An Application of L-system and IFS in 3D Fractal Simulation

DENG FANG , XI LI-FNEG
College of Computer and Information
Zhejing Wanli University
Ningbo
CHINA

Abstract: - L-system and IFS are used to the plant's simulation. In this paper the theories of L-system and IFS were concisely described. The excellence and short of constructing Fractal image with L-system or IFS were discussed. So a method was put forward in this paper. At last the three-dimension tree model ceatrs successfully based on L-system and IFS.

Key-Words: - L-system, IFS, Fractal Simulation, OpenGL

1 Introduction

The plants are often seen in the nature. And to simulate those plants have been the hot topic in Computer Graphics. Computer plant simulate technique has an application foreground in entironment simulation, landscape design, Visual Reality , video and game making ,and so on. Because the plants have the characteristic of self-similarity, irregular and random; it is so difficulty to simulate by the traditional methods. Fractal geometry is a non-geometry rule for the scientific study. Fractal geometry can be used to simulate those complex objects.

There are many methods of creating fractal image. L-system[1] and IFS[2](Iterated Function System) are often used to simulate the plants. L-sytem must determine the creating ruler. It is simple but not flexible. The randomness of IFS is the most feature ,but the same time the random number isn't certain easily. There have been many people[3][4][5][6] to research the simulation of natural plant using L-system or IFS only. Because each method has its own limitations, it is hard to realize the natural form of plants and realistic effect.

The plant's simulation in this paper combines L-system and IFS. In order to get to the aim that the plants would be more realistic.

2 Principles

2.1 Fracal Theory

In 1975, mathematician Benoit Mandelbort publish a book——《Fratal：Form，Chance and Dimension,which marked the birth of the theory of Fractal.

Fractal is a sort of geometry that has a complex and sophisticated structure at arbitrary scale. Fractal geometry has the characteristic of self-similarity, which each graphic can be seen as part of the overall reduction of graphics. For example, The Koch snowflake is a typical fractal graphic in Fig.1.

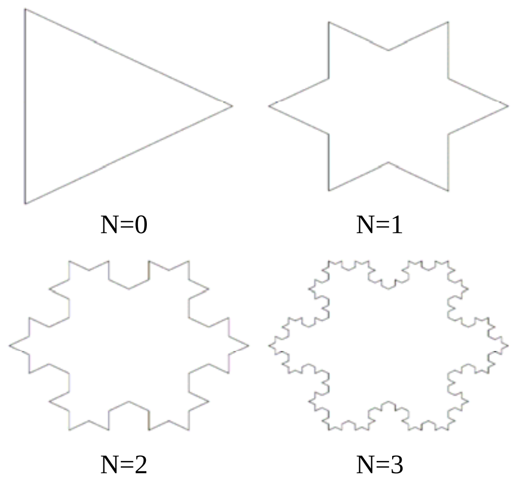


Fig.1 Koch snowflake

The birth of fractal theory form a modern geometry of a new branch – Fractal Geometry. The Basic idea of fractal geometry is objective things' self-similar structure.

The difference between fractal geometry and euclidean geometry is that fractal geometry is irregular, illimitable, self-similarity and simple ruler.

2.2 Iterated Function System (IFS)

IFS is an important part of fractal theory. Its theory and method are the theory base of fractal natural scene simulation.

IFS treat affine transformation as framework. produce by iteration according to the self-simulation of whole and parts.

A n-dimension's IFS is two parts. One is limited sets $W = \{W_1, W_2, \cdots W_N\}$ that is linear mapping .The other is probability sets $P = \{P_1, P_2, \cdots P_N\}$. Each P_i is connect with a W_i, and $\sum p_i = 1$.Generally IFS can be defined as following:

$$\{W_j, P_j : j = 1, 2, \cdots N\}$$

IFS's working process is as following:

Get any point Z_0 ,then choose the transformation W_i according to P_i ,and do transform $Z_1 = W_i(Z_0)$. Do it as above, do transform $Z_2 = W_i(Z_1)$. Then there is a infinite point sets. The method is to find the right mapping sets, probability sets and original point .then those created point sets could simulate certain scene. If there exist a enough large number N, the number could let the iterated result A= $Z_k(k \geq n)$ not to be changed. This A, existing and unique, is the attractor of IFS. This A can be scribed as

$$A = \bigcup_{j=1}^{N} w_j(A) = w_1(A) \cup w_2(A) \cdots \cup w_N(A)$$

so the attractor's structure is ontrolled by the contractive mapping. The robability sets $\{p_1, p_2, \cdots, p_N\}$ determinates measure of the attractor. It controls the probability of parts of A, and it is a important information that plot attractor.

There are two iterated algorithms to produce IFS attractor. They are determinate iterated algorithm and random iterated algorithm.

In this paper,I use random iterated algorithm.

Assume a IFS is composed with contractive mapping $X = \{X_i : w_{i1}, w_{i2}, \cdots w_{iN}, i = 1, 2, \cdots M\}$ and a probalilitysets $P = \{p_{i1}, p_{i2}, \cdots p_{iN}, i = 1, 2, \cdots M\}$. Here

$$\sum_{j=1}^{N} p_{ij} = 1, 且 p_{ij} \geq 0, j = 1, 2 \cdots N, i = 1, 2 \cdots M$$

$$w_{ij}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_{ij} & b_{ij} \\ c_{ij} & d_{ij} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_{ij} \\ f_{ij} \end{pmatrix},$$

$$j = 1, 2, \cdots N, i = 1, 2, \cdots M$$

$\forall i, \max(j) < N$,as all parameters that j is from $\max(j) + 1$ to N are zero.

Select any point $x_{i0} \in X(i = 1, 2, \cdots M)$ is original point, then iterate random algorithm select a point form this sets as $X_{in}, n = 1, 2 \cdots$,then

$$x_{in} \in \{w_{i1}(x_{i(n-1)}), w_{i2}(x_{i(n-1)}), \cdots w_{iN}(x_{i(n-1)})\}$$

At least, get the array $\{x_{in}\} \subset X_i$ .This array is converged in the attractor of IFS.

The mathematics expression of affine transformation can be expressed as :

$$\omega : \begin{pmatrix} x' = ax + by + e \\ y' = cx + dy + f \end{pmatrix}$$

Where ω is affine transformation. x, y is the position of original image. x', y' is the position of new image. There are six parameters[7] a, b, c, d . and $, e, f$ .We call them as affine transformation parameters.

The action of a, b, c, d are to accomplish linear transform and rotation transformation . The action of e and f are to accomplish to translational tr-ansformation.

A complex image could need several different affine transformation to carry out. Affine transformation controlls the image's structure and shape. Beacuse the form of affine transformation is same, the image's shape lies on the affine transformation system. So the probability P is very important.These paramete are called IFS code.

But not all affine transformation can be implemented by IFS. If it can do ,this affine transformation must be compression.In the other words, the distance of each element in the graphic would be reduce.We call this affine transformati-on as compression transformation.We know those compressed images are similar before they are compressed. And they come from old images.So when we simulate the plants, there would be some images by affine transformation , which is based on the external environment and the internal factors of plants. Then we collage those images[8]. So we know that if the collage is implemented, the IFS codes of affine transformation must be confirmed.

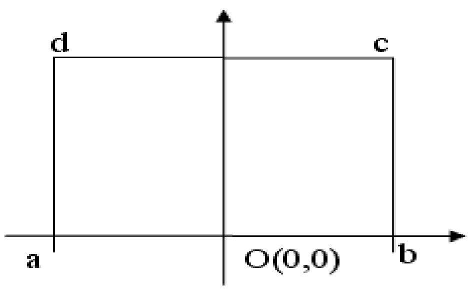
If to confirm the IFS code , it deponds on the priciple of plant's simulation. When the process of simulation is bottom-to-up, which is method deponding on experience, we use "interactive method " to confirm the IFS code. While those IFS codes are confirmed ,the affine transformation is being uninterruptedly . At the same time there occurs some sub-image.Then we collage those sub-image until the plant is simulated completely.

If the process of simulation is up-to- bottom, which's meaing is that each simulating step is unknown ,we need computer.We call it as " computer method". So we must establish a linear equations. Solving the equations , there can get the next simulating image.

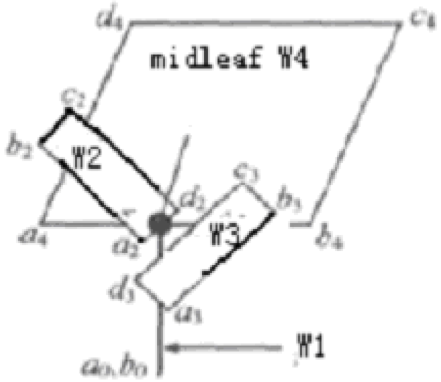
Table 1 is IFS code used to creat ferny leaf.And Fig. 2 is generation process of the ferny leaf.

Table 1 IFS code used to creat ferny leaf

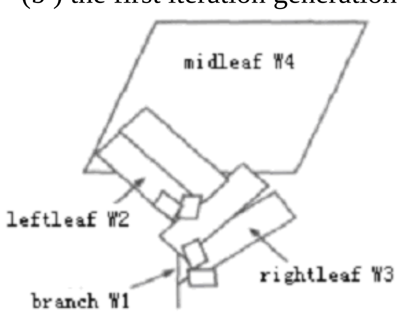
ω_i	a	b	c	d	e	f	p_i
ω_1	0	0	0	0.16	0	0	0.01
ω_2	0.85	0.04	-0.04	0.85	0	1.6	0.85
ω_3	0.2	-0.26	0.23	0.22	0	1.6	0.07
ω_4	-0.15	0.28	0.26	0.24	0	0.44	0.07



(a) original Matrix



(b) the first iteration generation



(c) the second iteration generation

(d) No. N-th iteration generation

Fig. 2 generation process of he ferny leaf

2.3 L-system

L-system is a re-write-system based on symbol .It was gived in 1968 by Aristid Lindenmayer. In 1984 and 1986, A.R.Smith and P.Prusinkiewicz introduced L-system to Graphics.There has a effectly method in natural simulation. It makes sure a complex object by permuting those parts of original object using a rewrite rule[9][10].

Supposed there is a string consist of two words. And the rewrite rule is:

b->a

a->ab

then the process of system evolve as follows:

b

a

ab

aba

abaab

abaababa

the increase of word amount in the string exactly is Fibonacci Number 1,1,2,3,5,8,13……

the sequence gratifies the recursive formula:

$$F_n = F_{n-1} + F_{n-2}$$

We use it that turtle crawls to descript the process of L-system fractal image.

Assuem (x, y) is the coordinate of turtle position, α is the angle of turtle's head, (x, y, α) is the original status, d is the forward increment, ∂ is angle increment ,the turtle moves according to those commands:

F: move forward , d is step-size, original state is (x, y, α) .

$$x_1 = x + d\cos\alpha$$

$$y_1 = y + d\sin\alpha$$

draw a line from (x, y) to (x_1, y_1)

E: move forward ,step-size is d , not to draw a line.
+: turn left ∂ angle,then turtle's next position is $(x, y, \alpha+\partial)$.
-:turn right ∂ angle,then turtle's next position is $(x, y, \alpha-\partial)$.

Repeat above ,there will be a fractal image.

Now we see a example that is a string.

$FFF+F+F-F+F-F+F-FFF$

$\delta=90^0$

The turtle graphic is Fig.3.

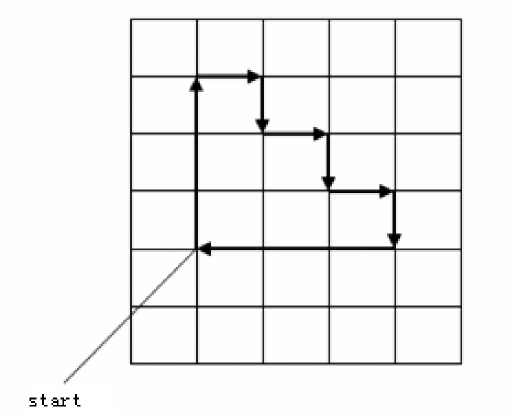


Fig.3 a turtle graphic of string

2.3.1 Three Dimension L-system

The reality world exists in the three-dimensional space. So L-system must be extended to three-dimensional.

In order to explain the 3D turtle[11], the key issue is using three vector H,L and U to express the turtle's current position. Here H represents the front, L represents the left, and U represents the top. The three vectors have unit length and direction orthogonal. Then the equation $H \times L = U$, the turtle's rotation can be express as

$$[H' \;\; , L' \;\; ,U' \;\;] = [H, L,U] R$$

Here

$$R_H(\alpha)=\begin{bmatrix}\cos\alpha & \sin\alpha & 0\\ -\sin\alpha & \cos\alpha & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$R_L(\alpha)=\begin{bmatrix}\cos\alpha & 0 & -\sin\alpha\\ 0 & 1 & 0\\ \sin\alpha & 0 & \cos\alpha\end{bmatrix}$$

$$R_U(\alpha)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\alpha & -\sin\alpha\\ 0 & \sin\alpha & \cos\alpha\end{bmatrix}$$

if we only use those symbol ＋、－ to express the turtle direction in the 3D ,there aren't enough. So there are some symbol that is uesd to change the direction of turtle.

$+(\delta)$: turn left δ angle around U

$-(\delta)$: turn right δ angle around U

$\&(\delta)$:turn down δ angle around L

$\wedge(\delta)$:turn up δ angle around L

$\backslash(\delta)$:turn left δ angle around H

$/(\delta)$:turn right δ angle around H

$|(\delta)$ ::turn back 180 angle around U

Fig.4 is the 3D direct and rotate map.

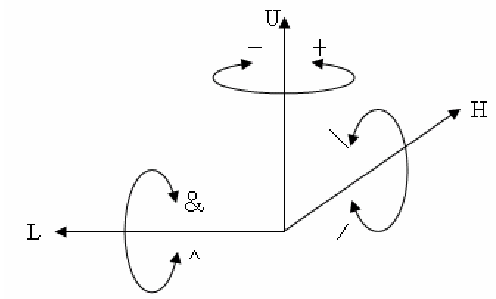


Fig.4 turtle's direct and rotaing map in 3D

2.3.2 L-system Simulates the Plant Structure

How to simulate a given tree by L-system ? L-system firstly defines a element set V, every elemnet in V is a node which has a function. Then there produces a orginal element W and growth rule P based on the tree's growth process. Those element (V,W, p) composes a L-system.

Due to the plant's furcating structure. There are some differentiation between plants structure and fractal image[12][13]. It needs to add the process that pushes the current information of furcated and pops the information.

We use axial-lead[14] tree to describe the process. The axial-lead tree includes parts as follows:

(1)root. It is a especial node. You can get to anyone end point form root.

(2) inner node. In path the node at least connects a subsequence ramification.

(3) terminal node. the end of final ramification.

(4) Straight branch and side branch.Straight branch is trunk. Side branch is ramification.

In order to describe the ramificate structure, I introduce two new words. They are :

[: push the present state information to a stack. The information includes the turtle's position and direction.

]:pop a state from the stack and treat it as the turtle's current state .

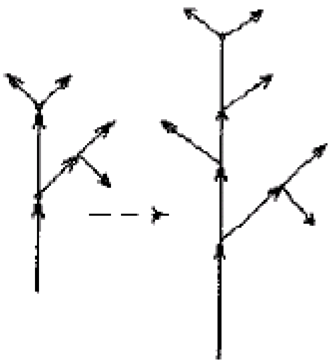


Fig.5axial-lead tree

In Fig.6 we see a sapling's growth process. The function of A is to be the Fig.2(b),and the function of B is to be double.According to the rule A must to be like this in Fig.2(d) some time later.The process can be showed by L-system code as following:

$$W : A;$$

$$P1: A->B[A][A]$$

$$P2: B->BB$$

A and B compose the set V; the orginal state W is A; P1 and P2 are the two growth rule.

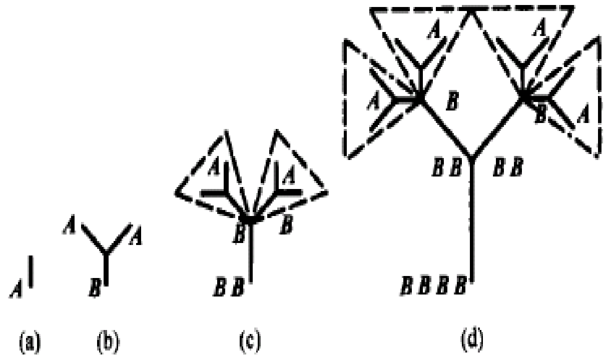


Fig.6 sapling's growth process

3 Generation Algorithm of Fractal Plant

3.1 Ideal of Algorithm

L-system has the advantage in topography structure of plant, for example, it can descript the ramdom growth pattern and struction morpholgy of leaf, flowers, fruits easily and freely. The reason is L-system's parallel mechinaism matchs with the plant's praallel growth pattern.But it has a short in texture flu.L-system just considers the self-similarity of whole plant,not to attente the parts of plant.which also have the self-similarity.

IFS's advantage is that the graphic is colored map,texture strcture is very good ,and the program implement easily. But it is short at controlling the plant's topography structure.The difference is very little in state of plants,the plants are very similar.

So if we only use L-system or IFS to simulate the plant, it is difficulty to simulate the complex plant's features of texturn, varialbility and randomness. Now I simulate a three-dimension tree by fusion of L-system and IFS.

The work of generating tree centers on trunk, branch and leaf.

First, we see the trunk. IFS can simulate the trunk factually. But with random iterated algorithm the treetop position isn't certainty easily. So I use L-system to generate the trunk and branch.

Second, it is that how to simulate leaf. It is easy by IFS algorithm now. But we need a lot of leaves not one leaf.So those leaves should have some feature as following[15][16][17][18] :

IFS is compose with same quantity of functions. Every leaf has belonged to itself IFS code. Of course, those leaves created by the IFS code should have same self-similarity. Those leaves should match the trunk and branch. For example, the leaf's angle of inclination should be relate with the tend of branch. The leaves on a branch should not spead to one same direction.

In order to solve those questions, the method is bringed forward that is L-system controlling IFS code's parameters. At last, when the tree's structure has been created, it must to do texture for the tree.

3.2 IFS Parameter Set

Now we give two concepts:

Definition 1: we name those parameter as inner parameters. Which controls these six parameters in random IFS .

Definition 2: we name those parameter as extern parameters. Which controls these iterated function in random IFS .

Before introducing inner parameters ,it must do parameters processing to IFS codes of random IFS. The process is as following:

At first, IFS code expresses as those parameters r, q, θ, f, e and f .The e, f 's meaning is changeless, the relation of the other's parameters and primary IFS code is :

$$\begin{cases} a = r\times\cos(\theta) \\ b = -q\times\sin(\theta) \\ c = r\times\sin(\theta) \\ d = q\times\cos(\theta) \end{cases}$$

then we can computer a set of numbers as original number.

Then ,to most leaves, analyse the graphic at the metric space $[0,1]\times[0,1]$,get every parameter's range. And solving the $k_{ij}(i=1,2,...6; j=1,2,...n)$ as the formula k_{ij} = (threshold –threshold lower limit)/(threshold upper limit –threshold lower limit). Beacuse every parameter has its range, k_{ij} 's value is $0\le k_{ij}\le 1$.

Thirdthly, consider the k_{ij} 's distributing relativing to $\frac{1}{2}$ .Under the formula premise $e_j+\frac{b_j}{2}=e_j'+\frac{b_j'}{2}, f_j+\frac{d_j}{2}=f_j'+\frac{d_j'}{2}$, variance is close to zero by ajusting the original parameters and those range.And confirm the range of the inner paramters $k_{ij}(i=1,2,...6; j=1,2,...n)$.

At last, match the IFS's inner parameters $k_{ij}(i=1,2,...6; j=1,2,...n)$ for all IFS code based on the range.

IFS code's form structured, evrey IFS code is relating and restrict by importing those inner parameters $k_{ij}(i=1,2,...6; j=1,2,...n)$ .The same time,because of the difference of parameters's range, to every two IFS parameters, the parameter's change is differce though importing same k . So all parameters can be evaluated unifiedly while controlling those parameters. It cann't affect the generation effect, the same time , it can impove the algorithm efficiency. $6\times n$ parameters's storage reduces to 1. IFS code's form structured, evrey IFS code is relating and restrict. With the k 's change ,the whole system will change,and it cann't occur aberration to the graph generated.And it could occur some different form's fractal graph.

3.3 Generation Process

The generation process has two parts.

First,when leaf has been simulated via IFS, the importance is how to put it on the branch which has been generated via L-system. In other words, how to join the branch's coordinate and leaf's. The generation theorem of L-system is recursiveness.It would be has a twig of branch when we draw a line using same grammar at each recursiveness. So when a recursiveness finishes ,the coordinate of the treetop pops form stack. The position $(xpos, ypos)$ was transfered to the IFS function in order to draw a leaf.. While we compute the IFS code of leaf, in order to spread the image efficaciously ,the parameter e and f are unequal to zero synchronously . So it would bring a result that the foremost several points could overfill the area of leaf,and the leaf couldn't join with the branch.

Second,the leaves is on the branch, the trend of branch will affect the position of leaf. With the development of branch ,each leaf has different processing mode.So ,while the branch generates,it offers the extern parameters to IFS function.Then the shape of leaf will be adjusted beacuse of the change of branch.

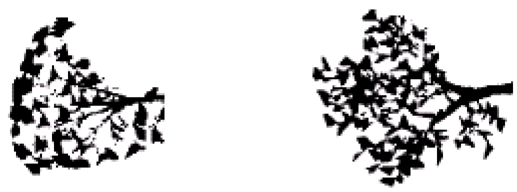
Affine transformation controls the value of extern parameters.the parameter $a_{ij}(i=1,2 and j=1,2)$ is determinted by the direct of L-system and iterated times.as

$$a_{11}=\cos(new\delta), a_{12}=-\sin(new\delta)$$
$$a_{21}=\sin(new\delta), a_{22}=\cos(new\delta)$$

$new\delta$ is a new angle that string rewrites in L-system . it is the incline angle between ending of branch and y-axis negative direction in fractal trees. As this method, those parameters were confirmed, and then transfer to IFS function.

When IFS function gets those extern parameters, it will do rotation, translational or liner transform. And the effect is very perfectly that the branches by L-system unite the leaves by L-system controlling IFS code's parameters.

In Fig.7 it shows the difference that Branches created before and after L-system controlling IFS code's parameters. The leaves in Fig.7 (a) haven't the controlling of parameters, so they seem to be stiff. But in Fig.7 (b) the leaf's extern parameters have been controlled by L-system. So these leaves look seem to be reality.



(a) creating leaf without parameters (b)creating leaf with parameters

Fig.7 Branches with Leaf

3.4 L-system Improvement

In order to pursue the natural effect[19], it requires the parts of plant, for example, trunk, branch and leaves, should change randomly on thickness, length, color and direction. L-system should adjust the transfer parameters of all parts of plant, which IFS generate according to actual needs while the process of generation.

Generation algorithm of traditional L-system is that confirming an original string, rewriting according to growth rule, and to upping limit of iterative. At lastly explain the long string and draw the graph. This method of finally one-time graphics isn't enough flexble.It is difficult to implement the random controll.

Hence, the algorithm should be impoved.That is step-by-step drawing. In other words, once iterate ,once draw.Every drawing is to draw the new generate part.

To imporve L-system, there are three draw-line symbols. They are H,G and F. H is a symbol of drawing line. G is a symbol of drawing line and substitution. F is a symbol of drawing line and substitution and transfering IFS function.

The tree's trunk and branck generate by H. The last graphic is no divarication. The divarication is by G,then there will have be less divarication. The treetop and leaves is by F.

The graphic in Fig.8 shows the steps of drawing plants. n is iteration time.

(1) In Fig.8(a), n=1, draw the parent branch according to ruler. The point symbol as 1,2,3 is the start point of the first iteration. Those points are called as growing point. After parent branch has been drawed, these growing points's position and direction can be recorded in order to draw the first generation subbranch.

(2) In Fig.8(b),n=2,draw the first generation subbranch at the older generation point according to ruler. Of couse ,the second generation subbranch's informations of 1~9 must be writed down.

(3) In Fig.8(c) ,it is the third subbranch at the gorwing points at step2.

Followed as above, a complex plant should be drawed until the iteration limit.

The improved algorithm uses this method of step-by-step drawing. This method can controlled the growth more flexiblely.For example,according to your needs, it can controll the number of growth point randomly based on the iteration times in the process of simulation. And at the growth point, it can adjust those parameters of thickness,length,color

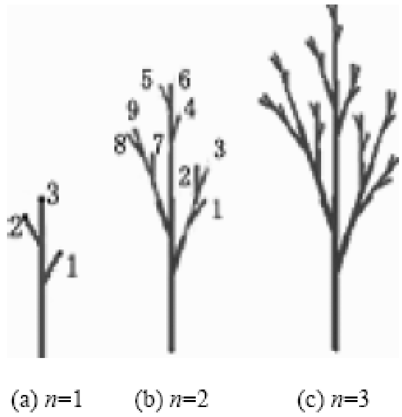


Fig.8 the step of draw a plant

and growth direction ,and so on. And it can controll the next simulation results ,which is branch or leaf.

3.5 Step of Create Realistic Tree

The steps of creating the tree as following:

Step 1: set an original point; create the branch by L-system. Then push the position of the point to the stack.

Step 2: pop the position $(xpos, ypos)$ form the stack.

Step 3: if the stack has been the bottom .turn to step 7.

Step 4: if the position $(xpos, ypos)$ isn't treetop, draw a thick or thin branch at the position turn to step 2.Otherwise turn to step 5.

Step 5: i++,then name a random number K to $(xpos, ypos)$ at the position $(0,1)$. Confirm the extern parameters by L-system and transfer them to IFS function.

Step 6: call the IFS function. Let the point $(xpos, ypos)$ as the original point,confirm the IFS code by K,then draw the leaf. Turn to step 2.

Step 7: add texture mapping[20] to the tree .the end.

3.6 Texture Mapping to the Tree

Texture mapping is an important part of realistic graphic. It can express the surface detail of plant in the 3D scene[21].

The realistic tree's thread structure is clear. There are different colors on different parts. For example, the trunk and branch's color is different with leaves. But when fractal image was created, it wasn't consider the color, gray and texture. So it considers not only the topography structure but also the colored, texture, and so on. We call the process as rendering.

IFS's determinant is iterative parameters. So the IFS fractal image's color is processed by it.

The operation can begin from the probability, at the same time considers the different fractal affine transformation control the different parts of fractal image. Assume $\{w_i : p_i \mid i = 1, 2, ..., n\}$ is a IFS code set , it can assure that exist one and only attractor G and probability P. The probability of G's subset B can be describe as $P(B)$. The attractor G and probability P determine the graphic that would to draw, we call it as basic model (G, P) . G's structure is controlled by the IFS code's transform $\{\omega_1, \omega_2, ..., \omega_n\}$, P is controlled by the IFS code's probability $\{P_1, P_2, ..., P_n\}$ .The color distribution based on probability is to determine the attractor's every area's color according to the changeless probability of IFS attractor.The color is detemined by counting the image point number of every area.

3.7 Implementation of Algorithm

While tree grows, its state is indefinite and random because it is affected by the wind or gravity, and so on. In order to implement the tree more truly, I use Guess random function to show the tree's rando-micity while the algorithm is implemented .For inst-ance, add some random distrbing to the branch's dip angle ,length and branch size during simulating each branch. Then the dip angle ,length and diameter will be created around a range.

Fig.9 shows a tree without random and controlling of parameters. The tree is so symmetrical that seems too stiffness and sameness. The trees in Fig. 10 and Fig.11 are added random number. And it is generated by L-system controlling IFS code's parameters. They look seem more natural and realistic than the tree in Fig. 9.

In Fig.12(a)(b),they are two trees using the improving algorithm.

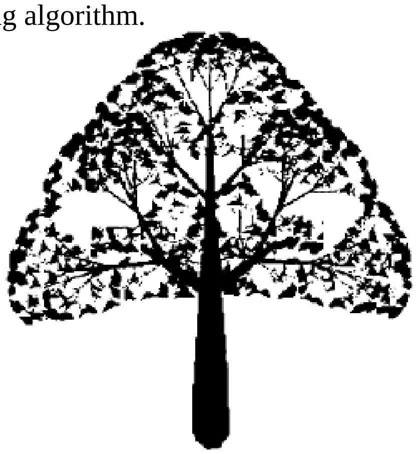


Fig. 9 creating tree without random

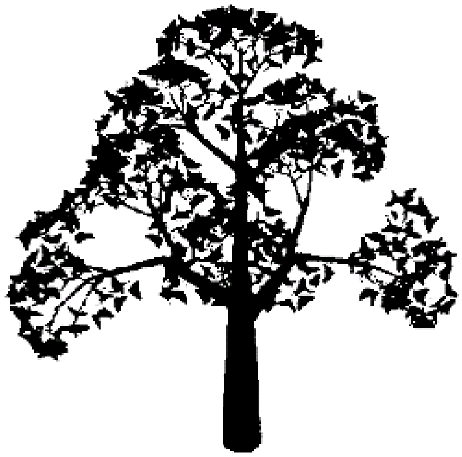


Fig. 10 a tree generated by L-system controlling IFS code's parameters(1)

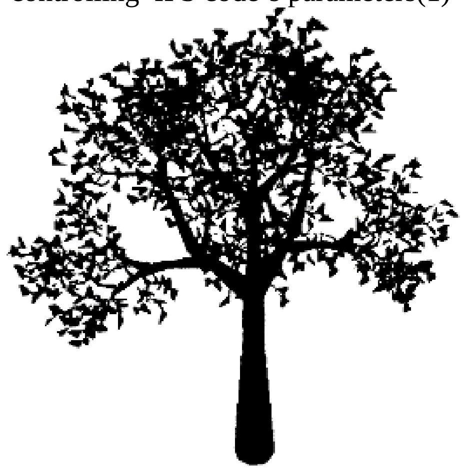
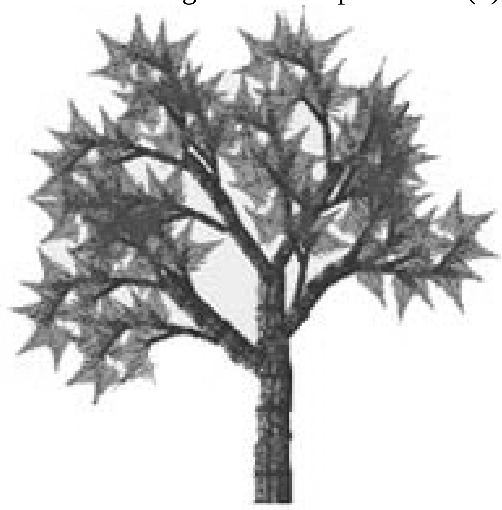
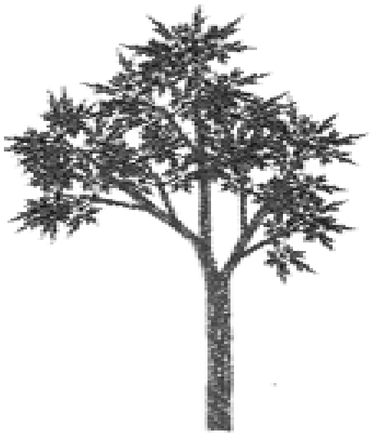


Fig. 11 a tree generated by L-system controlling IFS code's parameters(2)



(a)



(b)
Fig.12 fractal tree

4 Conclusion

In this paper, the tree's generation is a fusion of L-system and IFS. For the different characteristics of L-system and IFS. The leaf is simulated via IFS, and the branch is generated via L-system. To be a whole tree, a generate method comes up. The method is to generate tree by L-system controlling IFS code's parameters. this method has some merit .For example, implementation is easy ,and the random number can be made certain more easily. Via this method the trees looks more realistic .The shape detail of the leaf is more active.

Fractal geometry plays an important role at irregular object's realistic generation. And OpenGL[22] provides a stability flat to simulate the realistic graph. Using functions of OpenGL, static simulated scene is implemented by translation, list, light, color, blend, antialiasing, fog and texture .

The trees that we have seen above are implemented by Visual C++6.0 and OpenGL [23][24].

References:

[1] Hutchinson j, Fractals and self-similarity, *University Journal of Mathematics*, Vol.30, No.2, 1981, pp.713-747.

[2] Barnsley M F, Iterated function system and the global construction of fractals , *Proc Roy Soc London,Ser A*, 1985,399:243-275

[3] Jim Hanan. Virtual plants-integrating architectural and physiological models. *Environmental Modeling & Software*, Vol.1 No.12 1997,pp 35-42

[4] Slawomir S.Nikiel, True-colour images and iterated function systems. *Computer & Graphics*, vol. 5, NO.22 1998, pp 635-640.

[5] Wang Hao-peng, Research on dynamic simulation method of plants based on arithmetic of IFS, *J.Changchun Inst.The.(Nat.Sci.Edi)*,Vol.6 NO.2 2005,pp 49-52

[6] Zhang Shu-bing,Wang Jian-zhong, Improvement of Plant Strucure Modeling Based on L-system, *Journal of Image and Graphics*,Vol.7(A) NO.5 2002,pp 457-460

[7] Yan Yu-bao, Parameter Design of IFS in Fractal, *Journal of Jiangsu Polytechinc University*, Vol.15, No.4 2003, pp.47-50

[8] Zeng Feng, Xu Bo-qin,Zhang Li-ming, Simulation of True-color Plant Based on the Iterated Function System, *IT Age*,NO.10 2005,pp6-8

[9] Prusinkiewicz P, Lindenmayer A. The algorithmic beauty of plants . *Springer-Verlag*,1990, pp 235-238

[10] Glassner A S, Geometric Substitution :A tuotorial, IEEE Computer Graphics and Application, Vol.12 NO.1 1992, pp 22-26

[11] HU Hai-ying, CAO Xiu-ge, LIN Bin, Three Dimensional L – System**,** *Journal of Engineering Graphics*,NO.3 2003,pp 89-93

[12] Zhang Mingshu, Zhang Yan, Computer Simulation Algorithm Research of Three-Dimensional Plant, *Computer Technology and Development,* Vol.16No.10 2006, pp.108-117.

[13] Feng Li, Wang Li, Algorithm of 3D Fractal Plants Based on L-system and Its Implementation, *Computer Simulation,* Vol.22 No.11 2005, pp.205-208

[14]Kong Yong , QU Bai-qing, Application of L-system in Plant Simulation, *Journal of Agricultural Mechanization Research*,NO.4 2007 , pp 136-138

[15] Zhang Gan-yuan , ZhangYi-shun , Wang Jian-guo, Improvement and Analysing of Random Iterated Algorithm of IFS，*Journal Of Southern Institute of Metallurgy*,Vol.26, No.4 2005, pp.50-55.

[16] Liu Xiangdong, Liao Xin, Yu Hai, Zhu Weiyong , Research on the Parameters Control of the Attractor of IFS and the Simulation of the Tree，*Computer Engineering and Applications*, No.5 2000, pp.28-29.

[17] Zhong Lan-fen, Wang Yan, Cheng Lei , IFS Generation Algorithm of 3D Fractal Trees, *Journal of Shenyang Institute of Technology* ,Vol.24, No.1 2005, pp.28-31.

[18] Zhu Yue , Zou Bei-ji ,Tree Simulation Based on the 3D Geometric Modeling Method， *Computer Engineering & Science*, Vol.28, No.7 2006, pp.46-52

[19] Cheng Shao-jiou.An Approach to Plant Structure Modeling Based on L-System, *Journal of Computer-Aided Design & Computer Graphics*,Vol.12 NO.8 2005,pp 571-574

[20] TAN Ru, WANG Xiang-hai, XIN Ling, A Geometry Texture Generating Approach, *Microelectronics & Computer*,Vol.24 NO.6 2006,pp 185-187

[21] Lu Man-sha ,Compounded texture mapping based on OpenGL standard library, *Computer Engineering and Design*,Vol.27 NO.8 2006 pp 1501-1508

[22] Guo Yan-li , Xia Gui-song , Peng Jian-yi , Realization of Generating Method of Fractal Graphics Based on OpenGL, *Computer Knowledge and Technology,* No.11 2005,pp 120-124

[23] Qin Zhong-bao, Shi Huai-yu, Methods of building the realistic graphics of natural objects with fractal using OpenGL, *Computer Applications*,NO.1 2005,pp202-205

[24] Li Da-jin, An Object-Oriented L-system for Simulating Tree Growth, *Computer Simulation* ,Vol.24, NO. 2 2007,pp 183-186