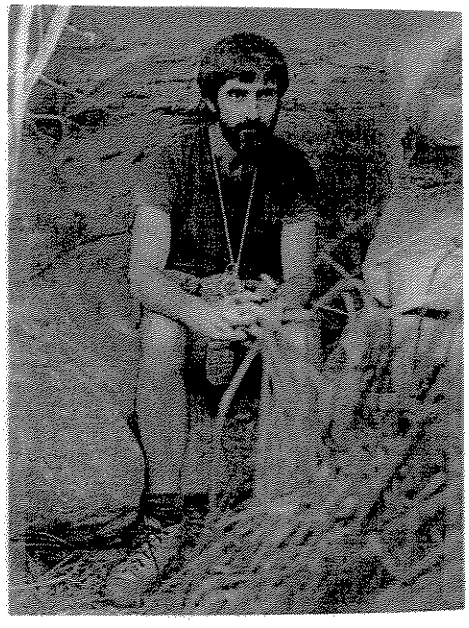
Part V: Fractals in the Computer Age

Cantor's, Koch's and Sierpinski's ways of constructing their fractals are easy to explain and you or I can draw the first few steps by hand. However, it is excessively difficult for a computer to implement these original algorithms. There's a "new" (to people of my age) way to think of fractal construction, which was put on a solid foundation in a paper by John Hutchinson, *Fractals and Self-Similarity*, Indiana University Math Journal **30**, 1981, pages 713-747.

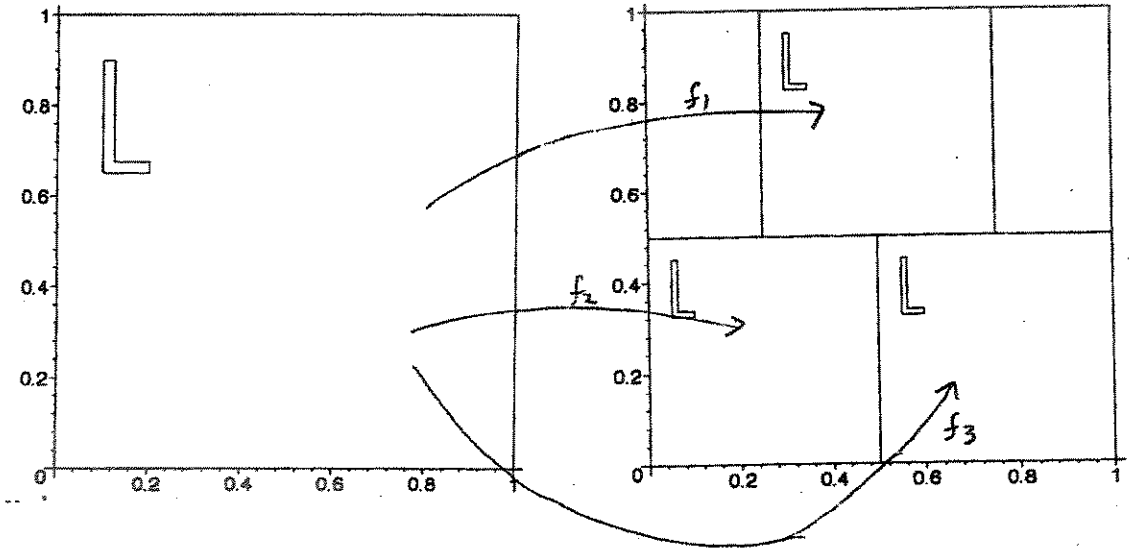
Hutchinson realized that there is a general framework in which to understand a large class of fractals, and this framework is actually related to the scaling ideas we've just been using to discuss dimension. Hutchinson's way of understanding fractals also leads to an amazingly easy algorithm with which computers can generate fractal pictures, which you will get to experience. This has led to the use of fractals in widespread computer graphics applications. We teach the theoretical underpinnings with which to understand Hutchinson's ideas completely in the senior-level math analysis course Math 5210. But it is possible to get a good idea of what's going on without being completely precise....



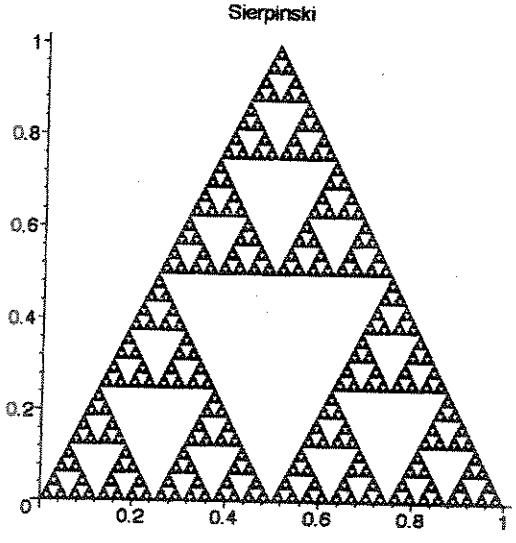
John Hutchinson in 1986

We illustrate what Hutchinson realized with the Sierpinski Triangle example. Think about how we rescaled this object in order to figure out its dimension, and now think about shrinking rather than expanding:

Let f_1 , f_2 , f_2 , be the three affine functions as shown below:



e.g. $f_1\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1/3 \\ 1/2 \end{bmatrix}$



Notice:
(1) The Sierpinski triangle S is exactly the union of its three transformed copies, i.e.

$$S = f_1(S) \cup f_2(S) \cup f_3(S).$$

Equivalently, consider the "cloning and shrinking" transformation applied to arbitrary sets B (for Bob), given by

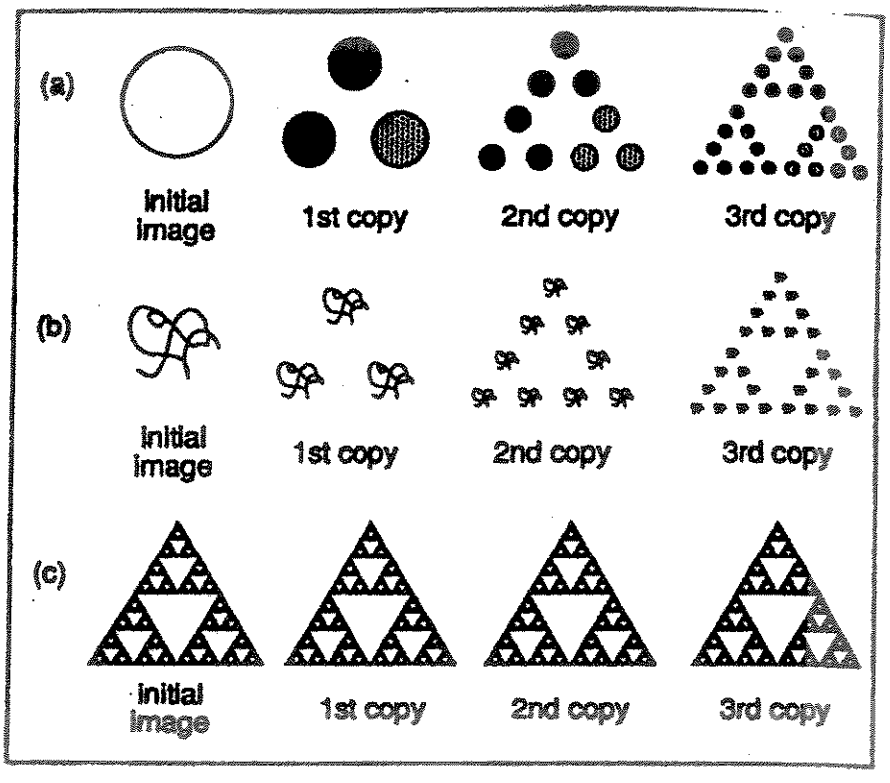
$$F(B) := f_1(B) \cup f_2(B) \cup f_3(B).$$

Then F(S)=S, i.e. Sierpinski's triangle doesn't get changed by the set transformation F.

(2) Each of the transformations f_1, f_2, f_3 shrinks Euclidean distances between pairs of points by the scaling factor of ½ .

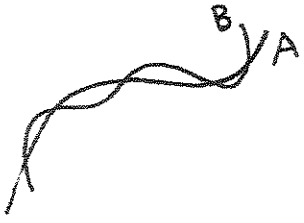
(3) What happens if we successively apply the set transformation F to arbitrary subsets? Can you explain the picture below, using (1) and (2)? It looks like if we clone and shrink Bob enough times, he'll turn into Sierpinski's triangle too!

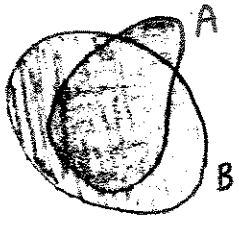
B F(B) F(F(B)) F(F(F(B))) ...

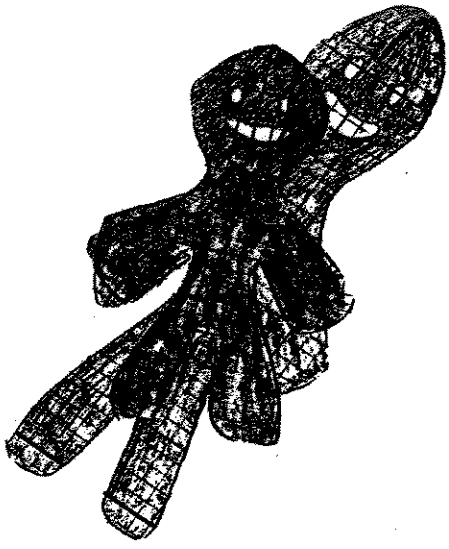


Explanation: First, we need a practical way to say that two sets, A and B in the plane, are close to each other. One way that works is to see if every point in A is close to some point in B, and vise-verse.

Precisely, we will say that the "distance" between A and B, dist(A,B), is either the maximum distance points in A can be from (nearest points in) B; or, the maximum distance points in B can be from A; dist(A,B) is the larger of these two numbers. In fact, this distance is well known to advanced mathematics students, and is called the "Hausdorff distance", after Felix Hausdorff. Here are some sets to look at in order to understand Hausdorff distance:







Using Hausdorff distance we can explain what happened in the Sierpinski Triangle example:

Because each of the transformations f_1 , f_2 , f_3 shrinks distances between points by a factor of ½ , the set transformation F also shrinks Hausdorff distance between sets:

$$\text{dist}(F(A),F(B)) \leq \tfrac{1}{2}\,(\text{dist}(A,B))$$

always holds. Since the Sierpinski triangle S is left fixed by F, the Hausdorff distance from Sierpinski to (the cloned and shrunken) F(Bob) is only half as large as the original distance from Bob to S. If we compare F(F(Bob)) to S, they are only ¼ the distance apart as B and S were initially. After 10 steps, the iterated Bob will be only $1/2^{10}$ as far from Sierpinski as he started. Certainly to our eyes, the two sets will be indistinguishable.

That's why we can begin with any initial set, the easiest of which would be a single point, and simply iterate F a large number of times to create a set which is practically indistinguishable from the real Sierpinski triangle.

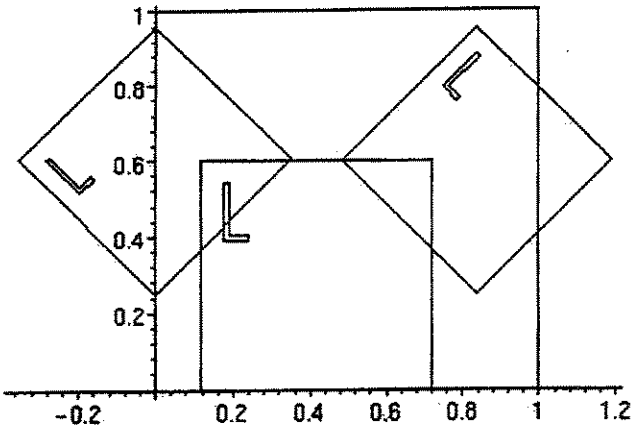
The rest of Hutchinson's brainstorm: You can begin with ANY finite collection of transformations, f_1 , f_2 , f_3 … f_n , such that each transformation f_i shrinks Euclidean distances between pairs of points by at least a factor m_i less than 1. (These transformations don't have to be uniform scalings – for example they can be more general affine transformations as long as they have the distance contracting property.) Then, as with the Sierpinski example, define the cloning-shrinking transformation

$$F(B) := f_1(S) \cup f_2(S) \cup \ldots \cup f_n(S).$$

This transformation will shrink the Hausdorff distance between pairs of transformed sets, by a factor which is no larger than the largest shrinking factor for any of the f_i . Thus, as you successively iterate F on any two sets A and B, the corresponding transformed sets get closer and closer, at a geometric rate.

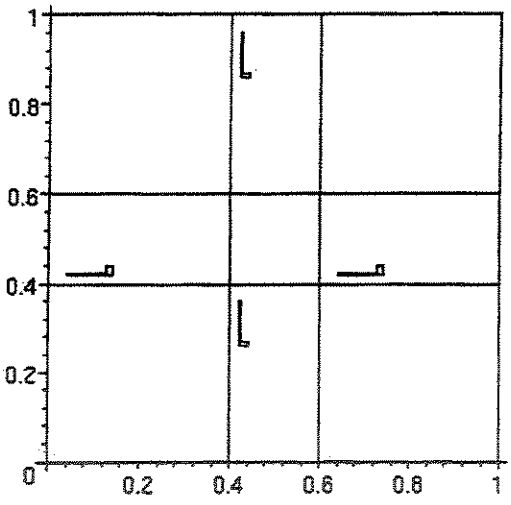
In fact, we can start with any set A, and the iteration sequence {A, F(A), F(F(A)) …} converges to a unique limit fractal determined by the choice of f_1 , f_2 , f_3 … f_n. (The Hausdorff distance between this fractal and the elements of the iteration sequence converges to zero as the number of iterations approaches infinity). SEE EXAMPLES

fractal template

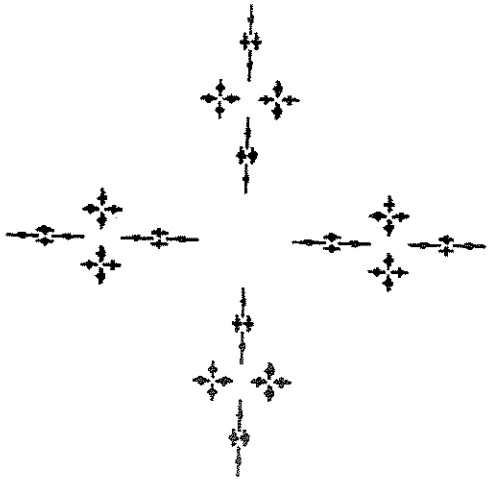


algae

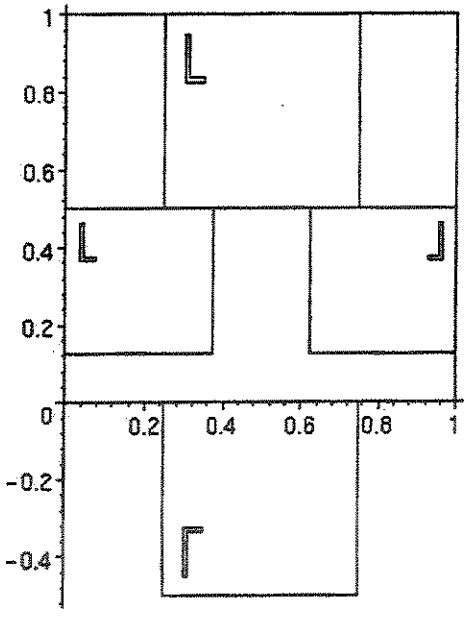
fractal template



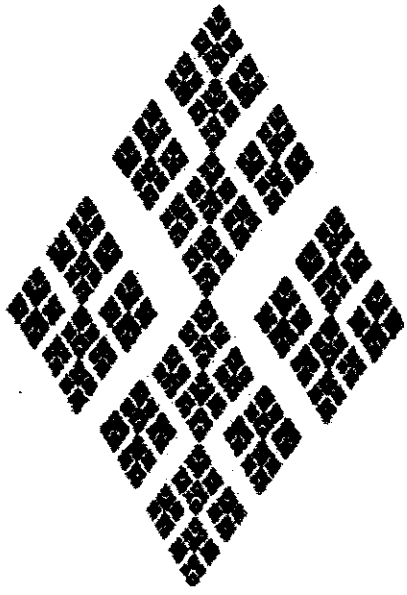
cross-type example



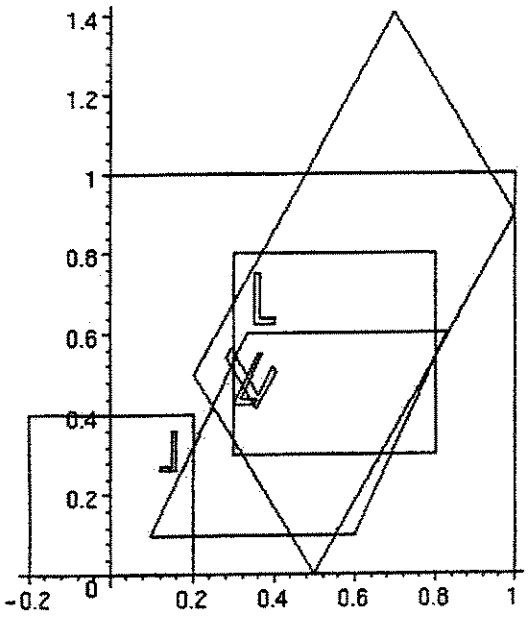
fractal template



diamond pattern

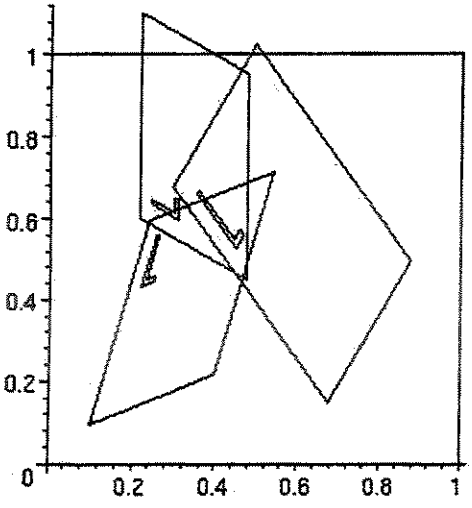


fractal template



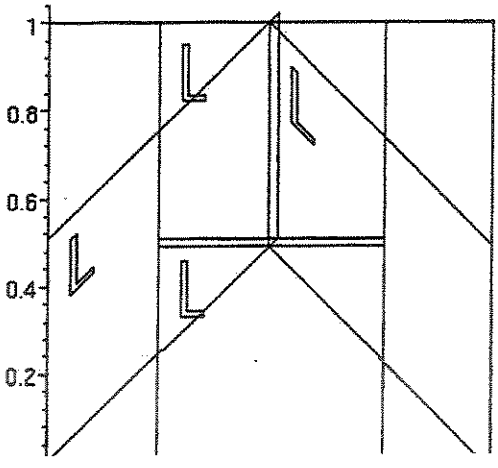
flame wave

fractal template



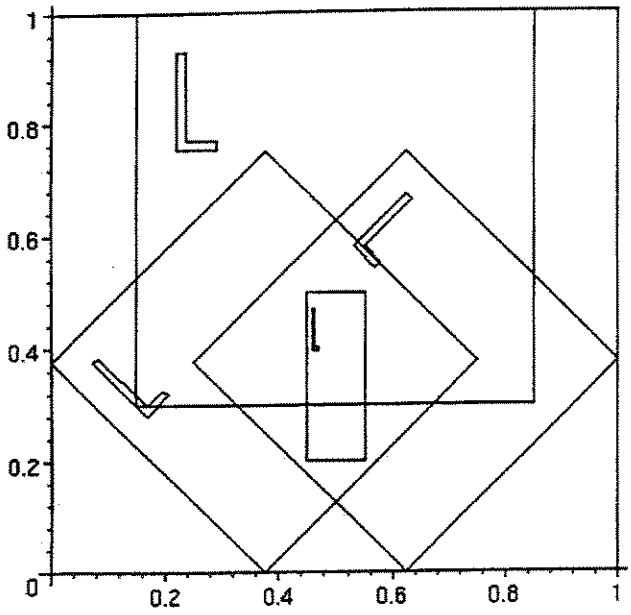
Leaves

fractal template

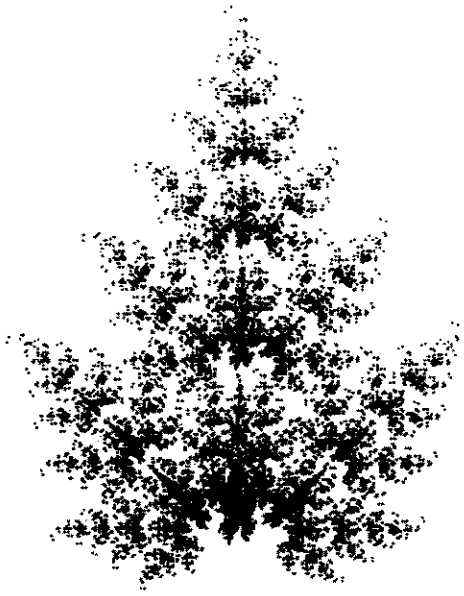


Travis' Mountain

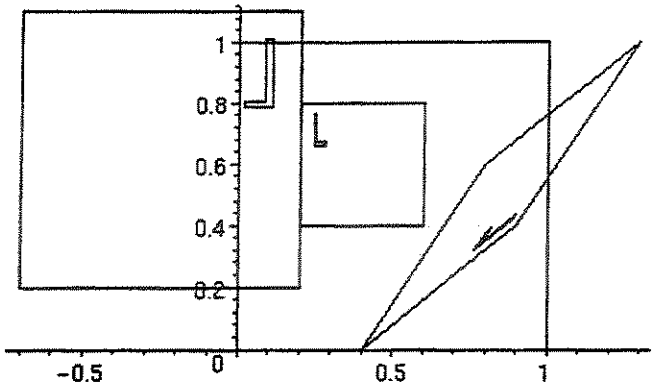
fractal template



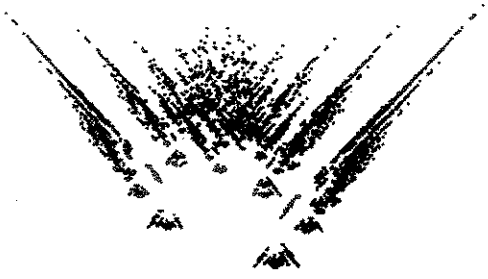
pine tree



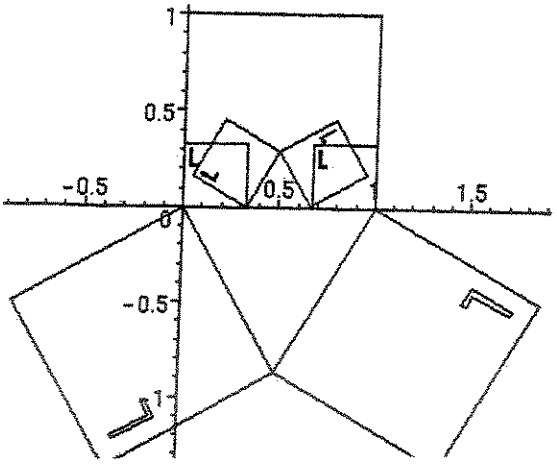
fractal template



Punk Kid



fractal template



snowflake