Simulation Model of Flower Using the Integration of L-systems with Bezier Surfaces

Peiyu Qin,† and Chuanbo Chen ††,

Institute of Computer, Huazhong University of Science and Technology, Wuhan, China

Summary
This paper presents a simulation model of plant flower. Based on the integration of L-systems representing flower topology with Bezier surfaces depicting flower geometry, a plant flower image generation algorithm is given. The proposed technique is illustrated using the apple flower. The realistic image result reproduces the visual beauty of nature. It is especially suitable for game, entertainment, education and so on.
Key words:
L-systems, Bezier surfaces, flower, realistic image synthesis

1. Introduction

Virtual plants are computer models that recreate the structure and simulate the development of plants. In 1968, Aristid Lindenmayer introduced a formalism for modeling and simulating the development of multicellular organisms [1], subsequently named L-systems. As a general framework for plant modeling, this formalism immediately attracted the interest of computer scientists, botanists and mathematicians. They developed many methods and models for generating plant images, such as the various enhanced L-systems [2], IFS [3], reference axis technique [4], fractal method [5], particle systems for forest scenes simulation [6] and dual-scale automation model [7]. Flower is an important part of plant, so flower simulation plays an important role and also poses an interesting and important challenge in plant modeling. It has a great number of components, such as calyxes, petals, stems, and stamens, which take on highly varied three dimensional shapes and which are connected with intricate structures. The geometric and topologic complexity makes this a difficult and time-consuming task. Realistic plant flower simulation, same as true flower in the nature, has the captivating appeal of reproducing the visual beauty of nature. But, general L-systems only represent plant topology. Realistic effect of plant simulation can not be obtained without plant geometry. Prusinkiewicz developed a method using predefined surfaces [2] to simulate plant organs and got the satisfying results. Based on this method, this paper presents a model for simulating plant flower integrating L-systems with Bezier surfaces.

2. Extensions to L-systems

L-systems are string-rewriting systems. An L-system consists of an alphabet, V, an axiom, ω, and a set of productions, P, defined over V. Each production in P replaces one or more letters of V with zero or more letters in V. A word, x, in the system is a sequence of letters in V. The system's current state is represented by a word. The axiom, ω, is a special word, which represents the system's initial state. At each time step, the production rules are applied in parallel to each of the letters in the current word to produce a new word. But general L-systems were introduced to represent plant topology and only presented two dimensional graphic. To express the form manifested by such properties as the orientation and the length of flower components, the strings must be assigned a geometric interpretation in three dimensions. A LOGO-style turtle interpretation is employed to extend the general L-systems [2] for realistic plant flower simulation. The turtle is represented by its state, which consists of turtle position and orientation in the Cartesian coordinate system, as well as additional attributes, such as current line width. The position is defined by a vector P, and the orientation is defined by three vectors H, L, and U, indicating the turtle's heading and the directions to the left and up (Figure 1). These vectors have unit length, are perpendicular to each other, and satisfy the equation $H \times L = U$. Consequently, rotations of the turtle can be expressed by the equation:

$$[H' \ L' \ U'] = [H \ L \ U] R \quad (1.1)$$

where R is a 3×3 rotation matrix. Specifically, rotations by angle α about vectors U , L and H are represented by the matrices:

$$R_U(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

$$R_L(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad (1.3)$$

Manuscript received February 25, 2006.
Manuscript revised February 28 , 2006.

$$R_H(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \quad (1.4)$$

The turtle is initially located at the origin of a Cartesian coordinate system, with the heading vector H pointing in the positive direction of the y axis, and the left vector L pointing in the negative direction of the x axis. The turtle's actions and changes to its state are caused by interpretation of specific symbols, each of which may be followed by parameters. If one or more parameters are present, the value of the first parameter affects the turtle's state. If the symbol is not followed by any parameter, default values specified outside the L-system are used. The following list species the basic set of symbols interpreted by the turtle for realistic plant flower modeling in the paper.

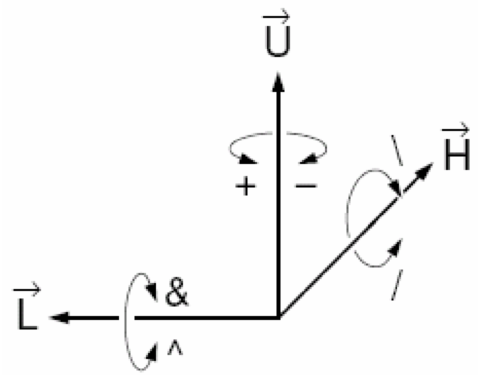

Fig. 1 Controlling the turtle in three dimensions.

F(s): Move forward a step of length s and draw a line segment from the original to the new position of the turtle.
f(s): Move forward a step of length s without drawing a line.
+(α): Turn left by angle α around the U axis. The rotation matrix is $R_U(\alpha)$.
-(α): Turn right by angle α around the U axis. The rotation matrix is $R_U(-\alpha)$.
&(α): Pitch down by angle α around the L axis. The rotation matrix is $R_L(\alpha)$.
^(α): Pitch up by angle α around the L axis. The rotation matrix is $R_L(-\alpha)$.
/(α): Roll left by angle α around the H axis. The rotation matrix is $R_H(\alpha)$.
\(α): Roll right by angle α around the H axis. The rotation matrix is $R_H(-\alpha)$.
[: Push the current state of the turtle (position, orientation and drawing attributes) onto a pushdown stack.
]: Pop a state from the stack and make it the current state of the turtle. No line is drawn, although in general the position and orientation of the turtle are changed.
~ X(s): Draw the surface identified by symbol X, scaled by s, at the turtle's current location and orientation.
#(w): Set line width to w, or increase the value of the current line width by the default width increment if no parameter is given.
!(w): Set line width to w, or decrease the value of the current line width by the default width decrement if no parameter is given.

3. Representation of plant flower using L-systems

The method for various flowers simulation using L-systems is alike, while different plant flowers have different structures. The example in this paper refers to apple flower simulation. The following working steps must be carried out for recreating apple flower.
(1) Observe the structure and development of apple flower and get the abstraction of the structure and development;
In the beginning, flower is composed of bud and stem, and then bud turns into blossom which contains several same segments. A calyx, a petal and a stamen constitute a segment.
(2) Represent the structure and development in symbols;
According to the analysis in step (1), L-system (1) is given below:
ω : AB
p1: B $\rightarrow$ DDDDDD (1)
p2: D $\rightarrow$ CPS
axiom ω : Flower contains stem A and bud B.
production p1: Bud B is composed of six same segments D.
production p2: Segment D contains calyx C, petal P and stamen S.
(3) Explain the whole symbols of apple flower using turtle interpretation and simulate flower in three dimensions in computer.
L-systems (1) represent the topologic information of apple flower without length, line width, relative position and so on. Based on L-system (1), L-system (2) is given below:
initial line width = 20
line width increment = 5
lenA = 500
ω : /(154) &(72)#A!B
p1: B $\rightarrow$ D/(60)D/(60)D/(60)D/(60)D/(60)D (2)
p2: D $\rightarrow$ [C][P][S]
axiom ω : Symbols /(154) and &(72) are set for observation with an appropriate angle at the compute screen. Symbol # increases the value of the current line

width by the default width increment and symbol ! decreases the value of the current line width by the default width increment.
production p1: Symbol /(60) means every two segments is separated by 60 degree.
production p2: Symbol [and] mean pushing the current state of the turtle onto a pushdown stack and popping a state from the stack.
L-systems (2) do not include the geometric information of stem A, calyx C, petal P and stamen S. L-systems (3) are given below:
initial line width = 20
line width increment = 5
lenA = 500
lenF = 100
ω :/(154) &(72) #A!B
p1:B → D/(60)D/(60)D/(60)D/(60)D/(60)D
p2:D → [C][P][S]
p3:A → F(lenA) (3)
p4:C → ~c
p5:P → ~p
p6:S → F(lenF)[-(18)F(lenF)][＋(18)F(lenF)]
production p3: Stem A is expressed as F(lenA) (i.e. a line segment with length lenA).
production p4: Calyx C is interpreted as drawing the surface identified by symbol c(see detailed discussion in section 4)
production p5: Petal P is interpreted as drawing the surface identified by symbol p(see detailed discussion in section 4)
production p6: Stamen S is seen as a combination of F(lenF) and [-(18)F(lenF)][＋(18)F(lenF)] with the length lenF.

4. Sepal and petal simulation based on Bezier surfaces

The standard computer graphics method for defining arbitrary surfaces makes use of bicubic patches. A patch is defined by three polynomials of third degree with respect to parameters u and w. The following equation defines the x coordinate of a point on the patch:

$$x(u,w) = a_{33}u^3w^3 + a_{32}u^3w^2 + a_{31}u^3w + a_{30}u^3 + a_{23}u^2w^3 + a_{22}u^2w^2 + a_{21}u^2w + a_{20}u^2 + a_{13}uw^3 + a_{12}uw^2 + a_{11}uw + a_{10}u + a_{03}w^3 + a_{02}w^2 + a_{01}w + a_{00} \quad (4.1)$$

Analogous equations define y(u, w) and z(u, w). Complex surfaces are composed of several patches.
Bezier-form bicubic patches [8] are employed in this paper. Let P_{ij}(i=0,1,2,3; j=0,1,2,3) be 4×4 points in space, then a Bezier-form bicubic patch is defined as:

$$S(u,w) = \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,m}(u)B_{j,n}(w)P_{ij} \quad u,w \in [0,1] \quad (4.2)$$

where $B_{i,m}(u)$ and $B_{j,n}(w)$ are Bernstein basis functions. A Bezier surface is denoted in matrix form as:

$$S(u,w) = UM_zB_zM_Z^TW^T \quad u,w \in [0,1] \quad (4.3)$$

where

$$U = [u^3 \quad u^2 \quad u^2 \quad 1] \quad (4.4)$$

$$W = [w^3 \quad w^2 \quad w^2 \quad 1] \quad (4.5)$$

$$M_Z = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (4.6)$$

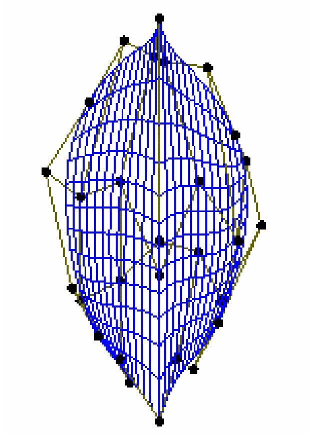

Fig. 2 Caylx

Calyx C (Figure 2) of apple flower is composed of two Bezier-form bicubic surfaces ($S_1(u,w)$ (Figure 3) and $S_2(u,w)$ (Figure 4)) joined with C^0 continuity.

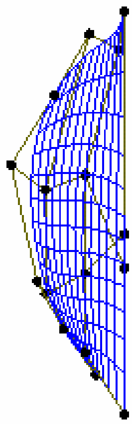

Fig. 3 S_1 (u,w).

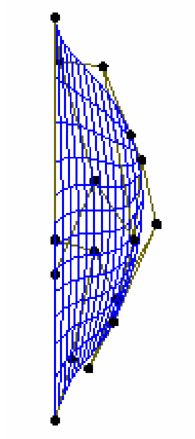

Fig. 4 $S_2(u,w)$.

Petal P (Figure 5) is denoted as a Bezier surface.

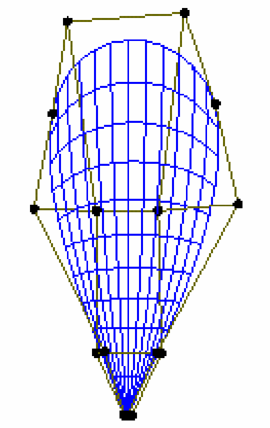

Fig. 5 $S_2(u,w)$.

Finally, an apple flower (Figure 6) forms using a combination of a stem, six calyxes, six petals with six stamens.

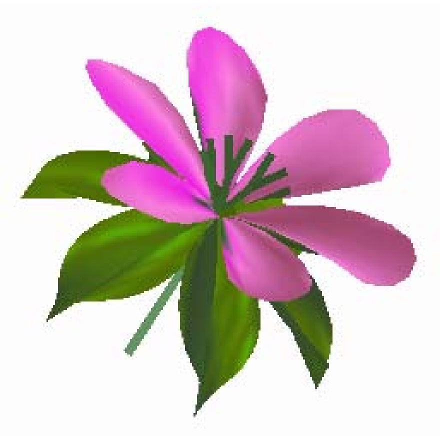

Fig. 6 Apple flower

5. Conclusions

L-systems are good at representing the topologic information of plant flower, while Bezier surfaces excel at depicting geometric information of flower. So the paper puts forward a model for plant flower simulation integrating L-systems with Bezier surfaces. Based on this model, a satisfying realistic image is obtained. This method is not only suitable for plant flower, but also for other plant organs. It is valuable for game, education, advertisement, horticulture et al. However, further research needs to be carried out to develop the method.

Acknowledgments

The authors would like to thank Hui Xia and Guanghui Liu for their discussions pertinent to L-systems-based plant modeling. Many thanks to Yanfen Hu for her editorial help.

References

[1] A. Lindenmayer. Mathematical models for cellular interaction in development, Parts I and II. Journal of Theoretical Biology, 18:280~315, 1968
[2] P. Prusinkiewicz, A. Lindenmayer. The Algorithmic Beauty of Plants. New York: Springer-Verlag, 1990
[3] S. Dekmo, L. Hodges, B. Naylor. Construction of fractal objects with iterated function systems. Computer Graphics, 19(3): 271-278, 1985
[4] F . Blaise, J. F. Barczi, M. Jaeger, P. Dinouard, P. de Reffye. Simulation of the Growth of Plants Modeling of Metamorphosis and Spatial Interactions in the Architecture and Development of Plants. Cyberworlds, Tokyo: Springer-Verlag, 81-109, 1998
[5] P. Openheimer. Real time design and animation of fractal plants and trees. Computer Graphics, 22(4): 55-64,1986
[6] W. Reeves. Approximate and probabilistic algorithms for shading and rendering structured particle systems. Computer Graphics, 19(3):313-322, 1985
[7] X. Zhao, P. de Reffye, F. L. Xiong, B. G. Hu, Z. G. Zhan. Dual-scale automaton model for virtual plant development. Chinese Journal of Computers, 24(6): 608-615, 2001 (in Chinese)
[8] J. G. Sun. Computer Graphics, Beijing: Tsinghua University Press, 1998(in Chinese)