

Linear Fractal Shape Interpolation

Brandon Burch* John C. Hart

School of EECS
Washington State University
Pullman, WA 99164-2752
hart@eecs.wsu.edu

Abstract

Interpolation of two-dimensional shapes described by iterated function systems is explored. Iterated function systems define shapes using self-transformations, and interpolation of these shapes requires interpolation of these transformations. Polar decomposition is used to avoid singular intermediate transformations and to better simulate articulated motion. Unlike some other representations, such as polygons, shaped described by iterated function systems can become totally disconnected. A new, fast and image-based technique for determining the connectedness of an iterated function system attractor is introduced. For each shape interpolation, a two parameter family of iterated function systems is defined, and a connectedness locus for these shapes is plotted, to maintain connectedness during the interpolation.

Keywords: Fractal Geometry, Iterated Function System, Mandelbrot Set, Morphing, Shape Interpolation.

1 Introduction

The basis of key-framed animation is shape interpolation, a.k.a. morphing. Recent advances in shape interpolation have discovered that alternative geometric representations yield new features. The iterated function system represents a shape using only transformations, modeling a shape out of smaller copies of itself, yielding a compact description of a highly-detailed, and often fractal, object. This paper explores the use of the iterated function system as a geometric representation for shape interpolation.

An iterated function system [Barnsley & Demko, 1985] is defined as a finite collection $\{w_i\}_{i=1}^N$ of N maps $w_i : \mathbf{R}^n \rightarrow \mathbf{R}^n$. An IFS models an object by constructing it out of smaller copies of itself. Symbolically, an IFS $\{w_i\}_{i=1}^N$ models the set $A \subset \mathbf{R}^n$, called the *attractor* of the IFS, as the union of *attractorlets* $A_i = w_i(A)$ — as

the solution A of

$$A = \bigcup_{i=1}^N A_i. \quad (1)$$

By a theorem of Hutchinson [1981], if all the maps of an IFS are contractions, then the attractor A exists and is unique and non-empty.

This paper uses iterated function systems as a representation for two-dimensional shapes as sets. As such, it does not require the association of probabilities with each map that are often used to define a measure on sets. This paper also focuses on iterated function systems consisting of affine transformations. The family of attractors of such iterated function systems are commonly called linear fractals.

Section 2 examines some of the previous work in shape interpolation of 2-D polygons and linear fractal models. Section 3 reviews the polar decomposition for interpolating affine transformations, and applies it to the affine transformations of the IFS. Section 4 examines the connectedness of IFS shapes and presents a new image-based algorithm for determining path connectedness. Section 5 parameterizes all of the transformations of an IFS with two variables and plots a corresponding connectedness locus, such that connectedness is maintained during the shape interpolation. Section 6 demonstrates the technique and examines its effects. Section 7 concludes, and offers directions for future research.

2 Previous Work

Morphing is an interpolation between images animated over time. Whereas much of the previous research has concentrated on warping images, e.g. Beier & Neely [1992], several have focused on transforming the model depicted in the image. Such shape morphing can be further divided into 2-D versus 3-D representations. This paper focuses only on the 2-D representation, and its previ-

*Current address: Sapien Health Network, Portland, Oregon, brandonb@sisc.com

ous research shows that alternative representations offer new insight and benefits into shape metamorphosis.

2.1 Polygon Morphing

Two-dimensional simple polygonal shapes were morphed by Sederberg & Greenwood [1992] using an energy-minimizing physically-based algorithm to avoid shape inversion and self intersection. A least-work solution was proposed when the initial and final polygons contain a different number of vertices.

Sederberg *et al.* [1993] continued this work by replacing the standard vertex representation of a polygon with a turtle-graphics representation. Morphing this new representation allows the intermediate shapes to retain their intrinsic qualities, but required special attention to maintaining a closed polygon.

Goldstein & Gotsman [1995] employed a multiresolution representation for polygonal shapes to identify feature based correspondences and for smoother shape interpolation.

2.2 Linear Fractal Morphing

Plate 9.8.5 of Barnsley [1988] demonstrates an animation sequence of clouds based on changes to the transformations of its IFS representation. The following theorem proved that small changes in the IFS maps morph the IFS attractor continuously, and provided the basis for IFS-based fractal morphing.

Theorem 1 [Barnsley, 1988] *Let $\{w_i(t)\}_{i=1}^N$ be an IFS whose maps are parameterized by a single bounded variable $t \in \mathbf{R}$. Then the function $A(t)$, that maps the parameter t into the attractor of the IFS parameterized by t , is continuous.*

IFS interpolation has appeared in several animations. Hart & Das [1990] interpolated a Sierpinski's tetrahedron into a 3-D dragon by elementwise linear interpolation of the matrices representing the linear transformations of the IFS. Hart [1991] interpolated IFS representations of trees and Platonic solids by interpolating the coefficients used in their modeling, such as the parameters of the scaling, rotation and translation operations that were composed to create each map of each IFS. Hart [1992] controlled only the scaling parameters of the IFS maps to simulate a fractal fade-in and dissolve.

Bowman [1995] explored the effects of changing scaling and rotation coefficients of the IFS maps. Prusinkiewicz *et al.* [1993] interpolated L-systems based on turtle geometry (which can be used to represent linear fractals) for the application of animating continuous plant development.

3 Interpolating Affine Transformations

A transformation is affine if and only if it takes parallel lines to parallel lines. Affine transformations consists of a linear transformation (which may rotate, scale, stretch and shear) followed by a translation. In computer graphics, the two-dimensional affine map is typically represented by the 3×3 homogeneous transformation matrix.

Morphing of shapes described by the IFS representation involves interpolation of the IFS maps. Previous explorations show that interpolation of the individual parameters (rotation angle, scale factor, translation displacement, shear amount) used in the composition of each IFS map provide the best level of control over the morph. In a general key-frame animation system, such parameters may not be available, and the IFS representation may consist of nothing more than the affine transformation matrices. Polar decomposition extracts the individual transformation components from a general affine transformation matrix, and provides a basis for more controlled shape interpolation.

Linear transformations can be composed of translation, scale, shear, and rotation. *Polar decomposition* separates a linear transformation matrix M into an orthogonal matrix Q that contains its rotation and reflection components, and a symmetric positive definite matrix S that contains its scale and shear components [Shoemake & Duff, 1992].

The factor Q is computed by repeatedly averaging the matrix with its inverse transpose. Let $Q_0 = M$ and define

$$Q_{i+1} = \frac{1}{2}(Q_i + Q_i^{-1T}). \quad (2)$$

This sequence eventually converges to the orthogonal component Q of the linear transformation matrix M .

Given Q , and M , one can solve for S by inverting (transposing) Q

$$S = Q^{-1}M. \quad (3)$$

After polar decomposition, S holds the scale and shear parameters while Q holds the rotation and reflection parameters. The arctangent of the first column of Q yields the angle of rotation. Interpolation of the Q matrix consists of constructing rotation matrices based on the interpolation of the angle. Interpolation of differing reflection factors (1 vs. -1) remains an open problem. The matrix S may be interpolated elementwise.

4 Connectedness

There are many different kinds of connectedness in topology. The following definitions assume a two-dimensional

complete space \mathbf{R}^2 under the topology induced by the Euclidean metric. Since the attractors of iterated function systems are closed, the following discussions apply to closed sets in this metric space.

The definition of connectedness is based on the inability to find two disjoint open sets that form a disconnection of a set.

Definition 1 *The set $A \subset \mathbf{R}^2$ is connected if and only if there does not exist two disjoint, non-empty open sets $B, C \subset \mathbf{R}^2$ such that $A \cap B \neq \emptyset, A \cap C \neq \emptyset$ and $A \subset B \cup C$.*

The definition of path connectedness requires the ability to find a path between any two points in the set.

Definition 2 *The set $A \subset \mathbf{R}^2$ is path connected if and only if for any two points $\mathbf{x}, \mathbf{y} \in A$ there exists a continuous map $f : [0, 1] \rightarrow A$ such that $f(0) = \mathbf{x}$ and $f(1) = \mathbf{y}$.*

These definitions are based on those by Munkres [1975], which also contains several examples of sets that meet one but not the other definition.

4.1 An Object-Based Connectedness Algorithm

Barnsley [1988] presents the following algorithm for determining the connectedness of an arbitrary IFS attractor. The algorithm tries to find a separation between disjoint components of a given set, and was used to explore the connectedness locus of families of IFS attractors to better understand their dynamics.

Let $d(x, A) = \min_{y \in A} d(x, y)$ define the distance from a point to a set, and let

$$d(A, B) = \min(\min_{x \in A} d(x, B), \min_{y \in B} d(y, A)) \quad (4)$$

define the Hausdorff distance between two sets.

The value n balances the accuracy of the connectedness determination versus the computation time; the greater the value of n is, the greater the accuracy of the algorithm and the greater the time to compute. The algorithm's overhead is primarily due to the computation of the Hausdorff distance (4), which is quadratic in the number of points in \tilde{A}

4.2 Path Connected Attractors

The image-connectedness algorithm is based a particular rendering algorithm, and depends on its viewing window and display resolution. It defines connectedness by determining if the components of the IFS attractor overlap.

```

Initialize
  s = max_i Lip w_i.
  R sufficiently large such that B(R) ⊃ A.
   $\tilde{A} = \{(0, 0)\}$ .
For n iterations
  For each IFS map w_i
    Let  $\tilde{A}_i = w_i(\tilde{A})$ .
  For each IFS map w_i
    Let  $d_i = \min_{j \neq i} d(\tilde{A}_i, \tilde{A}_j)$ .
    If  $d_i > 2s^{n+1}R$  then
      Return disconnected.
  End For.
  Let  $\tilde{A} = \bigcup_i \tilde{A}_i$ .
End For.

```

Figure 1: Object-based connectedness algorithm [Barnsley, 1988].

From (1) we have that the attractor is defined as the union of its images under the IFS maps. If these images overlap to form a path, then the resulting attractor is connected. We use the edges of an N -vertex graph to track the paths of overlap, and determine the image-connectedness of the attractor.

Definition 3 *The graph G is an attractorlet-overlap graph of an IFS $\{w_i\}_{i=1}^N$ if and only if G contains N vertices and an edge in G connects vertex i to vertex j if and only if $A_i \cap A_j \neq \emptyset$.*

One should not confuse the overlap of attractorlets with the overlapping construction described by Barnsley [1988]. In particular, these definitions differ with “just touching” attractors, whose attractorlets overlap but are not of overlapping construction.

The attractorlet-overlap graph indicates the path connectedness of the attractor.

Theorem 2 *Let A be an IFS attractor in a complete metric space, and let G be its attractorlet-overlap graph. Then A is path connected if and only if G is connected.*

Proof: Assume G is connected. Then G shows there exists a sequence of overlapping attractorlets from any attractorlet A_i to any other attractorlet A_j . In each attractorlet A_i , there exists a sequence of overlapping sub-attractorlets $A_{ik} = w_i(A_k) \subset A_i$ between any two sub-attractorlets in A_i . Hence, any two points \mathbf{x}, \mathbf{y} in A are connected by a sequence of overlapping attractorlets, which in turn are connected by a sequence of overlapping sub-attractorlets, which, in the limit, yield a path connecting \mathbf{x} to \mathbf{y} .

Assume G is disconnected. Then there exists two vertices i and j in G not connected by any path of edges, and these vertices correspond to two attractorlets A_i and A_j that can not be connected by any sequence of overlapping attractorlets. Since the attractorlets are closed and subsets of a complete space, any two attractorlets that do not overlap are therefore not path connected. Hence, there exist no path in A connecting any point in attractorlet A_i with any point in attractorlet A_j . \square

4.3 An Image-Based Path Connectedness Algorithm

Computer graphics defines a discretized shape $A \subset \mathbf{Z}^2$ to be connected if and only if there exists a *path* of pixels connecting every two pixels in A . We infer the path connectedness of $A \subset \mathbf{R}^2$ by the path connectedness of $A \subset \mathbf{Z}^2$. In fact, typical computer graphics applications are only concerned with such properties of the image, regardless of the properties of the object the image represents.

The algorithm in Fig. 2 uses the chaos game to plot points in the attractor labeled by their attractorlet. When two points in different attractorlets are plotted in the same location, an edge is added to the attractorlet-overlap graph. The path-connectedness algorithm represents the attractorlet-overlap graph using an adjacency matrix.

Definition 4 A graph G with N vertices is represented by a symmetric $N \times N$ adjacency matrix X if and only if $X(i, j) = 1$ if an edge connects vertex i with vertex j and $X(i, j) = 0$ otherwise.

The following theorem provides an algorithm for determining the connectedness of the attractorlet-overlap graph.

Theorem 3 [Deo, 1972] If X is the adjacency matrix for a graph G with N vertices, and $Y = X + X^2 + \dots + X^{N-1}$, then G is disconnected if and only if at least one element of Y is zero.

The algorithm uses the chaos game to find points on the attractor, but any rendering algorithm could be used [Hep-ting *et al.*, 1990]. The algorithm also assumes a uniform distribution of points, and could be improved by selecting maps randomly based on the probabilities commonly used to make the measure on the attractor more uniform.

The algorithm runs in space proportional to the resolution of the image, which is related to the number of iterations in that higher resolutions require more iterations. Likewise the algorithm's time complexity is proportional to the number of points plotted which is similarly related to the resolution of the image.

```

Initialize
  x such that  $\mathbf{x} = w_1(\mathbf{x})$ 
   $i = 1$ .
  A unit  $N \times N$  adjacency matrix  $X = I$ .
  The index buffer  $v(\mathbf{p}) = 0$  for all pixels  $\mathbf{p}$ .
For  $n$  iterations
  Let  $i \in [1, N]$  be a random integer.
  Let  $\mathbf{x} = w_i(\mathbf{x})$ .
  Let  $\mathbf{p} = w_2v(\mathbf{x})$ .
  If  $v(\mathbf{p}) \neq 0$  and  $v(\mathbf{p}) \neq i$  then  $X(i, v(\mathbf{p})) = 1$ .
   $v(\mathbf{p}) = i$ .
End For.
Let  $Y = X + X^2 + \dots + X^{N-1}$ .
If  $Y(i, j) = 0$  for any  $i, j$  then
  Return disconnected.
Else
  Return connected.

```

Figure 2: The Index Buffer algorithm for determining path connectedness.

5 Interpolation of IFS Attractors

As the parameters of an IFS change, such as during shape interpolation, the connectivity of the resulting attractors can also change. During shape interpolation, certain intrinsic properties of the shape should be preserved [Sederberg *et al.*, 1993], and the most important of these is connectivity.

A connectedness locus maps the parameter space of a representation. This map depicts regions of parameters whose resulting shape is connected, and other regions of disconnectedness. These regions themselves may be simply connected, multiply connected or totally disconnected.

Decompose the affine map w_i of an IFS into the polar component matrices Q_i, S_i and a vector translation component T_i . Let θ_i be the angle of the rotation represented by Q_i and let the operator $Q(\theta)$ return a rotation matrix by the angle θ (e.g. $Q_1 = Q(\theta_1)$). Hence, we decompose the IFS map

$$w_i(\mathbf{x}) = Q(\theta_i)S_i\mathbf{x} + T_i \quad (5)$$

The parameter space for the connectedness locus is spanned by two parameters (u, v) such that the point $(0, 0)$ will indicate the parameters of the IFS representing the initial shape, and the point $(1, 1)$ will indicate the parameters of the IFS representing the final shape. Let θ_i^0, S_i^0, T_i^0 parameterize the maps of the initial IFS and let θ_i^1, S_i^1, T_i^1 parameterize the maps of the final IFS. Then the two variables (u, v) parameterize a family of iterated

function systems with N maps of the form

$$w_i(\mathbf{x}) = Q((1-u)\theta_i^0 + u\theta_i^1)((1-u)S_i^0 + uS_i^1)\mathbf{x} + (1-v)T_i^0 + v(T_i^1). \quad (6)$$

Hence, the parameter u interpolates (in polar coordinates) the linear part of the IFS maps, and the parameter v interpolates the translation part of the IFS maps.

The connectedness locus is plotted over the domain $[-1, 2] \times [-1, 2]$. Interpolation between two IFS attractors can then be described by any path from the point $(0, 0)$ to the point $(1, 1)$ in this connectedness locus. If $(0, 0)$ and $(1, 1)$ belong to the same connected component (with respect to the domain), then connectivity of the IFS attractors can be maintained during the shape interpolation.

6 Results

The two contributions of this research are a more sophisticated shape interpolation between IFS attractors, and a fast determination of the connectivity of IFS attractors. This section demonstrates both contributions with the results of several experiments.

6.1 Morphing Effects

The effects of the enhancements to shape morphing are more clearly understood by comparing corresponding frames from still sequences of the shape interpolation.

Figure 3 shows the naive approach to morphing two attractors. The method linearly interpolates each element of the IFS affine transformation matrices. Both the initial attractor and final attractor are described by three-map iterated function systems. The only difference is that the lower left map of the initial attractor is rotated 180° with respect to its corresponding map in the final attractor.

The figure indicates that the interpolation causes the lower left map to become singular midway through the morph. Figure 4 interpolates the same attractors, but uses polar decomposition to avoid the singularity. Although the initial and final attractors are connected in Figures 3 and 4, all of the intermediate attractors are disconnected.

Figure 5 demonstrates a simple linear interpolation of the polar decomposition parameters. Notice that the initial and final attractors are connected, but some of the intermediate attractors become disconnected, particularly in the latter half of the morph.

Figure 6 uses a path through the connectedness locus to guide the morph and avoid becoming disconnected when possible.

Figure 7 shows the connectedness locus for the morphs in Figures 5 and 6. The morph in Figure 5 corresponds to a

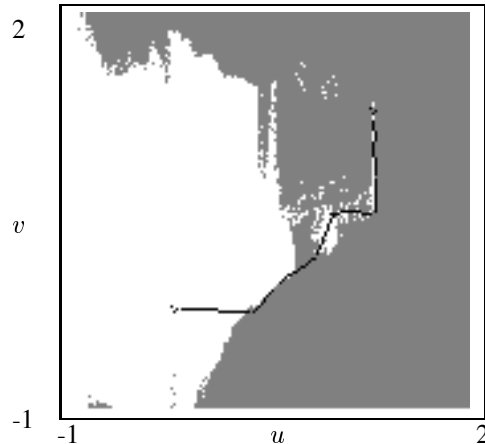


Figure 7: The connectedness locus for the Dragon-to-Gasket interpolation. White indicates parameters of path-connected attractors.

straight-line path from the two endpoints in Figure 7. The morph in Figure 6 follows the path indicated in Figure 7 to better maintain connectedness.

6.2 Fast Connectedness Algorithm Performance

The effectiveness of the image-based connectedness algorithm is measured by plotting the accuracy of its results against the resolution it uses for measurement.

Resolution	Scale
75^2	0.47850
150^2	0.48750
300^2	0.49000
600^2	0.49815
theoretic	0.50000

Table 1: Accuracy of image connectedness.

The algorithm uses a rendering method and checks for pixels multiply plotted from different mappings. A family of Sierpinski's gaskets tested the image-based path-connectedness algorithm at various screen resolutions. If the scaling components of the gasket's self-maps is reduced below one-half, then the resulting gasket is topologically disconnected. Experimentation of the image connectedness algorithm show it was able to distinguish connectedness better at higher screen resolutions, as shown in Table 1.

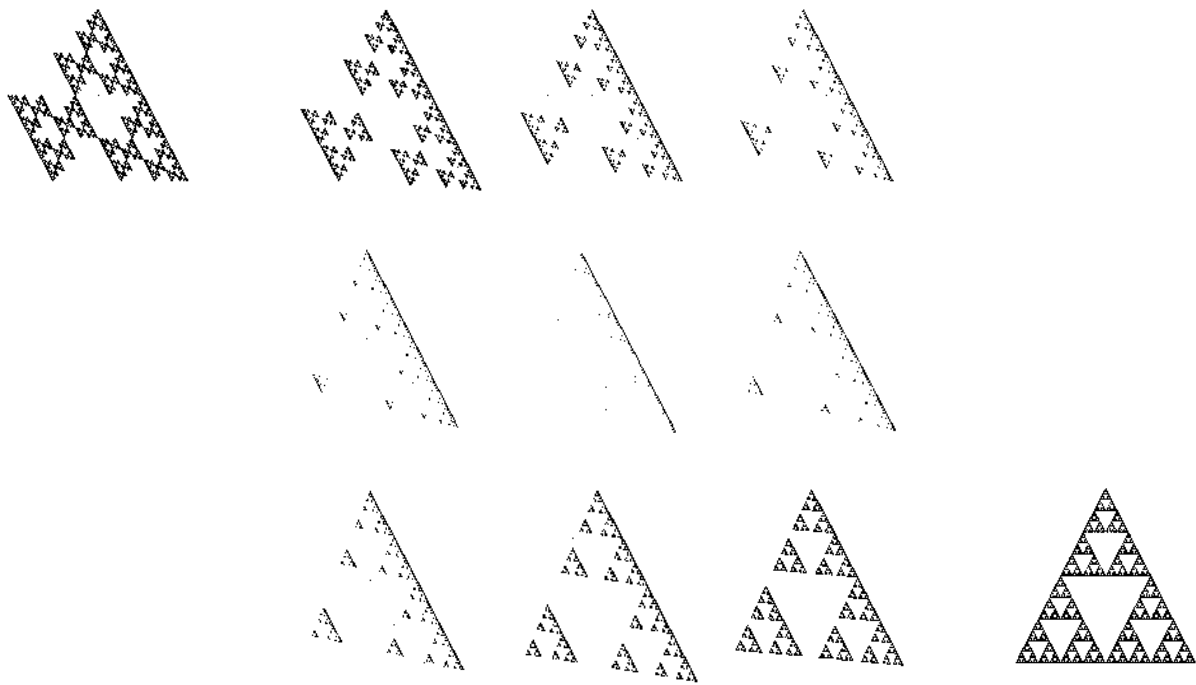


Figure 3: Linear interpolation of IFS transformation matrix elements.

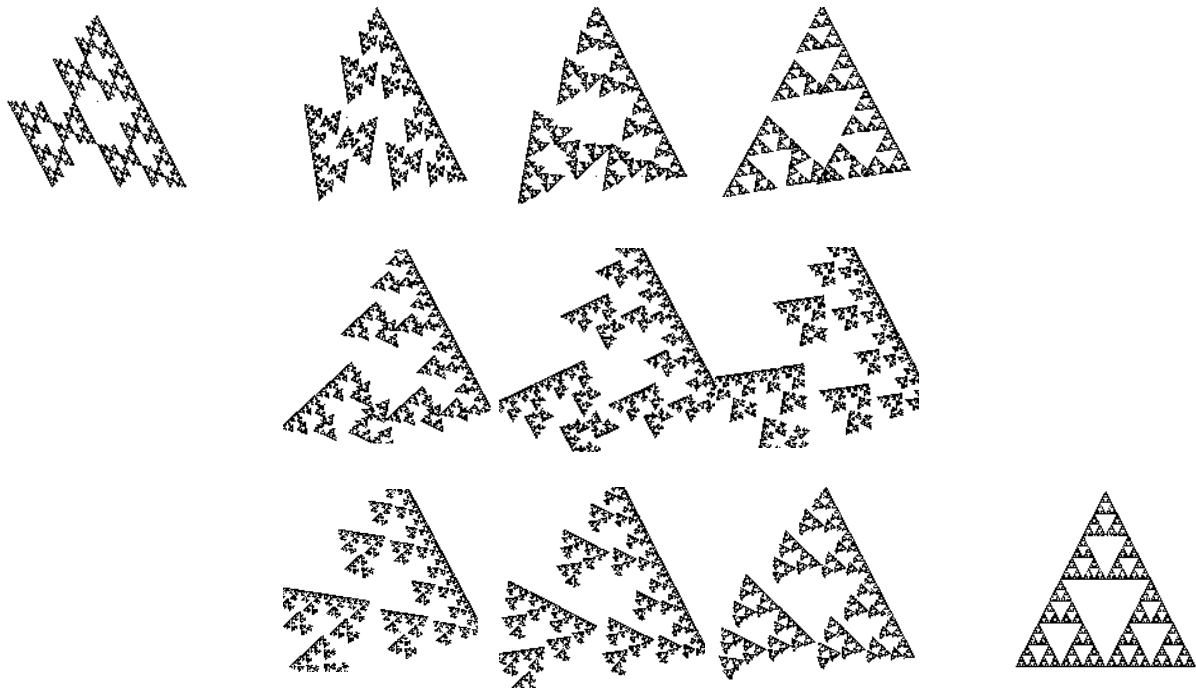


Figure 4: Polar interpolation of IFS transformation matrices.

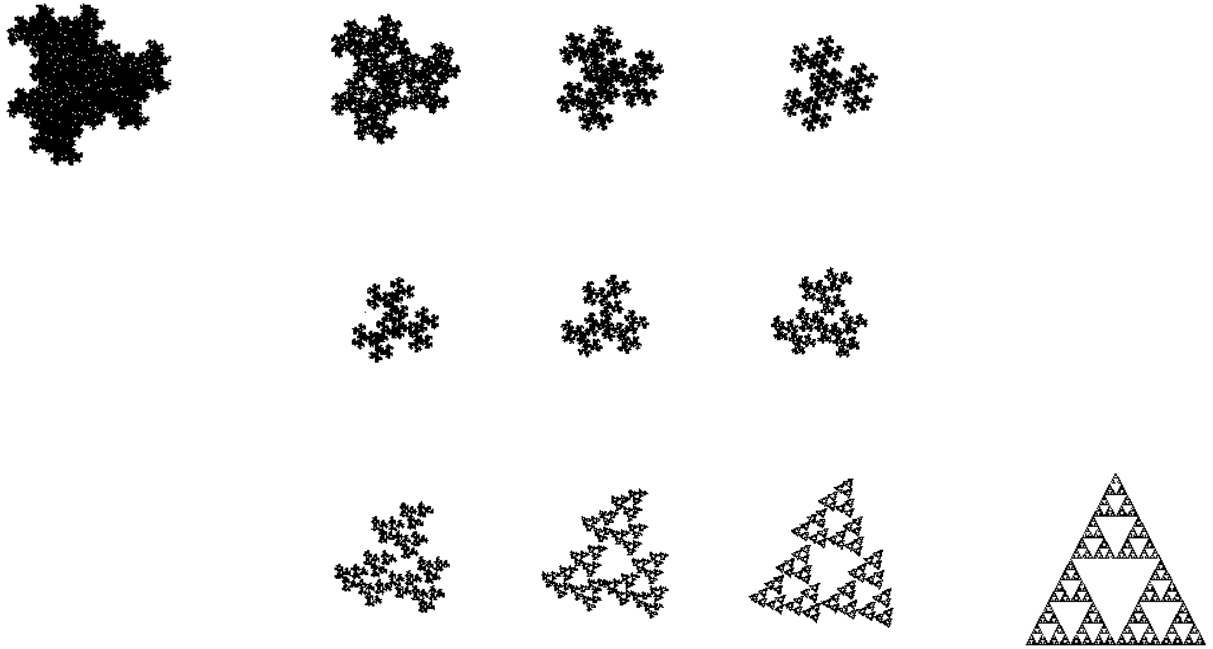


Figure 5: Polar interpolation of IFS attractors ignoring connectedness.

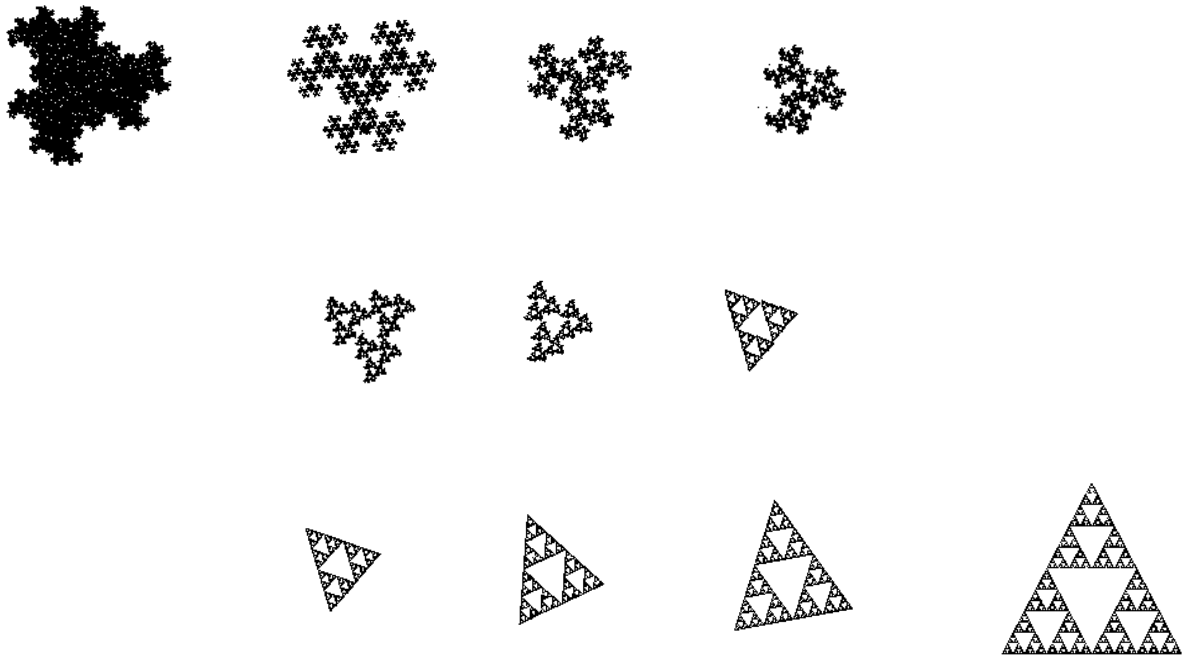


Figure 6: Polar interpolation of IFS attractors avoiding disconnectedness.

7 Conclusion

The image definition of connectedness results in the fast generation of a connectedness locus, which is then used to maintain connectedness during shape transformation. While the image connectedness locus does not determine topological connectedness, it does perform well at predicting the appearance of connectedness of the rendered attractor. Since the rendered attractor appears in the animation sequence, the image-based definition of connectedness seems better suited for maintaining connectedness during shape interpolation.

7.1 Future Work

This paper focused on interpolation of attractors whose iterated function systems contained the same number of maps, with *a priori* correspondences between these maps. Automatic determination of map correspondences remains an open problem, as does the problem of interpolating between iterated function systems with different numbers of maps.

The recurrent iterated function systems (RIFS) constrains an iterated function system with a directed graph controlling map composition. Interpolation of RIFS attractors poses the problem of interpolating directed graphs, possibly even with a different number of vertices.

Although such a case has not been encountered in this research, there is no guarantee that there will be a path through the connectedness locus between the interpolation endpoints. Using more than two parameters yields higher dimensional connectedness loci that have a greater chance at yielding such paths, but create the problem of navigating through multidimensional spaces.

7.2 Acknowledgments

This research is part of the recurrent modeling project, funded in part by the National Science Foundation grant #CCR-9529809 and a gift from Intel. Thanks also to Dietmar Saupe and the GI reviewers for helpful comments.

References

- [Barnsley & Demko, 1985] Barnsley, M. F. and Demko, S. G. Iterated function schemes and the global construction of fractals. *Proceedings of the Royal Society A* 399, 1985, pp. 243–275.
- [Barnsley, 1988] Barnsley, M. F. *Fractals Everywhere*. Academic Press, New York, 1988.
- [Beier & Neely, 1992] Beier, T. and Neely, S. Feature-based image metamorphosis. *Computer Graphics* 26(2), July 1992, pp. 35–42.
- [Bowman, 1995] Bowman, R. Fractal metamorphosis: A brief student tutorial. *Computer and Graphics* 19(1), 1995, pp. 157–164.
- [Deo, 1972] Deo, N. *Graph Theory with Applications to Engineering and Computer Science*. Prentice Hall, Englewood Cliffs, N.J., 1972.
- [Goldstein & Gotsman, 1995] Goldstein, E. and Gotsman, C. Polygon morphing using a multiresolution representation. In *Proc. Graphics Interface*, May 1995, pp. 247–254.
- [Hart & Das, 1990] Hart, J. C. and Das, S. Sierpinski blows his gasket. *SIGGRAPH Video Review* 61, 1990. (Animation).
- [Hart, 1991] Hart, J. C. unNatural Phenomena. *SIGGRAPH Video Review* 71, 1991. (Animation).
- [Hart, 1992] Hart, J. C. Fun with octrees: Graph topologies of the recurrent cube. *SIGGRAPH Video Review* 87, 1992. (Animation).
- [Hepting *et al.*, 1990] Hepting, D., Prusinkiewicz, P., and Saupe, D. Rendering methods for iterated function systems. In *Proc. of Fractals '90*. IFIP, 1990.
- [Hutchinson, 1981] Hutchinson, J. Fractals and self-similarity. *Indiana University Mathematics Journal* 30(5), 1981, pp. 713–747.
- [Munkres, 1975] Munkres, J. R. *Topology: a first course*. Prentice-Hall, Englewood Cliffs, New Jersey, 1975.
- [Prusinkiewicz *et al.*, 1993] Prusinkiewicz, P., Hammel, M. S., and Mjolsness, E. Animation of plant development. *Computer Graphics (Annual Conference Series)* 27, Aug. 1993, pp. 351–360.
- [Sederberg & Greenwood, 1992] Sederberg, T. W. and Greenwood, E. A physically based approach to 2-D shape blending. *Computer Graphics* 26(2), July 1992, pp. 25–34.
- [Sederberg *et al.*, 1993] Sederberg, T. W., Gao, P., Wang, G., and Mu, H. 2-d shape blending: An intrinsic solution to the vertex path problem. In *Computer Graphics (Annual Conference Proceedings)*, 1993, pp. 15–18.
- [Shoemake & Duff, 1992] Shoemake, K. and Duff, T. Matrix animation and polar decomposition. In *Proc. of Graphics Interface '92*, May 1992, pp. 258–264.