

MRCMによるフラクタル画像の生成

小島菜摘 (指導教官: 金子 晃)

1 はじめに

自然界のあらゆる場面では、フラクタル図形（どんなに拡大しても複雑な図形）が見られるとされ、自然科学の新たなアプローチ手法とされてきた。更に、コンピュータグラフィックスにおける自然物形状の生成アルゴリズムや、株価の動向など社会的な現象などへの応用も注目されている。今回はフラクタル描画手法の一つとして、IFS を利用したものを取り上げ、その生成原理を調べる。平面でのフラクタル画像を計算し、描画することを目指す。

2 反復関数系 (Iterated function system, IFS) によるフラクタル図形生成

反復関数系は、フラクタルの描画や計算に用いられる。IFS による図形の写像は、自身のいくつかのコピーの和集合からなり、各コピーは関数によって変形されている。一般にそれらの関数は縮小写像であり、自身の縮小コピーを（場合によっては重ね合わせて）まとめたもので、各部を拡大してみると、もとの画像が含まれており、それが永遠に続くため、フラクタルとしての自己相似性が生じているといえる。

3 MRCM : Multiple Reduction Copy Machine (多重縮小複写機)

フラクタル画像生成方法の一つとして、反復関数系の原理を用いた MRCM がある。MRCM とは、いくつかのレンズを使い、原本となる画像を縮小した後、指定した位置に複写した画像を返すシステムである。ここで簡単な例 (図 1-4) としてコンピュータで実際に実装した結果を挙げる。図 1 を原画とし、3 枚の 1/2 縮小レンズを使って相似変換を行い、図 2 のような配置に設定し、システムを反復する。その結果、Sierpinski Gasket (図 4) にたどりつく。同じシステム設定の上で反復すると、原本の画像が何にせよ、一つの共通の画像に近づく。その性質を stability (安定性) という。また、縮小の縦横比率に差をもたせることでアフィン変換を、凹凸レンズを用いることで非線形変換を実現できる。



図 1: 原本 (白黒)

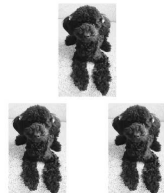


図 2: 1 回変換後

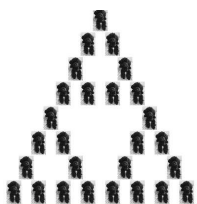


図 3: 3 回変換後

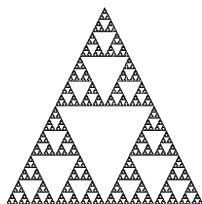


図 4: 8 回変換後

4 アフィン変換

MRCM のレンズシステムは、平面でのアフィン線形変換を用いている。座標を x 軸、 y 軸とし、その平面上の点を P とする。よって、 $P = (x, y)$ と書ける。

すべての点は実数による四則演算が可能であり、

$$P_1 + P_2 = (x_1 + x_2, y_1 + y_2)$$

$$sP = (sx, sy)$$

と表される。

線形変換 F は、すべての点 P に $F(P)$ を対応させるもので、縮小・回転・反転を実現でき、行列により表現される。さらに、点 Q の座標値を与えて加えることで、平行移動変換もできる。よって、アフィン線形変換 w は 6 つの変数によって与えられ、 $P = (x, y)$ 、 $w(P) = (u, v)$ とするとき、

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

または、

$$w(x, y) = (ax + by + e, cx + dy + f)$$

と表現できる。

5 定義

MRCM のレンズシステムは、アフィン変換の集合、 w_1, w_2, \dots, w_N で説明される。初期画像 A が与えられた時、縮小アフィンコピー、 $w_1(A), w_2(A), \dots, w_N(A)$ が作られる。最終的に、それらすべてのコピーを 1 つのイメージ画像とし、それを出力 $W(A)$ とすると、

$$W(A) = w_1(A) \cup w_2(A) \cup \dots \cup w_N(A).$$

で表される。この時、 W は Hutchinson operator と呼ばれる。式から分かるように、MRCM を動かすことで演算子 W が反復される。これは、IFS の核心である。初期集合 (画像) A_0 から始めると、 $A_1 = W(A_0)$ 、 $A_2 = W(A_1)$ 、 \dots が得られる。これを一般化すると、

$$A_{k+1} = W(A_k), \quad k = 0, 1, 2, \dots,$$

となる。このようにして、IFS は最終画像 A_∞ に至る。この時、 A_∞ を IFS (または MRCM) のアトラクターと呼び、IFS によって不変となる：

$$W(A_\infty) = A_\infty.$$

つまり、 A_∞ は W の不変集合といえる。

6 実装結果 1 (平面の場合)

実際に、IFS と MRCM の原理を用いて、コンピュータ上でいくつかの例を実装した。6 つの変数 a, b, c, d, e, f によって 1 つの変換レンズが与えられるとし、その値

を変えることで、多種多様なフラクタル構造をもつ画像を生み出せる。また、変換レンズの枚数を増やすことでも描画の可能性を上げられる。図5,6は、初期画像を図1として、表1,2の変数を使い画像変換し、反復した結果である。図5は3つの変換を、図6は4つの変換を用いて描画されたものである。また、同じ変換を用いたとき、初期画像が何であっても、最終的に同じ画像(アトラクター)にたどりつくことを実験で確認した。

更に、カラー画像を用いて実装した場合、図9で示したように、反復するたびに色の判別が難しくなっていく、一見、最終画像が白黒に近づいていくように見える。白黒画像を用いて反復した時とは違い、反復すればするほど全体画像が見えづらくなっていく。

	a	b	c	d	e	f
1	0.387	0.430	0.430	-0.387	0.2560	0.5220
2	0.441	-0.091	-0.009	-0.322	0.4219	0.5059
3	-0.467	0.020	0.113	0.015	0.4000	0.4000

表 1: 木の枝 (図 5) を描画するための変数

	a	b	c	d	e	f
1	0.255	0.000	0.000	0.255	0.3726	0.6714
2	0.255	0.000	0.000	0.255	0.1146	0.2232
3	0.255	0.000	0.000	0.255	0.6306	0.2232
4	0.370	-0.642	0.642	0.370	0.6356	-0.0061

表 2: クリスタル (図 6) を描画するための変数

	a	b	c	d	e	f
1	0.000	-0.500	0.500	-0.000	0.5000	0.0000
2	0.000	0.500	-0.500	0.000	0.5000	0.5000
3	0.500	0.000	0.000	0.500	0.2500	0.5000

表 3: クリスマスツリー (図 7) を描画するための変数

	a	b	c	d	e	f
1	0.849	0.037	-0.037	0.849	0.0750	0.1830
2	0.197	-0.226	0.226	0.197	0.4000	0.0490
3	-0.150	0.283	0.260	0.237	0.5750	-0.0840
4	0.000	0.000	0.000	0.160	0.5000	0.0000

表 4: シダの葉 (図 8) を描画するための変数

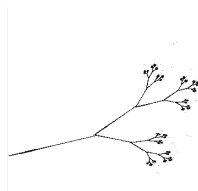


図 5: 木の枝



図 6: クリスタル

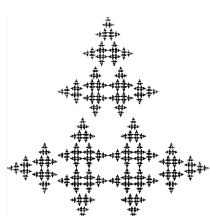


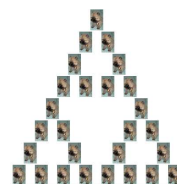
図 7: クリスマスツリー



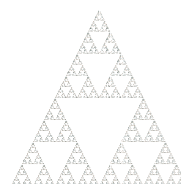
図 8: シダの葉



図 9: 原本 (カラー)



3 回変換後



10 回変換後

7 実装結果 2 (立体の場合)

MRCMの原理を用いて、平面における反復写像の生成を実装でき、その生成された写像は、フラクタルの性質を持つことを確認できた。

更に、描画の可能性を上げるべく、今まで紹介した平面でのフラクタル描画だけでなく、3次元での描画にも踏み込んだ。例えば、平面での描画に成功した Sierpinski Gasket(図 4 参照) の図形を 3次元で表現する場合について考える。平面で実装した時は、3枚の変換レンズを用いたが、立体で表現する場合、4枚の変換レンズが必要となる。つまり、図 10 を初期画像とすると、図 11 のように 4つの写像が生成されるようにする。この時の変換として、それぞれ 1/2 の縮小を施した後、以下の値 (表 5) によって平行移動を行った。

	x	y	z
1	0.000	0.000	0.354
2	0.577	0.000	-0.204
3	-0.289	0.500	-0.204
4	-0.289	-0.500	-0.204

表 5: 3D Sierpinski Gasket 生成のための平行移動変数

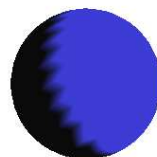


図 10: 原本

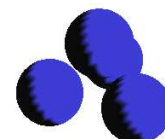


図 11: 1 回変換後

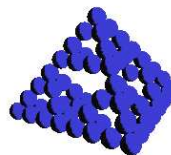


図 12: 3 回変換後

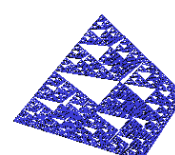


図 13: 7 回変換後

このような変換を反復していくと、図 12 のような立体 Sierpinski Gasket を描くことができる。また、キー操作によって物体を回転させ、様々な角度から立体を見ることで、よりフラクタル構造を確認しやすい形で実装ができた。他にも、木の枝やクリスタルなどの 3D 化の実現も考えられる。

参考文献

- [1] Heinz-Otto Peitgen, Hartmut Jurgens, and Dietmar Saupe: "Fractals for the classroom", Springer-Verlag, New York, 1992, 318pp.
- [2] Hutchinson, J., Fractals and self-similarity, Indiana University Journal of Mathematics 30, 1981, 713-747.