Metamorphic Animation of 3D Fern-like Fractal Images based on a Family of Transitional 3D IFS Code Approach

Tedjo Darmanto[1], Iping Supriana Suwardi[2], Rinaldi Munir[3]

[1]Sekolah Tinggi Manajemen Informatika & Komputer, AMIK Bandung Jl. Jakarta 28. Bandung
[2,3]School of Electrical Engineering and Informatics, Bandung Institute of Technology. Jl.Ganesha 10 Bandung
[1]tedjodarmanto@gmail.com, [2]iping@informatika.org, [3]rinaldi@informatika.org

ABSTRACT

Based on a family of transitional 3D IFS code, the attractive metamorphic animation of 3D fractal images between two kinds of 3D fractal images can be conducted. To show an animation which is looked natural, two kinds of 3D fern-like IFS fractal are chosen with small difference of orientation in both x-axis and z-axis directions or horizontally and also in y-axis direction or vertically. There are two ways of animation is conducted, those are with and without spin animations. The spin animation exhibits the three dimensional point of view and the other animation exhibits the accuracy of gradual changing from the start image to the target image without lost of orientation. The 3D IFS code sets as the start and target objects should be in a family of transitional 3D IFS code, otherwise the result of the metamorphic animation will show a strange transition effect

Keywords: *Metamorphic Animation, 3D IFS code, 3D fern-like IFS fractal, transitional 3D IFS code, spin*

I. Introduction

The morphing animation techniques is always interesting, on both 2D and especially 3D images. Metamorphic animation of 3D IFS code fractal can be accomplished by interpolating the IFS code coefficients between the start and target images. The problem is the transitional images as the results of metamorphic animation are not always looked natural to represent the in between appearance of both start and target images. Fortunately there is a solution to overcome the problem, that is by choosing the right order of functions in the IFS code as the representation of affine transformation functions of both start and target fractal images

II. Related Work

The work on IFS fractal interpolation for 2D and 3D visualization that is conducted by Wittenbrink [1] has tremendous speed advantage over ray tracing and the interpolations may be used to give an indication of the uncertainty of the data and may allow more accurate data analysis. Through the work on fractal-based algorithm for metamorphic animation that is studied by Chen et.al. [2], the size of objects can be measured, for which the traditional way based on Euclidean geometry is failed. An image morphing based on pixel transformation approach that is proposed by Rahman et.al. [3] depict the transformation of pixels with their neighborhoods; this method is organized with the replacement of the pixel values of a source image and convolving the neighbor with the help of a mask that is fast and efficient for image morphing. An interesting work on L-system and 3D IFS fractal simulation that is conducted by Fang and Lifeng [4] can create a realistic 3D tree model successfully through the combination between generating the branch by L-system and generating the leaf by IFS. In their paper Wang et.al [5] propose a new method of drawing 3D plants. Through the observation, plants have the golden sections around 34.4 and 55.6 degrees of a branch angle 90 degree as a standard. The bamboo, herb and poplar tree as the examples of 3D plants IFS are reconstructed under OpenGL environment. In their paper Zhuang et.al [6] study a morphing IFS fractal by calculating local attractor's coarse convex-hull and selecting rotation matching between IFSs. The morphing procedure of two IFS's fractal attractors is done by interpolating the parameters of the iterated function.

III. Fractal in 3D IFS Code

A. Fractal and IFS Code

Normally object has one, two or three dimensions, such as point, line, or filled rectangle. If an object has fractional dimension, then it is classified as a fractal object [7], such as Koch curve, Sierpinsky gasket, and many others. One way to encode fractal object is by IFS code [9] [10] and the most famous IFS object fractal is Barnsley fern fractal which has four contractive functions of IFS code [10]. Actually there is another way to construct plant-like fractal objects, such as by L-system [4], but for the case of morphing animation is more suitable using the IFS model that can be realized by interpolating between the coefficients of the IFS code of the source and target objects [2].

B. 3D IFS Code

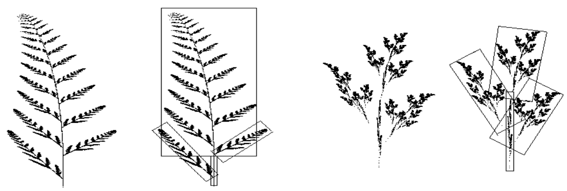
3D Fractals can be encoded into 3D IFS code based on 3D affine transformation function, the collage theorem and the self similarity property of fractal [8]. In 3D IFS code form, 3D fractal has twelve coefficients, the first nine coefficients determines the form of fractal object in three directions (3D), and the last three other coefficients determines the position of fractal object relatively from the origin of space. Mathematically the typical 3D IFS fractal can be represented by the-**a, b, c, d, e, f, g, h, i, j, k** and **l** coefficients of the affine transformation function [10] as described in figure-1 below.

$$w\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} j \\ k \\ l \end{bmatrix}$$

Figure-1. 3D Affine transformation functions

C. 3D Fern-like IFS Code

The formal example of IFS code fractal is fern fractal. The fern-like fractals have four contractive functions. One function with the smallest probability factor represents the skeleton of leaf (usually in the vertical direction), two functions represent the right and left side parts of leaf and the last one with the largest probability factor represents the top of leaf in vertical direction as described in the figure-2 (a & b) below as an illustration of collage theorem and in table-I and table-II as an example of 3D fern-like (1) IFS code and 3D fern-like (2) IFS code



(a) (b)

Figure-2. Fern-like fractal: model-1 (a), model-2 (b)

TABLE-I An example of 3D fern-like (1) IFS code

a	b	c	d	e	f	g	h	i	J	k	l
0.00	0.00	0.01	0.00	0.26	0.00	0.00	0.00	0.05	0.00	0.00	0.00
0.20	-.26	-.01	0.23	0.22	-.07	0.07	0.00	0.24	0.00	0.80	0.00
-.25	0.28	0.01	0.26	0.24	-.07	0.07	0.00	0.24	0.00	0.22	0.00
0.85	0.04	-.01	-.04	0.85	-.09	0.00	0.08	0.84	0.00	0.80	0.00

TABLE-II An example of 3D fern-like (2) IFS code

a	b	c	d	e	f	g	h	i	j	k	l
0.05	-.00	-.00	0.00	0.60	0.00	-.00	-.00	0.05	0.00	0.00	0.00
0.45	-.22	0.22	0.22	0.45	0.22	-.22	0.22	-.45	0.00	1.00	0.00
-.45	0.22	-.22	0.22	0.45	0.22	0.22	-.22	0.45	0.00	1.25	0.00
0.49	-.08	0.08	0.08	0.49	0.08	0.08	-.08	0.49	0.00	2.00	0.00

D. Family of Transitional 3D IFS Code

For a single IFS code fractal image, the order of contractive functions can be written in random manner and will not change the form of fractal images as the result of decoding IFS code through the random iteration algorithm. For the sake of the natural visualization of morphing animation from one IFS code fractal image to the other IFS fractal image, the order of the functions of both fractals should be in a family of transitional IFS code which depends on the type of fractal image. For the 3D fern-like fractal the order of functions can be sorted by the value of probability factor and the sign of coefficients in the x-axis and the y-axis directions [11]. To describe the idea of the family of transitional 3D IFS code more clearly, table-I and II show two kinds of 3D fern-like IFS code fractal as an example of the IFS code pair of the start and target images pair that have the self-affine function order based on the representation of the same part of object, the first order is the skeleton representation, the second and the third orders are the right and left sides representation and the last is the representation of top part of object. The order of both IFS code sets also are set based on the increasing value of the probability values of the self-affine functions

IV. SPIN AND MORPHING

A. Spin Around y-axis

One way of animation that can be used to explore the visualization of 3D effects is to rotate the fractal object around one of the space axis's. The best way to explore the visualization of 3D fern-like fractal that stands at the x-z plane is to rotate it or make it spinning around the y-axis. The speed of rotation can be controlled by three subtle changes of angle in space clock wisely or encounter clock wisely. In mathematics expression the rotation around the y-axis can be expressed by the equation [12] as described in figure-3 below

$$a' = a.\cos(x) + c.\sin(x)$$
$$d' = d.\cos(x) + f.\sin(x)$$
$$g' = g.\cos(x) + i.\sin(x)$$
$$c' = c.\cos(z) - a.\sin(z)$$
$$f' = f.\cos(z) - d.\sin(z)$$
$$i' = i.\cos(z) - g.\sin(z)$$

Figure-3. Spin around y-axis

B. Transition of Metamorphic Animation

The metamorphic transition between the start image and target image can be made based on the interpolation of the affine transformation coefficients between the two images. The rate of metamorphic changes between the two images is determined by the iteration number in one cycle of metamorphic animation. It is a good idea to set the speed of rotation mentioned in above subsection and the rate of metamorphic changes in the same fashion, so periodically the metamorphic changes and the rotation effect can be seen simultaneously, as presented in the figures (4 and 5) in the next section. An interesting result is the gradually change of visualization between the right and left parts of the objects. The movement direction of the left and right sides of the source fern object are in the opposite directions. The left side moves up and the right side moves down.

V. Animation

A. *Metamorphic Animation without Spin*

The first simulation of animation that shows the metamorphic changes of the two fern-like fractals is conducted without the spin animation around the axis's or in the steady state as can be seen in the figure-4 below, as the illustration of the first simulation (the direction of the top of object is changed slightly from left to right)

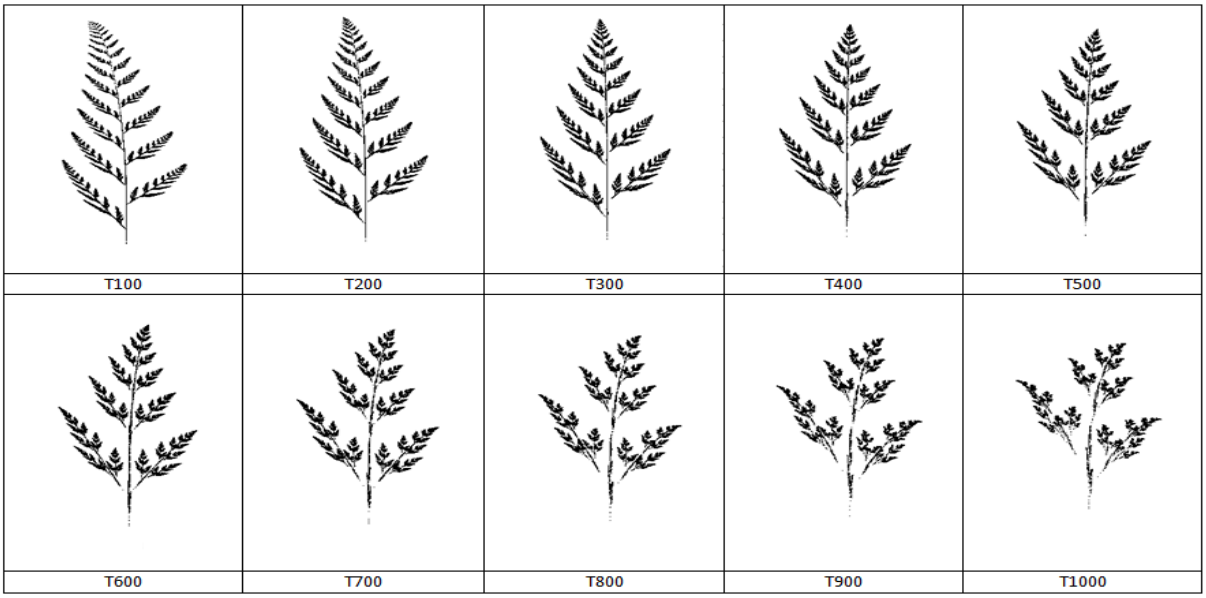


Figure-4. Transitional images of metamorphic animation of fern-like fractals without spin

B. *Metamorphic Animation with Spin*

The second simulation of animation that shows the metamorphic changes of the two fern-like fractals in 3D visualization effect is conducted with the spin animation around the y-axis as can be seen in the figure-5 below, as the illustration of the second simulation.

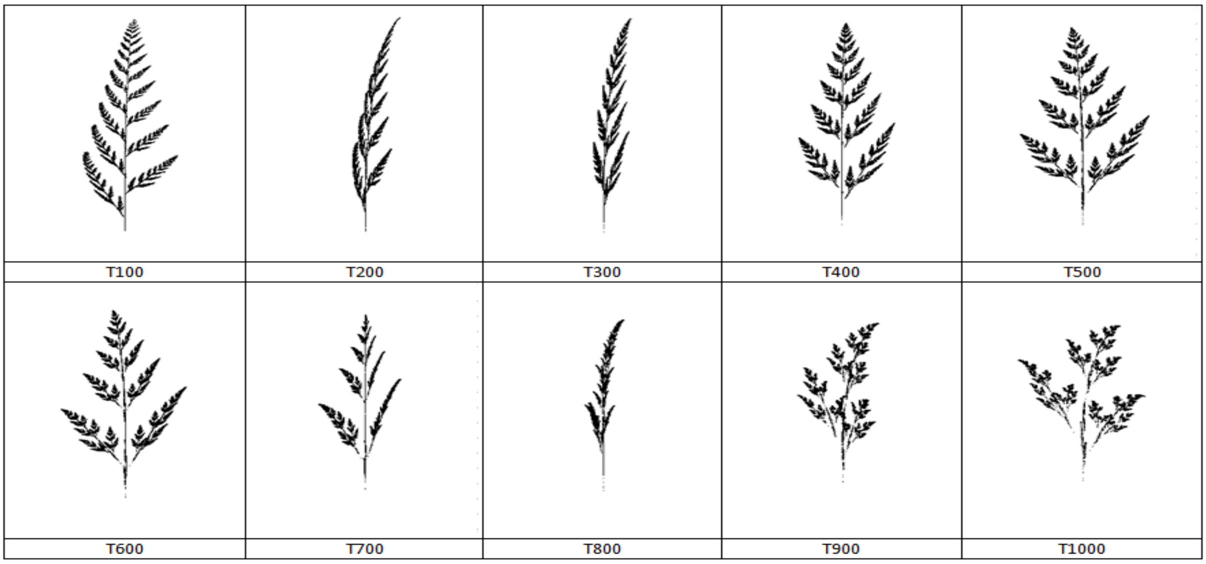


Figure-5. Transitional images of metamorphic animation of fern-like fractals with spin

VI. Conclusion

A metamorphic animation on a pair of 3D fractal objects can exhibit a natural animation if the order of self-affine functions in the 3D IFS code is set as a family of transitional 3D IFS code, which relatively represents the same part of object. The animation on a pair of 3D fractal objects with and without spin or rotation around the y-axis exhibit two kinds of aspect. The spin metamorphic animation shows 3D perspective and the no spin metamorphic animation shows the gradually changes between the source and target images more clear, especially in showing the movement of the left and right part of the source object (up and down in the opposite direction) and also the direction movement of the top (left to right).

VII. Future Work

For the related future work of this study, the research can be extended to study the metamorphic animation of more than two images of 3D fractal images based on the family of multi-transitional 3D IFS code version cyclically

References

[1] Wittenbrink, Craig M, 1995, "IFS Fractal Interpolation for 2D and 3D Visualization", Computer Engineering & Information Sciences, University of California.

[2] Chuanbo Chen,Yunping Zheng & M Sarem, 2006, "A Fractal-based Algorithm for the Metamorphic Animation", 0-7803-9521-2/06 ©2006 IEEE, pp.2957-2963

[3] M.T. Rahman, Al-Amin, M.A. Jobayer, A.R. Chowdhury & M.A Bhuiyan, 2007, "A Novel Approach of Image Morphing Based on Pixel Transformation", 1-4244-1551-9/07 ©2007 IEEE.

[4] Deng Fang, Xi Lifeng, 2008, "An Application of L-system and IFS in 3D Fractal Simulation", *WSEAS TRANSACTIONS on SYSTEMS, No 4 Vol 7, 2008, ISSN:1109-2777, pp.352-361*

[5] Yan Wang, Lanling Zeng, & Yan Qi, 2008, "Golden Section in 3D Plants IFS Reconstruction", *The 9th International Conference for Young Computer Scientists, 978-0-7695-3398-8/08 ©2008 IEEE, pp.2962-2967*

[6] Zhuang, Yi-Xin, Xiong, Yue-Shan, Liu, Fa-Yau, 2011, "IFS Fractal Morphing based on Coarse Convex-Hull". *Information Technology and Artificial Intelligence Conference IEEE Joint International 2011*

[7] Mandelbrot, Benoit B. 1982. "The Fractal Geometry of Nature". W.H. Freeman and Company.

[8] Hutchinson, John E. 1979. "Fractals Self-similarity". *Indiana University Mathematics Journal 30*

[9] Barnsley, M. F., and Demko, S. 1985. "Iterated Function Systems and the Global Construction of Fractals". *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, Vol. 399, No. 1817 (Jun. 8, 1985), pp. 243-275*

[10] Barnsley, Michael F. 1993. "Fractals Everywhere". 2nd edition. Morgan Kaufmann,. Academic Press

[11] Darmanto, T., Suwardi, I.S., & Munir, R., 2012, "Cyclical Metamorphic Animation of Fractal Images Based on a Family of Multi-transitional IFS Code Approach", *IEEE Conference on Control, Systems & Industrial Informatics 2012 , 878-1-4673-1023-9/12 ©2012 IEEE, pp.231-234*

[12] Oliver, Dick, 1992, "Fractal Vision: Puts Fractals to Work for You"; Sam Publishing co.