Fractal Animation Metamorphosis Based on Polar Decomposition

Hussein Karam Aboul Ella Hassanien Masayuki NAKAJIMA
Graduate School of Information Science & Engineering, Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo, 152-8552 Japan
{huss,abo,nakajima}@cs.titech.ac.jp

Abstract

This paper presents a new, fast and image-based technique for determining the connectedness of an iterated function system attractors. The technique generates a smooth continuous shape transformation sequence from a given two *IFS* attractors, initial and final. For each shape interpolation, a two parameter family of iterated function systems is defined, and a connectedness locus for these shapes is plotted. Interpolation is performed by decomposing the *IFS* attractors using polar decomposition. Polar decomposition is suggested to use for interpolation because it avoids singular intermediate transformations, better to simulate articulated motion, and its factors have physical and visual interpretation which are not found in other decomposition methods.

Keywords: fractal geometry, iterated function system, computer animation and entertainment, polar decomposition, image morphing

1 Introduction

Image metamorphosis algorithms have been widely used in creating special effects for television commercials, music videos such as Michael Jacksons Black or White, and movies such as Willow and Indiana Jones and the Last Crusade [1,3]. It has proven to be a powerful visual effects tool in animation seen in the entertainment and broad casting industry. Recent advances in shape interpolation have discovered that alternative geometric representations such as iterated function system yields 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 objects often, called fractal. Mandelbrot [2] formally defines a fractal to be a set whose fractal dimension exceeds its topological dimension.

The aim of this paper is, given two *IFS* attractors, initial and final, we have to generate a smooth continuous shape transformation sequence from the initial attractor to the final attractor. Such sequence is generated by decomposing both the initial and final *IFS* attractors using polar decomposition which preserve smoothness for such sequence. Moreover, this paper explores the use of the iterated function system as a geometric representation for shape interpolation using polar decomposition. It uses *IFS* as a representation for two-dimensional shapes as sets, and doesn't require the association of probabilities with each *IFS* map attractor that are often used to define a measure on sets. It also focus on iterated function systems consisting of affine transformations. The family of attractors of such iterated function system are commonly called linear fractals. Linear fractals are shapes that can be constructed from finitely many smaller copies of themselves. Sierpinski triangle, Koch curve, dragon and C-curve are good examples of linear fractal shapes.

The paper is organized as follows. Section 2 reviews some of the previous work in shape interpolation of linear fractal models. Section 3 provides a brief summary of iterated function systems. Section 4 give an overview of polar decomposition computations, advantages, and applies it to the affine transformations of the *IFS* using our proposed method. Finally, experimental results, conclusion and direction for future work are discussed in section 5.

2 Related Work

Iterated function system interpolation has appeared in several animations. Das et al. [14] interpolated a

Sierpinski's tetrahedron into a 3-D dragon by elementwise linear interpolation of the matrices representing the linear transformation of the IFS. Hart [6] controlled only the scaling parameters of the IFS maps to simulate fractal fade-in and dissolve. Prusinkiewicz et al. [12], interpolated L-systems based on turtle geometry (which can be used to represent linear fractals) for the application of animating continuous plant development. Bowman [13] explores the effect of changing scaling and rotation coefficients of the IFS maps. Hart [7] 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 creat each map of each IFS.

The previous work 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 transformation. Unfortunately, in a general key-frame animation, such parameters may not be available, and the IFS representation may consist of nothing more than the affine transformation matrices. Polar decomposition is used for interolation to avoid singular intermediate transformations by extracting the individual transformation components from a general affine transformation matrix and provides a basis for more controlled shape interpolation. Moreover, it generates a continuous smooth transformation sequence that reflects the interpolation correspondence for the IFS attractors.

3 Iterated Function Systems

One of the more common ways to generate fractals is through iterated function systems. The mathematical theory of IFS has a unique advantages for addressing a broad class of modeling problems including the modeling of natural objects and scenes. The feasibility of using IFS theory in computer graphics was reviewed previously at SIGGRAPH'85 [15] and [10]. Iterated function system represents a shape using only transformations and modeling a shape as smaller copies of itself. It is defined as a pair $\{X; T_n, n = 1, 2, \cdots, N\}$, where X is a complete metric space and each T_n are affine contractions, that is,

$$T_i(x) = C_i x + B_i \tag{1}$$

where, C_i is a square matrix with n rows and B_i is a vector with n elements. By a theorem of Hutchinson [8] there exists for each IFS, a single compact non-empty set $\hat{A} \subset R^n$, called its attractor which is the union of attractorlets $\hat{A}_i = T_i(\hat{A})$ under the IFS maps, that is,

$$\hat{A} = \bigcup_{i=1}^{N} T_i(\hat{A}) \tag{2}$$

The Hutchinson operator $W : R^n \to R^n$ is a convenient shorthand notation given by

$$W(\cdot) = \bigcup_{i=1}^{N} T_i(\cdot) \tag{3}$$

that allows us to simplify the definition of an IFS attractor of equation (2) as

$$\hat{A} = W(\hat{A}) \tag{4}$$

On the other hand, 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 3×3 homogeneous transformation matrix. On the other hand, given two IFS attractors the problem is to compute a continuous shape transformation sequence from one to the other. These operation is known variously as shape averaging, shape interpolation, shape blending, shape evolving and metamorphosis. Metamorphosis of shapes described by the IFS representation involves interpolation of the IFS maps. In this paper, interpolation of the IFS maps will be performed using polar decomposition with the aid of the following property,

Property 1 *Let $\{T_i(k)\}_{i=1}^{N}$ be an IFS whose maps are parameterized by a single bounded variable $k \in R$. Then the function $f(k)$, that maps the parameter k into the attractor of the IFS parameterized by k, is continuous.*

The above property prove that small changes in the IFS maps transform the IFS attractor continuously, and provided the basis for IFS-based fractal transformations.

4 Interpolation of IFS attractors

Shape interpolation is a problem which has been motivated by different applications and attacked in several different ways. Transformation of shapes described by the IFS representation involves interpolation of the IFS maps. Iterated function system define shapes using self-transformations, and interpolation of these shapes requires interpolation of these transformation. The interpolation is given in terms of individual parameters such as (rotation angle, scale factor, translation displacement, shear amount), which are the composition of each IFS map. Given two IFS attractors the problem is to compute a continuous shape transformation from one to the other. In this paper we shall decompose the affine map T_i of an IFS attractors using polar decomposition.

4.1 Polar Decomposition: Computations and Advantages

In recent years interests in the polar decomposition have increased because it is useful for a variety of purposes, including matrix animation and interactive interfaces. In this paper we will use it for matrix animation. The polar decomposition [4,11,17] is a generalization to matrices of the familiar complex number representation $z = re^{i\theta}, r > 0$. It is an iterative algorithm for computing the orthogonal polar factor of a non singular $m \times n$ matrix M. The algorithm is based on the well-known Newton-iteration to compute the square root of a number.

Definition 4.1 (Polar Decomposition) *Let M be an $m \times n$ matrix, then there exists a matrix U of order $m \times n$ and a unique Hermitian positive semi-definite matrix H of order $n \times n$ such that $M = UH$. If rank(M)=n then H is positive definite and U is uniquely determined.*

Polar decomposition [5] separates a linear transformation matrix M into an orthogonal matrix U that contains its rotation and reflection components and a symmetric positive definite matrix H that contains its scale and shear components [9]. The factor U is computed by repeatedly averaging the matrix with its inverse transpose. Let $U_0 = M$ and compute

$$U_{i+1} = 1/2(U_i + (U_i^{-1})^T), \tag{5}$$

until $U_{i+1} - U_i \approx 0$. This sequence eventually converges to the orthogonal component U of the linear transformation matrix M. Finding the U matrix of a 2×2 matrix is easy. Suppose for example,

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{6}$$

Then,

$$U = M + sign(det(M)) \begin{pmatrix} d & -c \\ -b & a \end{pmatrix} \tag{7}$$

Where, det denotes the determinant of the matrix M and $sign$ denotes the (positive or negative) symbol of the resulting determinant of matrix M. Now, given U and M, then, H can be calculated as:

$$H = U^{-1}M \tag{8}$$

After computing the polar decomposition of a matrix M, then H will holds the scale and shear parameters while U holds the rotation and reflection parameters. The arctangent of the first column of U yields the angle of rotation. Interpolation of the U matrix consists of constructing rotation matrices based on interpolation of the angle. Polar decomposition is suggested for interpolation because of some reasons:

• It extracts the individual transformation components from a general affine transformation matrix and provides a basis for more controlled shape interpolation.
• The polar decomposition factors are unique, coordinate independent, closeness, simple and efficient to compute.
• The orthogonal matrix U is the closest possible orthogonal matrix to M, a property which is also coordinate independent. that is, U satisfies the following conditions: Find Q minimizing $||U - M||_F^2$

subject to $U^TU - I = 0$, where, the symbol F denote the Frobenius matrix norm given by
$||U - M||_F^2 = \sum_{i,j}(u_{i,j} - m_{i,j})^2$.
• Polar decomposition is applicable to matrices of any size and shape, and its closeness property makes it good for matrix renormalization.
• Polar decomposition has a very physical interpretation as shown in figure 1.

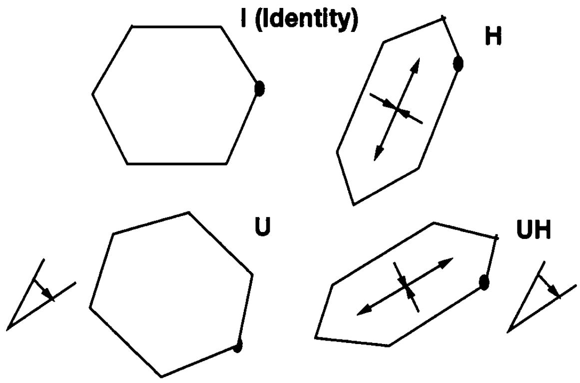

Figure 1. Physical View of Polar Decomposition

All these factors guarantees that small input perturbations will not produce large output variations.

4.2 *IFS* image-based Attractor Decomposition Algorithm

In this section, we will present a new image-based algorithm based on polar decomposition by parameterizing all the transformation of an *IFS* with two parameters. The algorithm plots a corresponding connectedness locus, such that connectedness is maintained during the shape interpolation. During shape interpolation, certain intrinsic properties of the shape should preserved [16], 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 simple connected, multiply connected or totally disconnected. As the parameter of an *IFS* change, during shape interpolation, the connectivity of the resulting attractors can also change. Decompose the affine map T_i of an *IFS* into the polar component matrices U_i, H_i, and a vector translation component B_i is given as follows:

Let θ_i be the angle of the rotation represented by U_i and let the operator $U(\theta)$ return a rotation matrix by the angle θ. We decompose the *IFS* maps $\{T_i\}_{i=1}^{N}$ of equation (1) as:

$$T_i(x) = U(\theta_i)H_i(x) + B_i \quad (9)$$

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. Suppose, θ_i^0, H_i^0, B_i^0 parameterize the maps of the initial *IFS* and let θ_i^1, H_i^1, B_i^1 parameterize the maps of the final *IFS*. Then the two variables (u, v) parameterize a family of an *IFS* with N maps as follows:

$$T_i(x) = U((1-u)(\theta_i^0) + u(\theta_i^1))((1-u)(H_i^0) + u(H_i^1))x + (1-v)(B_i^0) + v(B_i^1) \quad (10)$$

Hence, the parameter u interpolates the linear part of the *IFS* maps, and the parameter v interpolates the translation part of the *IFS* maps.

5 Conclusion and Experimental Results

This paper focused on interpolation of IFS attractors whose iterated function system contained the same number of maps using polar decomposition. For each shape interpolation, a two parameter family of iterated function system is defined and a connectedness locus for these shapes is plotted to maintain connectedness during the interpolation. The proposed algorithm generates smooth transformation sequence that reflects the interpolation correspondence for the *IFS* attractors. The most tedious part of shape metamorphosis is to define the *IFS* attractor correspondence automatically. Furthermore, interpolation between iterated function system with different number of maps is remains an open problem. Recently, we are working on an implementation to generate the in-between sequences for a given two *IFS* fractal shapes attractors automatically.

Figure 2, demonstrates a metamorphosis Dragon to Sierpinski triangle using polar decomposition. Both the initial and final attractors are described by three-map iterated function system. Figure 3, shows a smooth transformation sequence of two *IFS* attractors from sierpinski as an initial *IFS* attractor to a tree as a final *IFS* attractor. Moreover, the connectdness algorithm of using polar decomposition for interpolation is applied to demonstrates an animation sequence of clouds based on changes to the transformations of its *IFS* representation using polar decomposition as shown in figure 4. The smooth transition from frame to frame is a consequence of the continuous transformation result from using polar decomposition for interpolation.

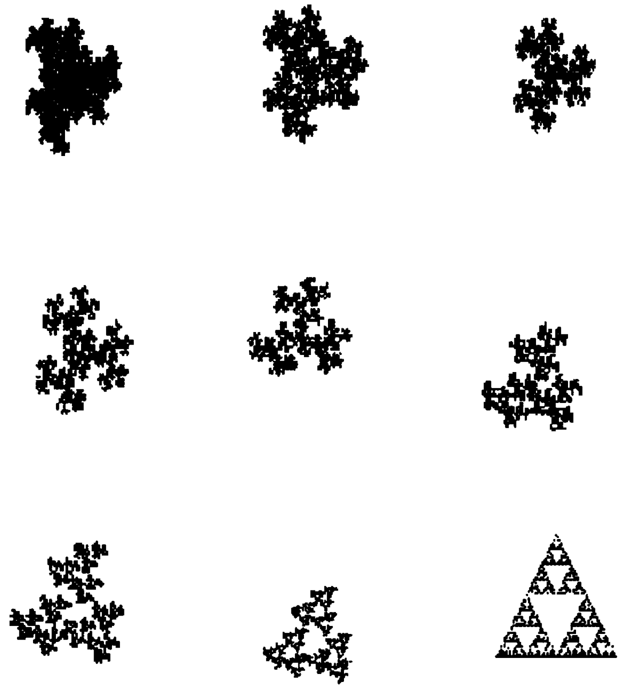

Figure 2. Dragon to Sierpinski triangle metamorphosis

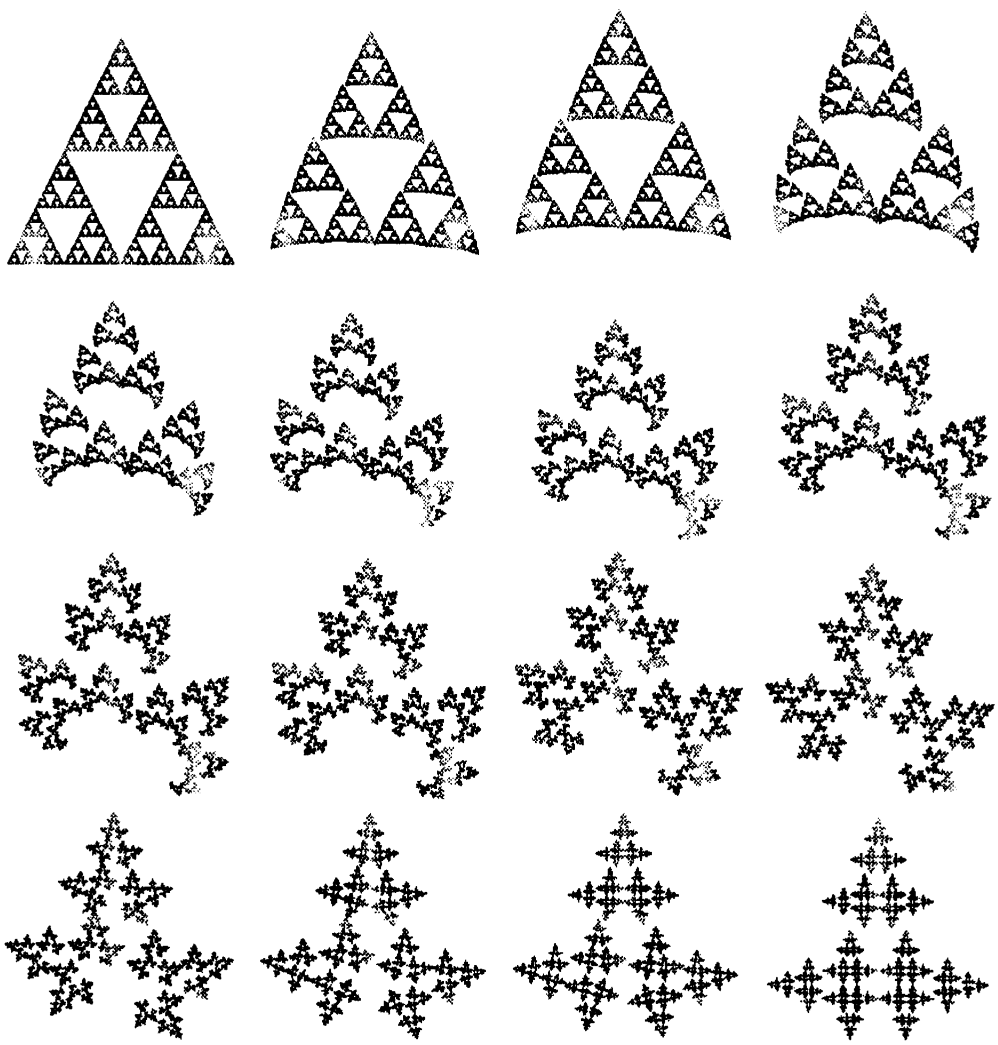

Figure 3. Sierpinski triangle to tree metamorphosis

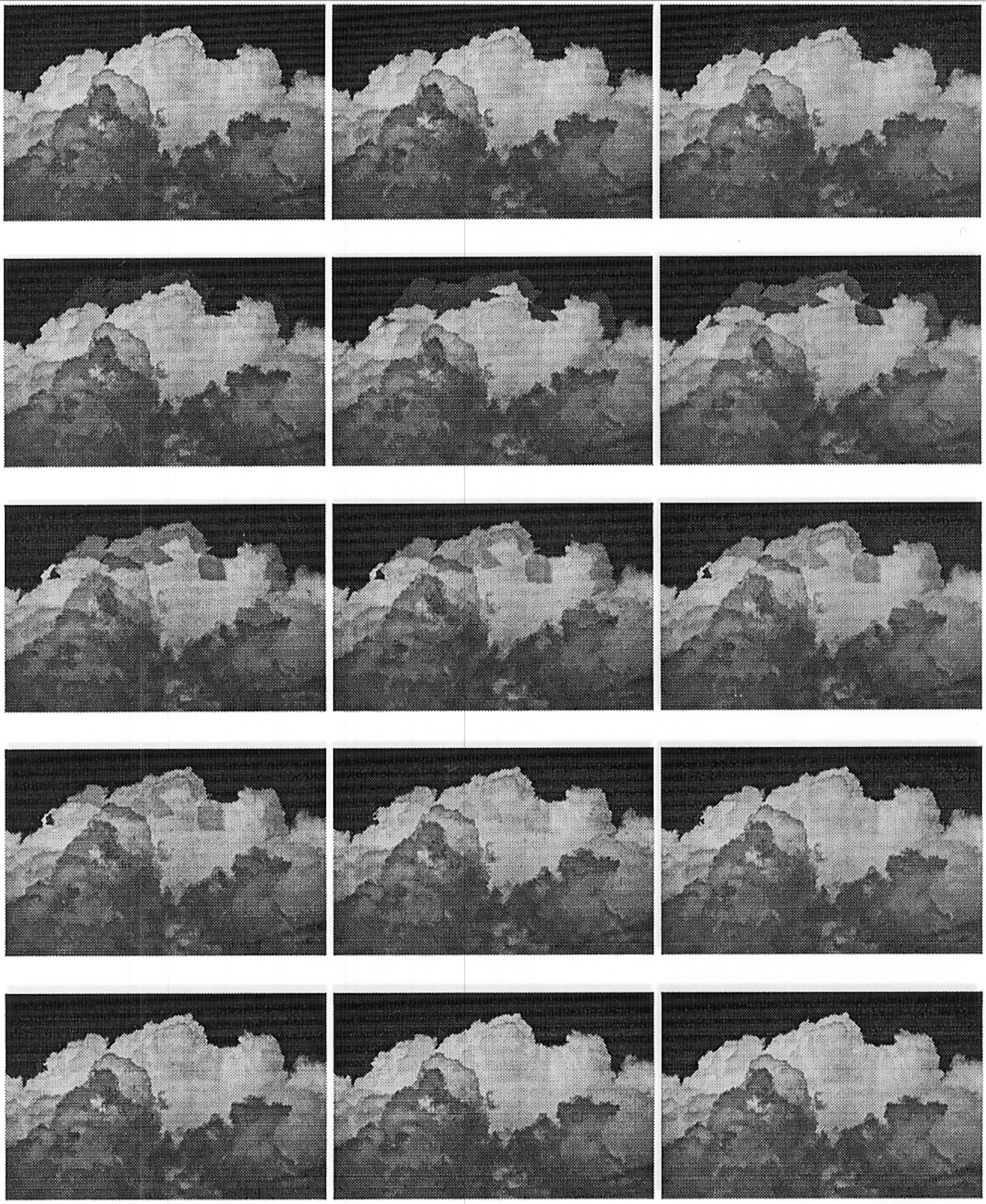

Figure 4. A sequence of frames of IFS encoded cloud using polar decomposition

References

[1] A. Hassanien, H. Karam and M. NAKAJIMA, "Image Metamorphosis Transformation of the Facial Images using Elastic Body Spline", Signal Processing Journal, Vol. 70, No. 2, pp. 129-137, 1998.

[2] B. Mandelbrot, "The Fractal Geometry of Nature", 1983, by B. Mandelbrot.

[3] G. Wolberg, Digital Image Warping, *IEEE Computer Society Press*, 1990.

[4] H. Karam, A. Hassanien, and M. Nakajima, “Polar Decomposition Interpolation for Linear Fractal Metamorphosis”, ITE’98, Annual convention, pp. 200-201, 1998.

[5] H. Karam, A. Hassanien, and M. Nakajima, “Fractal Metamorphosis Animation Using Polar Decomposition” Submitted to PCS’99, USA.

[6] J. Hart, “Fun with Octrees: Graph topologies of the recurrent cube”, SIGGRAPH Video Review 87 (Animation), 1992.

[7] J. Hart, "Unatuaral Phenomena", SIGGRAPH Video Review 71, (Animation), 1991.

[8] J. Hutchinson, "Fractal and self-similarity", Indiana Univ., Mathematics Journal, Vol. 30, No. 5, pp. 713-747, 1981.

[9] K. Shoemake and T. Duff, “ Matrix Animation and polar decomposition”, Graphics Interface’92, pp. 258-264.

[10] M. Barnsely, and S. Demko, “Iterated function systems and the global construction of Fractals”, Proceedings of the Royal Society of London Series A 399, pp. 243-275,1985.

[11] N. Higham, “Computing the Polar Decomposition with Applications”, SIAM J. Sci. and Comp., Vol. 7, No. 4, pp. 1160-1174, 1986.

[12] P. Prusinkiewicz, M. Hammel and E. Mjolsness, “Animations of plant development”, Computer Graphics (Annual Conference Series), Vol. 27, pp. 351-360, 1993.

[13] R. Bowman,"Fractal Metamorphosis: A brief student tutorial”, Computer and Graphics, Vol. 19, No. 1, pp 157-164, 1995.

[14] S. Das and J. Hart, “Sierpinski blows his gasket”, SIGGRAPH Video Review 61 (Animation), 1990.

[15] S. Demko, L. Hodges, and B. Naylor, ” Construction of Fractal Objects with Iterated Function Systems”, Computer Graphics, Vol. 19, No. 3, pp 271-278, SIGGRAPH’85 Proceedings.

[16] T. Sederberg and E. Greenwood, “ A physically based approach to 2-D shape blending”, Computer Graphics, Vol. 26, No. 2, pp 25-34, 1992.

[17] W. Gander, “Algorithms for polar decomposition” SIAM J. Sci. Statist. Comput., Vol. 11, pp. 1102-1115, 1990.