

# フラクタルコーディングによる対話型画像圧縮システム

2P-11

大石 義明<sup>†</sup> 山本 強<sup>†</sup> 青木 由直<sup>†</sup>  
 (†北海道大学工学部 †北海道大学大型計算機センター)

## 1. はじめに

フラクタルモデルを用いた画像圧縮法が提案されている<sup>[1]</sup>。圧縮の基本概念となるものはIFS理論であり、対象画像のパラメータ(IFSコード)を抽出することにより、それと乱数を用いて画像が再生可能となる。しかしIFSコードの完全自動抽出のアルゴリズムは未だ発見されておらず、対話的情報に依存し、試行錯誤で発見する必要がある。

そこで我々は対象画像を2値画像に限定し、IFSコードの対話的抽出支援システムを試作した。本稿では、本システムの概要と本システムに用いられているIFSコード抽出法について述べる。本手法では、対話的にIFSコードを与えた後、解像度を変化させながら対象画像と生成画像が可能な限り近似するような手続きを与える。対象画像と生成画像の近似評価はその差分を求めることにより行われる。

## 2. IFS(Iterated Function System)理論

### (1) 図形のモデリング

対象図形のモデリングはその図形のIFSコードの生成・抽出と同義であり、Collage Algorithm(以下C.A.)が用いられる。C.A.とは、対象図形 $T$ 自身をその図形の縮小・回転・平行移動した図形 $\tilde{T}$ で完全に被覆することである。個々の変換は affine 変換で記述可能であり、 $\tilde{T}$ は各々の affine 変換  $w_n$  ( $n:1, \dots, N$ ) の和集合として表される。

$$\tilde{T} = \bigcup_{n=1}^N w_n(T)$$

変換の数と各々の変換による重複部分はできるだけ少ない方が望ましい。この $N$ 個の affine 変換  $w_n$  と各々が適用される確率  $p_n$  のセットを IFS コードと呼ぶ。fig.1 のシダの葉の図形は table.1 の3組の IFS コードをもとにレンダリングされた結果である。

	$a_{11}$	$a_{12}$	$a_{21}$	$a_{22}$	$b_1$	$b_2$
$w_1$	0.836	0.044	-0.044	0.836	0.000	0.169
$w_2$	-0.141	0.302	0.302	0.141	0.000	0.127
$w_3$	0.141	-0.302	0.302	0.141	0.000	0.169

table.1 シダの葉の IFS コード

### (2) 図形のレンダリング

IFSコードを用いた図形のレンダリング方法には、いわゆる

ランダムウォークの手法が用いられる。初期座標  $(x_0, y_0)$  に対し、IFSコードの確率  $p_n$  に従い、一組の affine 変換  $w_n$  を選択し適用する。その後も変換された点に対して同様の操作を繰り返す。以上より求められた点の軌跡(集合体)が生成図形  $A$  (アトラクタ) となる。

### (3) 図形の近似度

対象図形と生成図形の近似の指標として Hausdorff 距離が用いられる。2次元空間領域  $X$  と  $Y$  の Hausdorff 距離  $h(X, Y)$  は以下のように定義される。

$$h(X, Y) = \max\{d(X, Y), d(Y, X)\}$$

ただし  $d(X, Y)$  は  $X$  から  $Y$  への距離を示す。各 affine 変換の Lipschitz 定数中の最大値を  $s$  とすると、対象図形  $T$  と生成図形  $A$  が十分に近似した状態であるとき、2つの図形は以下を満足する。ただし  $\epsilon$  は非負数である。

$$h(T, \tilde{T}) < \epsilon \text{ ならば } h(T, A) < \frac{\epsilon}{1-s}$$

## 3. IFSコード抽出における問題点

現状では、対話的に試行錯誤で IFSコードを与えなければ、期待される図形は生成不可能である。より具体的な問題として (a) 対象図形を構成する IFSコードの数が未知であること、(b) Hausdorff 距離の測定に膨大な計算量を必要とすること、等が挙げられる。

(a) については、与える IFSコードの数の組合せが無数であるから、あらかじめ人為的に決定する必要がある。(b) については、生成図形は点の集合体であるから、点の個数を  $n$  とすると Hausdorff 距離の算出に必要な計算量のオーダーは  $O(n^2)$  となり、 $n$  の値が大きい場合には発散する。従って、実際の抽出過程には Hausdorff 距離計算よりも簡略化された近似評価関数の導入が不可欠である。

## 4. 本システムの概要

我々が制作したシステムは、C.A.を対話的に実現し、IFSコード抽出を支援するものである。本システムにおける入力例を fig.1 に示す。ユーザは計算機上でマウスのみを用いて対象図形の IFSコードの抽出を行うことが可能である。さらにメニューを選択することにより、IFSコードの

修正・更新等も可能である。

基本的な操作手順は以下の通りである。(1) 圧縮対象図形  $T$  を画面上にロード。 $T$  を包含する最小の長方形  $R$  が描かれる。(2) ユーザはマウスを用い  $T$  に対して C.A. を適用。 $T$  を  $N$  個の平行四辺形で覆うようにする。(3) システムによる IFS コードの算出。(4) (3) の IFS コードによる図形のレンダリング。また、本システムでは入力した平行四辺形の面積比により、各々の変換を適用する確率を決定する。

5. 本システムにおける IFS コード抽出法

上述のような対話的な入力による生成図形  $A$  は、必ずしも対象図形  $T$  に対して十分近似しているとは限らない。そこでユーザによる入力の後に以下に示す手法により近似解を求める。(簡単化のため画像サイズを  $2^n \times 2^n$  とする。)

[Step 1] 解像度  $r$  ( $0 \leq r \leq n$ ) における  $T$  と  $A$  の画像  $T(r), A(r)$  を生成。各画像は  $2^r \times 2^r$  の行列  $a_{ij}(i, j : 1, \dots, 2^r)$  として表される (以下は  $r=2$  の例)。ただし 1 と 0 は黒および白の領域を示す。

$$T(r) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, A(r) = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

[Step 2]  $T(r)$  と  $A(r)$  の差分画像  $D(r)$  を生成。

$$D(r) = T(r) - A(r) = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

[Step 3]  $D(r)$  の各行列要素の二乗和  $S$  を計算。

$$S = \sum_{j=1}^{2^r} \sum_{i=1}^{2^r} a_{ij}^2 = 3$$

[Step 4]  $S$  を減少させるように IFS コードを変動。

以上の 4Step を  $S$  が最小になるまで繰り返す。その後、解像度を上げて ( $r = r + 1$ ) 同様の操作を繰り返す。

IFS コードの変動は一組ごとに行う。その優先順位は変換後の平行四辺形の面積の降順に従う。これは、生成図形  $A$  の形状が最大の平行四辺形に変換する IFS コードにより大きく左右されるという性質があることによる。そしてその段階における最適解の発見後に他の IFS コードの変動に移る。そのためにあらかじめ平行四辺形の面積の降順に IFS コードをソートしておく必要がある。

6. 今後の課題とまとめ

本手法における最適解抽出の要は多変量 (6 変数: IFS コード) 関数の最小化であるが、その計算量は初期値に依存し、最悪の場合には発散する恐れがある。そのため、IFS コードの変動方法に、より効率的な条件を与える必要がある。今後は各変数に対しての重み付け等を行い、計算の効率化を図るつもりである。

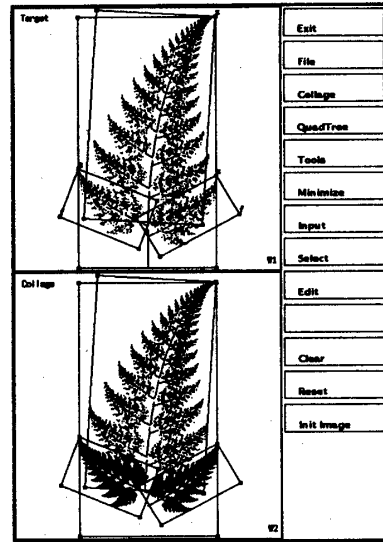


fig.1 対話型システムにおける入力例

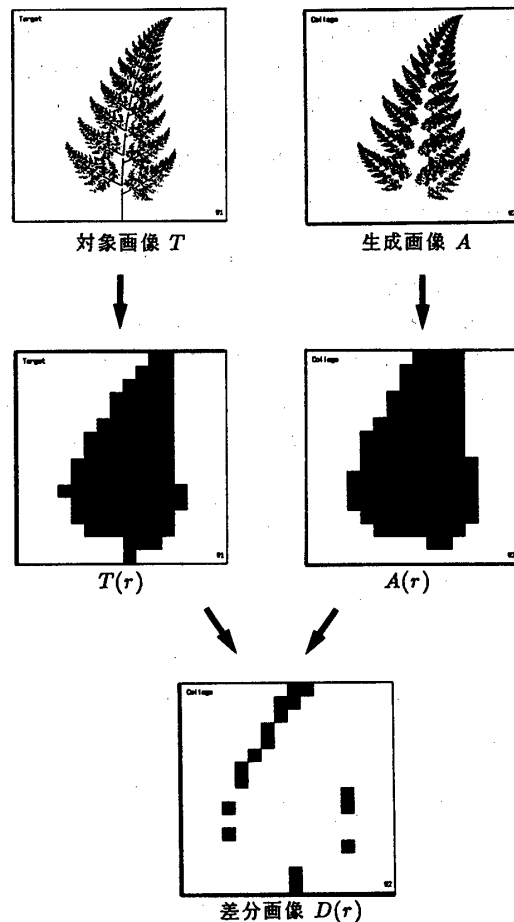


fig.2 本システムの IFS コード抽出手順

\* 参考文献 \*

[1] Barnsley, M.F et al., "Harnessing Chaos for Image Synthesis", SIGGRAPH '88 Conf. Proceedings, pp.131-140