2003－ＣＧ－113　(6)
2003／11／25

反復関数系を用いたベジエ曲線の生成

堀江 大輔†　望月 茂徳†　長峯 望††　蔡 東生†††

現在、コンピュータグラフィックスにおいて、テクスチャは不可欠なものとなっている。テクスチャは、立方体や球などの物体上のノイズと見なすことができる。本研究は、ベジエ曲面を用いてモデリングした物体を（確率付き）反復関数系(Iterated Function System : 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、物体のモデリングとテクスチャリングの両方を１つの IFS によって実現することを目的とする。本論文では、そのうち、主に IFS によるベジエ曲線の生成について扱う。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 Bezier Curve using Iterated Function System

Daisuke HORIE†　Shigenori MOCHIZUKI†　Nozomi NAGAMINE††　DongSheng CAI†††

Recently, texturing is an indispensable tool in computer graphics. A texture is a kind of a noise on geometry such as cube or ball. The goal of this research is transforming geometry modeled by Bezier surface into the attractor of Iterated Function System (IFS) with probability, and by using invariant measure of IFS as a texture, we are able to do both modeling and texturing by one IFS. In this paper, we discuss about generating Bezier curve using IFS. 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、テクスチャは不可欠なものとなっている。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、HyperTexture[1]等が挙げられるが、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、と言うことである。すなわち、テクスチャとは、物体のパラメータのノイズと見なすことができる。

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が、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。

Perlin によって提案された HyperTexture は、物体形状を３次元ノイズと見なし、これに密度調整関数を掛けることで、物体形状をテクスチャリングする技術で、雲のような半透明物体の生成に力を発揮する。しかし、HyperTexture の DMF 設計は経験的なものであり、任意のノイズ生成を行うことはできない。

そこで本研究では、物体のテクスチャリングに（確率付き）反復関数系（Iterated Function System : IFS）[3]を用いることを提案する。IFS with probability はアトラクタにより幾何形状を、Chaos Game Algorithm によりテクスチャ的ノイズを表すことが可能であり、形状、ノイズ共に任意の物を生成できることが Collage の定理によって保証されている。一方で、任意の形状やノイズを生成する手法は未だ確立されていない。そこで、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、物体の概形を生成し、不変測度をテクスチャとして用いることにより、物

†筑波大学 大学院 システム情報工学研究科
Graduate School of Systems and Information Engineering at University of Tsukuba
††筑波大学 大学院 理工学研究科
Master's Program in Science and Engineering at University of Tsukuba
†††筑波大学 電子・情報工学系
Institute of Information Sciences and Electronics at University of Tsukuba

体のモデリングとテクスチャリングの両方を１つの IFS によって実現することを目的とする。

本論文では、その経過報告として、IFS によるベジエ曲線の生成について扱う。また、本手法のベジエ曲面への適用を示す。

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。本研究では主に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、$0 \le t \le 1$

また、$B_i^n(t)$ は Bernstein 関数と言い、

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

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。

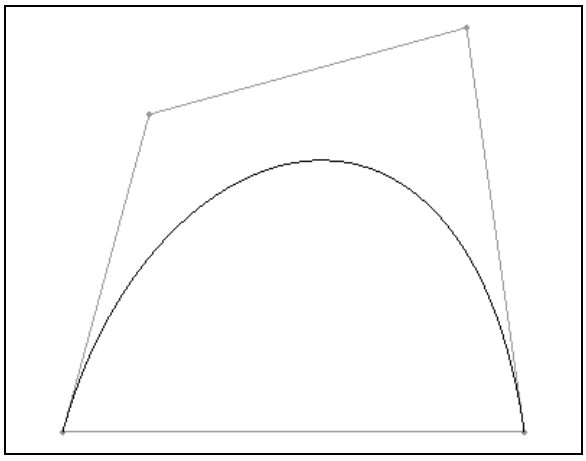

図 1　３次ベジエ曲線

ベジエ曲線の描画には、Bernstein 関数の媒介変数 t を変化させていくことによって求められる点列を直線で結んでいく方法の他に、ド・カステリョのアルゴリズムが知られている。これは、$n=2$ または 3 の時に用いることができるアルゴリズムで、図２のように、

$$P_i^0 = P_i$$
$$P_i^r = (1-t) \cdot P_i^{r-1} + t \cdot P_{i+1}^{r-1}$$

とした時、P_0^n はベジエ曲線上の点になることを利用している。t を変化させて求められる点 P_0^n を結んでいくことでベジエ曲線を描くことができる。

なお、ベジエ曲線は、P_0^n によって、２つの新たなベジエ曲線に分割される事が知られている。$n=3$ の場合、制御点 $\{P_0^0, P_1^0, P_2^0, P_3^0\}$ で定義されるベジエ曲線は、点 P_0^3 によって、それぞれ $\{P_0^0, P_0^1, P_0^2, P_0^3\}$、$\{P_0^3, P_1^2, P_2^1, P_3^0\}$ を制御点とするベジエ曲線に分割される。

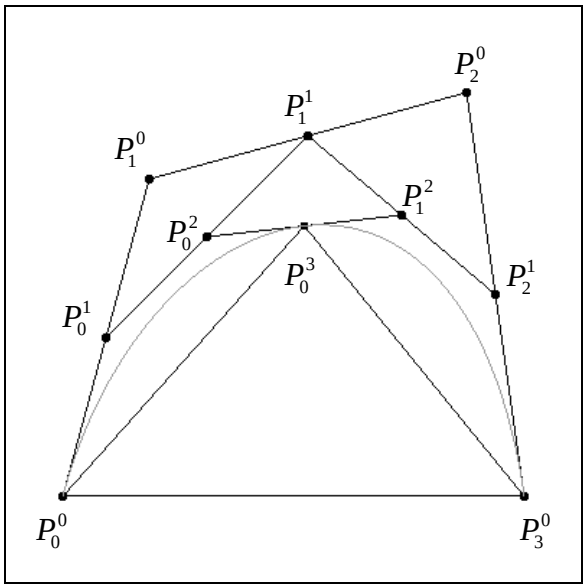

図 2　ド・カステリョのアルゴリズム($t = 1/2$)

３．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$\omega_i : X \to X$、確率p_iによって、

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

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

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

と定義される写像$W : H(X) \to H(X)$は縮小写像になる。よって、点列$\{W^{\circ n}(B)\}_{n=1}^{\infty}$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、

$$\lim_{n\to\infty} W^{\circ n}(B)=A$$

IFS のアトラクタを描画するアルゴリズムには、Deterministic Algorithm と Chaos Game Algorithm の２つがある。このうち、Deterministic Algorithm は、ある基本図形Sを、決められた回数s回アフィン変換したものを、全ての変換の組み合わせに対して描画するアルゴリズムである。

また、Chaos Game Algorithm 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（invariant measure）になる。

本論文では、直線を基本図形とする Deterministic Algorithm を用いた。これは、ステップ数sを小さくすることで、どのような写像が行われているかが確認できるためと、ベジエ曲面に拡張した際、法線ベクトルを与える時に、点よりも線（面）の方が都合がよいと考えたためである。

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。

４．ベジエ曲線から IFS への変換

それでは、実際にベジエ曲線から IFS コードを求める。まず、２次ベジエ曲線の IFS への変換について述べる。アイデアとしては、ド・カステリョのアルゴリズムによるベジエ曲線の分割を用いた。２次ベジエ曲線においては、この方法で容易に IFS コードを求めることができる。これは、２次ベジエ曲線の制御点が３点であることに起因する。つまり、平面を決定するために必要十分な数の点になっているのである。IFS コードを求めるためには、もとの制御点$\{P_0^0,P_1^0,P_2^0\}$を新たな制御点$\{P_0^0,P_0^1,P_0^2\}$及び$\{P_0^2,P_1^1,P_2^0\}$に移すようなアフィン変換を計算すればよい。

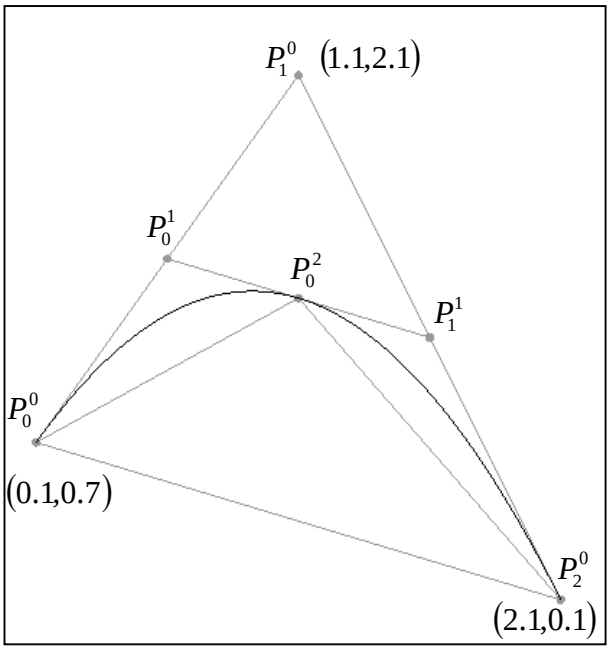

図 3　IFS による２次ベジエ曲線

例として、図３の２次ベジエ曲線から求めたアフィン変換を以下に示す。

$$\omega_1\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}0.5 & 0\\0.35 & 0.25\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}0.05\\0.49\end{pmatrix}$$

$$\omega_2\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}0.5 & 0\\-0.5 & 0.25\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}1.05\\1.125\end{pmatrix}$$

次に、３次ベジエ曲線の IFS への変換について述べる。３次ベジエ曲線の場合、２次元平面上に描きたい時のように、制御点が同一平面上にあると、４点からは平面を決定することができないので、２次ベジエ曲線と同じ方法でアフィン変換を見つけることができない。そこで本論文では、Bernstein 関数による４次元曲線を IFS 化し、それによっ

て得られた点列を、描画したい空間に写像する方法を取った。本手法の基本概念を図４に示す。

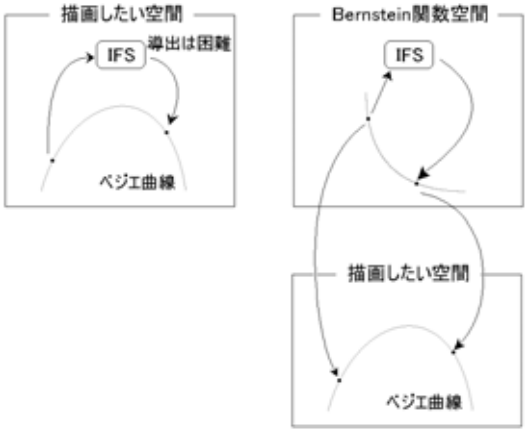

図 4　本手法の基本概念

５．Bernstein 関数曲線

まず、４次元実数空間を考える。この空間上で、曲線$B(t)$を、

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

と定義する。ただし、tは媒介変数で、$0\le t\le 1$
以後、この曲線$B(t)$を、Bernstein 関数曲線と呼ぶ。

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$t=\frac{1}{2}$で２つに分割すると、

$$\omega_1(B(t))=B\left(\frac{1}{2}t\right)$$
$$\omega_2(B(t))=B\left(\frac{1}{2}t+\frac{1}{2}\right)$$

となるはずである。ただし、

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

A_iは４×４の行列、B_iは４次元列ベクトルである。

後はそれぞれの項の係数が合うようにアフィン変換のパラメータを決めればよい。実際に求めたアフィン変換を次に示す。

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

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

ここで一つ注意しなければならないのが、ω_1の１行１列の値、及びω_2の４行４列の値である。これらはベクトルB_1,B_2の取り方によってはどんな値も取ることができるが、アフィン変換が縮小的でなければならないので、絶対値が１以上になってはならない。

これにより、Bernstein 関数曲線を、$\left\{R^4;\omega_1,\omega_2;\frac{1}{2},\frac{1}{2}\right\}$

という IFS によって表すことができた。実際に点をプロットする時は、制御点のベクトルP_iにより、IFS から出てきた点$b=(b_0,b_1,b_2,b_3)\in R^4$を、

$$F(b)=\sum_{i=0}^{3}b_i\cdot P_i$$

と変換して得た点$F(b)$をプロットすればよい。
本手法を用いて描いたベジエ曲線を図５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。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。

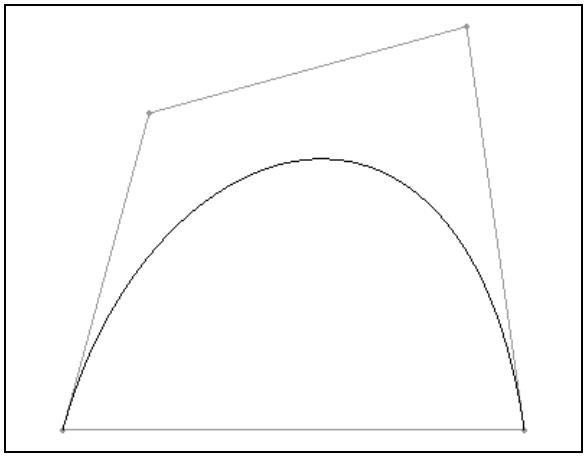

図 5　IFS によるベジエ曲線

描画する時とは逆に、ベジエ曲線上の点から Bernstein 関数曲線の点への写像$F^{-1}(x)$を求めることができる場合、これらを用いて、Bernstein 関数曲線を用いない$\{X;F^{-1}\circ\omega_1\circ F,F^{-1}\circ\omega_2\circ F,1/2,1/2\}$を見つけることができる。しかし、$F$は$R^4$から$R^2$もしくは$R^3$への１次変換であるため、その逆写像$F^{-1}(x)$を一意に求めることは困難である。

６．ベジエ曲面への適用

本論文で用いた手法を、双３次ベジエ曲面に適用した。

双３次ベジエ曲面$P(u,v)$は、１６個の制御点P_{ij}、媒介変数u、vを用いて、

$$P(u,v)=\sum_{i=0}^{3}\sum_{j=0}^{3}B_i^3(u)\cdot B_j^3(v)\cdot P_{ij}$$

と定義される。ただし、$0\le u,v\le 1$

これを、本手法に基づいて、１６次元実数空間上の曲面

$$B(u,v)=\left(B_0^3(u)B_0^3(v),\cdots,B_3^3(u)B_3^3(v)\right)$$

とし、

$$\omega_1(B(u,v))=B\left(1/2\,u,1/2\,v\right)$$
$$\omega_2(B(u,v))=B\left(1/2\,u+1/2,1/2\,v\right)$$
$$\omega_3(B(u,v))=B\left(1/2\,u,1/2\,v+1/2\right)$$
$$\omega_4(B(u,v))=B\left(1/2\,u+1/2,1/2\,v+1/2\right)$$

となるようなアフィン変換$\omega_1,\cdots,\omega_4$を求めた。IFS によって得られた双３次ベジエ曲面を図６に示す。

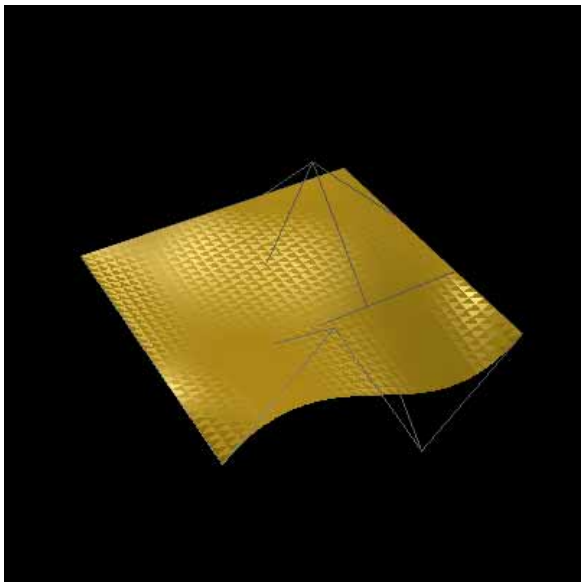

図 6　IFS による双３次ベジエ曲面

７．おわりに

本研究は、ベジエ曲面を用いてモデリングした物体を反復関数系(Iterated Function System : 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、物体のモデリングとテクスチャリングの両方を１つの IFS によって実現することを目的とした。本論文では、そのうち、主に IFS によるベジエ曲線の生成について扱った。

変換の方法として、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、ベジエ曲線の IFS を直接求めることができないために取られた物だが、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。

また、IFS は、Collage の定理により、任意のアトラクタ、及び不変測度を生成可能であることが保証されているが、そのためにいくつの縮小写像が必要なのかは明記されていない。しかし、本手法を用いれば、任意の点で任意の個数に分割できるため、任意の不変測度を生成することが可能であると考えられる。

今後の課題としては、Bernstein 関数曲線を用いない IFS の導出が挙げられる。これは、前述の関数$F^{-1}(x)$を何ら

かの方法で導出することで実現が可能である。

また、本論文の IFS によるベジエ曲線は、既存のベジエ曲線の描画アルゴリズムよりも描画速度が遅い。これは、IFS のアルゴリズムに Deterministic Algorithm を用いているためである。一般には Chaos Game Algorithm の方が高速であるが、前述の法線ベクトルの問題により、本論文では Deterministic Algorithm を使用している。しかし、これは、Point Rendering の概念を用いることで、Chaos Game Algorithm を使用し、高速に描画することが可能になると思われる。

本手法で導出した確率付き IFS において、Chaos Game Algorithm を用いる事により、テクスチャに似た効果を出すことが可能になる。特に本手法は、任意の点で任意の個数の縮小写像に分割可能という特長を持つため、任意の不変測度を設計することが可能であり、柔軟なテクスチャリングができると考えられるが、具体的なテクスチャ設計方法が確立されているわけではない。そのため、不変測度の設計手法に関する研究が必要である。

本手法では、求めた IFS コードのアフィン変換部分を変更することにより、既存の手法では困難な、形状モデルにおける微細構造の表現が可能になると予想される。アフィン変換の変更によるアトラクタの変化の連続性は、Blowing Theorem [3]によって保証されている。しかし、具体的に、アフィン変換をどのように変更すればどのような形状が得られるのかは、よく知られていない。

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、欠点として、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、ということが挙げられる。一方で、山や木など、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 をどのように変更しても、自然物のような統計的フラクタル形状を生成することができない。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こと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することができると考えられる。

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