APPLICATION OF ITERATED FUNCTION SYSTEMS AND SKELETONIZATION TO SYNTHESIS OF FRACTAL IMAGES

Ran Libeskind-Hadas
Department of Computer Science
University of Illinois
Urbana, IL 61801

and

Petros Maragos
Division of Applied Sciences
Harvard University
Cambridge, MA 02138

ABSTRACT. In this paper we introduce some new techniques for modeling fractal images using concepts from the theory of iterated function systems and morphological skeletons. In the theory of iterated function systems, a fractal image can be modeled arbitrarily closely as the attractor of a finite set of affine maps. We use the morphological skeleton to provide us with sufficient information about the parameters of these affine maps. This technique has applications for fractal synthesis, computer graphics, and coding. Images that exhibit self-similarity, such as leaves, trees, mountains, and clouds, can be easily modeled using these techniques. Slight perturbations in the parameters of the model create variations in the image that can be used in animation. Finally, the small number of parameters in the model allows for very efficient image compression.

1 Introduction

"I wonder about trees" —Robert Frost

Many natural objects exhibit a very high level of geometrical complexity. When a small region of such an object is examined closely, a surprising level of detail is revealed. Such objects are known as *fractals* (a fractal is more formally defined by Mandelbrot [8] as a set whose Hausdorff-Besicovitch dimension is greater than its topological dimension) and a variety of techniques have been developed to model these objects.

For a large class of natural images such as leaves, trees, mountains, and clouds, the detail of the object closely resembles the object as a whole. This paper discusses a technique for modeling such *self-similar* fractals.

Demko, Hodges, and Naylor [4] provide a summary of the previously used techniques to model self-similar objects. Some of the techniques are highly specialized in the sense that they effectively model only one type of object, while other techniques are not intuitive and thus are ill-suited for interactive design and modeling. Many of the fractal techniques offer very limited control, such as the specification of the fractal dimension, and are therefore of limited use in modeling specific fractal objects.

The technique introduced here is based on the theory of iterated function systems. An important contribution to this area is Barnsley's Collage Theorem which states that if an image can be roughly covered with smaller affine transformations of itself then an approximation to the image can be reconstructed by computing the *attractor* of the set of affine transformations.

While this theory is potentially very useful in modeling a large class of images, until now there has been no systematic method for actually finding the parameters of the affine transformations. Our method is based on first computing the *skeleton*, a thinned caricature of the image, and using the information provided by the skeleton to completely determine the affine transformations. This leads to a very simple interactive modeling tool and our algorithms are well-suited for complete automation of the modeling process.

In Sections 2 and 3 we review the necessary mathematical background. Section 2 reviews the theory of iterated function systems while Section 3 deals with the basic operations of mathematical morphology and the skeleton. In Section 4 we incorporate these concepts into a series of new algorithms for the interactive solution of the inverse problem. In Section 5 we summarize our results and make suggestions for future research.

2 Iterated Function Systems

Big whirls have little whirls;
That feed on their velocity;
And little whirls have lesser whirls,
And so on to viscosity.
— L. F. Richardson

In this section we begin with the classical definition of the Hausdorff metric and then introduce the notions of *contraction mappings*, *iterated function systems*, and *attractors*. Finally, we present a surprising result, due to Barns-

ley, which provides us with the mathematical basis necessary to model self-similar and "pseudo-self-similar" binary images.

Let $\mathcal{F}(X)$ be the set of all nonempty closed and bounded subsets of the metric space (X, d). Then the Hausdorff distance function is defined by

$$h(A,B) = \max\{\sup_{x\in A}\inf_{y\in B} d(x,y), \sup_{x\in B}\inf_{y\in A} d(x,y)\}$$

for $A, B \in \mathcal{F}(X)$. $f : X \to X$ is a *contraction* if there exists a number $\alpha < 1$ such that $d(f(x), f(y)) < \alpha d(x,y)$ for all $x, y \in X$. If $f_i : X \to X$, $1 \le i \le n$, are contraction mappings, then $\{X, f_i \mid 1 \le i \le n\}$ is said to be an *iterated function system*. Let us define $f_i(A)$ by $f_i(A) = \{f_i(x) \mid x \in A\}$, $1 \le i \le n$. Then, Hutchinson [5] has shown that the function $F : \mathcal{F}(X) \to \mathcal{F}(X)$ defined by $F(A) = \bigcup_{1\le i\le n} f_i(A)$, $A \in \mathcal{F}(X)$, is a contraction mapping with respect to the Hausdorff metric h.

Let us define $F^n(A)$ recursively by $F^n(A) = F(F^{n-1}(A))$ where $F^0(A) = A$. Then the *attractor*, $\mathcal{A}$, of the iterated function system $\{X, f_i \mid 1 \le i \le n\}$ is defined by $\mathcal{A} = \lim_{n\to\infty} F^n(A)$ for $A \in \mathcal{F}(X)$. An important property of the attractor is that $\mathcal{A}$ is independent of $A \in \mathcal{F}(X)$ [7].

Theorem 1 (Barnsley [2]) *Let $\{X, f_i \mid 1 \le i \le n\}$ be an iterated function system and let $S \in \mathcal{F}(X)$ with $h(F(S), S) < \varepsilon$, $\varepsilon > 0$. Then $h(\mathcal{A}, S) < \frac{\varepsilon}{1-m}$ where $\mathcal{A}$ is the attractor of the iterated function system and m is the smallest number such that $d(f_i(x), f_i(y)) \le m d(x,y)$ for all $x, y \in X$ and $1 \le i \le n$.*

This theorem states that if we can "approximately" cover an image S with smaller transformations of itself, then the attractor of these transformations approximates the image S.

We will find it convenient to generalize somewhat the standard definition of the iterated function system by adding a fixed set map to the system. That is, given an iterated function system $\{X, f_i \mid 1 \le i \le n\}$ with $F(A)$ defined to be $\bigcup_{1\le i\le n} f_i(A)$ and given a fixed set C, we can define a new iterated function system $\{X, C, f_i \mid 1 \le i \le n\}$ and an associated set map $G : \mathcal{F}(X) \to \mathcal{F}(X)$ defined by

$$G(A) = F(A) \cup C = \left[\bigcup_{1\le i\le n} f_i(A)\right] \cup C\,, \quad A \in \mathcal{F}(X)$$

Since $h(A\cup C, B\cup C) \le h(A,B)$ and F is a contraction map with respect to h, we see that $h(A,B) > h(F(A),F(B)) \ge h(F(A)\cup C, F(B)\cup C) = h(G(A),G(B))$ and thus G is a contraction mapping with respect to h. Therefore, the attractor of G is well-defined. Now, by Barnsley's Theorem, if we can "approximately" cover an image S with smaller transformations of itself *and* a fixed set, then the attractor of this system approximates the image S. For example, each tree in Figure 1 was produced by using a fixed set map for the trunk and affine transformations for the branches. Figure 2 demonstrates these transformations for a simple tree. The tree was computed as the attractor of three affine transformations (see the table in Figure 2) and a fixed set "trunk." The fixed set is comprised of two ellipses with major axis of length 60 and minor axis of length 10 centered at (0,0) and (0,20). The bottom of the "trunk" was later truncated to make the base of the "trunk" flat. The process of actually computing the attractor is discussed below.

In practice, it is inefficient to compute the attractor directly from its definition by computing $G(A)$, $G(G(A))$, etc. If A is much larger than the attractor, or the value of m in Barnsley's Theorem is close to 1, or there are many transformations in the system, then the number of iterations in this process may be quite large. Furthermore, in a finite number of iterations it is only possible to achieve an approximation to the attractor. However, Barnsley and Demko [1] propose an efficient and exact computation of the attractor that works as follows: Let $\{X, C, f_i \mid 1 \le i \le n\}$ be an iterated function system with attractor $\mathcal{A}$. Let $p = (p_0, \ldots, p_n)$ be a probability vector ($p_i > 0$ and $\sum_{i=0}^{n} p_i = 1$) and let $x \in X$. Choose C with probability p_0 and f_i with probability p_i at random. If C is selected then choose a new $x \in C$ at random and plot x. If f_i is chosen, apply f_i to x, plot $f_i(x)$, and set $x := f_i(x)$. This process is repeated for a fixed number of steps. For example, the tree in Figure 2 was computed using 500,000 iterations of this algorithm with equal probability assigned to each of the three affine transformations and the fixed set map.

If point x is initially chosen inside $\mathcal{A}$ then each iteration will keep the point inside $\mathcal{A}$ since points cannot "escape" from the attractor. If point x is initially chosen outside the attractor then we can first perform a few iterations *without* plotting to bring the point *inside* the attractor. This algorithm only plots points inside the attractor and the probability vector effects the distribution of points and not the geometry of the attractor. A sufficiently large number of iterations of this process will cover the entire attractor.

3 Mathematical Morphology

Mathematical morphology extracts geometrical information about an image (represented as a set) by performing a set transformation of the image with respect to a simpler and smaller object called a *structuring element* (represented by a compact set). The two most fundamental morphological transformations are *erosion* and *dilation*. Let A be an arbitrary set in $\mathbf{R}^2$, where $\mathbf{R}$ is the set of real numbers. Let B be a compact symmetric set (with respect to the origin) in $\mathbf{R}^2$. Then the erosion of A by B, denoted by $A \ominus B$, is defined to be $\bigcap_{b\in B} A_b$ where A_b is the translate of A centered at b (that is, $A_b = \{b + x \mid x \in A\}$). The dilation of A by B, denoted by $A \oplus B$, is defined to be $\bigcup_{b\in B} A_b$ [10]. If B contains the origin of the plane then erosion shrinks the image while dilation expands the image. The morphological *opening* of A by B, denoted by $A \circ B$, is defined to be $(A \ominus B) \oplus B$ [10]. An important property of the opening is that $A \circ B$ is the union of all translates of B which are contained in A.

If A is a set in the plane then the disk rD_x of radius r and centered at x is *maximal* with respect to A if it is contained in A and is not properly contained in any other disk contained in A. Blum [3] defines the skeleton (medial

axis) of A, denoted by $\mathrm{SK}(A)$, to be the set of centers of all disks maximal with respect to A. To each point, x, in the skeleton corresponds the radius of the maximal disk centered at x. The *skeleton function* $S(x)$ returns the radius of the maximal disk centered at x. The r^{th} *skeleton subset* of $\mathrm{SK}(A)$, denoted by $\mathrm{S}_r(A)$, is defined to be the set of all skeleton points x such that $S(x) = r$. An important property of the skeleton function is that it contains all the information necessary to reconstruct the original image. That is, $A = \bigcup_{r>0} [\mathrm{S}_r(A) \oplus rD]$. This fact has been used to efficiently encode binary images by their skeletons [9].

Lantuejoul [6,11] has shown that the skeleton can be alternatively defined by morphological operations as

$$\mathrm{SK}(A) = \bigcup_{r>0} \mathrm{S}_r(A) = \bigcup_{r>0} [(A \ominus rD) - (A \ominus rD) \circ drD]$$

where rD denotes the open disk of radius r, drD is the closed disk of infinitesmally small radius dr, and "$-$" denotes set difference.

The above concepts can be equally well applied to discrete binary images. If A and B are subsets of $\mathbf{Z}^2$, where $\mathbf{Z}$ denotes the set of integers, the definitions of erosion, dilation, and opening are still meaningful [11]. In order to generalize the concept of the discrete skeleton [9], we choose a finite discrete symmetric structuring element of approximately circular shape, D, and define this shape as having *size 1*. Now, rD, the disk of discrete *size* r, is defined to be

$$rD \stackrel{\text{def}}{=} \underbrace{D \oplus D \oplus \ldots \oplus D}_{r}$$

where $r = 0, 1, 2, 3, \ldots$. If $r = 0$, then $rD = \{(0,0)\}$ by convention. Then, the r^{th} skeleton subset is $\mathrm{S}_r(A) = (A \ominus rD) - (A \ominus rD) \circ D$, and the discrete skeleton is

$$\mathrm{SK}(A) = \bigcup_{r=0}^{n} \mathrm{S}_r(A) = \bigcup_{r=0}^{n} [(A \ominus rD) - (A \ominus rD) \circ D]$$

where $n = \max\{k \mid A \ominus kD \neq \emptyset\}$. Notice that the discrete skeleton may be disconnected. Maragos and Schafer describe an $\mathrm{O}(n)$ algorithm for computing the discrete skeleton of a set A, where $n+1$ is the number of skeleton subsets of $\mathrm{SK}(A)$ and the unit operation is an image erosion [9].

4 New Techniques

In order to apply Barnsley's Theorem, we must be able to find the set of transformations and the fixed set map that comprise the iterated function system. In this section we present some new techniques for solving this problem. Specifically, we use the information provided by the skeleton to find the affine transformations and the fixed set map.

Our study began with the investigation of several "synthetic" self-similar objects (These objects are called "synthetic" because they can be defined precisely by a simple construction rule). Perhaps most famous among such objects is the Triadic Koch Island [8], K, shown in Figure 3. The boundary of this fractal is generated by constructing an equilateral triangle whose edges are recursively defined in terms of smaller versions of themselves by the construction rule shown in Figure 4.

The Triadic Koch Island, K, is self-similar because it is defined in terms of smaller versions of itself. Notice that each of the six "protruding parts" is itself a Triadic Koch Island of size one-third that of the original. Therefore, this fractal can be *exactly covered* by six affine transformations of itself (Figure 5) and a fixed disk that covers the remaining portion or "bulk" of the image. The six affine transformations have the form:

$$f_i \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{3} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad 1 \leq i \leq 6$$

where $\begin{pmatrix} x_i \\ y_i \end{pmatrix}$ is the appropriate shift vector.

For the Triadic Koch Island, the problem of finding these shift vectors is a simple exercise. However, we will use this example to demonstrate an observation that will be of great importance in modeling more complex objects. The skeleton of K is shown in Figure 6. The central branch point (labelled in Figure 6) corresponds to the center of the image. Each of the outer branch points (labelled in Figure 6) corresponds to the center of a smaller Triadic Koch Island. Therefore, the six shift vectors are the six vectors emanating from the center and terminating at the branch points. If the coordinates of the center point and the six branch points are known then the six affine transformations can now be found.

In general, since self-similar objects are comprised of smaller rotated and shifted versions of themselves, we can use for the contraction mappings in the iterated function system affine transformations of the form:

$$f_i \begin{pmatrix} x \\ y \end{pmatrix} = r_i \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \end{pmatrix}, 1 \leq i \leq n$$

where $0 \leq r_i < 1$, $0 \leq \theta_i < 2\pi$. However, these affine transformations may not completely cover the image and it is convenient to cover the remaining bulk of the image with a fixed set comprised of disks. Disks are particularly convenient for two reasons. First, it is important that the fixed set be simple to encode. A disk can be easily encoded with three real parameters (radius, x- coordinate of center, and y- coordinate of center). Of course, the bulk could also be covered with affine transformations, but since the bulk is not in general similar to the original image, this approach is quite cumbersome. Furthermore, disks relate directly to the concept of the skeleton, and thus the bulk-covering disks can easily be found from the skeleton.

An Interactive Modeling Tool

Employing the ideas in the above discussion, we have developed an interactive tool for modeling fractal binary images. Given an image A to be modeled, we first compute the skeleton of the image. The user locates the central branch point, a, and a branch point, b, corresponding to the scaled, rotated, and shifted copy of the image, B. The shift element of the affine transformation carrying A to B is the vector from a to b. Since the ratio of the sizes of B to A is the ratio of sizes of the largest maximal disks in B and

A, respectively, the scaling factor of the transformation is $r = S(b)/S(a)$ where S is the skeleton function.

The problem of finding the angle of rotation, θ_i, is somewhat more difficult. If the image A is comprised of smaller rotated and shifted but not distorted copies of itself then we can apply the transformation, f_i, with several values for θ_i and find the value, θ_i^*, that provides the best fit. We solve the problem of finding this optimum value by minimizing a set-theoretic error criterion. Specifically, our method involves computing first $A \circ kD$, the k^{th} opening where $k = S(b)$, eliminating all detail up to disks of radius k from the image [9]. We then superimpose $f_i(A)$ on the opening and try to find the value of θ_i for which $(A \circ kD) \cup f_i(A)$ "best matches" A. This is done by minimizing the area of the symmetric set difference between $(A \circ kD) \cup f_i(A)$ and A (The *symmetric set difference* of sets X and Y is defined by $X \Delta Y \stackrel{\text{def}}{=} (X \cup Y) - (X \cap Y)$). The algorithm is given below.

Algorithm 1

begin
min.area := BIGNUMBER;
min.theta := 0;
opening := $(A \ominus kD) \oplus kD$;
for $\theta_i := 0$ **to** 2π **step** *incrementsize* **do**
begin
for all $(x, y) \in A$ **do**

$$f_i \begin{pmatrix} x \\ y \end{pmatrix} = r_i \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \end{pmatrix};$$

symmetric.difference := $(opening \cup f_i(A)) \Delta A$;
area.of.difference := area(symmetric.difference);
(The area function computes the*
*area (cardinality) of a set *)*
if *area.of.difference* < *min.area* **then**
begin
min.area := area.of.difference;
min.theta := θ
end
end;
return*(min.theta)*
end

The stipulation that the image be comprised of undistorted transformations of itself is in general only valid for synthetic self-similar images. Natural self-similar images are comprised of somewhat distorted copies of themselves and thus the "collage" of affine transformations will be imperfect. For such images, Algorithm 1 will not perform well since the best fit transformation may not correspond to the self-similarity transformation.

For these cases we use a different technique to compute the angle θ_i. The user locates the branch point, b, as well as several other points on the same branch. We then apply a least square fit to these points to give us the slope, m, of the branch and thus $\theta_i = \arctan(m) + k\pi$ where k is 0 or 1 depending on the orientation of the branch. Also, due to distortion, the x- and y- scalings of the self-similar region may be different and instead of a single scaling factor, r_i, we may consider seperate x- and y- scalings, $r_{x,i}$ and $r_{y,i}$, so that the transformation has the form:

$$f_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r_{x,i}\cos\theta_i & -r_{y,i}\sin\theta_i \\ r_{x,i}\sin\theta_i & r_{y,i}\cos\theta_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

where $r_{x,i},\ r_{y,i} < 1$. To compute $r_{x,i}$ and $r_{y,i}$ we use an algorithm similar to Algorithm 1. We first compute $A \circ kD$ and then superimpose $f_i(A)$ with varying values of $r_{x,i}$ and $r_{y,i}$, attempting to minimize $((A \circ kD) \cup f_i(A)) \Delta A$. This algorithm is given below.

Algorithm 2

begin
(User specifies central branch point, a,*
and a branch point b. LOFACTOR is a real number
less than 1 and HIFACTOR is a real number greater
*than 1. *)*
min.area := BIGNUMBER;
$r := S(b)/S(a)$;
for $r_{x,i}$:= *(r*LOFACTOR)* **to** *(r*HIFACTOR)*
step *incrementsize* **do**
for $r_{y,i}$:= *(r*LOFACTOR)* **to** *(r*HIFACTOR)*
step *incrementsize* **do**
begin
for all $(x, y) \in A$ **do**

$$f_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r_{x,i}\cos\theta_i & -r_{y,i}\sin\theta_i \\ r_{x,i}\sin\theta_i & r_{y,i}\cos\theta_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \end{pmatrix};$$

symmetric.difference := $((A \circ kD) \cup f_i(A)) \Delta A$;
area.of.difference := area(symmetric.difference);
if *area.of.difference* < *min.area* **then**
begin
$r^*_{x,i} := r_{x,i}$;
$r^*_{y,i} := r_{y,i}$
end
end;
return $(r^*_{x,i},\ r^*_{y,i})$
end

Once the affine transformations for the iterated function system are found, the fixed set can be found. The user views the original image with the affine transformations superimposed on the image and then fills the uncovered portion of the image with maximal disks. The maximal disks are found by locating skeleton points in uncovered regions of the image. Each skeleton point corresponds to the center of a maximal disk and the radius of the disk can be computed from the skeleton function.

Figure 7 demonstrates Algorithm 2 applied to a maple leaf. The attractor was computed using three affine transformations (see the table in Figure 7) and a fixed set disk of radius 20 centered at (0,-3). The attractor was then processed with a sequence of erosions and dilations to give an image more closely resembling the original leaf.

5 Conclusion

In this paper we have demonstrated an application of iterated function systems and mathematical morphology to

techniques for modeling self-similar images. Using iterated function systems alone, it is possible to construct self-similar images such as the trees in Figure 1. Using the techniques that we have described here, it is possible to solve the inverse problem and construct a model for any self-similar image.

Two obvious applications of this technique are to image compression and animation. The leaf in Figure 7 was modeled by three affine transformations and a bulk-filling disk corresponding to an information rate of approximately 0.038 bits per pixel.

Figures 8a, 8b, and 8c show three frames of a bending tree obtained by slightly varying the angular components in the affine transformations. This demonstrates a very nice application to animation.

Finally, the process that we have described here could potentially be fully automated. The difficulty of automation is primarily in the detection of branch points. This problem could perhaps be resolved by using a connected skeleton algorithm modified slightly to contain the information of the skeleton function. An automated algorithm for filling in the "bulk" is less difficult and could be implemented, for example, by choosing skeleton points (beginning with the last skeleton subset) and drawing the corresponding maximal disks until the "bulk" is completely covered.

Acknowledgements.
We wish to thank Prof. Roger Brockett and the Harvard Robotics Laboratory, where this research was done, for their hospitality. P. Maragos was supported by the National Science Foundation under Contract CDR-85-00108, and partially by a National Science Foundation PYIA under Grant MIP-8658150 with matching funds from Bellcore, IBM, and Xerox.

References

[1] Barnsley, M. F. and S. Demko, "Iterated Function Systems and the Global Construction of Fractals," *Proc. Royal Soc. London*, A399, pp. 243-275, 1985.

[2] Barnsley, M. F., V. Ervin, D. Hardin, and J. Lancaster, "Solution of an Inverse Problem for Fractals and Other Sets," *Proc. National Academy of Sciences*, vol. 83, pp. 1975-1977, April 1986.

[3] Blum, H., "A Transformation for Extracting New Descriptors of Shape," in *Models for The Perception of Speech and Visual Forms*, W. Wathen-Dunn, Ed. Cambridge, MA: M.I.T. Press, 1967.

[4] Demko, S., L. Hodges, and B. Naylor, "Construction of Fractal Objects wjects with Iterated Function Systems," in *Proc. of SIGGRAPH '85*, vol. 19, 1985.

[5] Hutchinson, J. E., "Fractals and Self-Similarity," *Indiana Univ. Math. J.*, vol. 30, pp. 713-747, 1981.

[6] Lantuejoul, C., "Skeletonization in Quantitative Metallography," in *Issues of Digital Image Proc.*, R.M. Haralick and J.C. Simon, Eds., The Netherlands: Sijthoff and Noordhoff, 1980.

[7] Libeskind-Hadas, R., "An Application of Iterated Function Systems and Skeletonization to the Modelling of Fractal Images," A.B. Honors Thesis, Harvard University, Cambridge, Massachusetts, 1987.

[8] Mandelbrot, B. B., *The Fractal Geometry of Nature*, New York: W.H. Freeman, 1983.

[9] Maragos, P., and R. W. Schafer, "Morphological Skeleton Representation and Coding of Binary Images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1228-1244, Oct. 1986.

[10] Matheron, G., *Random Sets and Integral Geometry*. New York: Wiley, 1975.

[11] Serra, J., *Image Analysis and Mathematical Morphology*, New York: Academic Press, 1982.

Figure 1. Trees constructed from iterated function systems.

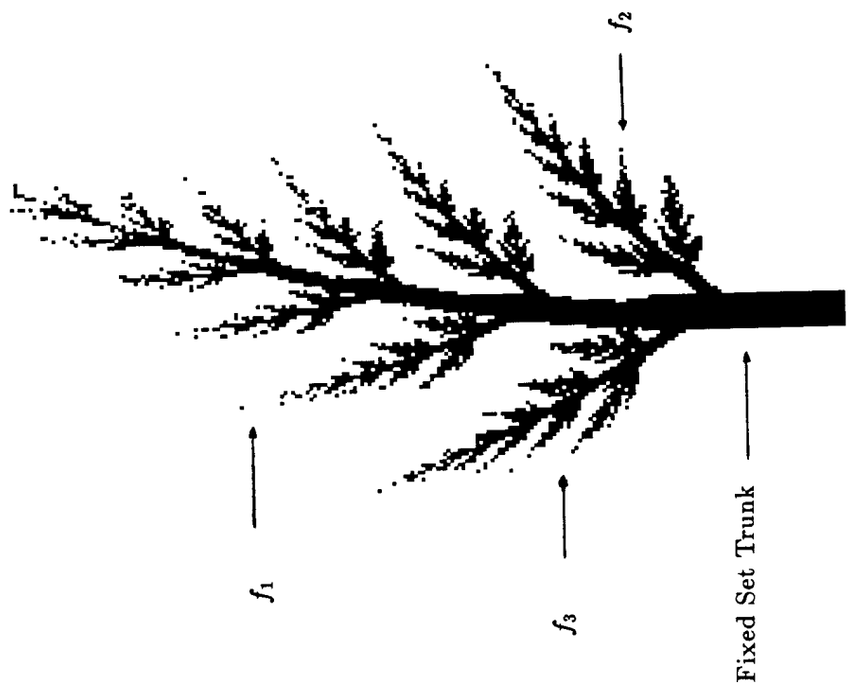


Table of affine transformations

f_i	$r_{x,i}$	$r_{y,i}$	θ_i	x_i	y_i
f_1	0.7	0.85	−0.1	0	50
f_2	0.3	0.4	−0.7	5	20
f_3	0.25	0.4	0.6	−5	30

$$f_i\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r_{x,i}\cos\theta_i & -r_{y,i}\sin\theta_i \\ r_{x,i}\sin\theta_i & r_{y,i}\cos\theta_i \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

Figure 2. Construction of one tree.

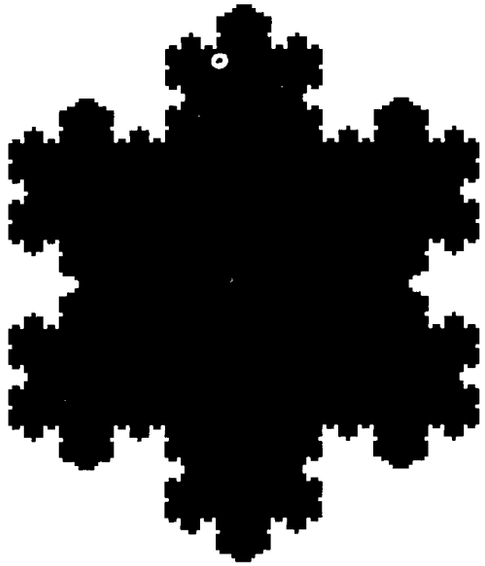
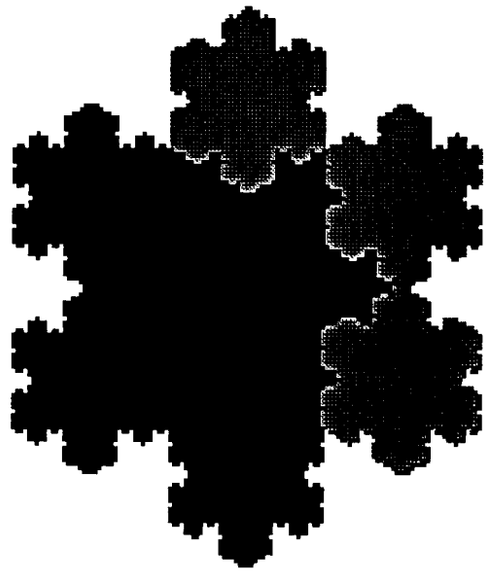


Figure 3. Triadic Koch island.



Three of the six affine transformations

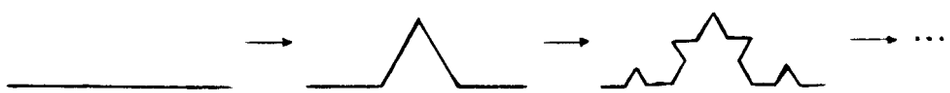


Figure 4. Iterative process to form the Koch island.

Table of affine transformations for K

f_i	r_i	θ_i	x_i	y_i
f_1	0.332	0	1	46
f_2	0.332	0	41	22
f_3	0.332	0	41	−22
f_4	0.332	0	1	−46
f_5	0.332	0	−41	−22
f_6	0.332	0	−41	22

The fixed set is a disk of radius 40 centered at (0,0).

Figure 5. Affine maps for the Koch island.

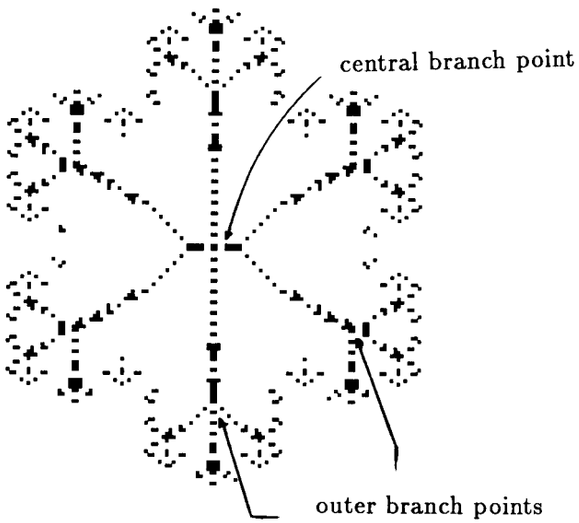


Figure 6. Skeleton points of the Koch island.

Digitized Leaf

Skeleton

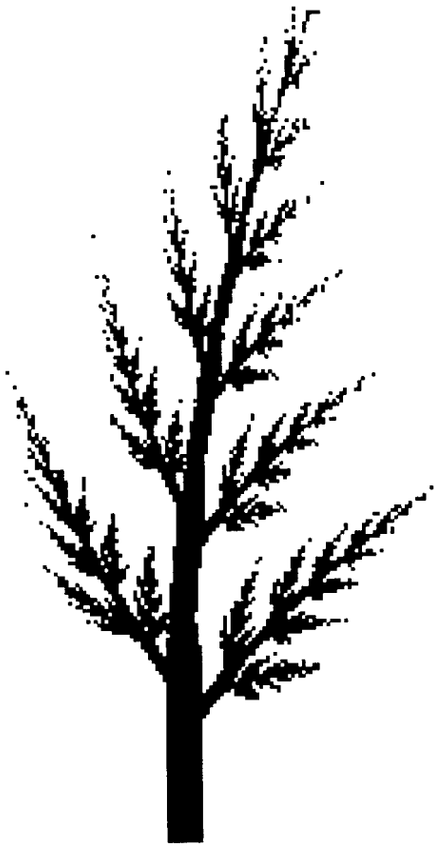
Table of affine transformations

f_i	$r_{x,i}$	$r_{y,i}$	θ_i	x_i	y_i
f_1	0.7	0.7	0	2	15
f_2	0.4	0.8	−0.92	17	−8
f_3	0.4	0.7	0.79	−15	−7

Attractor after 500,000 iterations

Morphologically postprocessed attractor

Figure 7. Modeling a maple leaf.

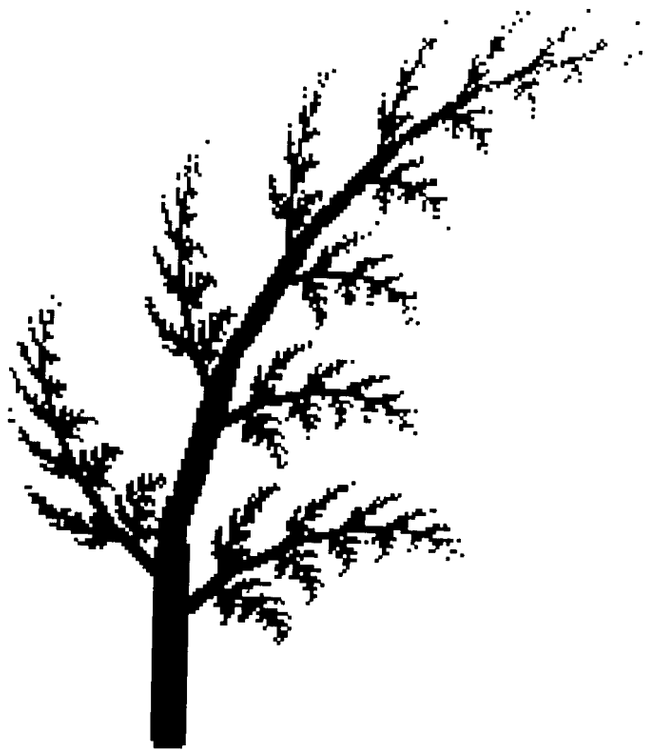


Frame 1

f_i	$r_{x,i}$	$r_{y,i}$	θ_i	x_i	y_i
f_1	0.7	0.85	−0.1	0	50
f_2	0.3	0.4	−0.7	5	20
f_3	0.25	0.4	0.6	−5	30

Frame 2

f_i	$r_{x,i}$	$r_{y,i}$	θ_i	x_i	y_i
f_1	0.7	0.85	−0.2	0	50
f_2	0.3	0.4	−0.8	5	20
f_3	0.25	0.4	0.7	−5	30



Frame 3

f_i	$r_{x,i}$	$r_{y,i}$	θ_i	x_i	y_i
f_1	0.7	0.85	−0.3	5	55
f_2	0.3	0.4	−0.9	5	20
f_3	0.25	0.4	0.8	−5	30

Figure 8. Animation of tree bending.