Computer-Aided Design and Applications

http://www.cadanda.com

Interactive Evolutionary 3D Fractal Modeling with Modified IFS

Wenjun Pang[1] and K. C. Hui[2]

[1]The Chinese University of Hong Kong, wjpang@mae.cuhk.edu.hk
[2]The Chinese University of Hong Kong, kchui@mae.cuhk.edu.hk

ABSTRACT

In this research we present a technique for the automatic generation of 3D art form, which allows designers to get access to large number of 3D shapes that can be altered interactively. This is based on a revised evolutionary algorithm using a modified Iterated Function System, which is a combination of transformations in 3D Euclidean space, and which can provide tunable geometric parameters. The operations employed by the evolutionary algorithm are single-point crossover, arithmetic Gaussian mutation and inferior elimination. The system proposed can greatly enhance the productivity of visually appealing fractal greatly. Experimental results of the study demonstrated the effectiveness of the proposed modified IFS formula and evolutionary system. Examples in applying the proposed method in the design of jewelry and decoration are also presented.

Keywords: 3d fractal, IFS, genetic algorithm, interactive evolutionary design.
DOI: 10.3722/cadaps.2009.55-67

1. INTRODUCTION

Works on the application of fractal in the field of art and design started almost three decades ago. M. Barnsley and S. Demko [2] put forward an Iterated Function System (IFS) to build fractals. J.C. Sprott [18] extended the system to build fractals art and evaluated fractal aesthetic value, which leads to a series of related research and art works subsequently. These methods are mainly confined to producing 2-D art image, and the productivity of visually appealing fractal generated using IFS with popular Random Iteration Algorithm [3] is low. One possible way to solve this problem is to build an evolutionary system with a suitable selection function.

An automatic generator of 3D fractals with acceptable aesthetic value is built in this research, which also allows user modification of the generated fractals. This system is based on a revised evolutionary algorithm with a modified IFS formula which provides satisfactory performance in the generation of artistic fractals. The fitness of fractal, which sorts fractals by aesthetic value, is formulated by considering the mathematical characteristics of fractal, including the fractal dimension and Lyapunov exponent. The proposed technique has various potential and promising applications, such as in jewelry design, decoration design and animation.

2. FRACTAL MODELING

2.1 Fractal Art

Fractal art is usually created by calculating fractal objects and representing the results as still images, animations, music, textures etc.. Fractal objects in fractal art fall into four main categories, escape time fractals, Lindenmayer systems, stochastic synthesis and Iterated function systems. Among various application of fractal, 2D still fractal images, which are mainly used as art collection, texture and screen saver, are most popular and are investigated by many researchers.

Fractal Flame [15], put forward by Scott Draves, creates images using a two-dimensional IFS by plotting

the output of a chaotic attractor directly on the image plane with non-linear functions, log-density display, and structural coloring. Based on the fractal flame, Electric Sheep [14], is widely used for automatically generates distributed screen-saver using genetic algorithm. More recently, fractal method is also applied in jewelry design. Somlak W. etc. [21][22] proposed to use an aesthetic-driven evolutionary approach to create two dimensional art forms for user-centered jewelry design.
Fractals have various applications especially in art and aesthetic field, however, research on applying fractals to three dimensional problems, is still rare. Fractal Cosmos [5] have tried some 3-D fractal generation and use them in virtual scene.

2.2 Evolutionary Design

Evolution is Nature's design process. Artificial evolutionary design applying natural evolution mechanism can create diverse interesting art works effectively. In evolutionary art system, evolution acts as a form generator and can provide designer with much more design alternatives. The evolutionary process continuously generates the new art forms based on the individuals' fitness from the previous generations. There have been several researches [4][19] focusing on design applications, such as art and aesthetic forms generation.

2.3 Iterated Function System

Iterated Function System was introduced by [2] as a unified way of generating a broad class of fractals based on probability measures associated with functional equations. IFS is an effective method for modeling and generating self-similar fractals. The fractal is made up of the union of several copies of itself with each copy being transformed by a function. In general, IFS fractals can be of any number of dimensions, but are commonly computed and drawn in 2D. An IFS is a set of affine transformations, which are constituted of any combination of scaling, rotation, shearing and translation of point sets.
An IFS consists of a complete metric space (χ, d) together with a set of contraction mappings ω_n: X→X with respective contraction factors S_n, $|S_n|\leq 1$ for n=1,2,3,…,*N*, where n is the index for each affine map. Among the finite number of contraction affine maps, one affine transformation of a point set in the Euclidean three dimensional space is defined as a map ω, written in matrix form:

$$\omega_n(x,y,z)=\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix}+\begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}=A_n X + D_n \tag{2.1}$$

where the real numbers a_{ij} control scaling, rotation and shearing, and d_i control translation.
IFS and IFS genotypes are not widely used in evolutionary art and design, there are a few researchers [9][21] investigated on this kind of evolutionary art systems.

2.4 Fitness Evaluation

Fitness evaluation is the process of evaluating whether a generated fractal satisfies certain fitness criterion. Physicist J.C. Sprott demonstrated a relationship between aesthetic judgments of fractal and their fractal dimensions [18]. Aks and Sprott [1] investigated the effect of Lyaponov exponent (quantifying the dynamics that produce fractal patterns) on visual appeal. Spehar etc. [17] and Minita etc. [13] prove that the value of aesthetics, the compactness and connectivity of fractals mainly depends on fractal dimensions and the Lyapunov exponent.

3. EVOLUTIONARY 3D FRACTALS MODELING ALGORITHM

3.1 Modified IFS Formulation

A modified IFS formula is defined in this research to better adapt fractal modeling to three dimensional space and possesses direct geometric meaning, which is the foundation of user interaction. Users and designers can intuitively adjust an existing fractal model to the desired shape by tuning the parameters.

3.1.1 Fractal Self-similar Condition

Let $M_d(\mathrm{R}^d)$ denote the set of d×d matrices and let A be a finite family of $\mathrm{A}_n \in M_d(\mathrm{R}^d), n = 1,2,\ldots N$, of expanding matrices (i.e., all the eigenvalues of A_n have moduli > 1). Let $D = \{\vec{d_1}, \vec{d_2}, \ldots, \vec{d_N}\} \subset R^d$ be a set of d dimensional vector with real number component.

We define the affine maps $\omega_n(X) = A_n^{-1}(X + \vec{d_n})$, and call $\{\omega_n(X)\}_{n=1}^{N}$ a self-affine iterated function system. If there is an attractor $T = T(A, D)$ satisfying $T = \cup_{n=1}^{N} \omega_n(T)$, T is called a self-affine set. If all the A_n's are similar matrices (i.e., $A_n = \rho_n R_n$, where $\rho_n > 0$ and R_n is an orthonormal matrix), T is a self-similar set [6]. If the A_n's are expanding similarity matrices, the self-similar set always exists; but it is not the case for the self-affine sets. Nevertheless for the special case that all A_n equal to A, the self-affine set T always exists under the expanding condition. In general, T or its boundary ∂T (if T has non-void interior) are fractal sets.

3.1.2 Modified IFS Formulation

In general, rotation, translation and scaling are the most commonly seen transformation of objects. The modified IFS formulation is derived by using a combination of rotation, translation and scale in three dimensional Euclidean space to build fractals. Rotations about the principle axes are used for transforming an object to a desired posture and position, or for transforming between different coordinate systems. In the following discussion, the angle of rotation is specified with a right hand coordinate system (x forward, y to the right, and z upwards, see Fig. 1). The rotation about the z axis will be referred to as *roll*, rotation about the y axis as *yaw*, and rotation about the x axis as *pitch* [20]. A rotation will be considered positive if it is clockwise when looking down the axis towards the origin. Under the self-similar conditions described above, we define the IFS Eqn. (2.1) as

$$\omega_n = A_n \bullet \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \tag{3.1}$$

$$A_n = \rho \begin{pmatrix} \cos(\alpha_r)\cos(\alpha_y) + \sin(\alpha_r)\sin(\alpha_p)\sin(\alpha_y) & \sin(\alpha_r)\cos(\alpha_p) & -\cos(\alpha_r)\sin(\alpha_y) + \sin(\alpha_r)\sin(\alpha_p)\cos(\alpha_y) \\ -\sin(\alpha_r)\cos(\alpha_y) + \cos(\alpha_r)\sin(\alpha_p)\sin(\alpha_y) & \cos(\alpha_r)\cos(\alpha_p) & \sin(\alpha_r)\sin(\alpha_y) + \cos(\alpha_r)\sin(\alpha_p)\cos(\alpha_y) \\ \cos(\alpha_p)\sin(\alpha_y) & -\sin(\alpha_p) & \cos(\alpha_p)\cos(\alpha_y) \end{pmatrix} \tag{3.2}$$

Where ρ is the amplification factor, which ranges from 0 to 1; α_p is pitch, rotation angle about x axis; α_y represent yaw, rotation angle about y axis; α_r is roll, rotation angle about z axis. α_p, α_y and α_r range from -Π to Π. This formula has a geometric meaning that A_n is a rotation in 3 dimensional space. The order of rotations applied here is to rotate about the y axis first (yaw), then the x axis (pitch), and then the z axis (roll). A_n^{-1} is an expanding orthonormal matrix, and A_n's are similar matrices. According to Eqn. (3.2), the self-similar set always exists.

3.1.3 Interactive Fine-tuning using Modified IFS

While the fractal is built by evolutionary system automatically, users might want to further modify the fractal pattern to obtain a desired shape.

In fact, users can modify the 3D fractal pattern by tuning each parameters in Eqn. (3.2): tuning α_p, α_y and α_r control the rotation about each axis, which will result in the fractals' encircling around the axis; if p is large, the fractal object departs the x axis far as shown in fig. 10(b). Tuning ρ is to influence the scale of the fractals, and d_1, d_2 and d_3 are offsets along the x, y, z axes applied in each iteration, which adjust the fractal compactness along one axis; if d1 is large, the fractal is loose along x axis. Besides, the selection probability of each chromosome can be changed to adjust the effect of one affine map in phenotype mapping. Users can employ any or any combination of these methods to explore a wonderful fractal world. Furthermore, only a few second is required for reproducing the fractal after a parameter is changed.

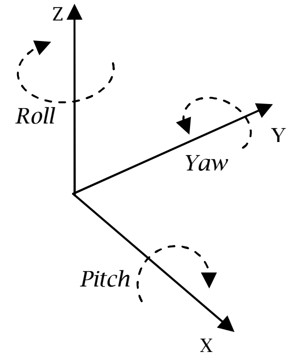

Fig. 1: Coordinate system employed in Eqn. (3.1, 3.2).

3.2 Mapping IFS Genotype to Fractal Phenotype

The genotype is the genetic information or code describing an individual [16]. The phenotype is the individual itself, or the results generated with the development rules and the genotype. Expression is the process of mapping the genotype to phenotype. A gene represents an affine map and there are at least two genes in a chromosome, while a chromosome corresponds to a fractal phenotype. The genotype is mapped to a phenotype by a probability selection process. Here we applied a modified Random Iteration Algorithm (RIA) [3]. The phenotype will be rendered as volume point or reconstructed as mesh surface.

3.2.1 IFS Genotype

As described in Sec. 3.1, an IFS can be encoded as a genetic genotype in the form of an N by 7 matrix (see Fig. 2). M.Barnsley and S. Demko [2][3] and J.C. Sprott [18] etc. adopted two to three affine maps in IFS. We applied a value of 2 or 3 for N. This N by 7 chromosome matrix can be easily processed in evolutionary algorithm.

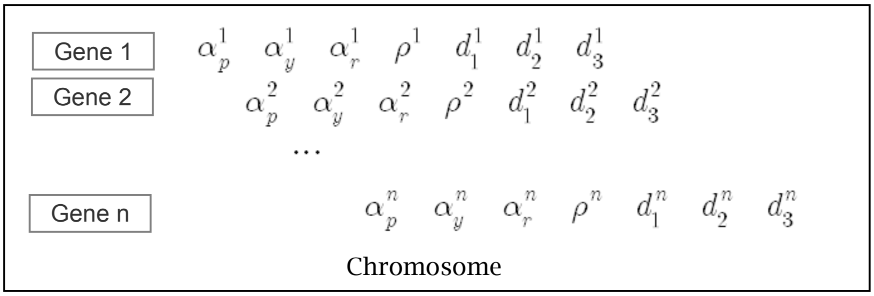

Fig. 2: IFS Genotype.

3.2.2 Revised Random Iteration Algorithm

Random Iteration Algorithm, put forward by Barnsley [2], provides a method for selection probability computation based on convergence rate as shown in Eqn. (3.3).

$$P_i = \frac{\left|\det A_i\right|}{\sum_{i=1}^{N}\left|A_i\right|}, \text{and } \sum_{i=1}^{N} P_i = 1 \text{ and } P_i > 0 \tag{3.3}$$

Using these probabilities can lead to fast convergence in generating a fractal. However, in the experiment we conducted, we found that the selection probability of an affine map may be too small to have effect on the selection. If a two genes fractal is under such condition, there will be only one affine map working which has severe impact on the diversity of the IFS pattern. If we only use random probability [21], the convergence rate can not be assured. We propose to introduce a minimum probability $P_{\min}$ as shown in Eqn. (3.4), where α is influencing factor larger than 1. The influencing factor α is used to adjust the diversity of the fractals. If α is large, the effect of a certain gene might be reduced and the phenotype of generated fractal might be less diverse.

$$P_{\min} = 1/\left(\alpha * N\right) \tag{3.4}$$

If the probability P_i of one affine map is below P_{min}, the corresponding probability will be adjusted to P_{min} and the map with the highest probability is adjusted to $P_{max}+P_i-P_{min}$ so as to reconcile the conflict between the convergence rate and fractal pattern diversity.

3.3 Evolutionary Algorithm

In this research, we propose to conduct fractal evolution by genetic algorithm. Selection is the process first applied, the fitness of phenotypes is determined during this process. Quality of phenotype is reflected by fitness, which is a reference used to decide the likelihood of survival and select the parents for generating successive generation. This step is crucial because it applies the basic concept of survival (i.e. the fittest in evolutionism) and enforces the process to convergence. The second step is the reproduction process in which new genotypes are generated from parents' genotypes. Typical genetic operators used in this process are crossover and mutation, also inferior elimination mechanism is introduced. These operators are used to reproduce new genotypes from existing ones. The crossover operator randomly recombines the chromosomes of two or more selected parents and thus creates a new offspring. The mutation operation generates a random variable for each allele in sequence with a certain occurrence probability, and is used to increase population diversity in the evolution. (Allele is one alternative form in a gene and is found at the same place on a chromosome.) The occurrence probability is to control the stability of the process. In general, if the probability is too high, the average quality of the fractals will not be stable. The evolutionary system will terminate when a predefined maximum number of generations is reached, or when the users are satisfied. The process of proposed EA is shown in Fig. 3.

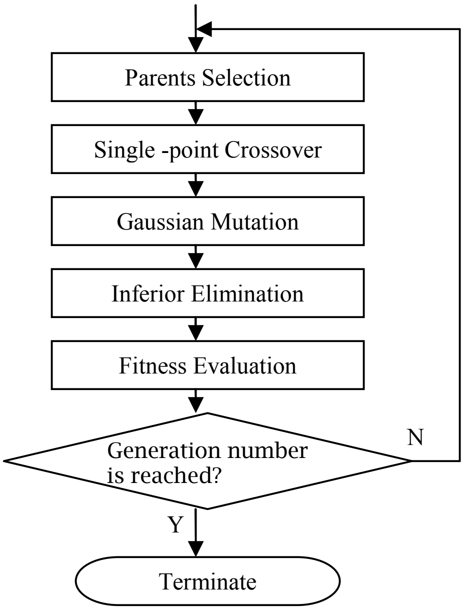

Fig. 3: Flowchart of the EA process.

3.3.1 Single-point Crossover

Crossover occurs with crossover probability P_c. Crossover, also called recombination, is used to vary the given chromosome from one generation to the next by recombining portions of good individuals. Crossover position is randomly generated, however it is banned to locate the position in the intervals between gene (j=7 in this chromosome matrix O), which means the genes will not be exchanged wholly.

$$O_1' = \begin{pmatrix} C_1(1:i,1:j) & C_1(1:i,j:7) \\ C_2(i:N,1:j) & C_2(i:N,j:7) \end{pmatrix}$$
$$O_2' = \begin{pmatrix} C_2(1:i,1:j) & C_2(1:i,j:7) \\ C_1(i:N,1:j) & C_1(i:N,j:7) \end{pmatrix} \tag{3.5}$$

where C_1 and C_2 are the chromosome matrix of parents, O_1' and O_2' are the chromosomes of the offspring, C(1:i, 1:j) is part of matrix C from row 1 to row i and column 1 to column j, and (i, j) is the crossover point.

3.3.2 Arithmetic Gaussian mutation

Mutations create variation within the gene pool. Less favorable mutations can be reduced in frequency by the selection process, while more favorable mutations may accumulate and result in adaptive evolutionary changes. Mutation might bring uncontrollable changes to gene, and hence is strictly controlled by mutation probability P_m. Mutation is conducted to the level of allele.

$$O = O' + M,\ M(i,j) = \begin{cases} k, & if\, prob \ge P_m \\ 0, & else \end{cases} \tag{3.6}$$

where O is the new offspring, O' is the chromosome matrix after crossover operation, k is a Gaussian distribution random number with mean 0, variance 1 and standard deviation 1.

3.3.3 Inferior Elimination

There might be a sudden change in the quality of generated fractals due to randomness in the evolution process. An inferior elimination mechanism is introduced to avoid this. Fractals generated by the previous two steps will be evaluated for their fitness; if fractal fitness is below a certain cut-off score it will be viewed as unqualified and is eliminated.

3.3.4 Fractal Fitness Function

Parents used to generate offspring are selected by their fitness. Fractal Fitness refers to the compactness, connectivity and most importantly the aesthetic appeal. There have been several measures in the field, such as fractal dimension and Lyapunov exponent. Fractal dimension is a statistical quantity that gives an indication of how completely a fractal appears to fill space. In this research we adopted capacity dimension (D_0), correlation dimension (D_2) and Largest Lyapunov exponent (LLE).

D_0 is a way of determining the fractal dimension of a set S in a Euclidean space R^d, it expresses the density of a fractal and how much a fractal fills up the space. D_0 is calculated using box-counting method [10]. D_2 is a dimensional measure of the space occupied by a set of random points. Compared with other fractal dimension, such as D_0, D_2 also measures the contraction rate of the points that land on a fractal [21]. D_2 is calculated through the G. Peter and P. Itamar's correlation integral method [8] and TSTool [12]. D_0 and D_2 generated in our experiments ranges from 0 to 3, and we normalize it to [0, 1] by dividing by 3. Lyapunov exponent is a quantity that characterizes the rate of separation of infinitesimally close trajectories, the largest Lyapunov exponent is the one dominating in the Lyapunov exponent spectrum. We use J. P. Eckmann and D. Ruelle's ergodic theory [7] and LET [1] tool to calculate the LLE. A lower LLE indicates a larger separation between two nearby points; the fractal pattern becomes disconnected and less compact [22].

We employed Gaussian function to build the fitness function. Correlation analysis and linear regression is used to analyze thousands of cases for studying the relationships between D_0, D_2, LLE and the fitness. The normalized fitness function is expressed as:

$$fit(S) = 1.493 \times e^{-\left(\frac{(D_0-0.334)^2}{2\times 0.840^2}+\frac{(D_2-0.587)^2}{2\times 0.698^2}+\frac{(LLE+0.057)^2}{2\times 1.092^2}\right)} - 0.5 \tag{3.7}$$

where *fit(S)* is the fitness value of a 3d fractal S, which ranges in [0, 1].

4. EXPERIMENT RESULTS AND DISCUSSION

We construct a chromosome library of 100 fractals for the initial selection of parents. This library will be refreshed each time when new offspring are generated; newly generated chromosomes will be added to the chromosome library automatically.

While mapping IFS genotype to fractal phenotype, according to Eqn. (3.4), the influencing factor α is set to 4 and then the minimum mapping probability P_{min} is equal to 0.125 for N=2; and P_{min} is 0.083 for N=3; and the number of iterations used to build a visual fractal is 5000.

In the evolution process, the mutation probability P_m is set to 0.5, crossover probability P_c is set to 0.75. The cut-off score used in inferior elimination is set to 0.6. The evolutionary system terminates when it reaches a predefined maximum number of generations, which is set to 15 in the experiment. The experiments are conducted on a PC with Intel Pentium 4, dual CPU 3.40 GHz, 2GB RAM. Average running time of reproducing a qualified fractal is 7.15 sec.

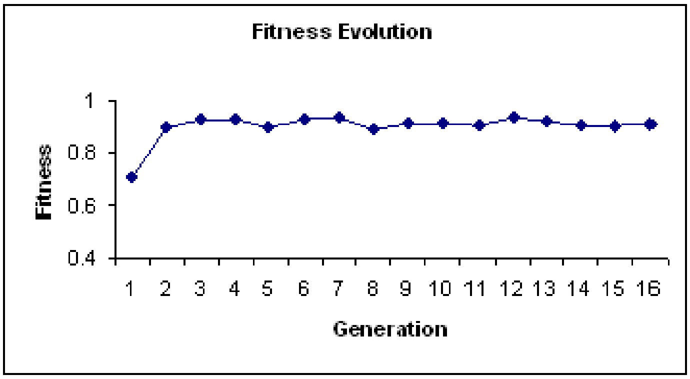

Fig. 4: Fitness value in the evolution process.

As we can see from Fig. 4, the average fitness values of offsprings reproduced are all above 0.8, which means that the fitness of fractals generated by employing evolutionary mechanism is relatively higher and hence the productivity of excellent fractals is higher. We systematically examined all the fractals generated in the evolution process and found that they are nearly all visually interesting.

To explore the effectiveness of different formulas and evolutionary system, we performed a comparison on three cases. In the first case, Eqn. (2.1) is used under contraction condition (i.e. A is random orthonormal matrix with |detA| less than 1) to generate a hundred fractals. In the second case a hundred fractals are built using Eqn. (3.1, 3.2) without applying genetic algorithm. In the third case, fractals are built by Eqn. (3.1, 3.2) in evolutionary method, and there are 15 generations with a population of 10 for each generation. Considering the evolutionary system is a stochastic system, the experiments were conducted 10 times, and all the values are the average value from the ten experiments. The average running time of generating fractals is 332.48 sec. for case 1, 441.42 sec. for case 2, and 1072.35 sec. for case 3.

From Fig. 5, we can observe that the average fitness and excellent rate are increasing from case 1 to case 2 and to case 3, which demonstrates that Eqn. (3.1, 3.2) used with evolutionary system can effectively enhance the productivity of fractals generation.

5. 3D FRACTALS RENDERING AND APPLICATION

The 3D fractal generated from the system is represented by a set of voxels. Such fractal voxels lack of topology and distribute non-uniformly, and the density of some part might be very low or even zero. We use volume point as rendering primitives to display the generated fractal. This approach is very fast and can be used to provide a coarse image of the fractals. To obtain a better image quality, we use marching cubes to reconstruct its mesh surface and topology.

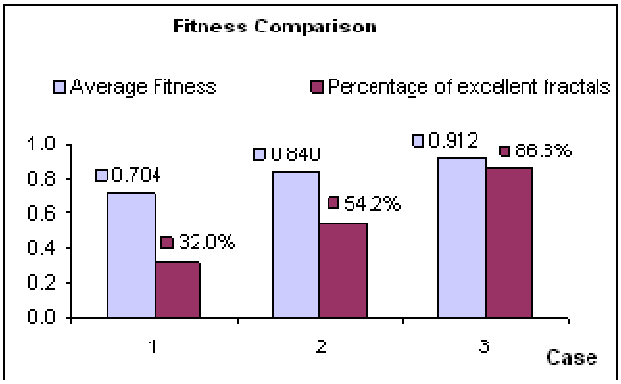

Fig. 5: Fitness Comparison between three cases.
*: Excellent fractals means fractals with fitness value larger than 0.85.

5.1 Volume Point Rendering and Application in Decoration

Volume point is used as rendering primitives and the rendering is obtained using OpenGL. For the coloring of 3-D fractals, we employ a gradient diagonal matrix α (x,y,z), of which the diagonal elements are the ratios of x, y, z to corresponding coordinate maximum range. The r, g, b channel are associated with x, y, z coordinate respectively. It can be written as $C(x,y,z) = (1,1,1) \times \alpha(x,y,z)$, where *C* is the color vector (r, g, b) at the position. If a dual-tone linear gradient scheme is applied, the computed color *C* is $C(x,y,z) = C_1 - C_2 \times \alpha(x,y,z)$, where C_1 and C_2 are the base colors.
The examples shown in Fig. 6 are produced in the evolutionary system with 5000 points and rendered with linear gradient coloring and dual-tone gradient coloring. Their corresponding gene code is in Table 1. The Christmas tree shown in Fig. 7 consists of six kinds of fractal patterns. There are 20000 voxels for rendering the tree and 5000 voxels for each decoration object on the tree. The tree is olive green, and the stars, ribbon and conch are colored by using the dual-tone gradient coloring scheme based on red, orange and orchid respectively; others are colored by using linear gradient coloring scheme.

Index	α_p	α_y	α_r	ρ	d_1	d_2	d_3
06328	2.2237	1.7187	0.3243	0.7794	0.2761	1.0610	1.7711
	1.2230	-0.1873	0.0150	0.8465	1.3295	1.1357	-0.0139
	-2.1695	0.9688	0.0180	0.2624	1.2817	1.0508	-1.1341
06425	2.9523	2.9343	2.9154	0.2095	-0.5115	0.2966	-0.0109
	0.1554	-2.4184	-0.1682	0.5514	0.0268	-0.8664	1.1950
	-0.3223	3.0522	2.3830	0.8780	-0.4476	-1.2096	-0.9740
06447	-1.1886	1.2925	2.6916	0.0397	-1.0043	-1.9483	-1.9868
	0.0203	0.1246	1.2773	0.9802	-0.5522	-0.4016	0.9956
	-0.7206	-1.9475	0.4351	0.0965	1.1693	0.0640	0.2261
06582	-2.5900	-0.2154	1.9586	0.2793	-0.9179	-1.2826	0.5331
	0.7182	-0.9256	3.0556	0.9677	0.8829	-0.3807	0.0029

Tab. 1: The gene code for fractal examples in Fig 5.

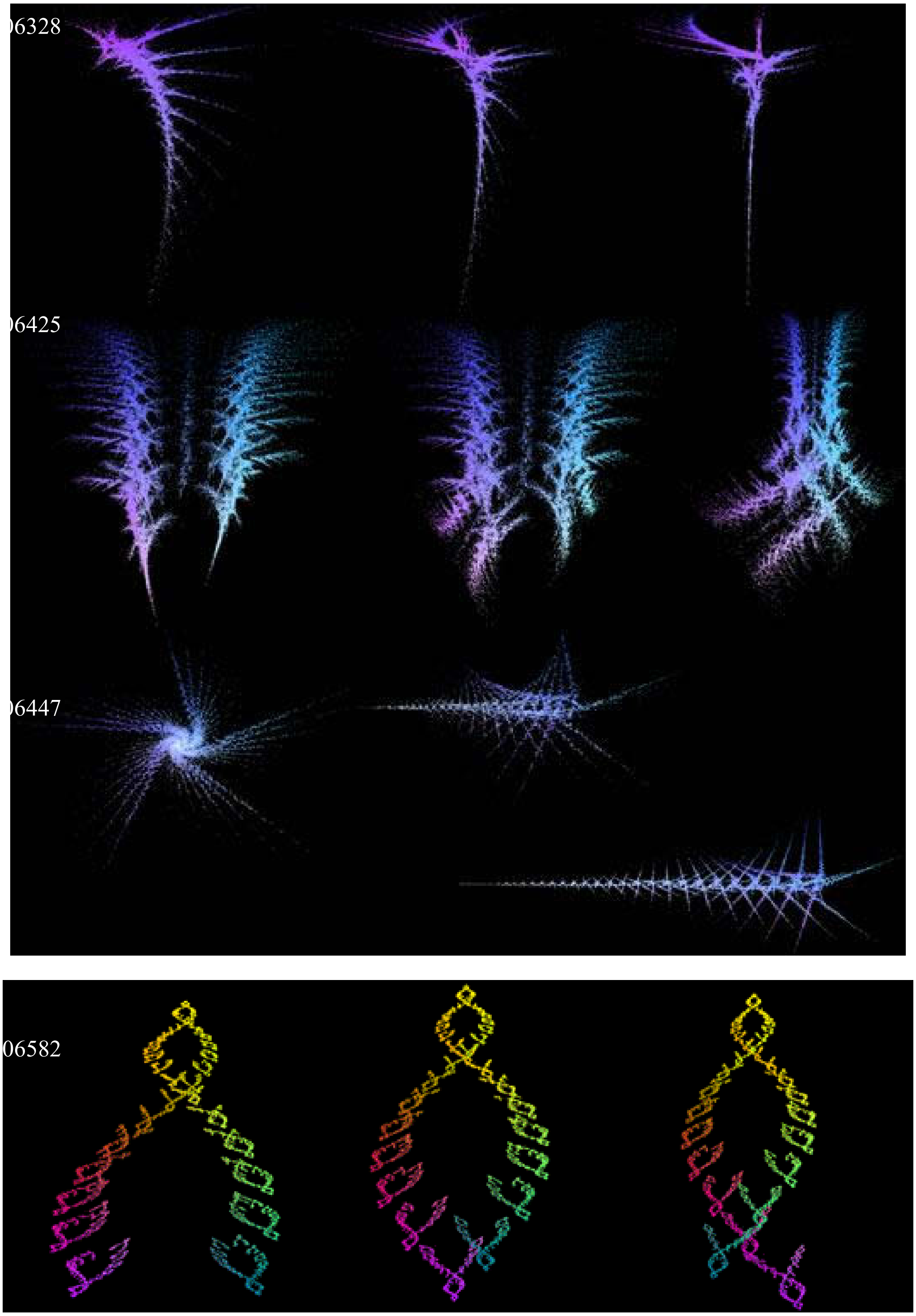

Fig. 6: Examples of 3D fractals generated in the evolutionary system using linear gradient coloring.

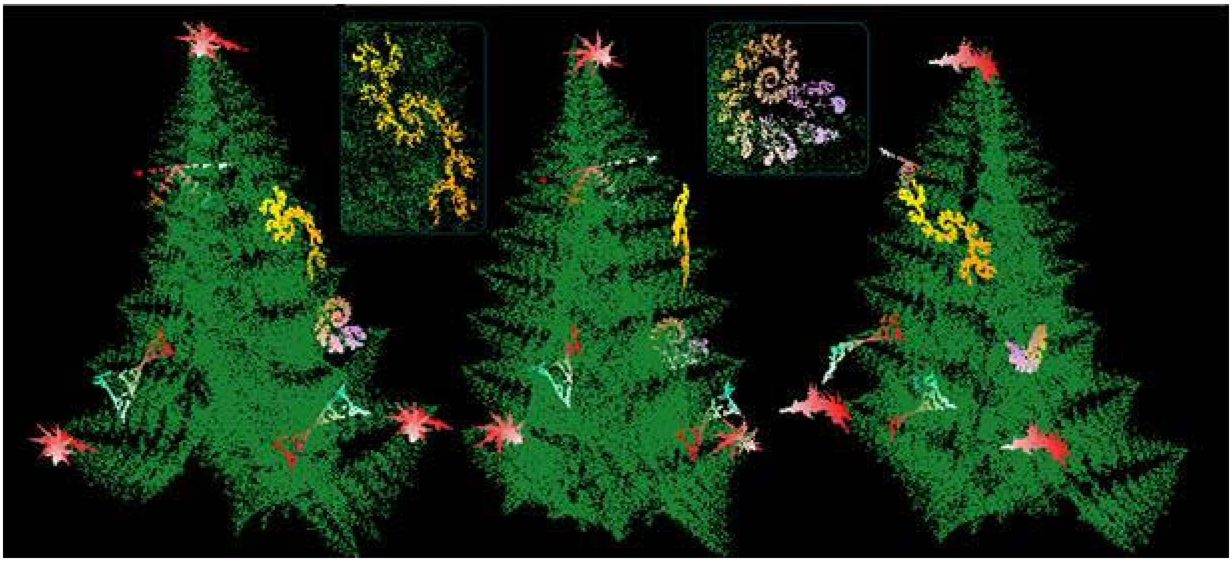

Fig. 7: A Christmas tree with decoration generated using volume point rendering.

5.2 Application in Jewelry Design

The jewelry pattern as shown in Fig. 8 and Fig. 9 is obtained by reconstructing mesh surface from voxels using the Marching Cube method [11]. Here the cubic grids used in the example are 85×199×144 for J0698 and 200×156×126 for J0619. After converted to mesh surface by Marching Cubes, the fractal is further refined and rendered in Maya to obtain better rendering effect for the fractal surface.

Fig. 8: Different views of 3D fractal jewelry J0698, rotate clockwisely from left to right.

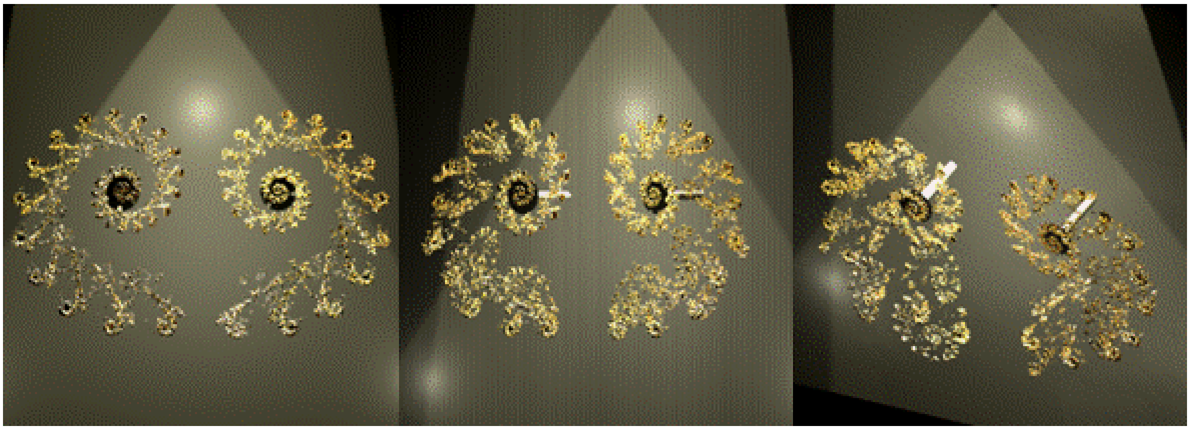

Fig. 9: Different views of 3D fractal jewelry J0619, rotate clockwisely from left to right.

5.3 Transforming Property

According to Eqn. (3.1, 3.2), the shape of the fractals transforms continuously while the variables are changing. Fig. 10(a) shows the original appearance of a fractal. Fig. 10(b, c, d) shows the result of changing the pitch, yaw, and roll by adding a value of $\pm\Pi/60$ in each step. Fig. 10(e) illustrates the result of changing d_1 by a value of 0.25. Fig. 10(f) shows the effects of reducing ρ by 3% in each step. There are 10000 voxels used for the rendering.

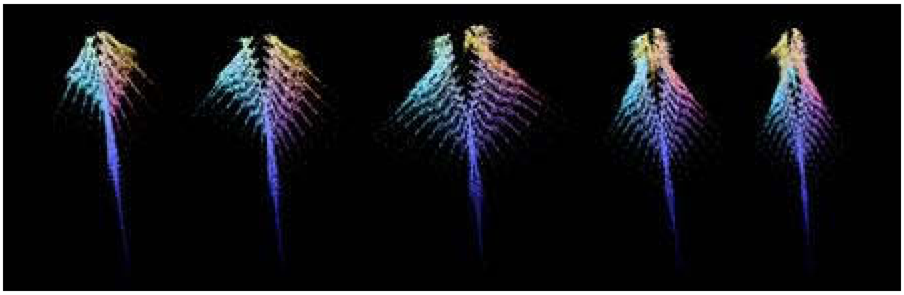

Fig. 10(a) Multi-views of a 3D fractal, rotated counterclockwise from left to right.

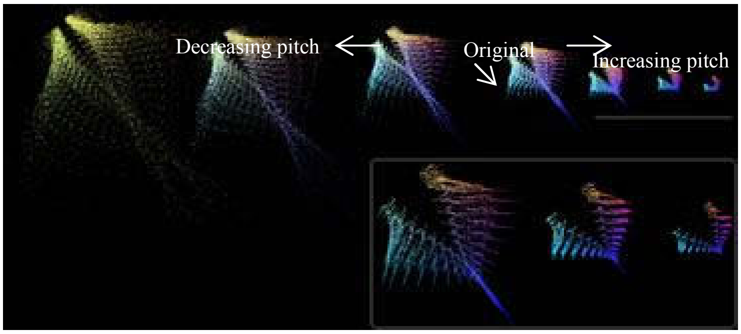

Fig. 10(b): Effect of changing pitch, bottom-right corner is enlarged view of the most right three fractals.

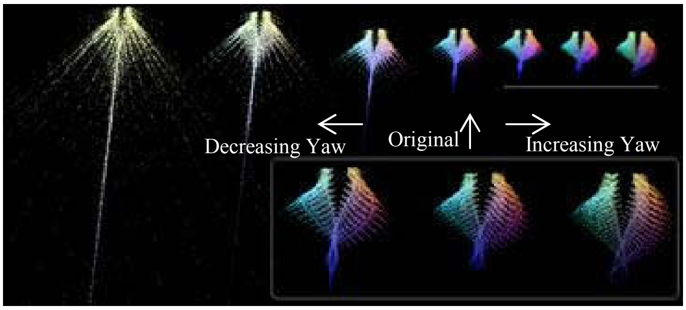

Fig. 10(c): Effect of changing yaw, bottom-right corner is enlarged view of the most right three fractals.

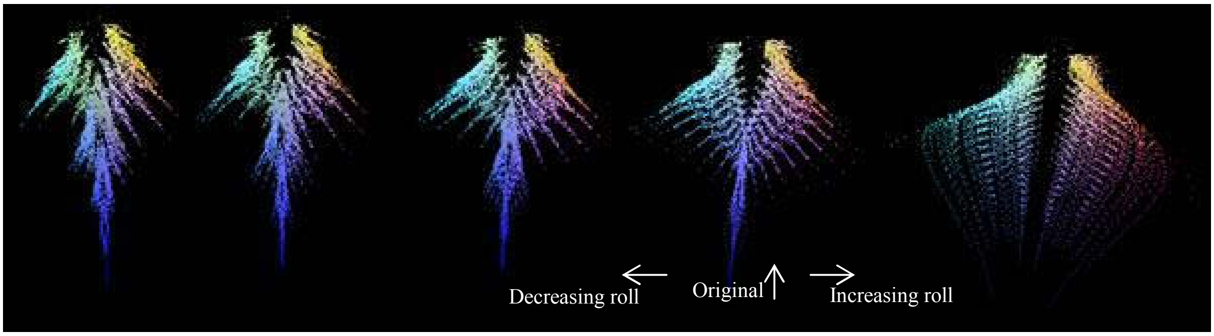

Fig. 10(d): Effect of changing roll.

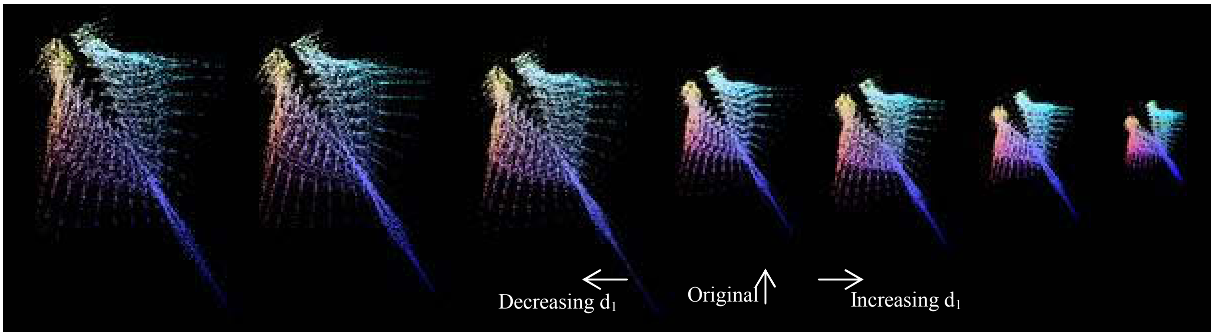

Fig. 10(e): Effect of changing d_1.

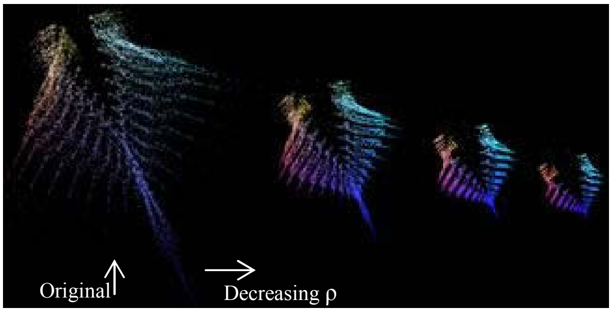

Fig. 10(f): Effect of changing ρ.

6. CONCLUSION

An extended IFS formula under fractal self-similar condition is proposed in this paper. 3-D fractal pattern built using this formula is controllable and can be interactively altered by users and designers through tuning corresponding parameters. Evolutionary system is applied to produce 3-D fractal patterns based on this modified IFS formula, which offers a diversity of design. Genetic operation includes crossover, arithmetic Gaussian mutation and inferior elimination are employed to provide variation of the chromosomes in the evolutionary process. This system aims to function as a 3-D fractal builder and automatically generate large number of fractal patterns with acceptable aesthetic value. Fitness of fractal art pattern is measured and formulized by characteristic parameters in fractal theory.

7. ACKNOWLEDGEMENT

This work is affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies, and is partially supported by a grant from the Research Grants Council of the Hong Kong Special Administration Region (Project No. 412106) and a Direct Grant from the Chinese University of Hong Kong (Project No. 2050413).

8. REFERENCE

[1] Aks, D.; Sprott, J. C., Quantifying aesthetic preference for chaotic patterns, Empirical Studies of the Arts, 14, 1996, 1–16.
[2] Barnsley, M.; Demko, S.: Iterated Function Systems and the Global Construction of Fractals, Proc. Royal Soc. of London, A 399(1817), 1985, 243-275.
[3] Barnsley, M.: Fractals Everywhere, 2nd ed., Academic Press, San Diego, USA, 1993.
[4] Bentley, P.: Evolutionary design by computers, Morgan Kaufmann Publishers, San Francisco, U.S.A, 1999.
[5] Berkowitz, J.: Fractal Cosmos: The Art of Mathematical Design, Amber Lotus, Portland, 1998.
[6] He, X. G.; Lau, K. S.: On a generalized dimension of self-affine fractals, Math. Nachr, 281(8), 2008, 1142-1158.
[7] Eckmann, J. P.; Ruelle, D.: Ergodic Theory of Chaos and Strange Attractors, Rev. Mod. Phys., 57(3),1985, 617-656.
[8] Grassberger, P.; Procaccia, I.: Measuring the Strangeness of Strange Attractors, Physica D: Nonlinear Phenomena, 9(1-2), 1983, 189-208.
[9] Joye, Y.: Evolutionary and cognitive motivations for fractal art in art and design education, The International Journal of Art & Design Education, 24(2), 2005, 175–185.
[10] Kruger, A.: Implementation of a fast box-counting algorithm, Computer Physics Communications, 98(1-2), 1996, 224-234.
[11] Lorensen, W. E.; Cline, H. E.: Marching Cubes: A high-resolution 3D surface construction algorithm, Computer Graphics, 21(3), 1987, 163-169.
[12] Merkwirth, C.; Parlitz, U.; Wedekind, I.; Lauterborn, W.: TSTOOL MatLAB toolbox, http://www.physik3.gwdg.de/tstool/, 2002.
[13] Mitina, O. V.; Abraham, F. D.: The use of fractals for the study of the psychology of perception: psychophysics and personality factors, a brief report, International Journal of Modern Physics C, 4(8), 2003, 1047–1060.
[14] Scott D.: The electric sheep, ACM SIGEVOlution, 1(2), 2006, 10-16.
[15] Scott D.: The Fractal Flame Algorithm, 2004, http://flam3.com/flame.pdf.
[16] Sims, K.: Artificial evolution for computer graphics, Computer Graphics, 25(4), 1991, 319-328.
[17] Spehar, B.; Clifford, C.W.G.; Newell, B.R.; Taylor, R.P.: Universal aesthetic of fractals, Computer & Graphics, 27(5), 2003, 813–820.
[18] Sprott, J. C.: Automatic Generation of Iterated Function Systems, Computer & Graphics, 18(3), 1994, 417-425.
[19] Todd, S.; Latham, W.: Evolutionary Art and Computers, Academic Press, London, 1992.
[20] Vince, J.: Mathematics for Computer Grahics, 2nd ed., Springer-Verlag, 2006.
[21] Wannarumon, S.; Bohez, E. L. J.: A new aesthetic evolutionary approach for jewelry design, Computer-Aided Design & Applications, 3(4), 2006, 385–394.
[22] Wannarumon, S.; Bohez E. L. J.; Annanon, K.: Aesthetic evolutionary algorithm for fractal-based user-centered jewelry design, AI EDAM, 22(1), 2008, 19-39.