Plant Simulation Based on Fusion of L-System and IFS

Jinshu Han

Department of Computer Science, Dezhou University, Dezhou 253023, China
lihanjinshu@tom.com

Abstract. In this paper, we present a novel plant simulation method based on fusion of L-system and iteration function system (IFS). In this method, the production rulers and parallel rewriting principle of L-system are used to simulate and control the development and topological structure of a plant, and IFS fractal graphics with controllable parameters and plenty subtle texture are employed to simulate the plant components, so that the merits of both techniques can be combined. Moreover, an improved L-system algorithm based on separating steps is used for the convenient implementation of the proposed fusing method. The simulation results show that the presented method can simulate various plants much more freely, realistically and naturally.

Keywords: Fractal plant; L-system; Iteration function system.

1 Introduction

Along with the computer graphics development, the computer simulation for various natural plants as one of the research hotspots in computer graphics domain has the widespread application prospect in video games, virtual reality, botanical garden design, and ecological environment simulation. Nevertheless, at present, commercial image and graph processing software based on Euclidean geometry only can draw some regular geometric shapes, and can't describe the natural plants with complex structures, subtle details and irregular geometric shapes realistically. Although we can use complex techniques such as curves or surfaces to approximate to such natural objects, we must ensure providing adequate memory spaces and high rendering speeds. On the other hand, fractal geometry takes irregular geometric shapes as research objects. One of the fractal typical application domains is natural objects simulation. In plant simulation, there are two common methods: L-system and iteration function system (IFS). Previous works[1,2,3,4,5] mainly focus on plant simulation only by L-system or only by IFS, and can't provide realistic and natural representations for plant constructs and shapes due to the limitation of both methods. Obviously, combining them may be better. However, how to combine them is worth researching further. In response to such problem, a novel plant simulation method based on fusion of L-system and IFS is presented. The simulation results show the feasibility and validity of the presented method.

2 Characteristics of L-System and IFS

L-system[2] is a rewriting system based on symbols. It defines a complex object through replacing every part of the initial object according to rewriting rules. For a certain natural plant, the simulation processes are as follows: Firstly, set of production rules is abstracted according to shapes and development rules of the plant. Secondly, starting from the initial string, a new string is created recursively according to the production rules. Thirdly, the created string is interpreted using turtle graphics algorithm and the corresponding geometric object is drawn.

In plant simulation, L-system can better simulate development rules and topological structure of a plant, which is the main merit of L-system. For example, L-system can describe the development rules and structures of branches, leaves, flowers and seeds freely, because of the fact that the parallel principle of L-system is similar to the parallel growth course of plants. However, comparing with a natural plant, the plant graphics drawn by L-system is lack of texture. The key reasons are as follows: Firstly, the plant branches are represented by lines. Secondly, L-system only considers the self-similarity of the topological branching structure of the whole plant, and doesn't consider that the self-similarity also exists in every component of plant, such as a bark or a leaf.

IFS[3] represents a plant shape based on collage theorem. For a certain plant image, several sub-images generated by contraction affine transform are the little scaled version of the original image. They try to cover original image accurately, and they are permitted to overlap each other partly. So a set of contraction affine transforms are as follow:

$$W = \{R^2 : \omega_1, \omega_2, \dots, \omega_n\} \quad \omega_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r \cos \theta & -q \sin \phi \\ r \sin \theta & q \cos \phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}, i = 1, 2, \dots, n \quad (1)$$

Where, r, q are scaling factors in x or y direction, and θ, ϕ are rotation angles between sub-images and the original image. e, f are translation distances of sub-images in x or y direction. In addition, based on proportion of sub-images' areas, an array of accumulative probabilities associated with W is gotten. It is

$$P = \{p_1, p_2, \dots, p_n\}, \text{ where } p_i > 0, \sum_{i=1}^n p_i = 1, i = 1, 2, \dots, n \quad (2)$$

The above W and P form IFS. Commonly, the random iteration algorithm is used to draw IFS fractal graphics. Firstly, (x_0, y_0) is the initial point. Next, an affine transform is selected from W according to P , and new coordinate point (x_1, y_1) is computed. Then (x_1, y_1) is looked as initial point. Repeat the process, and the orbit of all points form a shape similar to initial image. We call it the attractor of IFS.

In plant simulation, IFS can better represent the subtle texture of a plant and the implementation program is simple, which is the main merit of IFS. Nevertheless, IFS is lack of representation and control for the topological structure of a plant. In other words, IFS simulates a plant only according to the features of the plant's geometric

shapes, and IFS can't represent physiological features of the plant. Therefore simulated plants are machine-made, have few differences from each other. The key reason is: for the plant simulated by IFS, both the whole and the parts have strict self-similarity. Namely, the length and amount of branches of trunks obey the same strict self-similarity rules as those of branches or sub-branches, and the self-similarity rules exist in more little detail sequentially.

Therefore, it is difficult in simulating a natural plant which has plenty texture and various complex structure and shape only by L-system or only by IFS.

3 Plant Simulation Method Based on Fusion of L-System and IFS

Based on above analysis, a novel plant simulation method based on fusion of L-system and IFS is presented. The basic idea is as follows: L-system is used to simulate the diverse development rules and topological structure of a plant, and IFS is employed to simulate the textured plant components, such as branches and leaves. Therefore, the merits of both techniques can be combined. For one thing, through L-system, we can freely control and simulate the development rules and topological structure of plant. For another, through IFS we can quickly simulate the self-similarity and plenty texture of the plant components.

The process of combining L-system with IFS is similar to that of building the building blocks. For one thing, the production rules and parallel rewriting principle of L-system are regarded as building rules for building blocks. For another, IFS fractal graphics with controllable parameters and plenty subtle texture are regarded as various building blocks. Therefore, we can build the IFS building blocks according to L-system rules, so that realistic simulated plant shapes are created. Next, taking Fig.1 as the example, we introduce the implementation process of the novel method.

Fig.1 (a) is a leaf created by IFS[6]. It is regarded as a leaf building block. The IFS algorithm of the leaf can be encapsulated as a function defined as $leaf(x, y, r, k_1, k_2)$. Where x, y is the value of x-coordinate or y-coordinate of the petiole' bottom endpoint, r is the rotation angle of the whole leaf, k_1, k_2 is the scaling factor in x or y direction. If we change the values of x, y, r, k_1, k_2 , or the five controllable parameters, we can create various leaves with different sizes, directions and coordinate positions. Obviously, the leaf building block is not invariable, on the contrary, its size, direction and coordinate position are changeable and controllable.

Fig.1 (b) shows branches created by L-system. L-system is a formal language, which models objects through giving geometric interpretation to the character in string. For instance, character F is a basic symbol in L-system, while drawing, F is interpreted as Moving forward by line length drawing a line. However, in this paper, character F can be interpreted as a novel meaning. For example, some F in the character string can be interpreted as Moving forward drawing a leaf by calling the function $leaf(x, y, r, k_1, k_2)$. It means that the present values of coordinate position and drawing direction or angle are regarded as actual parameters of function $leaf()$. Therefore, An IFS leaf replaces the traditional L-system' simple line. At the same time, next coordinate values and angle are computed to prepare for next drawing step.

Under the plant growth rules shown in Fig.1 (b), the IFS leaf building block with controllable parameters shown in Fig.1 (a) is called, so a plant with leaves in certain growth position is created as shown in Fig.1 (d).

In order to simulate textured branches, we define a branch building block shown in Fig.1(c). The branch building block is created by IFS[7].The IFS algorithm of the branch is encapsulated as a function defined as $trunk(x, y, r, k_1, k_2)$, the meaning of every parameter is similar to that of function $leaf()$. Fig.1 (e) is a modeling plant using L-system to combine IFS leaves and IFS textured branches together.

Fig. 1. Process of simulating plant by combining L-system with IFS: (a) An IFS leaf. (b) A modeling branch with L-system. (c) An IFS textured branch. (d) Plant created with modeling branches and IFS leaves. (e) Plant created with modeling branches, IFS leaves, and IFS textured branches.

Through above process, we may see that as far as the method based on fusion of L-system and IFS, IFS is used to create various graphics building blocks representing components of plant, and we can freely modify set of production rules of L-system, namely, building rules for building blocks can be modified. Actually, L-system determines the development rules and the whole structure of a plant, IFS determines the modeling and texture of every basic part of a plant. Through combining the merits of both techniques, we may simulate the topological structure and subtle texture of a plant freely.

4 Improved L-System Implementation Algorithm

In order to create a realistic and natural modeling plant, we commonly request that any component of the plant, such as a trunk, a branch or a leaf can be changed freely in width, length, color and growth direction. Therefore, in implementation process of the fusing method, according to actual demands, L-system must freely adjust or control the function parameters of plant components created by IFS.

The implementation algorithm of traditional L-system is as follows: Firstly, beginning with the initial string, according to set of production rulers, the string is rewritten repeatedly to create a long string until the iteration number is reached. Secondly, the long string is interpreted and drawn. This kind of final one-off drawing process is not flexible enough. It is difficult in controlling the parameters of plant components freely. Therefore, this paper presents the improved implementation algorithm of L-system. The final one-off drawing process is separated into several steps, namely, the string is rewritten once, then it is interpreted and drawn once, only the new created parts are drawn every time. Fig.2 shows the steps of drawing plant based on improved algorithm, where n is iteration number.

Fig.2 (a) shows the father branches, where $n = 1$. In the figure, numbers 1, 2, 3 label the points called growth points where the first generation sub-branches start to draw. Once the father branches have been drawn, the position and direction of these growth points are recorded, preparing for drawing the first generation sub-branches in next step.

Fig.2 (b) shows the first generation sub-branches drawn from the growth points according to production rules, where $n = 2$. At the same time, the growth points of the second generation sub-branches, labeled by numbers from 1 to 9, are recorded.

Fig.2 (c) shows the third generation sub-branches drawn from growth points recorded in Fig.2 (b).

Repeat above steps, and a plant with complex shape and topological structure is created.

Fig. 2. Process of drawing plant based on the improved L-system algorithm: (a) $n = 1$. (b) $n = 2$. (c) $n = 3$.

The improved algorithm based on separating steps can control the drawing plant process flexibly. For example, during the process of simulating plant, according to actual needs, we can freely control the number of growth points based on the value of n . We can control and adjust the parameters of width, length, color and growth direction of certain branches freely. We can also freely control that the next generation to be drawn is leaf or flower based on n . Therefore, L-system and IFS are combined into an organic whole, and can realize better plant simulation freely.

5 Results

Based on the fusion algorithm of L-system and IFS, many plant simulation experiments have been done. And during the experiments, for every drawing parameter involved in L-system and IFS, modest random factor is added in the drawing process. Fig.3, Fig.4, and Fig.5 are parts of the typical experiment results.

Fig.3 shows the simulation results of four maples. In Fig.3 (a)(c)(d)(e), the development rulers of L-system are used to combine IFS branch building blocks with IFS maple leaf building blocks[8] shown in Fig.3(b), and random factors are added in the process. The simulation results show the four maples have similar natural structure and subtle texture, but their branches and leaves in different ranks have modest differences in width, length, color and growth direction.

Similarly, Fig.4 and Fig.5 are also created by the improved methods presented in this paper. The development rulers of L-system are used to organize IFS branch building blocks and IFS leaf building blocks into an organic whole.

Fig. 3. Plant simulation results 1: (a) Maple 1. (b) Maple leaf. (c) Maple 2. (d) Maple 3. (e) Maple 4.

6 Conclusion

This paper presents an improved method for plant simulation based on fusion of L-system and IFS. In this method, L-system is used to simulate the random development and topological structure of plant, IFS is used to simulate the self-similarity and subtle texture existing in components of plant, so that more natural and realistic simulated plant can be created.

In addition, based on improved method presented in this paper, three dimension L-system and lighting effects can be employed to create much more realistic and natural simulated plants.

Fig. 4. Plant simulation results 2: (a) Arbor. (b) A arbor leaf[6]. (c) Bamboo. (d) A bamboo pole. (e) A bamboo leaf.

Fig. 5. Plant simulation results 3: (a) Shrub. (b) A flower.

Acknowledgments

We thank Q.Z Li and anonymous reviewers for their suggestions and comments.

References

1. Jim Hanan.: Virtual plants-integrating architectural and physiological models. In: Environmental Modeling & Software 12(1)(1997)35-42
2. Prusinkiewicz P.: Modeling of spatial structure and development of plants: a review. In: Scientia Horticulturae 74(1998)113-149

3. Slawomir S. Nikiel.: True-color images and iterated function systems. In: Computer & Graphics 22(5) (1998)635-640
4. Xiaoqin Hao.: The studies of modeling method of forest scenery for three dimension iterated function system. In: Chinese Journal of Computers 22(7) (1999)768-773
5. Zhaojiong Chen.: An approach to plant structure modeling based on L-system. In: Chinese Journal of Computer Aided Design and Computer Graphics 12(8) (2000)571-574
6. Sun wei, Jinchang Chen.: The simple way of using iteration function system to obtain the fractal graphics. In: Chinese Journal of Engineering Graphics22(3) (2001)109-113
7. Chen qian, Naili Chen.: Several methods for image generation based on fractal theory. In: Chinese Journal of Zhejiang University (Engineering Science) 35(6) (2001)695-700
8. Hua jie Liu, Fractal Art. Electronic edition. Electronic Video Publishing Company of Hunan. Hunan China(1997)chapter 5.4

Appendix

The related data and IFS code of the figures in the paper are as follows: where $a = r \cos \theta, b = -q \sin \varphi, c = r \sin \theta, d = q \cos \varphi$.

Table 1. Affine transform parameters in Fig.1 (c)

ω	a	b	c	d	e	f	p
1	0.5	0.5	0.0	0.0	0	0	0.15
2	0.5	0.5	0.0	0.0	50	0	0.35
3	0.5	0.5	0.0	0.0	50	50	0.35
4	0.5	0.5	0.0	0.0	50	50	0.15

In Fig.4 (d), p of ω_3 in Table 1 is changed into 0.15. Other parameters aren't changed.

Table 2. Affine transform parameters in Fig.4 (e)

ω	a	b	c	d	e	f	p
1	0.29	0.4	-0.4	0.3	0.28	0.44	0.25
2	0.33	-0.34	0.39	0.4	0.41	0.0	0.25
3	0.42	0.0	0.0	0.63	0.29	0.36	0.25
4	0.61	0.0	0.0	0.61	0.19	0.23	0.25

In Fig.5 (b), the flower is created recursively based on the equations as follows:

$$\begin{cases} X_{n+1} = bY_n + F(X_n) \\ Y_{n+1} = -X_n + F(X_{n+1}) \end{cases}, F(X) = aX + 2(1-a)X/(1+X^2) \quad (3)$$

where, $a = 0.3, b = 1.0$