IFS FRACTAL ANIMATION

Nevin M. Darwish
Faculty of Engineering, Cairo University,
Cairo, Giza, Egypt.

Reda Abd-El-Wahab
Institute of Statistical Studies and Research,
Cairo, Giza, Egypt.

ABSTRACT

This paper presents the outline of a system for scientific animation. In particular a set of fundamental procedures based on fractals is given. It is known that the (Iterated Function System) IFS fractal remains IFS under linear deformations. Moreover the parameters of the deformed fractals could be conducted from the original parameters and the linear transforms causing these deformations. A short discussion of how to control the IFS fractals through their parameters is included. As a general rule in fractal-based systems, objects pertain their natural looking under linear deformations because their fractal dimensions do not change under these transformations. As a result, the resolution of the digital images does not change with respect to changes in scale. The projections of 3-D IFS fractals on any plane could also be described by IFS code saying that IFS fractals are isotropic images. The control of 3-D fractals is simply an extension to what is said about 2-D ones. The procedures have been successfully implemented. A set of examples is given at the end. The work concludes with a recommendation for the use of the procedures for scientific animation and an open question for future research.

Keywords

Animation, Fractal, IFS, Linear Transforms, isotropic objects.

1. INTRODUCTION

As proposed in [4], traditional methods of motion specification do not work very well with procedural models in computer animation. It is usually not possible to use key framing or scripting techniques with the models that these techniques produce, because of the large number of primitives they require and the use of pseudo random numbers. One remedy to a classical object oriented approach for animation was using a delegation technique and a system recently reported in [8]. A common requirement for procedural models in which images are represented by parameters and an algorithm acting on them is to do all transformations through the model parameters rather than doing them to image points. This allows the control of the image while it is still represented by the same model. The issue is to have a closed system of representation in which all operators map from one space of representation to the same space. This will pertain the major characteristics of the images generated by the underlying model. In the case of fractal images, this can greatly preserve the natural look of the image.

In the following, we introduce an outline of a system based on the above fact. Next we give an overview of the mathematical details as found in [1, 2] and finally we present some results of its successful implementation and an open question for future work.

2. *Object Oriented System for an IFS Animation*

The IFS fractal representation provides easy ways to handle motion necessary for animation. The fact that the IFS fractal is transformed linearly to a known IFS fractal makes it easy to represent the IFS fractal object as a class in languages supporting Object Oriented Programming techniques. The class contains -as a data structure -the IFS parameters. All methods related to its animation are applied to the fractal parameters-changing them accordingly. The class contains a method that displays the fractal object on digital screens. All changes happening to the fractal object during its motion will update the fractal parameters only. The result of changes is an object belonging to the same class. Of course object oriented programming helps greatly to resolve the complexity problems facing geometric modeling systems. Thence, an IFS fractal based system can be a part of a general system based on object oriented programming for modeling and display software [4]. This part has the advantage of being a highly compressed code. Actually it is not needed to store more than fractal parameters whether a change to an object happens or not.

We have developed a 3-D fractal system based on the above fact. Drawing is mainly performed using a draw function. The draw function is based on the chaos game algorithm of generating fractal objects from a specified set of parameters. The points are immediately projected perspectively on the screen with the eye residing at the system origin. The eye position can be changed by applying linear transformations to the fractal parameters so as to see different views. The parallel projection of a 3-D fractal on a given plane can be represented as will be seen shortly by a 3-D fractal with thickness approaching to zero along this plane and hence can be described by IFS fractal parameters.

3. IFS FRACTALS UNDER LINEAR TRANSFORMS

It was shown by [1] that the IFS fractal appearance change continuously with change in parameters. This fact can be exploited in animating fractal objects. A derived relation showing the effect of linear transforms on fractal parameters supports this fact. Moreover it gives a deterministic way of fractal animation.

3.1. *Mathematical details*

In this part we start by briefly discussing the IFS theory of fractal generation and the effect of continuous change of parameters on the images as introduced by [1]. Next we present the new approach given in [2]. In this parameter change under linear transformations are calculated. This is shown to allow for motion specification of the fractal-based images contrary to many other procedural models in which the generation process usually involves random numbers preventing the animator from directly specifying the motion[6].

From the derived relations, the common 2-D and image transforms could be done in parameter spa including scaling, rotation, translation, shearing, many other complicated but linear ones. The projection on any plane is shown to still be an IFS derived parameters. The basic definitions [4,5,6,7] first be given followed by the main result. Fo complete derivation, the reader is referred to [2].

3.2. *Basic definitions*

Definition 1 (IFS fractal)

An IFS fractal A defined by the parameters of transforms W_i, $i=1..N$ is the set of points $x \epsilon A$ s

that $A = \bigcup_{i=1}^{N} W_i(A)$

where $W_i(A)$ is the set coming from applying W_i each $x \epsilon A$. .

Definition 2. (contractivity of a transform)

A transformation $f: X \rightarrow X$ on a metric space (X,d) called
contractive if there is a constant $0 \leq s < 1$ such th

$$d(f(x), f(y)) \leq s.d(x,y) \, \forall x, y \epsilon X$$

Any such number s is called a contractivity factor f.

Definition 3. (contractivity of IFS)

A (hyperbolic) iterated function system IFS consists a complete metric space (X,d) together with a finite of contraction mappings W_n: $X \rightarrow X$ with respecti contractivity factors

$$S_n, \text{ for } n = 1,2,..,N$$

The contractivity factor of the IFS is

$$s = \max \{S_n \; n = 1,2,..,N\}$$

Definition 4. (Hausdorff distance)

Let (X,d) be a complete metric space, then t Hausdorff distance between points A,B $\epsilon H(X)$, compact subsets of X, is defined by:

$$h(A,B) = \max\,(d(A,B),\ d\,(A,B))$$

where

$$d(A,B) = \max_{x \in A}\ (\min_{y \in B}(d\,(x\,,y))$$

3.3. *The Effect of Continuous Change in Parameters on Fractal Images*

Let (X,d) be a metric space. Let $\{X;\ W_i,\ i = 1,2,...,N\}$ be a hyperbolic IFS (with condensation) of contractivity s. For n = 1,2,...,N let W_n depend continuously on a parameter $p \in P$, where P is a compact metric space. Then the attractor $A(p) \in H(X)$ depends continuously on $p \in P$ with respect to the Hausdorff metric h(d).
This means - as Barnsley said - that we can smoothly interpolate between attractors provided they remain hyperbolic and this is useful for image animation [1].

3.4. *Linear Transformations to IFS fractals*

It was shown in [2] that the IFS fractal $\{A;\ W_i\ i=1,2,...,N\}$ with linear transforms $W_i(x) = A_i\ x+b_i$ is transformed to another IFS fractal $\{\bar{A};\bar{W}_i, i=1,2,\ldots N\}$ under the linear transform $T(x)=\ A_T(x)+b_T$

Where:

$$\bar{W}_i(x) = \bar{A}_i(x) + \bar{b}_i$$

$$\bar{A}_i = \ A_T\ A_i\ A_T^{-1} \qquad (1)$$

$$\bar{b}_i = \ A_T b_i +\ b_T -\ A_T A_i A_T^{-1} b_T$$

Proof

Let $x \in A$. Under T, x will be transformed to $\bar{x}$ where

$$\bar{x} = T(x) \qquad (2)$$

On the other hand, x is transformed to x_i; i = 1...N under each of the transforms W_i where

$x_i = W_i(x)$ and $x_i \in A$ (from the fractal definition).

The points x_i are also transformed to $\bar{x}_i$ under T where $\bar{x}_i = T(x_i)$ By noting that $\bar{X}$, $\bar{x}_i$ belong to the target image $\bar{A}$ and

$$\bar{x}_i = T(x_i) = T(W_i(x)) = A_T W_i(x) + b_T$$

$$= A_T\ (A_i\ x + b_i) + b_T$$

from (2):

$$x = A_T^{-1}\bar{x} - A_T^{-1} b$$

$$\bar{x}_i = A_T(A_i A_T^{-1}\bar{x} - A_i A_T^{-1} b_T + b_i) + b_T$$

$$= \bar{A}_i\ x + \bar{b}_i$$

where

$$\bar{A}_i = A_T A_i\, A_T^{-1}$$

$$\bar{b}_i = b_T + A_T b_i - A_T A_i A_T^{-1} b_T$$

$$= (I - A_T\ A_I\ A_T^{-1})\ b_T + A_T\ b_i$$

Now if W_i is contractive then $\bar{W}_i$ is also a contractive mapping. This is true since the transformation $A_T\ A_i\ A_T^{-1}$ is the transform of the matrix A_i when changing the basis vectors through A_T and the proof is complete.

Now, we can implement general procedures for scaling, rotation, ..etc. to the fractal by substituting in the above relation. For example when rotating a fractal about z axis, we can use the transformation T such that

$$A_T = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, b_T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

where θ is the angle of rotation.
The procedure of rotation accepts as a parameter the angle of rotation to construct the above matrix then by substituting in relations 1, the new parameters are calculated.

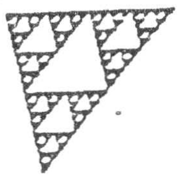

Fig i Sierpinski Triangle

```
Transform Parameters :

|0.50 0.00 0.00|   |15.06|   |0.50 0.00 0.00|   |15.30|   |0.50 0.00 0.00|   |15.30|
|0.00 0.50 0.00| , | 0.06| & |0.00 0.50 0.00| , | 0.00| & |0.00 0.50 0.00| , | 0.30|
|0.00 0.00 0.50|   | 0.00|   |0.00 0.00 0.50|   | 0.30|   |0.00 0.00 0.50|   | 0.30|
```

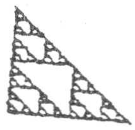

Fig ii
After rotation about x,y,z

```
| 0.50-0.00 0.00|   |15.06|   | 0.50-0.00 0.00|   |15.40|   | 0.50-0.00 0.00|   |15.46|
| 0.00 0.50 0.00| , | 0.01| & | 0.00 0.50 0.00| , | 0.03| & | 0.00 0.50 0.00| , | 0.23|
|-0.00 0.00 0.50|   |-0.06|   |-0.00 0.00 0.50|   | 0.14|   |-0.00 0.00 0.50|   |-0.08|
```

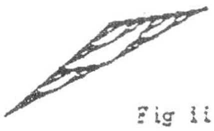

Fig iii (After more rotations)

```
| 0.50-0.00-0.00|   |14.99|   | 0.50-0.00-0.00|   |15.33|   | 0.50-0.00-0.00|   |15.17|
| 0.00 0.50 0.00| , |-0.08| & | 0.00 0.50 0.00| , |-0.26| & | 0.00 0.50 0.00| , |-0.45|
|-0.00 0.00 0.50|   |-0.00|   |-0.00 0.00 0.50|   |-0.03|   |-0.00 0.00 0.50|   |-0.20|
```

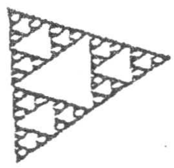

Fig iv
(After more rotations)

```
|0.50 0.00 0.00|   |15.07|   |0.50 0.00 0.00|   |15.34|   |0.50 0.00 0.00|   |15.38|
|0.00 0.50-0.00| , | 0.05| & |0.00 0.50-0.00| , | 0.03| & |0.00 0.50-0.00| , | 0.32|
|0.00-0.00 0.50|   |-0.02|   |0.00-0.00 0.50|   | 0.25|   |0.00-0.00 0.50|   | 0.16|
```

Figures 1. Sample output.

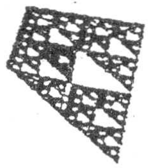

Fig v a Fractal pyramid

```
0.0-0.0| |15.0||0.5 0.0-0.0| |15.0||0.5 0.0-0.0| |15.1||0.5 0.0-0.0| |14.9|
0.5 0.0| |-0.1 |0.0 0.5 0.0| | 0.0 |0.0 0.5 0.0| |-0.3 |0.0 0.5 0.0| |-0.3|
0.0 0.5| | 0.1 |0.0 0.0 0.5| | 0.4 |0.0 0.0 0.5| | 0.3 |0.0 0.0 0.5| | 0.5|
```

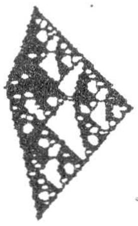

Fig vi Rotation of v

```
                |15.1|0.5 0.0 0.0|  |15.3|0.5 0.0 0.0|  |15.4|0.5 0.0 0.0|  |15.5|
0.0 0.0|        | 0.1|0.0 0.5 0.0|, | 0.1|0.0 0.5 0.0|, | 0.1|0.0 0.5 0.0|, | 0.3|
0.5 0.0|,       |-0.0|0.0 0.0 0.5|  |-0.2|0.0 0.0 0.5|  | 0.1|0.0 0.0 0.5|  |-0.0|
0.0 0.5|
```

Figures 1. sample output (Continue).

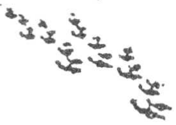

Fig vii a natural scene

```
| 0.39-0.01-0.27|   |18.62|  | 0.46 0.09-0.42|   | 16.12|
|-0.27 0.33 -0.1| , | 7.99|& |-0.43 0.39-0.04| , | 13.06|
|-0.04-0.29 0.29|   | 1.09|  | 0.05-0.42 0.34|   | -1.64|
```

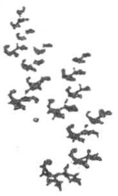

Fig viii Rotation of vii

```
| 0.15-0.25-0.07|   |25.59|  | 0.18-0.31-0.17|   | 24.51|
|-0.21 0.35 0.19| , | 6.02|& |-0.23 0.42 0.36| , |  7.02|
| 0.15-0.14 0.50|   |-4.41|  | 0.26-0.30 0.59|   | -8.19|
```

Figures 1. sample output (Continue).

3.5. Projection of IFS Fractal

If we apply a scaling transform in a given direction say z making the scaling factor very small then the result is a planner fractal-with derived parameters- in the x-y plane. This shows that the IFS fractal objects are isotropic recalling that an object is isotropic if it has the same fractal dimension in any direction. Many natural objects have this property.
Mathematically:

Let the parameters of the fractal be:

$$A_i = \begin{bmatrix} a_i & b_i & c_i \\ d_i & e_i & f_i \\ g_i & h_i & i_i \end{bmatrix}, b_i = \begin{bmatrix} j \\ k \\ l \end{bmatrix}$$

If we apply a transform:

$$A_T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \epsilon \end{bmatrix}, b_T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

where ϵ is a very small number, to the fractal image the result is a fractal of thickness approaching to zero which coincides with the parallel projection of the original fractal on the x-y plane. This projection can be described by a fractal with the following parameters:

$$\bar{A}_i = \begin{bmatrix} a_i & b_i & c_i/\epsilon \\ d_i & e_i & f_i/\epsilon \\ \epsilon g_i & \epsilon h_i & i_i \end{bmatrix}, \bar{b}_i = \begin{bmatrix} j \\ k \\ \epsilon l \end{bmatrix}$$

3.6. General Procedures

The system outline for animation starts by giving it the fractal of the object (or in general the procedure for calculating the fractal) as well as the required linear transformation both input is processed through matrix operations and the resultant fractal is produced which is passed to the drawing function based on the Chaos Game and finally the output image may be displayed in pixel form.

The fractal object class definition contains the space dimension variable, the number of fractal parameters, the fractal parameters themselves (a matrix and a vector per parameter) and the animation methods. The methods include a drawing function based on the chaos game algorithm, general transform methods that accept transform parameters for special transforms like rota scaling, translation and projection.

The general form of the procedures is then as follo

Step 1. Calculate the appropriate transformation ma
Step 2. Calculate the new parameters using equati (1).
Step 3. Use the draw function with the appropr parameters.

Obviously, the procedures are quite simple and yet v powerful and their implementation show promis forms. The advantage of using such a system comes fr the fact that all the motion frames could be coded fractal parameters and hence the system requires l memory space than traditional animation systems addition, the time required to animate an object cons of the time of matrix multiplications and additi required for the relations of the IFS fractal under transform and the time of drawing the IFS frac Obviously, the later time component will merely dep on the algorithm of drawing used as a measure for system speed.

4. RESULTS AND CONCLUSION

A set of images showing a sample of output of system are given in Figures (1). Different versions of same image are displayed but under linear transform is worth noting that all transforms including scaling any direction appear to be natural. Some of the ima are 2-D fractals while others are 3-D ones. Th samples recommend their use for scientific animat The question that still remain is whether the intersec of two fractals could also be represented as a fracta which case one would not need to return to image po at all.

REFERENCES

[1] Barnsley, M., "Fractals Everywhere", *Acade Press Inc.*, 1988.

[2] Darwish, N. El-Wahab R., "On the Inverse Fra Problem", *Journal of Faculty of Engineering, C University*, April 1993.

[3] Grant, E., Amburn, P. and Whitted,T., "Explo Classes in Modeling and Display Software", I *Computer Graphics and Applications*, Nov. 19

[4] M. Green, H. Sun, "A Language and System Procedural Modeling and Motion", IEEE *Comp Graphics and Application*, Nov. 1988.

[5] B. Mandelbrot, "The Fractal Geometry of Natu W.H. *Freeman and Co.*, CA,1982.

[6] Ed. Peitgen, H.O., Saupe, D., "The Science Fractal Images", Springer-Verlag, New York, 1

3.5. Projection of IFS Fractal

If we apply a scaling transform in a given direction say z making the scaling factor very small then the result is a planner fractal-with derived parameters- in the x-y plane. This shows that the IFS fractal objects are isotropic recalling that an object is isotropic if it has the same fractal dimension in any direction. Many natural objects have this property.
Mathematically:

Let the parameters of the fractal be:

$$A_i = \begin{bmatrix} a_i & b_i & c_i \\ d_i & e_i & f_i \\ g_i & h_i & i_i \end{bmatrix}, b_i = \begin{bmatrix} j \\ k \\ l \end{bmatrix}$$

If we apply a transform:

$$A_T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \epsilon \end{bmatrix}, b_T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

where ϵ is a very small number, to the fractal image the result is a fractal of thickness approaching to zero which coincides with the parallel projection of the original fractal on the x-y plane. This projection can be described by a fractal with the following parameters:

$$\bar{A}_i = \begin{bmatrix} a_i & b_i & c_i/\epsilon \\ d_i & e_i & f_i/\epsilon \\ \epsilon g_i & \epsilon h_i & i_i \end{bmatrix}, \bar{b}_i = \begin{bmatrix} j \\ k \\ \epsilon l \end{bmatrix}$$

3.6. General Procedures

The system outline for animation starts by giving it the fractal of the object (or in general the procedure for calculating the fractal) as well as the required linear transformation both input is processed through matrix operations and the resultant fractal is produced which is passed to the drawing function based on the Chaos Game and finally the output image may be displayed in pixel form.

The fractal object class definition contains the space dimension variable, the number of fractal parameters, the fractal parameters themselves (a matrix and a vector per parameter) and the animation methods. The methods include a drawing function based on the chaos game algorithm, general transform methods that accept transform parameters for special transforms like ro scaling, translation and projection.

The general form of the procedures is then as fol

Step 1. Calculate the appropriate transformation m
Step 2. Calculate the new parameters using equ (1).
Step 3. Use the draw function with the approp parameters.

Obviously, the procedures are quite simple and ye powerful and their implementation show prom forms. The advantage of using such a system comes the fact that all the motion frames could be code fractal parameters and hence the system requires memory space than traditional animation system addition, the time required to animate an object co of the time of matrix multiplications and add required for the relations of the IFS fractal unde transform and the time of drawing the IFS fr Obviously, the later time component will merely de on the algorithm of drawing used as a measure fo system speed.

4. RESULTS AND CONCLUSION

A set of images showing a sample of output o system are given in Figures (1). Different versions o same image are displayed but under linear transfor is worth noting that all transforms including scali any direction appear to be natural. Some of the im are 2-D fractals while others are 3-D ones. T samples recommend their use for scientific anim The question that still remain is whether the interse of two fractals could also be represented as a frac which case one would not need to return to image p at all.

REFERENCES

[1] Barnsley, M., "Fractals Everywhere", *Aca Press Inc.*, 1988.

[2] Darwish, N. El-Wahab R., "On the Inverse F Problem", *Journal of Faculty of Engineering, University*, April 1993.

[3] Grant, E., Amburn, P. and Whitted,T., "Exp Classes in Modeling and Display Software", *Computer Graphics and Applications*, Nov.

[4] M. Green, H. Sun, "A Language and Syste Procedural Modeling and Motion", IEEE *Co Graphics and Application*, Nov. 1988.

[5] B. Mandelbrot, "The Fractal Geometry of N W.H. *Freeman and Co.*, CA,1982.

[6] Ed. Peitgen, H.O., Saupe, D., "The Scie Fractal Images", Springer-Verlag, New York

[7] Pentland, A.P. "Fractal Based Description of Natural Scenes", IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), Vol. 6, No. 6, Nov. 1984.

[8] Zelezirik, R. et al, " An Object Oriented Framework for the integration of interactive Animation Techniques ", Computer Graphics (proc. Siggraph), Vol. 25, No. 4, July 1991.