを基本とする Chaos Game Algorithm が存在する。IFS の主な適用としては、フラクタル画像圧縮が挙げられるが、jpeg 等の既存手法より優れた結果が出ているとはいえないのが実情である。

そこで本研究では、IFS をテクスチャ生成に用いることを提案する。IFS を用いると、アトラクタで幾何形状を、不変測度でテクスチャを表現することが可能である。アトラクタ、不変測度は共にフラクタルになる。樹木など、多くの自然物にはフラクタル性が見られ、それらの表現に IFS を用いる事は妥当だといえる。また、アトラクタも不変測度もスケールに依存せず、視点の接近などの影響を受けない。また、描画に Chaos Game Algorithm を用いる事で、ポイントレンダリングも可能である。さらに、生成手法は確立されていないものの、任意のアトラクタや不変測度を生成可能であることが Collage の定理により保証されている。

本研究では、ベジエ曲面でモデリングされた物体を IFS

へ変換することで、物体の概形をアトラクタに持つ IFS を生成し、その不変測度をテクスチャとして用いることにより、物体のテクスチャリングと描画の両方を IFS によって実現することを目的とする。

2. ベジエ曲線

ベジエ曲線はパラメトリック曲線の一つである。 n 次ベジエ曲線は、 $n+1$ 個の制御点で記述される。本研究では 3 次ベジエ曲線を扱った。 P_0, P_1, P_2, P_3 をそれぞれ制御点の位置ベクトルとすると、3 次ベジエ曲線 $P(t)$ は、

$$P(t) = \sum_{i=0}^3 B_i^3(t) P_i$$

と定義される。ただし、 t は媒介変数で、 $0 \leq t \leq 1$

また、 $B_i^n(t)$ は Bernstein 関数と言い、

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} \cdot t^i$$

と定義される。特に $n=3$ の時は、

$$B_0^3(t) = (1-t)^3$$

$$B_1^3(t) = 3(1-t)^2 \cdot t$$

$$B_2^3(t) = 3(1-t) \cdot t^2$$

$$B_3^3(t) = t^3$$

となる。図 1 に、3 次ベジエ曲線の例を示す。

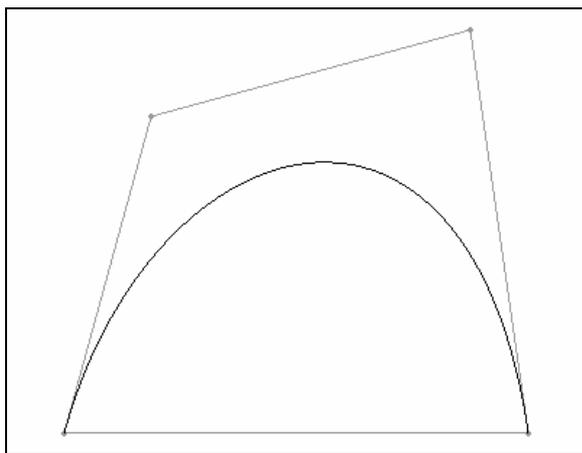


図 1 3 次ベジエ曲線

3. ベジエ曲面

ベジエ曲面はパラメトリック曲面の一つである。 $n \times m$ 次ベジエ曲面は、 $(n+1) \times (m+1)$ 個の制御点で記述される。

$0 \leq i \leq n$ 、 $0 \leq j \leq m$ について、 P_{ij} を制御点の位置ベクトルとすると、 $n \times m$ 次ベジエ曲面 $P(u, v)$ は、

$$P(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) \cdot B_j^m(v) \cdot P_{ij}$$

と定義される。ただし、 $0 \leq u, v \leq 1$

また、 $B_i^n(t)$ はベジエ曲線と同様、Bernstein 関数である。

なお、本研究では双 3 次ベジエ曲面を扱った。すなわち、 $n = m = 3$ 。

4. Iterated Function System (IFS)

(確率付き) IFS は、完備な距離空間 X 、 n 個の縮小写像 $\omega_i: X \rightarrow X$ 、確率 p_i によって、

$$\{X; \omega_1, \dots, \omega_n; p_1, \dots, p_n\}$$

と定義される。距離空間 X によるハウスドルフ空間を $H(X)$ 、 $C \in H(X)$ とすると、

$$W(C) = \bigcup_{i=1}^n \{\omega_i(c) : c \in C\}$$

と定義される写像 $W: H(X) \rightarrow H(X)$ は縮小写像になる。

よって、点列 $\{W^n(C)\}_{n=1}^\infty$ はある収束点 A を持ち、この点をアトラクタと言う。すなわち、

$$A = \lim_{n \rightarrow \infty} W^n(C)$$

また、 F をフィールド、 $A_i \in F$ (ただし $i \in \mathbb{N}$ 、 $A_i \cap A_j = \phi$) とすると、

$$\mu(\cup_{i=1}^\infty A_i) = \sum_{i=1}^\infty \mu(A_i)$$

を満たすような関数 $\mu: F \rightarrow [0, \infty)$ を測度 (measure) という。特に、 $P(X)$ をハッチンソン空間、 $\nu \in P(X)$ を正規化 Borel 測度とし、 $M: P(X) \rightarrow P(X)$ を

$$M(v) = \sum_{i=1}^N p_i \circ \omega_i^{-1}$$

としたとき、

$$\mu^* = \lim_{n \rightarrow \infty} M^{on}(v)$$

を IFS の不変測度 (invariant measure) という。

IFS のアトラクタを描画するアルゴリズムに、Chaos Game Algorithm がある。これは、ある点 $x_n \in X$ に、IFS コード内の確率に従って変換を選択し、 $x_{n+1} = \omega_i(x_n)$ とするアルゴリズムである。得られた点列をプロットしていくことで、アトラクタを近似的に描くことができる。この時、点の密度が不変測度になる。すなわち、 B を Borel 部分集合として、

$$\mu^*(B) = \lim_{n \rightarrow \infty} \left\{ \frac{B \text{ に含まれる点の数}}{n+1} \right\}$$

なお、IFS の写像としては、アフィン変換を用いるのが一般的である。本研究でもアフィン変換のみを扱っている。

5. IFS によるテクスチャリング

それでは、IFS によるテクスチャリングについて説明する。テクスチャリングにおいて、最も簡単なものはテクスマッピングである。テクスマッピングでは、2次元画像を生成し、その後にベジエ曲面のパラメータ u, v に従って曲面上にマッピングする。IFS はアトラクタとアトラクタ上に定義される不変測度を生成できるので、2次元実数空間上の IFS を作成し、その不変測度の $[0,1] \times [0,1]$ 部分をテクスチャ画像として用いれば良い。しかし、これでは描画の際に IFS の Chaos Game Algorithm を用いることができない。

Chaos Game Algorithm を用いるためには、ベジエ曲面の描画にも IFS を用いる必要がある。しかし、これも難しいものではない。IFS のアトラクタを $[0,1] \times [0,1]$ の空間全体と定義すれば、ベジエ曲面 u, v に対応するので、Chaos Game Algorithm で出てきた点 (u, v) を Bernstein 関数に代入すれば、ベジエ曲面を Chaos Game Algorithm で描画できる。

本研究ではこれをさらにもう一歩進め、ベジエ曲面自体をアトラクタに持つ IFS の導出を目指している。これにより、IFS コードに変更を加えることで、幾何形状そのもの

にフラクタル性を持たせることができると考えたためである。

6. ベジエ曲線から IFS への変換

ベジエ曲線の IFS への変換手法について説明する。説明には3次元空間上の3次ベジエ曲線を用いる。3次ベジエ曲線をアトラクタに持つ IFS の導出は一般に困難であるため、本研究では Bernstein 関数を IFS 化する手法を用いた。本手法の基本概念を図4に示す。

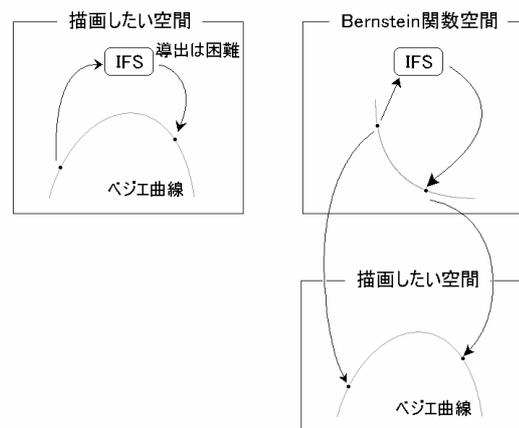


図 2 本手法の基本概念

まず、4次元実数空間を考える。この空間上で、曲線 $B(t)$ を、

$$B(t) = (B_0^3(t), B_1^3(t), B_2^3(t), B_3^3(t))$$

と定義する。ただし、 t は媒介変数で、 $0 \leq t \leq 1$ 以後、この曲線 $B(t)$ を、Bernstein 関数曲線と呼ぶ。この Bernstein 関数曲線によって、3次ベジエ曲線 $P(t)$ は次のように再定義される。

$$P(t) = B(t) \begin{pmatrix} P_0(x) & P_0(y) & P_0(z) \\ P_1(x) & P_1(y) & P_1(z) \\ P_2(x) & P_2(y) & P_2(z) \\ P_3(x) & P_3(y) & P_3(z) \end{pmatrix}$$

ただし、 P_i は制御点、 $i = 0, 1, 2, 3$

次に、縮小写像を計算する。縮小写像が存在すると仮定した時、仮に Bernstein 関数曲線を $t = 1/2$ で2つに分割できるとすると、

$$\omega_1(B(t)) = B\left(\frac{1}{2}t\right)$$

$$\omega_2(B(t)) = B\left(\frac{1}{2}t + \frac{1}{2}\right)$$

となるはずである。ただし、 $i=1,2$ に対して、

$\omega_i: R^4 \rightarrow R^4$ はアフィン変換であり、

$$\omega_i(x) = A_i x + T_i$$

A_i は 4×4 の行列、 T_i は 4次元列ベクトルである。

なお本手法では、Bernstein 関数曲線を、媒介変数 t による任意の点で任意の数に分割することができる。そのためには、Bernstein 関数曲線の t に $\frac{1}{2}t$ などの 1 次式を代入

し、合成関数を求めればよい。分割の位置や個数により、描画される曲線が変化することはない。

次に、先の仮定で立てた式を変数 t による恒等式と見なし、これを解くことでアフィン変換のパラメータを決める。本研究ではこれを行列の掃き出し法によって求めた。導出されたアフィン変換は、列ベクトル T_i の要素による 1 次式の形になる。

ここで適当な列ベクトル T_i を代入すれば、アフィン変換を求めることができる。しかし、これらのアフィン変換は代入する T_i によって縮小係数 s_i が異なり、場合によっては s_i が 1 以上の値になってしまう。これは $0 \leq s_i < 1$ という縮小写像の定義に反する。そこで本研究では、縮小係数を最小にする T_i を求める、という最適値問題を解くことで T_i を決定した。導出方法には山登り法を用いた。実際に求めたアフィン変換を以下に示す。

$$\omega_1 \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{4} & -\frac{3}{8} \\ -\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} \\ 0 & 0 & \frac{1}{4} & \frac{3}{8} \\ 0 & 0 & 0 & \frac{1}{8} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ \frac{1}{4} \\ 0 \\ 0 \end{pmatrix}$$

$$\omega_2 \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{8} & 0 & 0 & 0 \\ \frac{1}{8} & 0 & -\frac{1}{4} & -\frac{1}{4} \\ \frac{3}{8} & \frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{3}{8} & -\frac{1}{4} & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{4} \\ 0 \\ \frac{1}{2} \end{pmatrix}$$

このアフィン変換の縮小係数はどちらも $\frac{\sqrt{3}}{4}$ である。

これらにより、 $\{R^4; \omega_1, \omega_2; \frac{1}{2}, \frac{1}{2}\}$ という IFS によっ

て Bernstein 関数曲線を表すことができた。本手法を用いて描いたベジエ曲線を図 3 に示す。図 1 のベジエ曲線と同じ制御点を用いているため、図 1 と全く同じ曲線を描いていることが見て取れる。

なお、本論文では $t = \frac{1}{2}$ で 2 つに分割したが、これは分

割の 1 例である。本手法では、任意の t で、任意の数に分割することが可能である。

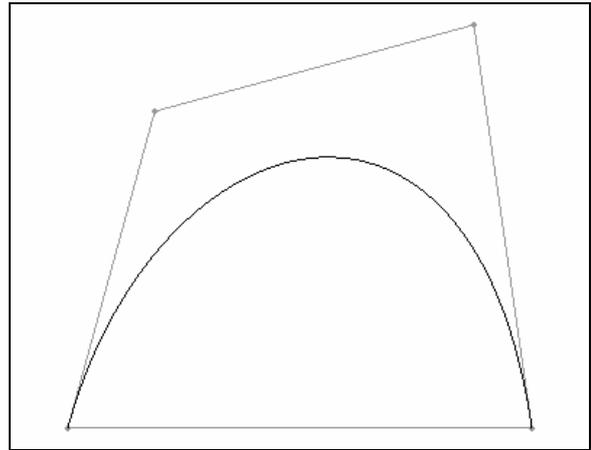


図 3 IFS によるベジエ曲線

7. ベジエ曲面への適用

先に述べたベジエ曲線の変換手法を、双 3 次ベジエ曲面に適用した。すなわち、2 つの Bernstein 関数 $B_i^3(u)$ 、

$B_j^3(v)$ をそれぞれ先の手法で IFS 化し、その組み合わせによってベジエ曲面を導出した。

図 4 に、本手法によって描画した Utah TeaPot を示す。描画には Chaos Game Algorithm によるポイントレンダリングを用いている。



図 4 IFS によるベジエ曲面(Utah TeaPot)



図 6 重なりを許した IFS によるテクスチャ

8. 不変測度によるテクスチャリング

上で示した Utah Teapot に対し、IFS の不変測度を使用したテクスチャリングを行った。

まず、最も簡単な例として、ベジエ曲面をランダムに格子状に区切る IFS を用いたテクスチャリングを行った。結果を図 5 に示す。

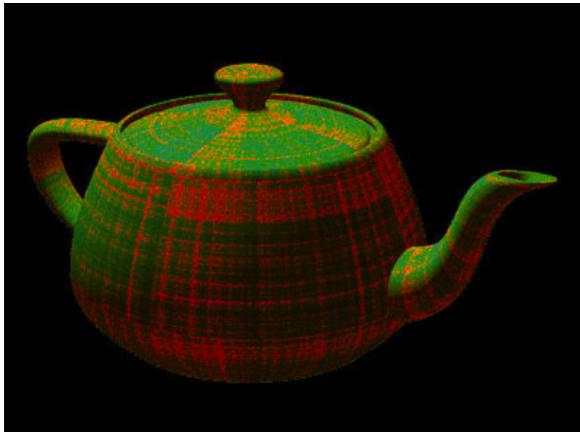


図 5 ランダム格子型 IFS によるテクスチャ

単に格子状に区切ただけであるため、パッチの縦横に線が走り、単純な模様になってしまっている。

次に、IFS の重なりを許した場合の例を図 6 に示す。

これは上下対称の IFS ではあるが、複雑な模様に見える。ただし、全てのパッチに同じ不変測度をマッピングしているため、各パッチに一つの模様という規則性が見えてしまっている。

なお、現段階では形状はベジエ曲面そのものであるため、図 6 は 2 次元平面状の IFS

$$\omega_1 \begin{pmatrix} x \\ y \end{pmatrix} = 0.5 \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0.125 \\ 0.125 \end{pmatrix} \quad p_1 = 0.24$$

$$\omega_2 \begin{pmatrix} x \\ y \end{pmatrix} = 0.5 \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0.375 \\ 0.125 \end{pmatrix} \quad p_2 = 0.24$$

$$\omega_3 \begin{pmatrix} x \\ y \end{pmatrix} = 0.5 \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0.25 \\ 0.375 \end{pmatrix} \quad p_3 = 0.24$$

$$\omega_4 \begin{pmatrix} x \\ y \end{pmatrix} = 0.5 \begin{pmatrix} x \\ y \end{pmatrix} \quad p_4 = 0.07$$

$$\omega_5 \begin{pmatrix} x \\ y \end{pmatrix} = 0.5 \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0.5 \\ 0 \end{pmatrix} \quad p_5 = 0.07$$

$$\omega_6 \begin{pmatrix} x \\ y \end{pmatrix} = 0.5 \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0.5 \end{pmatrix} \quad p_6 = 0.07$$

$$\omega_7 \begin{pmatrix} x \\ y \end{pmatrix} = 0.5 \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \quad p_7 = 0.07$$

をテクスチャマッピングしたものと等価になっている。

9. おわりに

本研究は、反復関数系 (Iterated Function System : IFS)

の持つ、フラクタル性や描画アルゴリズムのポイントレンダリングとの類似性に着目し、その不変測度をテクスチャとして用いることを提案した。また、対象となる形状をベジエ曲面で表現されたものとし、ベジエ曲面を IFS に変換することで、Chaos Game Algorithm を用いたポイントレンダリングによる描画と、不変測度によるテクスチャ生成を IFS で実現した。

ベジエ曲面から IFS への変換方法として、4次元空間上の Bernstein 関数曲線を考え、これを IFS 化し、出てきた点を変換して描画した。この方法は、IFS コードに変更を加えることで、幾何形状そのものにフラクタル性を与えるために取られた。

また、IFS の不変測度によるテクスチャの例を2つ挙げた。これにより、特に重なりを許した IFS の場合、極めて複雑なパターンを生成できうる事を示した。

今後の課題として、IFS コードを最適値問題で導出する際の解導出アルゴリズムの変更が挙げられる。本手法では、縮小係数が1未満に収まっていけば支障はない。しかし、この問題は多峰性であり、現在の山登り法で必ず縮小係数が1未満になるという保証はない。

また、ベジエ曲面への適用において、現在は2つの変数 Bernstein 関数を独立に IFS 化しているが、これを変更し4次元 IFS \times 2ではなく、8次元 IFS \times 1とすることで、より複雑な IFS が適用可能になると考えられる。

本研究では、幾何形状におけるフラクタル表現を可能にするため、単純な2次元画像生成ではなく、4次元の IFS を導出した。しかし幾何形状の変更はまだ試していないため、IFS コードを具体的にどのように変更すればよいのかを調べる必要がある。

IFS は、フラクタル形状を生成する際に有効な手法であるが、決定論的フラクタルしか生成できず、統計的フラクタルを生成することはできない、という欠点がある。それに対し、雲や樹木などの自然物の多くは統計的フラクタル形状である。そのため、本手法で求めた IFS をどう変更しても、実際の自然物のような統計的フラクタル形状を生成することはできない。しかしこれは、IFS に用いる変換をアフィン変換以外のものにする事で規則性を感じにくくしたり、IFS を拡張した LRIFS[4, 5]を用いて統計的フラクタル性を持たせたりすることによって改善できると考えられる。

また、本手法を拡張することにより、有理ベジエ曲線や

他のパラメトリック曲線を IFS 化することも可能であると考えられる。

参考文献

- [1] Ebert, D., Musgrave, F., Peachey, P., Perlin, K., and Worley, S., "Texturing and Modeling: A Procedural Approach, Third Edition", with contributions from W. Mark and J. Hart, Morgan Kaufmann, November 2002
- [2] "技術編 CG 標準テキストブック", 画像情報教育振興協会
- [3] M. F. Barnsley, "Fractals Everywhere", Academic Press, 1988.
- [4] Przemyslaw Prusinkiewicz and Mark Hammel, "Language-Restricted Iterated Function Systems, Koch Constructions, and L-systems", SIGGRAPH '94
- [5] Przemyslaw Prusinkiewicz and Mark Hammel, "Escape-time Visualization Method for Language-restricted Iterated Function Systems", Proceeding of Graphics Interface '92, pp.213-223, 1992