# The Sketch L-System: Global Control of Tree Modeling Using Free-form Strokes

Takashi Ijiri[1], Shigeru Owada[2], and Takeo Igarashi[3]

[1] Department of Computer Science, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 133-0033, Japan
`ijiri@ui.is.s.u-tokyo.ac.jp`
[2] Sony Computer Science Laboratory, Inc.
3-14-13, Higashigotanda, Shinagawa-ku, Tokyo 133-0033, Japan
`sowd@acm.org`
[3] Department of Computer Science, University of Tokyo / PRESTO, JST
`takeo@acm.org`

**Abstract.** L-system is a tool commonly used for modeling and simulating the growth of plants. In this paper, we propose a new tree modeling system based on L-system that allows the user to control the overall appearance and the depth of recursion, which represents the level of growth, easily and directly, by drawing a single stroke. We introduce a new module into L-system whose growth direction is determined by a user-drawn stroke. As the user draws the stroke, the system gradually advances the growth simulation and creates a tree model along the stroke. Our technique is the first attempt to control the growth of a simulation in L-system using stroke input.

## 1. Introduction

Fractals are commonly used in computer graphics, not only for creating visual effects, such as fires, clouds, and lightning, but also for modeling living organisms. Lindenmayer introduced L-system [8], which represents the fractal structure of plants using grammatical expressions. Starting with an initial structure, L-system constructs the fractal structure by applying rewriting rules sequentially. One important aspect of L-system is that the rewriting process closely mimics the growth of real plants, which makes L-system the most popular tool for modeling plants. However, the replacement of parts essentially involves a local description, and local changes strongly affect the global shape. This causes difficulty in creating a desired global shape using L-system.

Our goal is to allow the user to control the global shape of an L-system model easily and directly. In this paper, we propose a system that allows the user to specify the central axis of the plant by drawing a stroke. We also use the stroke to determine the depth of recursion; when the user draws a longer stroke, the system applies generating rule more times. A key idea is introducing a special module whose growth direction is determined by a user-drawn stroke. Figure 1 shows our prototype system. The system provides a predefined generating rule that consists of an internode, lateral apices, and the user-controlled top apex (Figure 1(a)). The user begins modeling by specifying the parameters of the generating rules, and then draws a free-form stroke

representing the central axis. As the user draws the stroke, the system advances the growth simulation by growing the top apex along the stroke and applying the generating rules to the lateral branches (Figure 1(c)).

Using our system, the user can control the axis and depth of recursion intuitively, by drawing a single stroke. Whereas the generating rule represents local structures, the stroke roughly specifies the global shape. Therefore, the user can easily create a variety of tree models that have the same local structure, but different global shapes, by drawing different strokes with the same generating rules. Figures 1 and 6 show tree models made using our system. The user can create these models easily by manipulating a few control points and drawing a stroke.
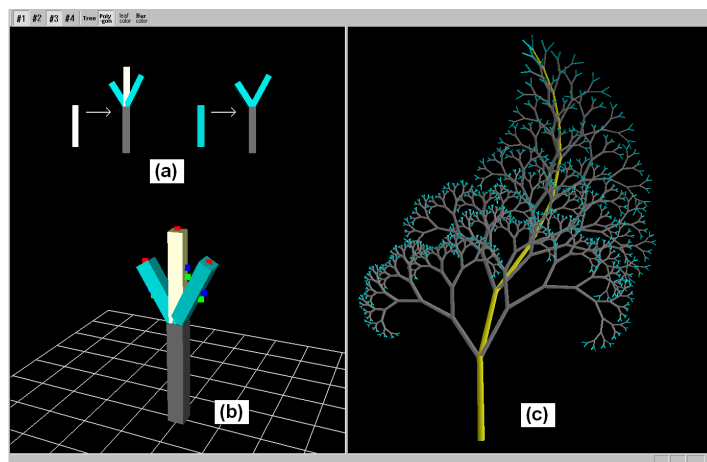


**Fig. 1.** A snapshot of the proposed system. The user first modifies the generating rules (a) by manipulating control points on a handle (b). Next, the user draws a single stroke representing the axis in the right panel (c). Then, the system creates a fractal structure along the stroke. The user-drawn stroke is highlighted in yellow.

## 2. Related Work

Our work is built upon knowledge in several different fields. Here we briefly review representative references in the three most relevant fields: L-System, plant modeling, and sketch-based interface for 3D modeling.

### 2.1. L-System

A. Lindenmayer originally introduced L-system in 1968 to formalize the development of multicellular organisms [8] and subsequently expanded it to represent higher plants and complex branching structures [5]. The framework of L-system consists of an *initial structure* and *rewriting rules* (or *generating rules*). The essence of development is parallel replacement using the rewriting rules. Starting from the initial structure, L-system replaces each part of the current structure by applying the rule sequen-

tially. Figure 2 shows a simple example, the development of a compound leaf [17]. This includes two module types: the *apices* (thin lines) and the *internodes* (thick line). In this example, there are two rewriting rules (Figure 2, top-left): one replaces an apex with an internode, two lateral apices, and a top apex, while the other replaces an internode with a longer one. The initial structure is a single apex. Using these simple rules, the system develops an intricate branching structure over a number of replacing steps. An interesting aspect of the system is that each replacement process corresponds to the growth of part of the plant. Therefore, L-system is not only a heuristic technique that creates fractal-like shapes, but is also a simulation of real-world plant growth.
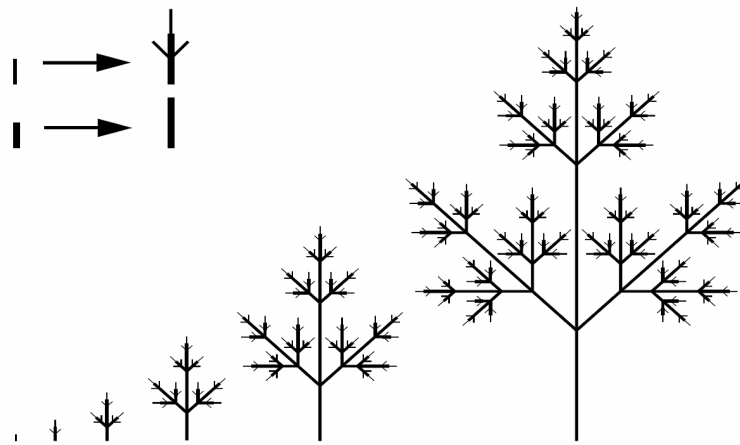


**Fig. 2.** Developmental model of a compound leaf. Beginning with the initial structure (leftmost thin line), the system generates a complicated structure by applying two generating rules (top-left) sequentially. This figure is from [17, page 4].

## 2.2. Plant Modeling

Lindenmayer [8] originally invented L-system, and Prusinkiewicz and Lindenmayer later introduced it to the computer graphics community [16]. L-system has been extended to simulate a wide variety of interactions between plants and their environments [9] [17] [19]. Subsequently, they designed a plant-modeling language called cpfg [18], which supports both context-free and context-sensitive generating rules. They also implemented the plant modeling software L-Studio based on cpfg [20]. Recently, Prusinkiewicz *et al*. [21] also proposed the use of positional information to control parameters along a plant axis. Boudon *et al*. [2] proposed an L-system-based process for designing bonsai tree models, which uses decomposition graphs to facilitate the manipulation of hierarchical parameters. These systems produce intricate, realistic plant models and illustrate the great potential of L-system. However, these systems focus only on manipulating the rules and parameters, and do not support direct manipulation, such as drawing an axis.

Streit *et al*. introduced a biologically based method for modeling plant variation from a basic plant model represented by L-system [22]. Their system mimics the underpinning of variation in real plants by simulating biological growth based on a feedback control system. They allow the user to specify the central axis and use it as an external stimulus that determines the growth direction. Onishi *et al*. proposed a tree-modeling system using 3D gestures [13] [14]. The user specifies the shape of dominant trunks using a 3D tracker, and then the system generates trunk models represented by a so-called *L-String,* which represents the initial structure of the standard L-system. Subsequently, the user attaches branches or leaves to the trunks using L-system. Both systems support the global control of the central axis of plants. However, these systems only modify the shape and do not control the growth of plants. Therefore, it is impossible for these systems to vary the depth of recursion depending on the user-specified axis.

Deussen and Lintermann developed the Xfrog system [4], which combines the power of a rule-based approach and intuitive user interfaces using a graph representation. Users design a graph representing the branching structures of a plant with 11 node types. This system offers an intuitive user interface and the resulting models are very realistic. However, their system does not support to control global shape directly and it is still difficult to design plants that have specific global appearances. Trees-Designer [12] is another commercial system for modeling plants. The user interface is based on supplying proxy geometry, such as boxes or quadrangles, to control the global appearance of trees. However, locating proxy geometry requires a traditional 3D CAD-like user interface that is substantially more difficult than sketching the central axis.

### 2.3. Sketch-based Interface for 3D Modeling

In recent decades, sketch-based modeling has become popular; instead of creating precise, large-scale objects, a sketching interface provides an easy way to create a rough model that quickly conveys the user's intentions. SKETCH [24] allows users to design 3D scenes consisting of simple primitives, while Teddy allows users to design free-form models [6]. Generating 3D curves through sketching is also a rich research domain; Pentland and Kuo [15] generated a 3D curve from its 2D projection using energy minimization, while Ijiri *et al*. [7] used a constant curvature. Another strategy for defining a 3D curve is to draw strokes twice. For example, combining a screen projection of a curve with its shadow is useful for disambiguating the 3D position [3] [23].

Recently, some systems have applied sketching interfaces to plant modeling. Okabe *et al*. [11] presented a sketch-based tree modeling system. They infer the 3D geometry of a tree using the assumption that a tree tends to maximize the distance between its branches. Ijiri *et al*. [7] introduced a flower-modeling system that separates the modeling process into structure definition and geometry creation. They used floral diagrams and inflorescences for structure definition and sketch interfaces for geometry creation. Neither method used L-system. Although these techniques support the propagation or reuse of existing branches or leaves, the user needs to design a tree in detail manually. Maya Paint Effects [1] allows the user to modify the shape of a

model by sketching its skeleton. Although this user interface offers an easy way to control the global shape of plants, it is essentially a deformation technique and does not simulate the growth of plants as our system does.

## 3. User Interface

Figure 1 shows a snapshot of our prototype system. The system consists of two panes; a control pane (left) and a creation pane (right). The user modifies generating rules in the control pane, and model trees in the creation pane. This section describes each of these user interfaces.

### 3.1. Manipulation of the Generating Rule and its Parameters

The user begins modeling with specifying generating rules of L-system in the control pane. Our system provides predefined generating rules (Figure 3(i)) that contain three module types: top apex (Figure 3a), internode (Figure 3b), and lateral apex (Figure 3c). Currently, our system supports the definition of two rules only. In the future, we will design it to support an arbitrary number of rewriting rules. In this generating rule, the left rule replaces a top apex by another top apex, an internode, and lateral apices, while the right rule replaces a lateral apex with an internode and lateral apices. Although the number of lateral branches is specified in a dialog box, the other parameters are set by directly manipulating control points (Figure 3(iii)). The change in parameters is reflected in the visual representation of the rule immediately. The controllable parameters are the lateral branch length ratio, width ratio, twisting angle, and
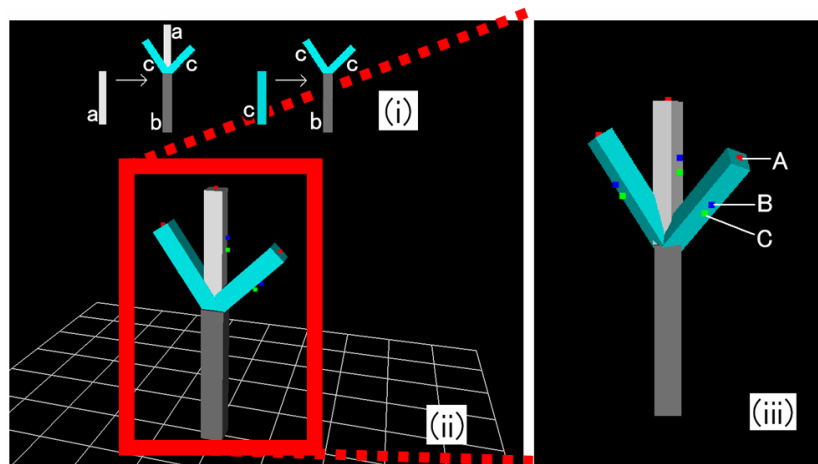


**Fig. 3.** A screenshot of the control pane. The generating rule is displayed above (i). The user modifies the parameters of the rule directly by manipulating control points of the handle (ii, iii). Control point "A" changes the orientation and the ratio of the branch lengths; "B" changes the twisting angle; and "C" changes the ratio of branch widths.

growth direction, and the top apex length ratio, width ratio, and twisting ratio. Here, "ratio" means the ratio between the current branch parameter and that of the parent branch. The top apex is different from modules in a standard L-system and its growth direction is determined in the subsequent generating process, described in the next section.

## 3.2. Geometry Creation by Drawing a Stroke

After setting the rules and parameters, the user creates tree geometry in the creation pane. The interface is very simple; the user draws a single stroke representing the central axis (Figure 4). The stroke corresponds to the top apex of the generating rule (Figure 3(i)). Since the top apex has a certain length, the system needs to resample the stroke.

When the user begins drawing a stroke, the system creates a straight line segment connecting the starting point and the current point. If the length of the line reaches a predefined value $l$, the system applies the rule using the line segment as a top apex (Figure 4(a)) and also updates the starting point of the line. Subsequently, when the line length reaches the value $l \times ratio$, the system reapplies the rule (Figure 4(b)). Here, "$ratio$" is the length ratio of the top apex determined in the previous process. As the user draws a stroke, the system repeats this process, growing the tree structure (Figure 4(d)–(h)). The length of the n-th line segment is $l \times ratio^{(n-1)}$. The growth of lateral branches is determined by the parameters specified in the previous process and is independent of the user drawn stroke. The system provides real-time feedback while the user is drawing a stroke.
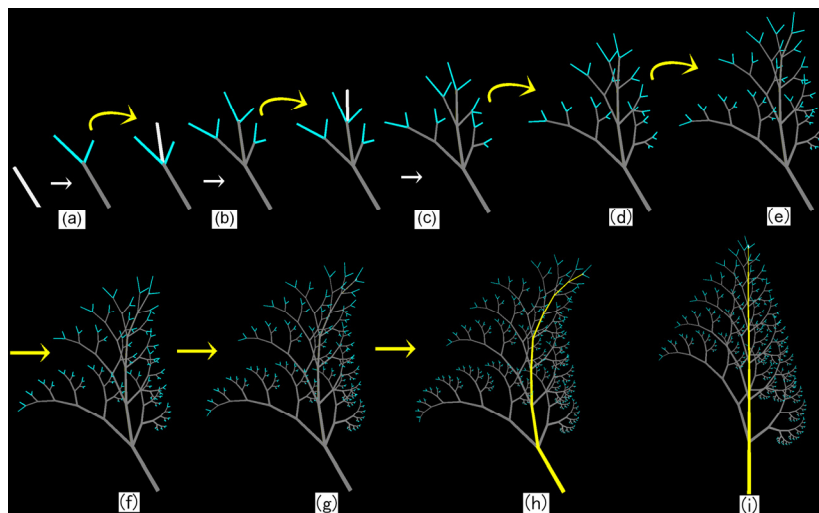


**Fig. 4.** Structure creation by drawing a stroke. The system interprets the user-dawn stroke (white bar) as the top apex, and applies a generating rule (a)–(c). Based on the stroke, the system generates the geometry (a)–(h). (i) The structure created by a straight stroke. In (h) and (i), the axis stroke is highlighted in yellow.

This framework allows the user to specify the depth of recursion, as well as the axis shape with a single stroke. If the user stops drawing when the stroke is short, the number of recursions is small and the resulting structure is thin. On the other hand, the longer the stroke is, the more iterations occur. To improve the visual appearance, the user can add a predefined leaf object on each terminal branch by pressing "tree button" (Figure 5).

Our system also allows modifying the generating rule and parameters *after* constructing the whole geometry. The changes in the rule and parameters are immediately applied to the tree model. This enables the user to determine desirable parameters through intuitive trial-and-error session. This is often difficult in standard L-System framework where the parameter setting and the geometry construction are separated.
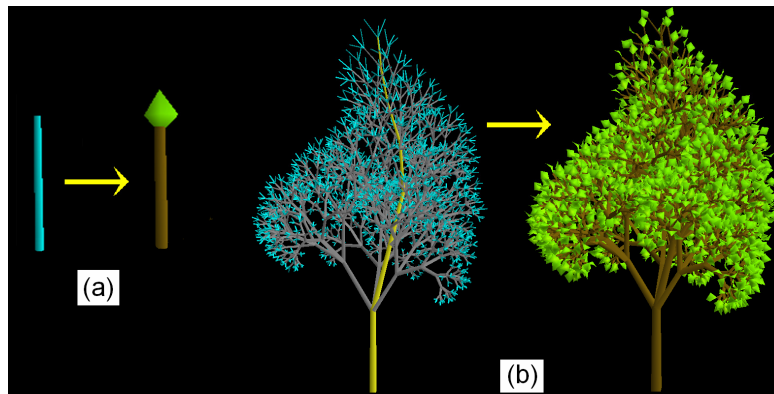


**Fig. 5.** Decorating a generated tree model. The system adds a leaf object to each edge of the terminal branch (a).

## 4. Discussion

In this paper, we have proposed a new tree-modeling system based on L-system. We introduced a special module to the generating rule of L-system whose growth direction is determined by a user-drawn stroke. Our system allows the user to control the global appearance and depth of recursion of the tree model easily and directly by drawing a single stroke.

Figure 1 and 6 show tree models designed using our system. Figures 6(f)–(h) show a variety of models created using a common rule set with different central strokes. Since our system provides a simple and easy-to-use interface, the user can complete each model in a few minutes. Furthermore, the data structure in our system is simple; each model is constructed by only two generating rules and one stroke. Therefore, tree models designed by our system will be easily exported to a standard L-System framework or to a general 3D modeling system. We believe that our system is useful not only for designing tree variations but also as a tool for artistic expression. A media artist, Professor Nagashima, developed a handheld device to interactively control

the creation of trees in collaboration with us and created art installation combining our system and music [10].

The most important aspect of our approach is the ability to directly control the global shape by drawing a stroke, while the local structure only emerges as a result of applying local rules in standard L-system. In the future, we would like to combine our approach with other advanced sketch-based plant-modeling systems [7] [11]. Since our system currently supports only a specific set of generating rules, we would also like to extend our system to support a wider range of rules.
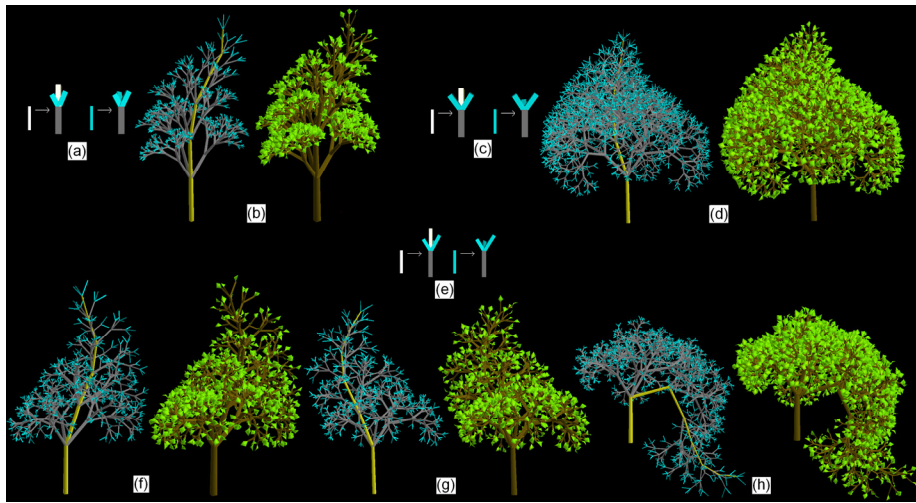


**Fig. 6.** Tree models and the corresponding generating rules. Each tree was designed in a few minutes using our system. Models (f)–(h) were created with the same rule

## Acknowledgements

## References

[1]    Autodesk and Alias. Maya. www.alias.co.jp
[2]    Boudon, F., Prusinkiewicz, P., Federl, P., Godin, C., and Karwowski, R. Interactive Design of bonsai tree models. Computer Graphics Forum, 22, 3, 591–599. (Proc. Eurographics'03)
[3]    Cohen, J., Markosian, L., Zeleznik, R., Hughes, J., and Barzel, R. An interface for sketching 3D curves. In Proceedings of ACM I3D 99, 17–21.

[4]  Deussen, O., and Lintermann, B. 1999. Interactive modeling of plants. IEEE Computer Graphics and Applications, 19, 1, 56–65.

[5]  Frijters, D., and Lindenmayer, A. A model for the growth and flowering of Aster novae-angliae on the basis of table (1,0) L-systems. In G. Rozenberg and A. Salomaa, editors. L Systems, Lecture Notes in Computer Science 15, 24–52. Springer–Verlag, Berlin, 1974.

[6]  Igarashi, T., Matsuoka, S., and Tanaka, H. Teddy: a sketching interface for 3D freeform design. In Proceedings of ACM SIGGRAPH 99, ACM, 409–416.

[7]  Ijiri, T., Owada, S., Okabe, M., and Igarashi, T. Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints. In Proceedings of ACM SIGGRAPH 2005, ACM, 720–726.

[8]  Lindenmayer, A. Mathematical models for cellular interactions in development, I & II. Journal of Theoretical Biology, 280–315.

[9]  Měch, R. and Prusinkiewicz, P. Visual models of plants interacting with their environment. In Proceedings of ACM SIGGRAPH 96, ACM, 397–410.

[10]  Nagashima, Y. Nature system. IAMAS, inter-college computer music concert. December 2005 in Japan.

[11]  Okabe, M., Owada, S., and Igarashi, T. Interactive design of botanical trees using freehand sketches and example-based editing. Computer Graphics Forum, 24, 3, C487–C496. (Proc Eurographics '05)

[12]  Olas, P. TreesDesigner. www.polas.net/trees/

[13]  Onishi, K., Hasuike, S., Kitamura, Y., and Kishino, F. Interactive modeling of trees by using growth simulation. Proc. of ACM Symposium on Virtual Reality Software and Technology, 66–72, 2003.

[14]  Onishi, K., Murakami, N., Kitamura, Y., and Kishino, F. Modeling of trees with interactive L-system and 3D gestures. Lecture Notes in Computer Science (Proc. of BioADIT 2006), 3853, 222–235, Jan. 2006.

[15]  Pentland, A., and Kuo, J. The artist at the interface. Vision and Modeling Technical Report 114, MIT Media Lab.

[16]  Prusinkiewicz, P., and Lindenmayer, A. The Algorithmic Beauty of Plants. Springer–Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.

[17]  Prusinkiewicz, P., Hammel, M., Hanan, J., and Měch, R. L-systems: from the theory to visual models of plants. In Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences, 1996.

[18]  Prusinkiewicz, P., Hanan, J., and Měch, R. An L-system-based plant modeling language. Lecture Notes in Computer Science 1779, 395–410. Springer–Verlag, Berlin, 2000.

[19]  Prusinkiewicz, P., James, M., and Měch, R. Synthetic topiary. In Proceedings of ACM SIGGRAPH 94, ACM, 351–358.

[20]  Prusinkiewicz, P., Karwowski, R., Měch, R., and Hanan, J. Lstudio/cpfg: a software system for modeling plants, 2000. Lecture Notes in Computer Science 1779, 457–464. Springer–Verlag, Berlin, 2000.

[21]  Prusinkiewicz, P., Mündermann L., Karwowski, R., and Lane, B. The use of positional information in the modeling of plants. In Proceedings of ACM SIGGRAPH 2001, 289–300.

[22]  Streit, L., Federl, P., and Sousa M.C. Modelling plant variation through growth. Computer Graphics Forum, 24, 3, C487–C596. (Proc. Eurographics '05)

[23]  Tobita, H., and Rekimoto, J. Flat3D: a shared virtual 3D world system for creative activities and communication. IPSJ, 44, 2, 245–255 (in Japanese).

[24]  Zeleznik, R.C., Herndon, K.P., and Hughes, J.F. SKETCH: an interface for sketching 3D scenes. In Proceedings of ACM SIGGRAPH 96, ACM, 163–170.