

*It may be true, as the poet said, that ". . . only God can make a tree,"
but through geometric modeling a computer graphics system can
produce a remarkable likeness.*

Botanical Tree Image Generation

Masaki Aono and Toshiyasu L. Kunii

University of Tokyo

Computer graphics provides an excellent way to represent natural objects or phenomena. The drawing algorithms now available for these representations are based largely on irregularity and fuzziness, as exemplified by the algorithms for fractal surfaces^{1,2} and particle systems.³⁻⁵

The underlying principle governing the development or construction of biological objects, however, does not necessarily appear to be based on their irregularity or fuzziness. On the contrary, biological objects exhibit aspects of shape that are inherently regular and deterministic throughout their life cycle.

This article presents botanical trees as models of biological objects, first by defining their developmental rules in a discrete grammatical form, then by defining them in continuous geometric forms. Having analyzed these models from several standpoints, we have developed an interactive synthetic tree manipulation system called the A-system.

There are several models for describing tree branching patterns, but we present an L-system model first because of its simple grammatical rules. The L-system was originally proposed by A. Lindenmayer,^{6,7} who applied it to the development of lower forms of plant life, such as red algae. The L-system, also called the developmental system, has since been applied in many fields, including formal language theory and biomathematics. After extensive study and experimentation, however, we found that the L-system is not powerful enough to represent complex three-dimensional branching patterns.

Many researchers⁸⁻¹³ have attempted to model botanical trees in geometric forms and test the results. The representations, for the most part, have not been flexible enough, nor have they been satisfactory in closely modeling specific plants. Moreover, these studies have not discussed applications or syntheses of the generated trees at any length.

We propose new geometric models (GMT1, GMT2, GMT3, and GMT4), analyzing and comparing them with the L-system approach. GMT1 is an elementary bifurcation model capable of representing simple tree shapes, and

it is a basis for other geometric models. We perform a parametric pattern analysis through this model. It cannot, however, deal with the effects of wind, sunlight, and gravity, nor with the genesis of more than two child branches from their mother branch. Furthermore, it cannot represent statistical variations of the child branches. GMT2 is a modified model that *can* represent these effects by using uniform and/or nonuniform deviators. It allows us to generate very realistic tree images. GMT3 is a ternary branching model. By combining it with GMT2, we can bring the generated images closer to reality. GMT4 is a statistical branching model that can represent variations of the branching angles with the growth level. We tested and evaluated the fitness of the trees generated by our models through computer graphics against real plants.

With the help of these studies, we developed the A-system. It enables us to organize the various types of tree images generated by the above geometric models and to deal with many applications. We enhanced the A-system by introducing the ability to incorporate such constructs as leaves, shadows, and shades, and to perform three-dimensional transformations to a tree. One of its applications is to assist with landscaping, which includes gardening and the design of street plants.

Representation of tree branching patterns

The branching patterns of higher plants are evident everywhere and are relatively easy to formalize, providing excellent examples for study.

From a morphological standpoint, there are only two basic categories of tree branching patterns: the dichotomous branching pattern and the monopodial branching pattern. With the dichotomous branching pattern, a branch divides in two and each branch goes in a different direction from the original. With the monopodial branching pattern, a branch divides in two at the growth point but one follows the direction of the main axis and the other goes in a different direction to form a lateral branch.

Pure dichotomous branching is rare in higher plants, although it is often seen in algae, a lower form of plant life. Most higher plants have monopodial branching. Typical examples are pines and cedars, whose main axes grow straight upward.

Two models that represent tree branching patterns are the L-system model and the geometric model.

L-system model. In his deterministic zero-interaction L-system (DOL-system), Lindenmayer^{6,7} showed that the branching patterns of primitive filamentous organisms can be formalized. He first applied his model to red algae. The branching rules are represented by parentheses, as illustrated in the example below. Though simple, the model can represent both dichotomous and monopodial branching patterns. The L-system does not support drawing rules by itself. Therefore, to give shapes to the generated strings, we must adopt some rules for branching angles and orientations.

The original DOL-system is defined as a triple $K = \langle G, w, P \rangle$, where G is a set of symbols, w is a starting symbol, and P is a set of production rules. The difference between the L-system and the Chomsky grammar¹⁴ is that in the L-system, rewriting rules are applied in parallel to all symbols in the current state. Changing the current state corresponds to growing one level forward.

Before summarizing the characteristics of the L-system, let's see how it describes branching patterns.

$$K1 = \langle G, w, P1 \rangle : \text{a monopodial branching}$$

$$K2 = \langle G, w, P2 \rangle : \text{a dichotomous branching}$$

$$G = \{g, d, (,)\},$$

$$w = g,$$

$$P1 = \{g \rightarrow d(g)g, \quad d \rightarrow d, \quad (\rightarrow (,) \rightarrow)\}$$

$$P2 = \{g \rightarrow d(g)(g), \quad d \rightarrow d, \quad (\rightarrow (,) \rightarrow)\}$$

$$\langle \text{level} \rangle \quad \langle \text{generated string for } K1 \rangle$$

$$1: \quad g$$

$$2: \quad d(g)g$$

$$3: \quad d(d(g)g)d(g)g$$

$$4: \quad d(d(d(g)g)d(g)g)d(d(g)g)d(g)g$$

.....

$$\langle \text{level} \rangle \quad \langle \text{generated string for } K2 \rangle$$

$$1: \quad g$$

$$2: \quad d(g)(g)$$

$$3: \quad d(d(g)(g))(d(g)(g))$$

$$4: \quad d(d(d(g)(g))(d(g)(g)))$$

.....

The following drawing rules are applied to give shapes to a set of generated strings for either type of branching pattern.

(1) Each alphanumeric character of the generated string denotes a filamentous cell; when drawn it is represented as a line segment. A pair of parentheses (left and right) in the string denotes a branch; the pair also specifies the direction of the branch, which is actually a set of line segments representing the subsequent cells. Except where confusion might result, no distinction is made between cells, characters, or line segments in the following discussion.

(2) The first alphanumeric character is drawn straight upward.

(3) Different alphanumeric characters have different cell lengths.

(4) A left parenthesis represents the beginning of the branch and a right parenthesis represents its end.

(5) For the string $\dots a(p)b(q)\dots$, branches p and q are drawn on opposite sides of the plane. The left side is drawn first. Each branch has a constant angle to the line segments a and b (Figure 1).

(6) For the string $\dots a(p)(q)\dots$, branches p and q are drawn on opposite sides of the plane (Figure 1).

If the notation $|a|$ means the length of cell a , one way to draw figures of the above examples is by setting $|d| = 2 * |g|$ and the branching angle = 45°. The strings generated at growth level 4 of these examples are illustrated in Figure 2.

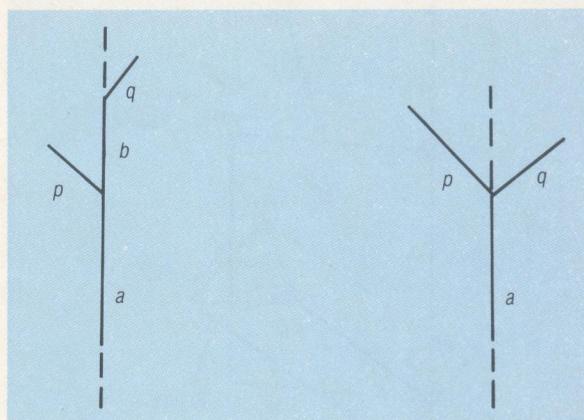


Figure 1. The strings $\dots a(p)b(q)\dots$ and $\dots a(p)(q)\dots$ are given shapes according to drawing rules 5 and 6.

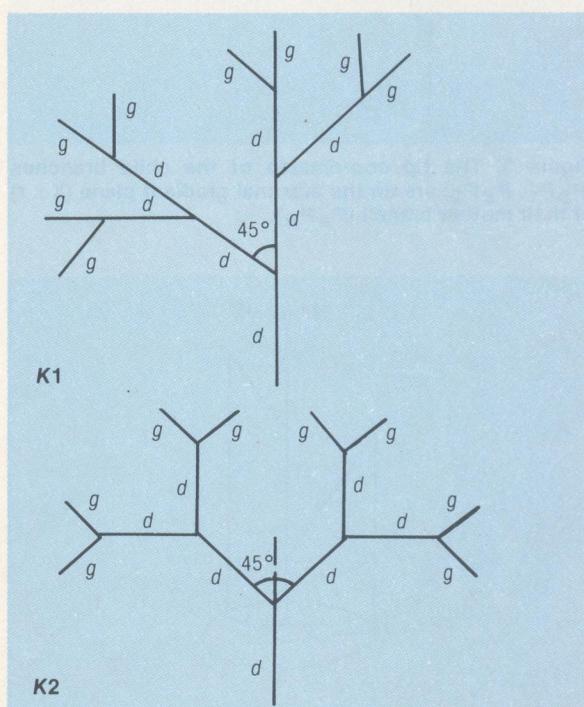


Figure 2. Examples of a monopodial branch (K1) and a dichotomous branch (K2), both at growth level 4.

The following is a summary of DOL-system characteristics.

- (1) The primitive unit is a cell, and the generated patterns are inherently one-dimensional linear arrays.
- (2) The time is specified discretely.
- (3) Morphologically or physiologically, at a given time, the subsequent state of a cell is a function of the current state and is determined uniquely.
- (4) The development of a cell corresponds to either its differentiation or its death.
- (5) Migration or expansion of cells cannot be handled.
- (6) Because cells and the states of cells are discrete entities, the DOL-system cannot deal with continuous quantities.

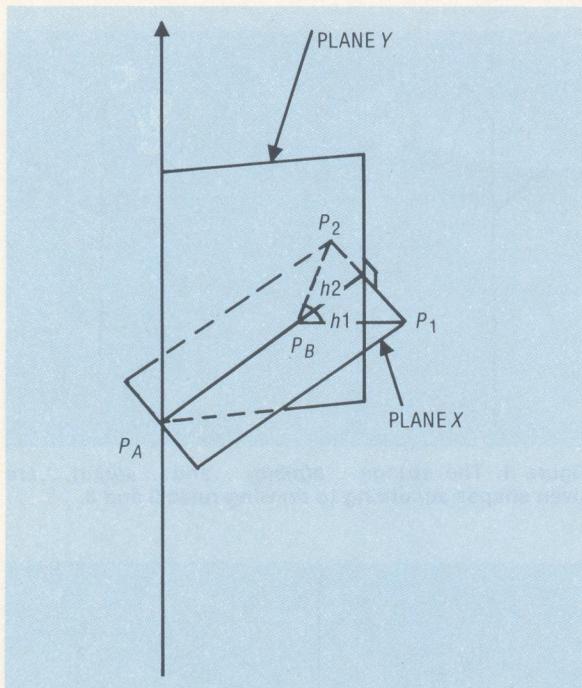


Figure 3. The tip coordinates of the child branches (P_BP_1, P_BP_2) are on the maximal gradient plane ($X \perp Y$) of their mother branch (P_AP_B).

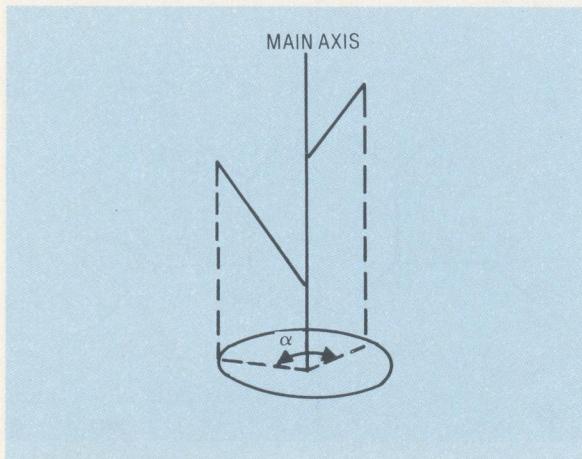


Figure 4. Divergence angle.

(7) Without suitable drawing and branching rules, the system cannot generate two- or three-dimensional developmental patterns.

Although the L-system has been extended to incorporate environmental effects (DIL-system) and a rule-switching mechanism (TIL-system),¹⁵ the fundamental difficulty in modeling the branching patterns of higher plants is that the L-system facility is at too low a level to describe such complex patterns.

3-D geometric model GMT1. The branching patterns generated by the L-system are nothing more than a set of strings, and they provide no clues to either their metric or topological information. To perform metric measurement or geometrical manipulation, complicated drawing rules are required.

To generate branching patterns as simply as possible, and to allow the generated patterns to be flexibly manipulated, we propose GMT1, a three-dimensional geometric model that consists of the following rules.

(1) Two child branches are generated by one branching (bifurcation).

(2) The length and diameter of the child branches become shorter at constant ratios and their branching angles are the same at any level of growth.

(3) Two child branches lie on the maximal gradient plane of their mother branch (Figure 3). The first two child branches in particular lie on the plane parallel to the direction of gravity.

(4) Branching occurs simultaneously at the tips of all branches.

Rules 2 and 3, which determine the geometry in 3-D space, do not appear in the L-system model. On the basis of these rules, the positions of child branches are determined as follows (see Figure 3). Let's use a branch represented by the two end positions: a "root" position and a "tip" position. We will denote the two branching angles as $h1$ and $h2$, where $h1 * h1 + h2 * h2 \neq 0$, and assume the following notations:

- P_AP_B : a mother branch
- P_BP_1, P_BP_2 : child branches
- $P_A = (x_A, y_A, z_A)$: a root of the mother branch
- $P_B = (x_B, y_B, z_B)$: a tip of the mother branch, and also the roots of its child branches
- $P_1 = (x_1, y_1, z_1)$: a tip of the child branch P_BP_1
- $P_2 = (x_2, y_2, z_2)$: a tip of the child branch P_BP_2
- R_1 : a contraction ratio of the child branch P_BP_1 to its mother branch P_AP_B
- R_2 : a contraction ratio of the child branch P_BP_2 to its mother branch P_AP_B

Then, the tip coordinates of the child branch i ($i=1, 2$) are given as follows:

$$x_i = x_B + R_i * (u * \cos(h_i) - S * T * v * \sin(h_i))$$

$$y_i = y_B + R_i * (v * \cos(h_i) + S * T * u * \sin(h_i))$$

$$z_i = z_B + R_i * w * \cos(h_i)$$

where

$$u = x_B - x_A$$

$$v = y_B - y_A$$

$$w = z_B - z_A$$

$$S = 1 / \sqrt{u * u + v * v}$$

$$T = \sqrt{u * u + v * v + w * w}$$

If $h1 \approx 0$ or $h2 \approx 0$, the branching pattern of the tree can be regarded as monopodial. Given the above rules, however, all the lateral branches develop near the same plane. To avoid this anomaly, we must introduce a divergence angle (Figure 4). The introduction of this divergence angle conforms to a notion in plant morphology called phyllotaxis.

Parametric pattern analysis of GMT1

Phyllotaxis, axiality, and apical dominance. The following three terms used in plant morphology are introduced to either describe external appearance or justify the validity of the model:

- **phyllotaxis**—the arrangement of branches (or leaves) on the main axis (stem),
- **axiality**—the quantity that represents how clearly the main axis can be detected, and
- **apical dominance**—the quantity that represents external appearance of the tree.

First let's concentrate on phyllotaxis. With GMT1, phyllotaxis can be specified only when the left or right branching angle is zero. There is no reason, when simulating a real plant, to specify the divergence angle at random.

In nature, a plant branches spirally with its growth rotating around the main axis. The pattern is regular for each species. Branches of two-lobed plants have a divergence angle of about 144 degrees. This angle is 2/5 of

360 degrees, which means that the sixth branch emerging from the main axis has the same position as the first one, only higher on the main axis.

The divergence ratio for a given plant, as a division of 360, belongs to one of the following sequences: 1/2, 1/3, 2/5, 3/8, 5/13, 8/21, 13/34, For example, grasses have a 1/2 ratio, roses and chrysanthemums have 2/5, cherry trees have 3/8, willows have 5/13, pines and ginkgos have 13/34, and so on. The fractions of this sequence constitute a “Fibonacci sequence” whose ratio is 0.618, which, as it turns out, is the golden ratio. This fact was discovered at the end of the nineteenth century and is often called Schimper-Braun's law,¹⁶ as noted by R. V. Jean.

The most important characteristic of phyllotaxis is its recursiveness, which accounts for the inclusion of phyllotaxis as a parameter in GMT1. In other words, one parameter is enough to specify phyllotaxis, regardless of the age of the tree. Note that our original assumption about a divergence angle can be modified to apply also in the case $h1 \neq 0$ and $h2 \neq 0$. This is seen in the experiments in axiality (Figures 5e and 5f), performed to point out the importance of this parameter. Compare them with Figures 5c and 5d, which show the inhibition of a divergence angle when $h1 \neq 0$ and $h2 \neq 0$.

We can classify the general characteristics of axiality and apical dominance through the experiments that follow. Table 1 summarizes the experimental results.

In our experiments—as far as axiality is concerned—where we varied $|h1|$ or $|h2|$ (branching angles of GMT1) while keeping their sum constant, we found that the larger the value of ($|h1| - |h2|$), the clearer the main axis becomes. This is shown in Figure 5. In the contrasting

Table 1.
Experiments on the GMT1 model.

| Fig. | View | $h1, h2$ | $R1, R2$ | R_diam | div | g_level | Standpoint |
|--|------|-----------|------------|--------|-------|---------|--------------|
| [I] $R1/R2$ and $ h1 + h2 $ fixed, $ h1 $ and $ h2 $ varied. | | | | | | | |
| 5a | TOP | (35, -35) | (0.9, 0.7) | (0.7) | (140) | (9) | --- |
| 5b | FRO | (35, -35) | (0.9, 0.7) | (0.7) | (140) | (9) | --- |
| 5c&e | FRO | (20, -50) | (0.9, 0.7) | (0.7) | (140) | (9) | - |
| 5d&f | FRO | (10, -60) | (0.9, 0.7) | (0.7) | (140) | (9) | + |
| 5g | FRO | (0, -70) | (0.9, 0.7) | (0.7) | (140) | (9) | +++ |
| [II] $R1/R2$ and $ h1 $ fixed, $ h2 $ varied. | | | | | | | |
| 6a | FRO | (0, -25) | (0.9, 0.7) | (0.7) | (140) | (10) | Sharp cone |
| 6b | FRO | (0, -40) | (0.9, 0.7) | (0.7) | (140) | (10) | |
| 6c | FRO | (0, -55) | (0.9, 0.7) | (0.7) | (140) | (10) | |
| 6d | FRO | (0, -70) | (0.9, 0.7) | (0.7) | (140) | (10) | Flat cone |
| [III] $ h1 $ and $ h2 $ fixed, $R1$ and $R2$ varied. | | | | | | | |
| 7a | FRO | (0, -45) | (0.9, 0.5) | (0.7) | (140) | (10) | Normal cone |
| 7b | FRO | (0, -45) | (0.9, 0.7) | (0.7) | (140) | (10) | |
| 7c | FRO | (0, -45) | (0.9, 0.9) | (0.7) | (140) | (10) | |
| 7d | FRO | (0, -45) | (0.7, 0.9) | (0.7) | (140) | (10) | Reverse cone |

Note:

Fig. — Figure number

$h1$ — Left branching angle (degrees)

$h2$ — Right branching angle (degrees)

div — Divergence angle (degrees)

$R1$ — Contraction ratio of left branch

$R2$ — Contraction ratio of right branch

R_diam — Contraction ratio of diameter

g_level — Growth level

TOP — Top view

FRO — Front view

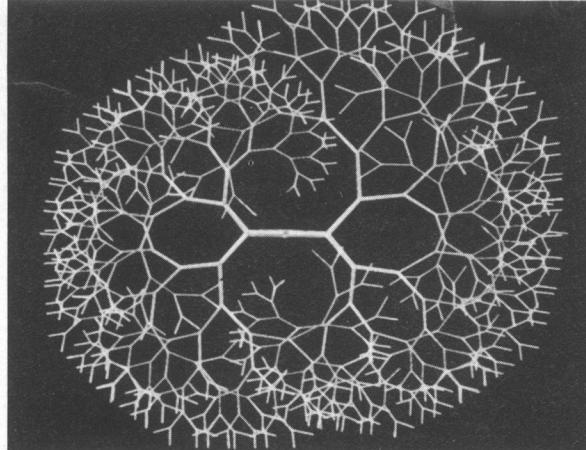


Figure 5a. $h_1 = 35^\circ$, $h_2 = -35^\circ$ (top view: pure dichotomous branching).

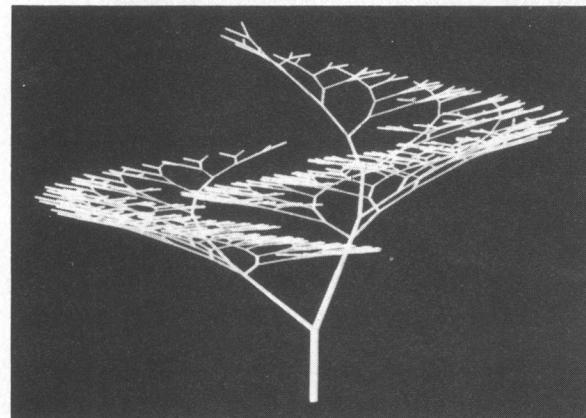


Figure 5c. $h_1 = 20^\circ$, $h_2 = -50^\circ$ (without a divergence angle).

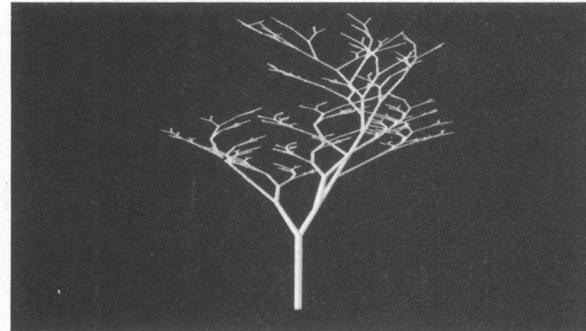


Figure 5e. $h_1 = 20^\circ$, $h_2 = -50^\circ$ (with a divergence angle).



Figure 5f. $h_1 = 10^\circ$, $h_2 = -60^\circ$ (with a divergence angle).

case, $|h_1| = |h_2|$ and the main axis is no longer noticeable (Figure 5a). This is a case of pure dichotomous branching.

Regarding apical dominance, in both experiments—with R_1/R_2 and $|h_1|$ fixed and with $|h_2|$ varied (Figure 6), and with $(|h_1| + |h_2|)$ fixed and with R_1/R_2 varied (Figure 7)—the contour changes from a tapering cone shape to a flat shape.

These experiments suggest that the absolute value of the branching angle is a key parameter in controlling both ax-

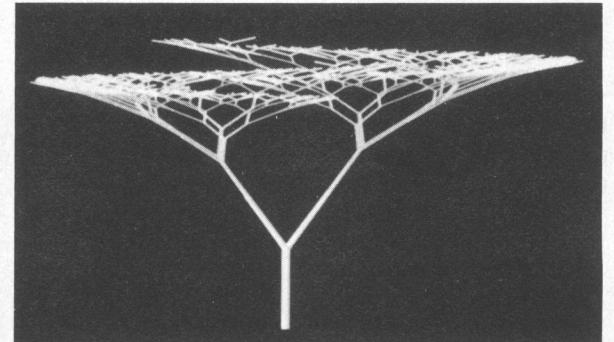


Figure 5b. $h_1 = 35^\circ$, $h_2 = -35^\circ$ (front view: pure dichotomous branching).

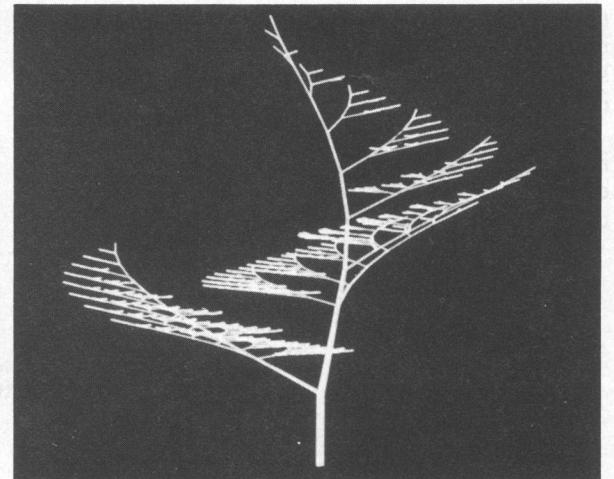


Figure 5d. $h_1 = 10^\circ$, $h_2 = -60^\circ$ (without a divergence angle).

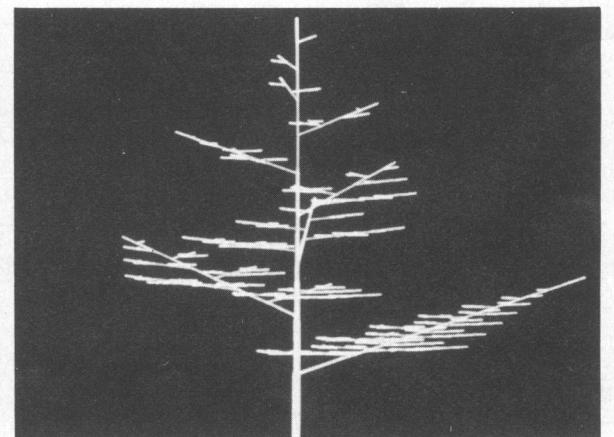


Figure 5g. $h_1 = 0^\circ$, $h_2 = -70^\circ$ (pure monopodial branching).

Figure 5. Botanical tree images generated by GMT1 with R_1/R_2 and $||h_1| + |h_2||$ fixed, $|h_1|$ and $|h_2|$ varied.

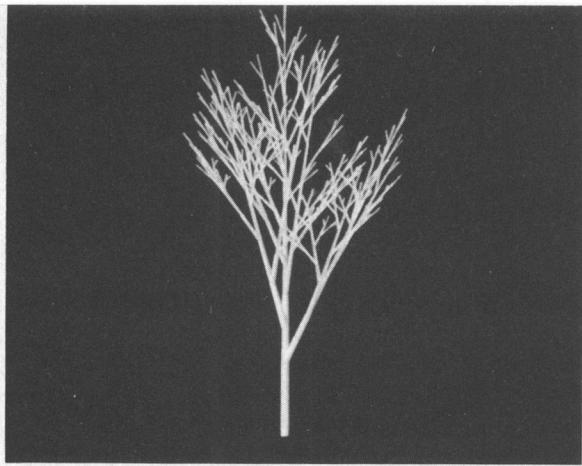


Figure 6a. $h1 = 0^\circ$, $h2 = -25^\circ$.

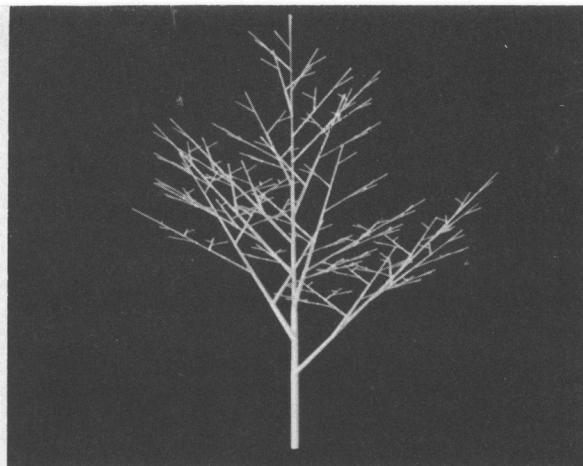


Figure 6b. $h1 = 0^\circ$, $h2 = -40^\circ$.

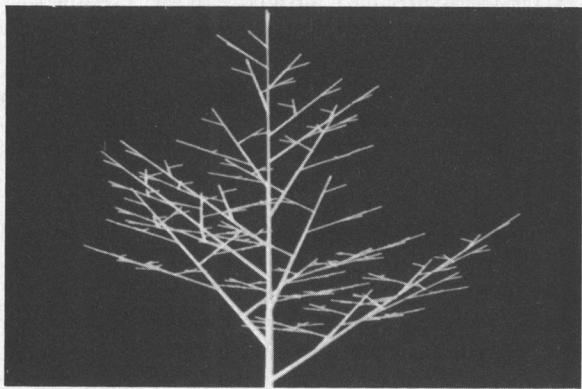


Figure 6c. $h1 = 0^\circ$, $h2 = -55^\circ$.

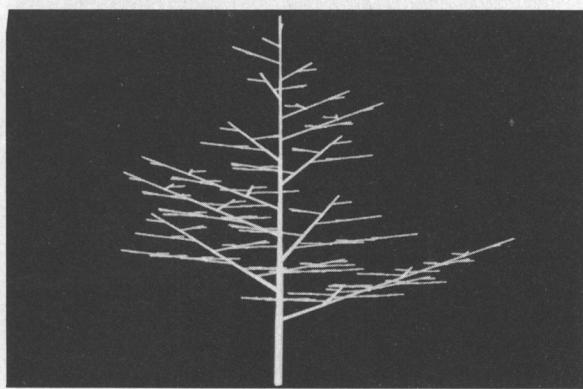


Figure 6d. $h1 = 0^\circ$, $h2 = -70^\circ$.

Figure 6. Botanical tree images generated by GMT1 with $R1/R2$ and $|h1|$ fixed, $|h2|$ varied.

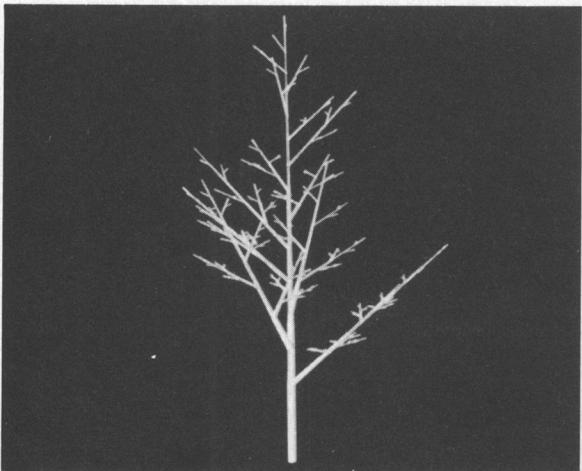


Figure 7a. $R1 = 0.9$, $R2 = 0.5$.

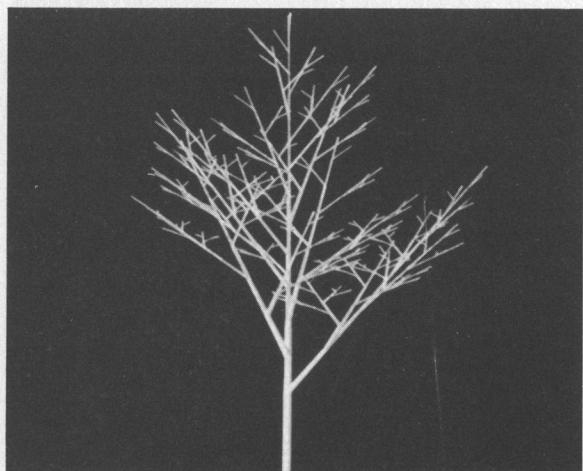


Figure 7b. $R1 = 0.9$, $R2 = 0.7$.

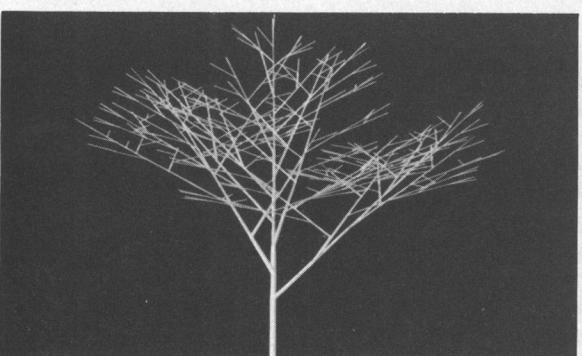


Figure 7c. $R1 = 0.9$, $R2 = 0.9$.

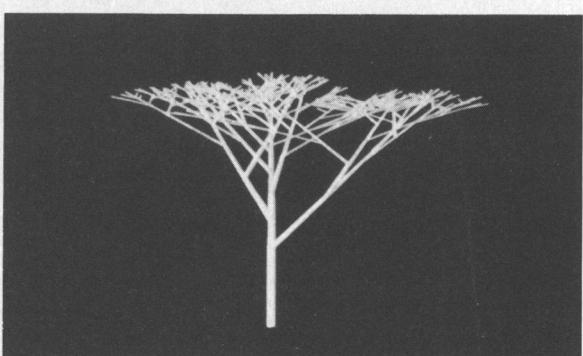


Figure 7d. $R1 = 0.7$, $R2 = 0.9$.

Figure 7. Botanical tree images generated by GMT1 with $|h1|$ and $|h2|$ fixed, $R1$ and $R2$ varied.

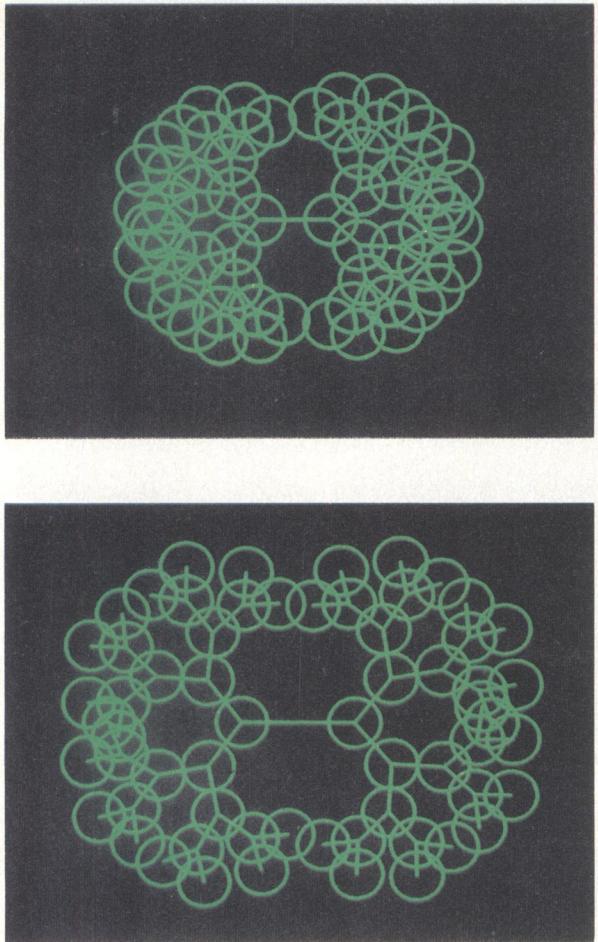


Figure 8. Approximation of leaves by a set of circles with the same radius. The top picture has smaller branching angles than the one below, and its overlapped area is larger.

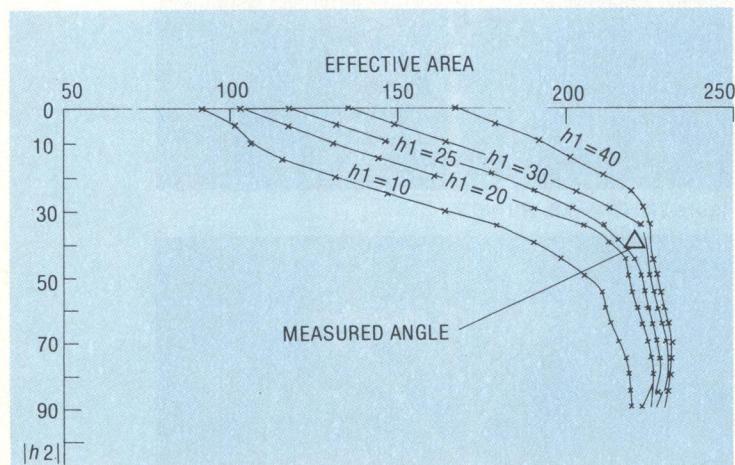


Figure 9. Comparison between the measured angle of *Aucuba japonica* and the theoretical angle from the standpoint of effective leaf area. The measured angle is near the maximum for the calculated effective area. For any representative plotting curves ($h_1 = 10, 20, 25, 30, 40$), there is a particular tendency: While $|h_2|$ is relatively small, the calculated effective area increases greatly with the growth of $|h_2|$, and after a certain threshold the ratio of the increase becomes small. The angle actually measured is located near the threshold.

iality and apical dominance. Generally speaking, as the angle becomes larger, the tree's horizontal growth is greater than its vertical growth; but as the angle becomes smaller, the tree grows more vertically. In practice, we often classify trees on the basis of their branching angles, albeit unconsciously.

Optimal branching pattern of a real plant. We have discussed and analyzed the branching patterns of trees by using two models, the L-system model and the GMT1 model. The teleological point of view will help explain why the patterns assume these branching angles (including divergence angle) and contraction ratios.

Many researchers have already proposed hypothetical branching patterns to optimize some values. One example is the ramification pattern of a blood vessel, which is so constituted that oxygen can be transported most efficiently.¹⁷ Another example, cited by Meinhardt,¹⁸ deals with the pattern of leaf veins, which are distributed so that the plant can transport nutrients as easily as possible. In reference to the tree branching pattern, B. B. Mandelbrot¹⁹ stated that, assuming the presence of self-similarity and space-filling characteristics, the contraction ratio becomes about $0.79 (1/\sqrt[3]{2})$, based on his fractal theory. J. B. Fisher,²⁰ who studied the tropical tree *Terminalia* in detail, based his study on maximal irradiation of its leaves by sunlight.

We experimented on the maximal sunlight theory, using the branching pattern model of the plant *Aucuba japonica*, which is familiar here in Japan. Leaves are approximated as a set of circles spread over a plane, and an efficient new algorithm is used for calculating the effective leaf area. The hypothesis is that the tree will exhibit a branching pattern that will allow its leaves to receive the greatest possible amount of sunlight.

The appearance of overlapped leaves in Figure 8 was obtained by approximating the leaves from the total area of circles. The net area of the leaves, excluding the overlaps, is the so-called "Dirichlet domain," formed by the center points of the circles.²¹

The following algorithms, from Honda and Fisher,²² can be used to calculate the Dirichlet domain of the circles: (1) Cut the net area with scissors and measure the weight, and (2) compute the actual Dirichlet domain. However, we have developed an efficient, simple approximation algorithm called the scan-line incremental method to calculate the net area.

Figure 9 shows the results of our experiments on *Aucuba japonica*. The branching pattern actually measured is almost the same as the optimal branching pattern for obtaining maximal sunlight.

There is a reason for the failure of the measured pattern to exactly match the optimal pattern. In its early stages, *Aucuba japonica* grows as tall as possible. It then branches out so that its child branches can obtain as much sunlight as possible. Another reason may be that after many branching processes, the plant still has to support its own weight; hence, from the mechanical standpoint, it cannot expand the branching area too far.

An assessment of GMT1 rules. On the basis of our experiments and study from the morphological standpoint, we can now assess our original rules individually.

Rule 1. GMT1 cannot deal with cases in which more than two child branches are generated from the mother branch.

Rule 2. It might be simpler to keep the branching angles and contraction ratios constant even though actual tree branching angles and their contraction ratios are not in any way constant. Every plant has a specific average angle and a specific average contraction ratio; thus, when actual values are substituted with the average, this rule can serve as a rough approximation for classifying trees. The main axis normally has the largest girth, and lateral branches are much smaller in girth than those derived from this rule. Girth depends, in fact, on the density of the tree's materials and seems to differ even within the same species growing at different places.

Rule 3. This rule does not hold for many kinds of trees, but for some specific species such as *Aucuba japonica* it is marvelously appropriate. For a more general model, child branch positions should deviate from the maximal gradient plane by δ around the axis of the mother branch, as illustrated in Figure 10. The initial branch and its two child branches do not necessarily lie on the plane parallel to the direction of gravity, since growth is often affected by wind, sunlight, or gravity. The main axis of some trees is not perpendicular to the ground. For example, trees along a river bank or on the side of a hill are often oriented toward open space or sunlight.

Rule 4. This rule cannot apply in the real world. For example, lateral branches near the ground are prone to drop, either for physiological reasons imposed by the tree itself or because of environmental factors such as being hidden from the sun or being buried under snow.

Improved geometric models

The original GMT1 model seems insufficient for generating natural tree shapes flexibly. We have introduced some improvements, and the results are depicted in Figures 11, 12, and 13.

Uniform and nonuniform deviation model (GMT2). The following are modifications to Rule 3 of GMT1.

(A) Taking into account the effects of wind, sunlight, and gravity, we find that child branch positions deviate uniformly by (dx, dy, dz) (uniform deviation).

(B) When we take into account point controllers (attractors and/or inhibitors) in space, we see that child branch positions deviate in proportion to the controllers' strength and in inverse proportion to the distance between the controllers and the original tip positions of the child branches. Whether a controller is an attractor or an inhibitor is determined by the sign of the strength; that is, if it is plus it is an attractor, otherwise it is an inhibitor (nonuniform deviation).

If we refer to the i th controller as a quadruple $\langle \text{factor}(i), qx(i), qy(i), qz(i) \rangle$, the new tip coordinate of a child branch at the j th growth level $(x'(j), y'(j), z'(j))$ is described as shown below, where

$\text{factor}(i)$ denotes the controller's strength, $(qx(i), qy(i), qz(i))$ is the controller's source coordinate, and $(x(j), y(j), z(j))$ is the original tip coordinate of the child branch at the j th growth level.

$$x'(j) = x(j) + dx + \sum \text{factor}(i) * (qx(i) - x(j)) / L(i)$$

$$y'(j) = y(j) + dy + \sum \text{factor}(i) * (qy(i) - y(j)) / L(i)$$

$$z'(j) = z(j) + dz + \sum \text{factor}(i) * (qz(i) - z(j)) / L(i)$$

where $L(i) =$

$$\sqrt{(qx(i) - x(j))^2 + (qy(i) - y(j))^2 + (qz(i) - z(j))^2}$$

Of course, if $(dx, dy, dz) = (0, 0, 0)$ and $\sum \text{factor}(i) = 0$, then GMT2 is exactly the same as GMT1. Note that we exclude the special case

$$(qx(i), qy(i), qz(i)) = (x(j), y(j), z(j))$$

Ternary branching model (GMT3). This modification to several rules of GMT1 deals with ternary ramifications. New rules for this model are as follows.

(A) At one branching process, three child branches are generated (ternary ramification).

(B) The contraction ratios of the three child branches are constant values $R0, R1$, and $R2$, where $R1$ and $R2$ are as in GMT1 and $R0$ denotes a center branch contraction ratio.

(C) The growth direction of the center branch is the same as its mother branch, and the other two are the same as before.

Statistical branching model (GMT4). Close observation of real plants may refine the original model. According to H. Hamano's²³ data for various trees, the probabilistic behavioral relationship between branching angles and the growth level, which serves as the index of tree ages, is roughly divided into five categories. Table 2 lists data gathered by Hamano on several actual trees, along with their categories. A general characteristic is that branching angles become smaller as the trees grow, regardless of their species. In other words, branches nearest the ground take the largest branching angles.

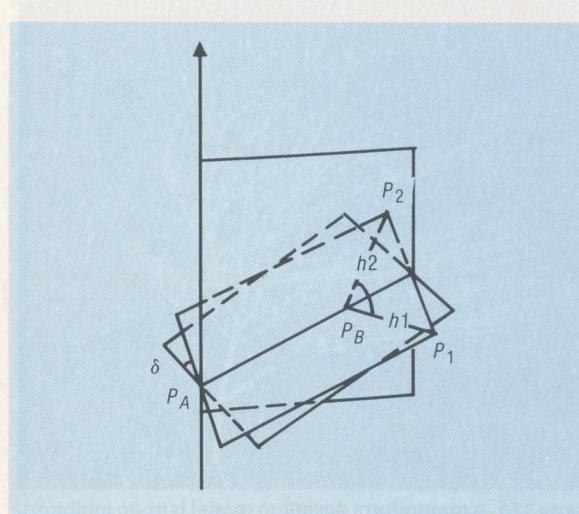


Figure 10. Child branch positions deviate from the maximal gradient plane by δ around the axis of the mother branch.

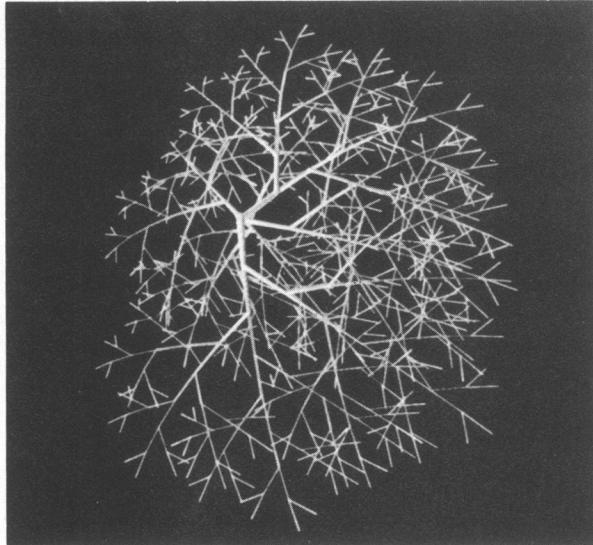


Figure 11a. A uniform deviation model (top view).

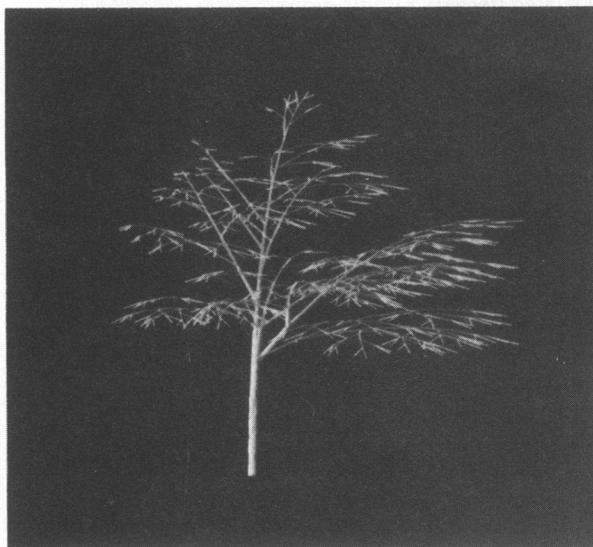


Figure 11b. A uniform deviation model (front view).

The following are five possible characteristics of branching angles:

- (1) They decrease at constant ratios from bottom to top with the growth level.
- (2) They decrease gradually until they reach the middle level, then they decrease sharply.
- (3) They decrease sharply until they reach the middle level, then they decrease gradually.
- (4) They decrease gradually until they reach the middle level, where they decrease sharply for a time, then they decrease gradually again.
- (5) They increase gradually until they reach the middle level, then they decrease.

Here we propose a tentative characteristic function for each category, as summarized in Table 3. When branching angles vary at constant ratios, we can easily formulate the characteristic function as a linear function. Otherwise, we have a choice of representing it by a polynomial function or by an analytic function (e.g., an exponential function).

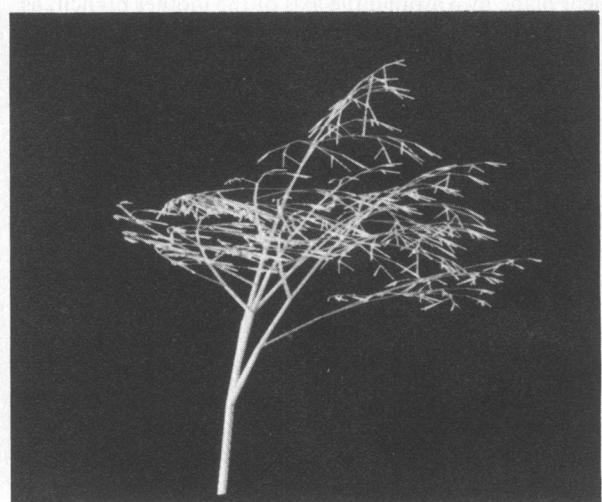


Figure 11c. A nonuniform deviation model (strong attractor).

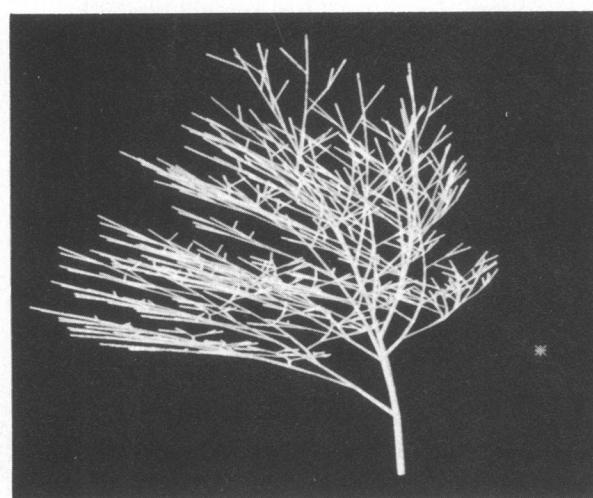


Figure 11d. A nonuniform deviation model (strong inhibitor).

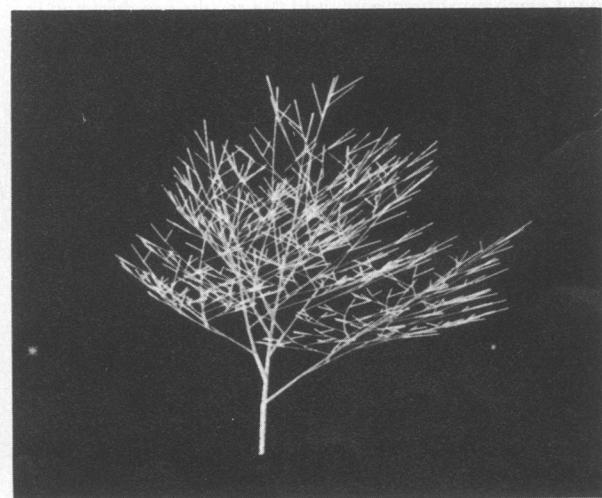


Figure 11e. A nonuniform deviation model (two inhibitors).

Figure 11. GMT2 examples. The star in Figure 11c represents an attractor. The stars in Figures 11d and 11e represent inhibitors. In each case, the size of the star corresponds to its strength.

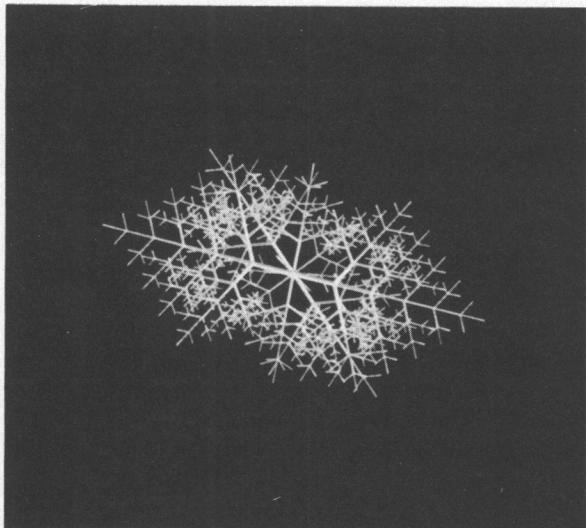


Figure 12a. A ternary branching model (top view: no deviation).

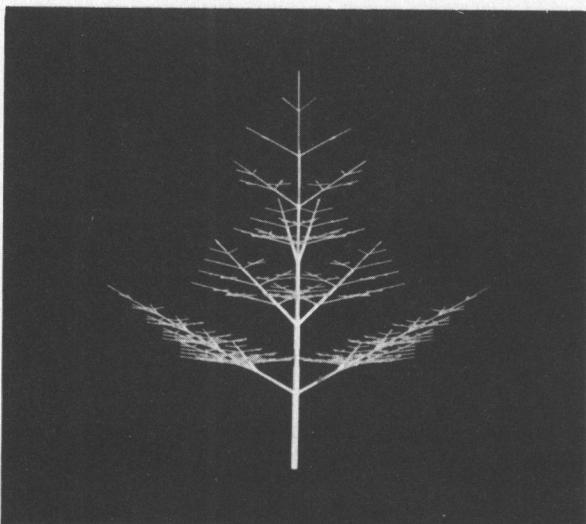


Figure 12b. A ternary branching model (front view: no deviation).

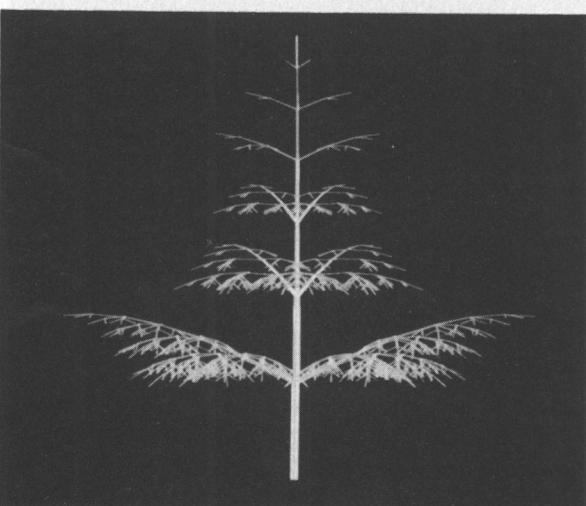


Figure 12c. A ternary branching model uniformly deviated toward the ground.

Figure 12. GMT3 examples combined with GMT2.

Table 4 tabulates our experiments with botanical tree generation using GMT2, GMT3, and GMT4. (See corresponding Figures 11, 12, and 13.) The examples show that GMT2 allows us to generate very realistic tree images. In particular, by utilizing the nonuniform deviators (at-

Table 2a.
Trees examined.

| No. | TREE NAME | SAMPLE CHECKED | TREE AGE | | |
|-----|--------------------------|----------------|----------|-------|-----|
| | | | Young | Adult | Old |
| 1 | Populus sp. Italiana. | Some | ☆ | ☆ | |
| 2 | Ginkgo bilbo | Many | | ☆ | ☆ |
| 3 | Cercidiphyllum japonicum | Some | ☆ | | |
| 4 | Osmanthus fragrans | Some | | | ☆ |
| 5 | Ilex rotunda | Two | | | ☆ |
| 6 | Chamaecyparis pisifera | Many | ☆ | ☆ | ☆ |
| 7 | Viburnum Awabuki | Some | | | ☆ |
| 8 | Betula platyphylla | Some | | | ☆ |
| 9 | Magnolia grandiflora | Some | | | ☆ |
| 10 | Aesculus turbinata | Some | | ☆ | |
| 11 | Styrax Obassia | Some | | | ☆ |
| 12 | Cornus controversa | Some | | | ☆ |
| 13 | Cedrus Deodara | Many | | ☆ | ☆ |
| 14 | Liquidambar formosana | Some | | | ☆ |
| 15 | Lithocarpus edulis | Some | | | ☆ |
| 16 | Metasequoia sp. | Many | | ☆ | |
| 17 | Myrica rubra | Some | | | ☆ |
| 18 | Liriodendron tulipifera | Some | | | ☆ |

Note: ☆ = Examined

Table 2b.
Data collected.

| TREE | AVER. | MAX. | MIN. | BOT. | TOP | HEIGHT | LATERAL | AGE | CATEGORY |
|------|-------|------|------|------|-----|--------|---------|-------|----------|
| 1 | 28 | 45 | 15 | 30 | 28 | 4.2 | 0.7 | Young | 1 |
| 1 | 30 | 50 | 15 | 31 | 20 | 13.2 | 3.3 | Adult | 1 |
| 2 | 53 | 86 | 35 | 73 | 45 | 11.8 | 9.1 | Adult | 3 |
| 2 | 58 | 85 | 40 | 65 | 59 | 6.8 | 2.9 | Old | 3 |
| 3 | 33 | 52 | 15 | 42 | 28 | 4.7 | 1.9 | Young | 1 |
| 4 | 55 | 78 | 30 | 65 | 43 | 3.1 | 2.2 | Old | 2 |
| 5 | 71 | 90 | 57 | 86 | 61 | 8.0 | 3.7 | Old | 1 |
| 6 | 63 | 81 | 45 | 53 | 54 | 4.4 | 1.7 | Young | 5 |
| 6 | 73 | 86 | 56 | 82 | 75 | 13.5 | 7.5 | Adult | 1 |
| 6 | 74 | 93 | 47 | 81 | 50 | 9.3 | 6.7 | Old | 1 |
| 7 | 55 | 83 | 30 | 73 | 44 | 5.0 | 3.5 | Old | 4 |
| 8 | 60 | 86 | 35 | 79 | 40 | 4.1 | 2.3 | Old | 1 |
| 9 | 51 | 72 | 28 | 57 | 46 | 6.7 | 4.9 | Old | 1 |
| 10 | 50 | 79 | 20 | 70 | 23 | 9.6 | 5.5 | Adult | 1 |
| 11 | 44 | 81 | 25 | 62 | 40 | 6.7 | 4.3 | Old | 3 |
| 12 | 60 | 88 | 40 | 74 | 44 | 3.9 | 3.9 | Old | 4 |
| 13 | 85 | 120 | 0 | 105 | 49 | 21.1 | 12.0 | Adult | 2 |
| 13 | 80 | 115 | 45 | 96 | 69 | 9.4 | 9.7 | Old | 1 |
| 14 | 64 | 99 | 25 | 88 | 30 | 5.3 | 3.0 | Old | 2 |
| 15 | 54 | 86 | 32 | 75 | 46 | 5.8 | 4.9 | Old | 1 |
| 16 | 53 | 90 | 11 | 67 | 42 | 13.6 | 8.5 | Adult | 1 |
| 17 | 60 | 85 | 30 | 74 | 51 | 5.0 | 4.4 | Old | 1 |
| 18 | 56 | 83 | 25 | 61 | 42 | 7.1 | 4.4 | Old | 2 |

Note:

TREE — Tree name (corresponding to the index of Table 2a)

AVER. — Average branching angle (degrees)

MAX. — Maximal branching angle (degrees)

MIN. — Minimal branching angle (degrees)

BOT. — Bottom branching angle (degrees)

TOP — Top branching angle (degrees)

HEIGHT — Height of the tree (meters)

LATERAL — Length of the lateral branch (meters)

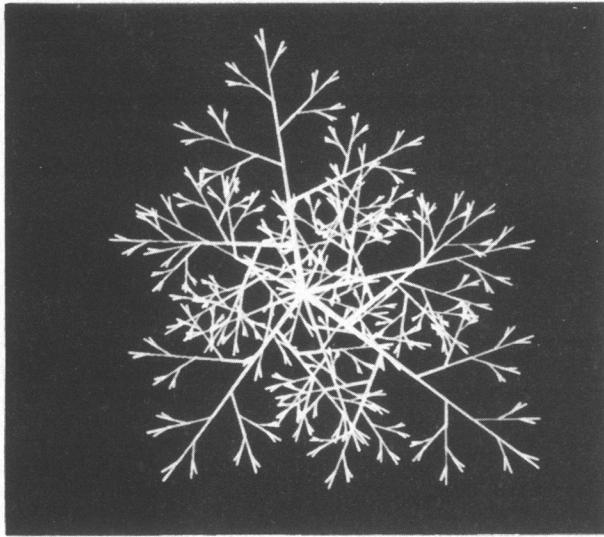


Figure 13a. Top view: category 1 in Table 3.

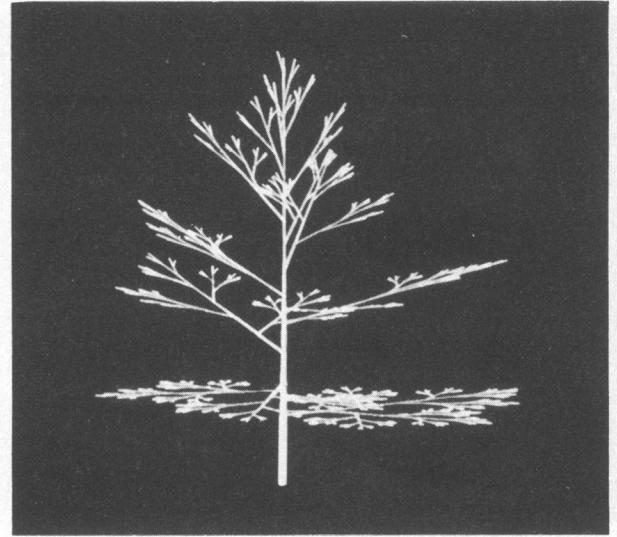


Figure 13d. Oblique view: category 3 in Table 3.

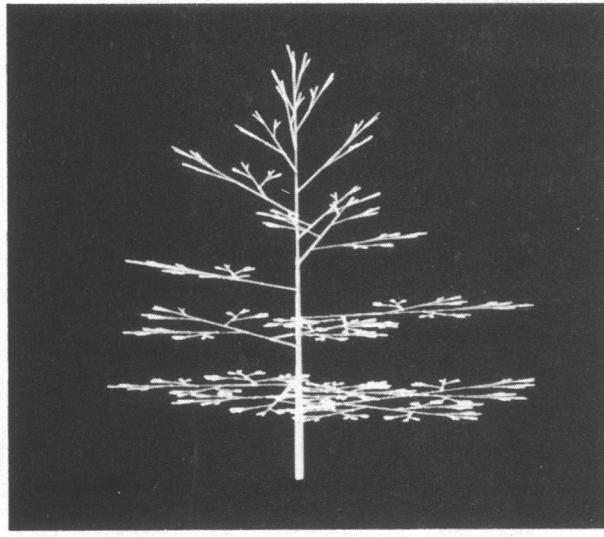


Figure 13b. Oblique view: category 1 in Table 3.

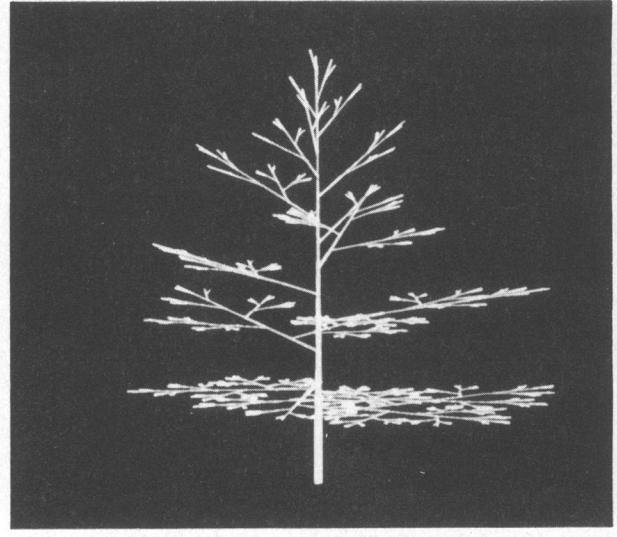


Figure 13e. Oblique view: category 4 in Table 3.

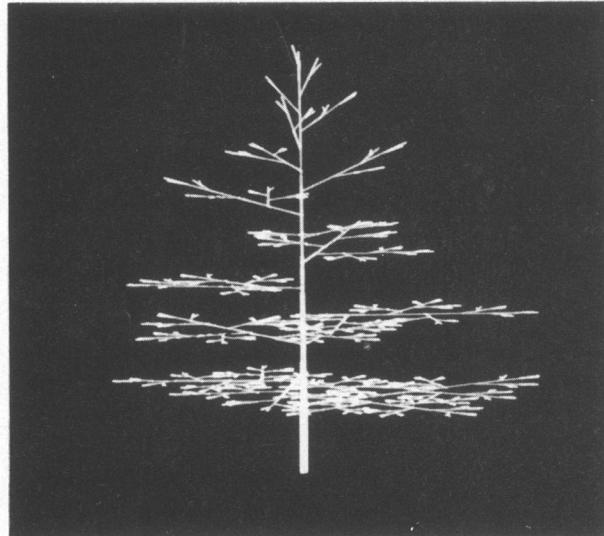


Figure 13c. Oblique view: category 2 in Table 3.

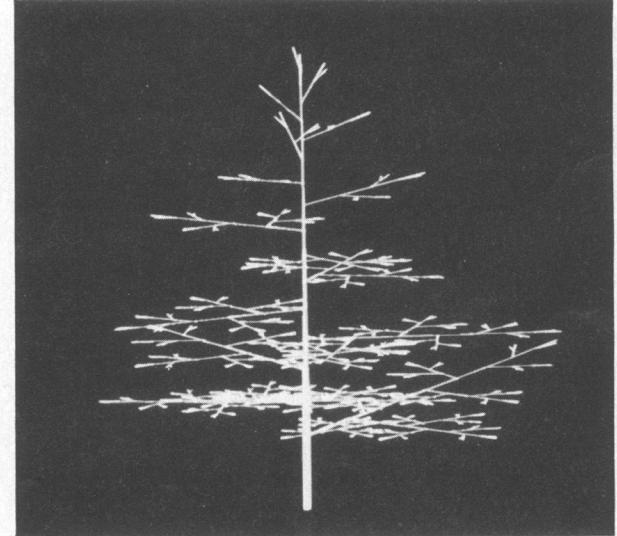


Figure 13f. Oblique view: category 5 in Table 3.

Figure 13. GMT4 examples.

Table 3.
Characteristic functions.

| CATEGORY | CHARACTERISTIC FUNCTION | CONDITION |
|----------|--|--|
| 1 | $\frac{\partial h}{\partial t} = -K$ | $K > 0$ |
| 2 | $\frac{\partial h}{\partial t} = -K*(t-n_1)$, $\frac{\partial h}{\partial t} = -K*(h-\varphi)$ | $n_1 = t_{\max}$, $K > 0$, $\varphi > 0$ |
| 3 | $\frac{\partial h}{\partial t} = K*(t-n_2)$, $\frac{\partial h}{\partial t} = K*h$ | $n_2 = t_{\min}$, $K > 0$ |
| 4 | $\frac{\partial h}{\partial t} = -K*(t^2 + n_3*t - n_4)$ | $K > 0$, $n_3, n_4 > 0$ |
| 5 | $\frac{\partial h}{\partial t} = -K*(t-n_5)$ | $K > 0$, $t_{\min} < n_5 < t_{\max}$ |

Note:

t = growth level of the tree ($t_{\min} < t < t_{\max}$)

h = branching angle ($h_{\min} < h < h_{\max}$)

tractors and/or inhibitors), we can bend the branches locally; that is, we can bend the branches at particular growth levels without affecting others, making it possible to interactively simulate an actual tree as closely as we like, placing the nonuniform deviators at various positions in space.

By combining GMT3 with GMT2, we can simulate an actual tree even more closely. The statistical model (GMT4) does not seem to add much more realism to the other models. In other words, the external appearance of the generated tree image is affected more by the absolute value of the branching angles than by their statistical variations.

Using the above modified models of GMT1 and experimental observations, we have attempted the reverse approach—fitting parameters to actual trees of our models.

Parameter fitting of geometric models to real trees

As a reverse process of image generation, we have examined some ways of fitting actual trees into our geometric models. We use statistical data and photographs of the original trees to extract their branching angles and the contraction ratios. Below we provide some typical examples of real trees for both dichotomous branching and monopodial branching. Note that the choice of geometric models (GMT1, GMT2, GMT3, and GMT4) to be applied

Table 4.
GMT2, GMT3, and GMT4 experiments.

GMT2 examples (uniform and nonuniform deviation model).

| Fig. | View | h_1, h_2 | R_1, R_2 | dx, dy, dz | $f_1 qx_1, qy_1, qz_1$ | $f_2 qx_2, qy_2, qz_2$ |
|------|------|------------|------------|------------------|------------------------|------------------------|
| 11a | TOP | (0, -45) | (0.9, 0.8) | (0.3, 0.3, -0.3) | | |
| 11b | FRO. | (0, -45) | (0.9, 0.8) | (0.3, 0.3, -0.3) | | |
| 11c | FRO. | (0, -45) | (0.9, 0.8) | | 1.6 (30, 0, 0) | |
| 11d | FRO. | (0, -45) | (0.9, 0.8) | | -1.0 (30, 0, 0) | |
| 11e | FRO. | (0, -45) | (0.9, 0.8) | | -0.6 (30, 0, 0) | -1.6 (-30, 15, 0) |

GMT3 examples (ternary branching model).

| Fig. | View | h_1, h_2 | R_0, R_1, R_2 | dx, dy, dz |
|------|------|------------|-----------------|------------------|
| 12a | TOP | (60, -60) | (0.9, 0.6, 0.6) | |
| 12b | FRO. | (60, -60) | (0.9, 0.6, 0.6) | |
| 12c | FRO. | (60, -60) | (0.9, 0.6, 0.6) | (0.0, 0.0, -0.6) |

GMT4 examples (statistical branching model).

| Fig. | View | h_1', h_2' | h_1'', h_2'' | R_1, R_2 | Category |
|------|------|--------------|----------------|------------|----------|
| 13a | TOP | (0, -90) | (0, -20) | (0.9, 0.7) | (1) |
| 13b | OBL. | (0, -90) | (0, -20) | (0.9, 0.7) | (1) |
| 13c | OBL. | (0, -90) | (0, -20) | (0.9, 0.7) | (2) |
| 13d | OBL. | (0, -90) | (0, -20) | (0.9, 0.7) | (3) |
| 13e | OBL. | (0, -90) | (0, -20) | (0.9, 0.7) | (4) |
| 13f | OBL. | (0, -90) | (0, -20) | (0.9, 0.7) | (5) |

Note:

TOP — Top view

OBL. — Oblique view

FRO. — Front view

R_0 — Contraction ratio of center branch

$|h_1'|$ — Maximal left branching angle (degrees)

$|h_2'|$ — Maximal right branching angle (degrees)

$|h_1''|$ — Minimal left branching angle (degrees)

$|h_2''|$ — Minimal right branching angle (degrees)

f_1, f_2 — Strength of a controller

In this table we assume that the contraction ratio of the diameter is 0.7 and the divergence angle is 140°.

to real trees depends on the data derived from the photographs of the trees.

Dichotomous branching. *Aucuba japonica*, popular in Japan, is an autogenous tree that originally grew in hilly districts. It is also planted in gardens. The tree grows about two meters high, and its branching pattern is generally regular. Figure 14 shows the average branching angles based on statistical data with an 80-percent confidence interval of the mean. Actual photographs are shown in Figure 15. The photo on the left presents the branching

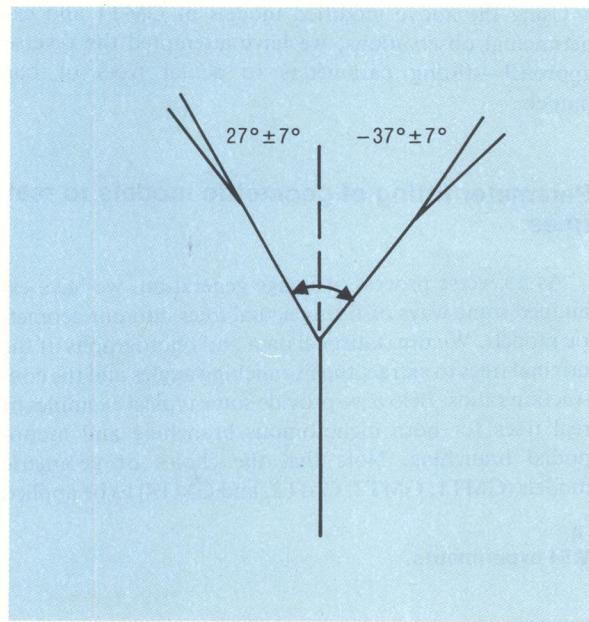


Figure 14. Average branching angle of *Aucuba japonica*.

angle at the first level and the photo on the right presents the branching angles at the second level. *Aucuba japonica* represents a typical dichotomous branching.

Statistical data showing the relation between branch lengths and the growth levels appear in Figure 16. The contraction ratio of *Aucuba japonica* is random and varies more at the early stages of its development than at the later stages. However, if we approximate the contraction ratio by a constant average value, we can calculate it as 0.786 plus or minus 0.25, with an 80-percent confidence interval of the mean.

The statistical data show the average maximal growth level of *Aucuba japonica* to be 6, and the average contraction ratio of its diameter is 0.7. The parameters of *Aucuba japonica* can be summarized as follows:

```
tree_type = GMT1 (bifurcation, no deviators)
maxlevel = 6
R_diameter = 0.7
h1 = 27 degrees (± 7 degrees)
h2 = -37 degrees (± 7 degrees)
R1 = 0.786 (± 0.25)
R2 = 0.786 (± 0.25)
```

The tree images generated by using the above parameters are shown in Figure 17.

Monopodial branching. With monopodial branching, we first describe how to make a general approximation of the parameters. We assume that the main axis of the tree is noticeable at any level of growth. Without loss of generality, the overview of the tree we are looking at is illustrated in Figure 18. In most cases, the branching angles can be easily determined from the photographs or from statistical data. Therefore, the problem is how to compute the contraction ratios of the child branches to their mother



Figure 15. Photo on the left shows first-level branching of *Aucuba japonica*; photo on the right shows second-level branching.

branch. For simplicity, let's suppose that the number of child branches generated by one branching is two (bifurcation) and the tree is not affected by deviators (environmental factors). Moreover, the symmetry of right and left branches is assumed to be normal.

As Figure 18 shows, here we assume the growth level of the tree to be n , the height of the tree to be H , the length from the first branching point to the top of the tree to be h , the length of the base section (from the root of the tree to the first branching point) to be H_0 , and the length from the first branching point to the farthest tip of the lateral child branches to be L . The following relations are easily obtained:

$$H = H_0 + h \quad (1)$$

$$h = H_0 * (R1 + R1 * R1 + \dots + R1^n) \quad (2)$$

$$L = H_0 * R2 * (1 + R1 + R1 * R1 + \dots + R1^{n-1}) \quad (3)$$

From Equations 2 and 3 we get

$$h:L = R1 * \frac{1 - R1^n}{1 - R1} : R2 * \frac{1 - R1^n}{1 - R1}$$

In particular, if n is big enough, from the fact that $0 < R1, R2 < 1$, $R1^n \approx 0$. Thus, $h:L \approx R1 : R2$. Here, from Equation 1,

$$H = H_0 * \frac{1}{1 - R1}$$

$R1$ and $R2$ are approximated roughly as follows:

$$R1 = 1 - \frac{H_0}{H}$$

$$R2 = \frac{L}{h} * R1$$

For the ternary branching model (GMT3), if we assume symmetry of left and right branches, $R0$, $R1$, and $R2$ are approximated as follows:

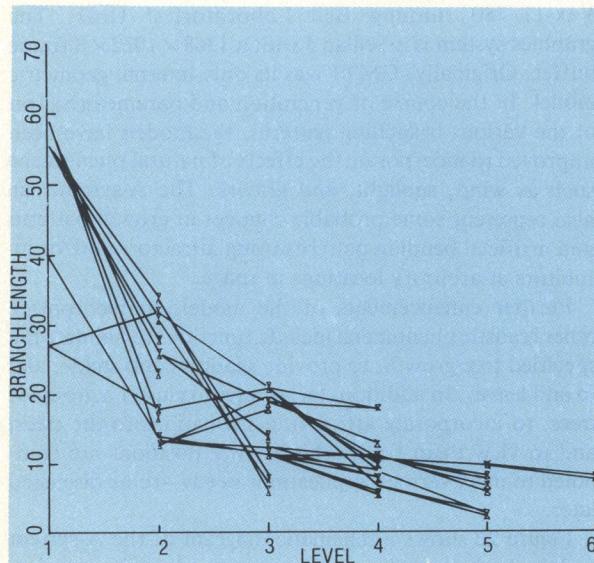


Figure 16. Relation between the length of each branch and the growth level, that is, the branching level of *Aucuba japonica*. The gradient of this graph represents the contraction ratio. A line segment between the adjacent levels corresponds to an actual branch. Typical branch length variation as well as the mean contraction ratio can be recognized even with this small data sample.

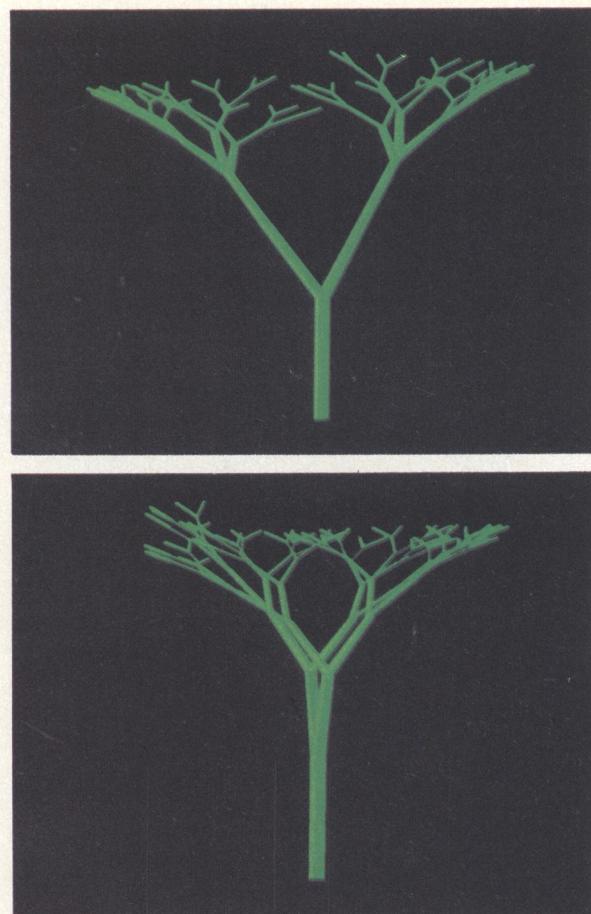


Figure 17. The parameter-fitted tree images of *Aucuba japonica*. Top: front view; bottom: side view. Compare these with Figure 15 left and right, respectively.

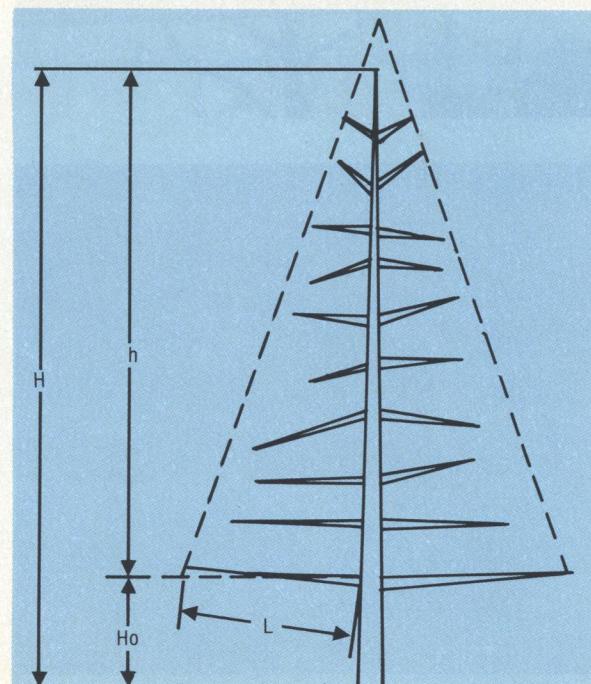


Figure 18. A typical tree representing monopodial branching.

$$R0 = 1 - \frac{H_0}{H}$$

$$R1 = R2 = \frac{L}{h} * R0$$

A young ginkgo tree can be used as an example for a GMT3 model. The branching angles can be approximated as 50 degrees, on the average. The ratios h/L and H_0/H are approximated as 0.75 and 0.174, respectively. Therefore, the parameters of *Ginkgo biloba* are

```
tree_type = GMT3 (ternary ramification, no
deviators)
maxlevel = 7
R_diameter = 0.7
h1 = +50 degrees
h2 = -50 degrees
R0 = 0.826
R1 = 0.620
R2 = 0.620
```

The generated tree image and the original photo are shown in Figure 19.

Examples of parameter fitting of GMT4 models can be obtained from the statistical data in Table 2. Fitting parameters of our models to real trees seems to work fairly well, even if we do not take into account such environmental effects as bending. If we try to fit trees that have been bent, we can interactively place a uniform deviator as well as nonuniform deviators (attractors and/or inhibitors) to increase realism. An example, a simulation of a pine tree using both uniform and nonuniform deviators, is shown in Figure 20 (top). To increase the realism, we put leaves on the tips of the branches (Figure 20, bottom).

We can also carry out parameter fitting by using a "glue" command in the A-system, which is outlined in the next section. Figure 21 shows a *Zelkova serrata*, a species of elm tree, and its simulation, generated by gluing a binary and a ternary tree.

Implementation of interactive tree generator (A-system)

On the basis of the work described so far, we have developed the A-system, an interactive synthetic system capable of manipulating the various trees already mentioned. It is implemented in the C language on a DEC Vax-11/780 running Bell Laboratories' Unix. The graphics system is a Seillac 3 with a $1368 \times 1022 \times 8$ frame buffer. Originally, GMT1 was its only internal geometric model. In the course of generation and parameterization of the various branching patterns, the models have been improved to incorporate the effects of natural phenomena such as wind, sunlight, and gravity. The A-system can also represent some probable changes in growth patterns and artificial bending patterns using attractors and/or inhibitors at arbitrary locations in space.

Further enhancements of the models to incorporate other realistic phenomena include functions to animate the specified tree growth, to provide shadows and shade, and to add leaves. In addition, techniques to cluster a group of trees, to incorporate affine transformations to the trees, and to view them from arbitrary eye positions are combined to meet various applications needs—to be discussed later.

Figure 22 shows a schematic diagram of the A-system and its basic operations. The command analyzer is implemented in both batch and interactive modes. The batch mode is intended for straight demonstration, while the interactive mode is intended for design and system enhancement. Our focus here is on the interactive mode. The full command syntax for the A-system appears in the Appendix on pages 32 and 33.



Figure 19. A young ginkgo tree (top) and its parameter-fitted tree (bottom).

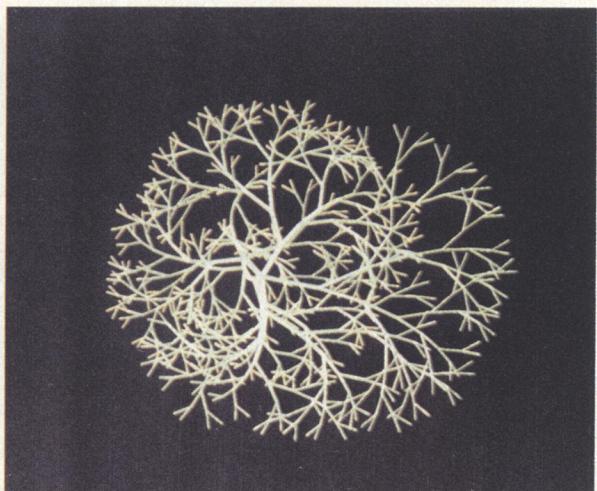


Figure 20. A uniform and nonuniform deviation model of a pine tree shown with and without leaves.

pdf | <http://www.cs.cmu.edu/~barney/pubs/1984/interaction.pdf> | <http://www.cs.cmu.edu/~barney/pubs/1984/interaction.html>

Internal data structure. The basic data structure of the A-system, coded in the C programming language, is shown in the box on page 27. The structure type *Tree* supports GMT1, GMT2, and GMT3 and is also extended to support GMT4, which generates probabilistic branching patterns. The initial state of the tree is maintained in the structure *Init*, which corresponds to a filamentous organism represented by a line segment. The structure *Vect* represents a uniform deviator and corresponds to the effects of wind or gravity. The structure *Ctrl* represents a nonuniform deviator. One *Tree* structure can have an arbitrary number of nonuniform deviators at arbitrary growth levels. A nonuniform deviator corresponds to an attractor or an inhibitor. When a tree is defined, the corresponding segment is initialized in the structure *Segment*, where transformations that occur afterward are stored. The class index, which makes it possible to cluster a group of *Trees*, is saved here as well. The structure *Emitter* represents a light source point in space and can specify the color of the shadow; the structure *Color* specifies a color for trees, leaves, and shadows.

Interaction handler. The A-system introduces the concept of a workstation based on the GKS^{24,25} and separate

Figure 21. *Zelkova serrata* (top) and its simulation, front view (middle) and top view (bottom).

physical devices from logical devices to achieve device-independence. The correspondence between physical and logical devices is given below; each logical device is represented by a physical device available for the A-system.

Pick device—light pen, joystick, tablet, digitizer

Keyboard device—keyboard

Locator device—joystick, tablet, digitizer

Valuator device—joystick, tablet

Display device—LCD monitor, display monitor, plotter

A general case of an interaction handling procedure might be

```
open(WORKSTATION,WORKSTATION_TYPE);
activate(WORKSTATION,LOGICAL_DEVICE1);

activate(WORKSTATION, LOGICAL_DEVICEn);
do {
    wait for user command input;
    switch(command) {
        case ...
        case ...
        ...
    }
} while (exit is not specified);
deactivate(WORKSTATION,LOGICAL_DEVICE1);

deactivate(WORKSTATION,LOGICAL_DEVICEn);
close(WORKSTATION);
```

As a final example, we will explain how a typical interaction algorithm works. We chose the algorithm for deforming a tree by specifying a controller interactively. It is cumbersome to input three-dimensional positions one at a time from the keyboard. This algorithm avoids the problem by using front and top views of a tree and by combining logical devices with an auxiliary line segment.

```
extern int maxlevel;
extern double R1, R2;
extern double h1, h2;
```

```
extern double factor;
extern struct INIT init;
deform(tree_segment_name)
NAME tree_segment_name;
{
    int i;
    double zoom_factor;
    double x,y,z;
    i = tree_search(tree_segment_name);
    tree_set(i,&init);
    controller_set(CTRL_SEGMENT,factor);
    evaluate_segment(TRANSLATION,CTRL_SEGMENT);
    report_position(&x,&y);
    evaluate_segment(ZOOM, CTRL_SEGMENT);
    get_segment_zoom(CTRL_SEGMENT,&zoom_factor);
    factor *= zoom_factor;
    line_set(LINE_SEGMENT,x,y);
    define_class(Temporary_CLASS,tree_segment_
        name,LINE_SEGMENT);
    looking_point(Temporary_CLASS,90,0,0);
    pick_segment(LINE_SEGMENT);
    report_position(&x,&z);
    delete_class(Temporary_CLASS);
    init_segment(tree_segment_name);
    draw_tree(maxlevel,R1,R2,h1,h2,init);
    register_tree(i,tree_segment_name,maxlevel,init);
```

First, the specified tree segment is retrieved from the tree database and the index is returned for later use. A standard-size controller (nonuniform deviator) is then displayed on the screen. At this point, the logical device *Valuator* is enabled to translate the controller. After the controller is located at the appropriate position on the screen (Figure 23a), *Valuator* is enabled to zoom it. Before this occurs, the controller's position is in the two-dimensional world of the *x-y* plane. After zooming, the tree is viewed from the top by rotating the view 90 degrees about the *x* axis, allowing the operator to specify the *z* position of the controller. Displaying a line segment that represents the *x-y* position determined earlier (Figure 23b) is helpful. By picking an arbitrary position of the line on the *z-x* plane, it is possible to determine the *z* position, resulting in precise specification of the three-dimensional position. The tree image is then drawn in its new shape (Figure 23c) and the basic tree information is saved.

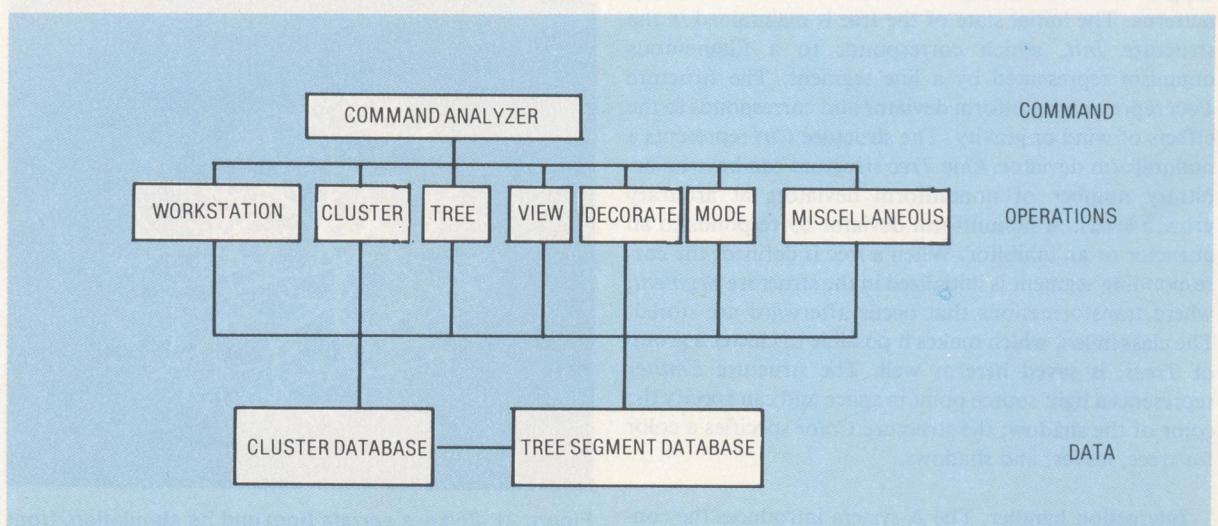


Figure 22. A-system architecture.

Applications

Parameter fitting (Figures 17, 19, 20, and 21), the introduction of leaves (Figures 20, bottom, and 26c), and shadowing (Figure 24) enhance the realism of the generated tree image. The ability to cluster a group of trees and

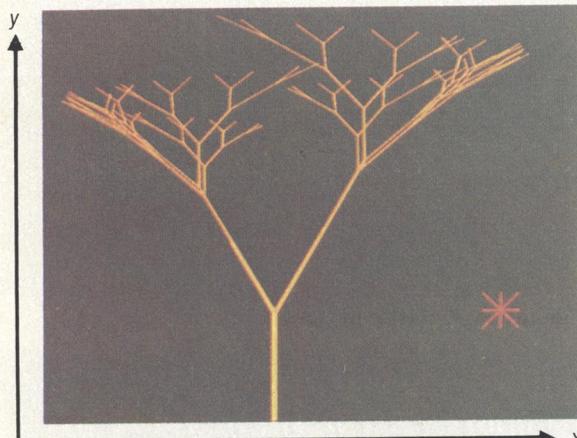


Figure 23a. An attractor (star) is located at the appropriate position ($x = a$, $y = b$).

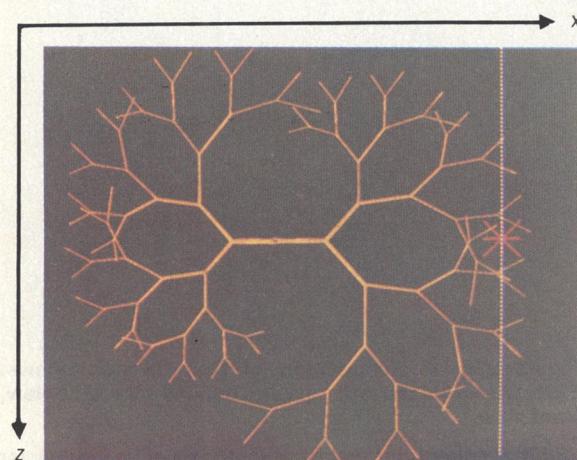


Figure 23b. The z position of the attractor ($z = c$) is determined by picking a point on a line segment ($x = a$, $y = b$), which represents the position specified in (a).

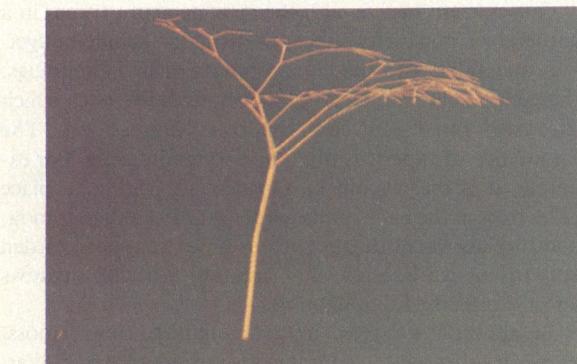


Figure 23c. The tree image, whose attractor is located at (a , b , c), is generated in its new shape.

Figure 23. An example of the interactive tree deformation algorithm.

A-system basic data structure

```

struct TREE {
    int tree_type;           /* inherent branching type of tree */
    int growth_level;        /* maximal growth level of tree */
    double R0;               /* center branch contraction ratio */
    double R1;               /* left branch contraction ratio */
    double R2;               /* right branch contraction ratio */
    double h1;               /* left branching angle */
    double h2;               /* right branching angle */
    struct INIT init;        /* initial filamentous organism */
    struct VECT vect;        /* uniform deviator vector */
    struct CTRL
        fact[FMAX];          /* nonuniform deviators */
    struct SEGMENT
        seg;                 /* transformation information */
};

struct INIT{
    double start_x1;         /* initial starting location (x) */
    double start_y1;         /* (y) */
    double start_z1;         /* (z) */
    double end_x1;           /* initial end location (x) */
    double end_y1;           /* (y) */
    double end_z1;           /* (z) */
};

struct VECT{
    double dx;               /* uniform deviator vector (x) */
    double dy;               /* (y) */
    double dz;               /* (z) */
};

struct CTRL{
    int lower_growth_level;  /* lower level affected */
    int upper_growth_level;  /* upper level affected */
    double factor;           /* strength of the controller */
    double source_x;          /* controller source (x) */
    double source_y;          /* (y) */
    double source_z;          /* (z) */
};

struct SEGMENT{
    double zoom;              /* zooming factor */
    double rotation;          /* rotation about the main axis */
    double trans_x;           /* translation along x-axis */
    double trans_y;           /* y-axis */
    double trans_z;           /* z-axis */
    int class;                /* class name of the tree segment */
};

struct EMITTER{
    double emit_x;            /* light source point (x) */
    double emit_y;            /* (y) */
    double emit_z;            /* (z) */
    struct COLOR
        shadow;               /* color of the shadow */
};

struct COLOR{
    int R;                   /* color contribution (red) */
    int G;                   /* (green) */
    int B;                   /* (blue) */
};

```

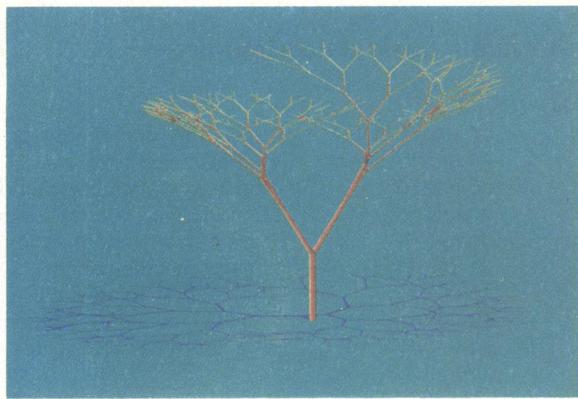


Figure 24. A pure dichotomous branching model with its shadow (oblique view).

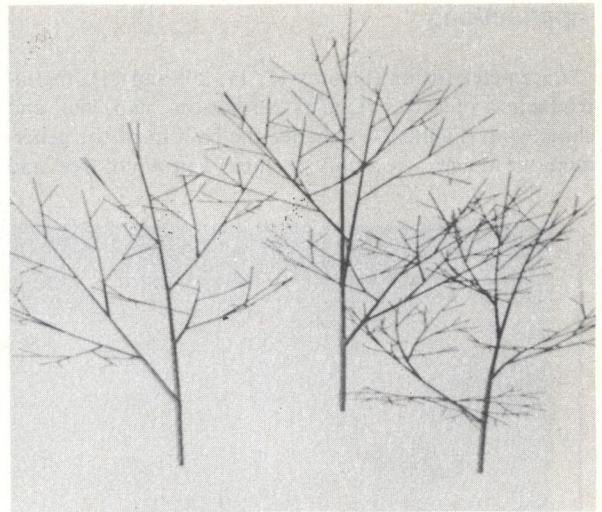


Figure 25a. A cluster of trees.



Figure 25b. Three clusters of trees.



Figure 25c. A composition of four different views. Clockwise from top left: top view, oblique view, side view, front view.

Figure 25. Clustering a group of trees and viewing trees from arbitrary directions make the synthesis of images easier.

to view trees from arbitrary directions makes the synthesis of images easier (Figure 25). These enhancements enable the A-system to be applied to many fields. Typical applications might include graphic design, the design of photo-gravures for magazines, or city planning (Figures 26a-f). In landscaping, the wrong trees are sometimes planted—trees that do not go well with the environment, or trees that grow so tall they block desired sunlight. The A-system can be used to make sure the right trees are planted.

Landscape gardening is interesting because of the novelty of introducing a CAD system into this type of work and because the so-called space planning problem cited by Eastman²⁶ can be applied to it.

A space planning problem consists of finding feasible or optimal solutions to spatial arrangement where performance is normally a function of distance, adjacencies, combined areas, and other complex arrangement parameters. Space planning is a key issue for solving problems through orthographic drawings. Examples of space planning problems include building floorplans, the spatial ar-

rangement of electronic circuits, site planning of a neighborhood, and the packing of equipment in a given space.

Our concern here is with the arrangement of trees in a garden or along a city street. Constraints include budgeting, adaptability to the circumstances (nearby buildings, the general climate, etc.), spatial boundaries over which the trees must not cast shadows, and so on. The knowledge of experts is often just common sense. For example, it is the custom in Japanese gardening to place taller trees at the back of the garden, far from the veranda, and to place small shrubs such as holly trees near garden lanterns or washbasins, thus creating artificial shadows and the effects of dim lighting.

In addition to trees, irregular-shaped stones, moss, grass, or other garden components are incorporated to arrive at a total gardening system.

We do not claim that the tree models of the A-system will represent all trees in Japan, much less on the earth.



Figure 26a. Simulation of bamboos.



Figure 26d. A simulation of Japanese maples.

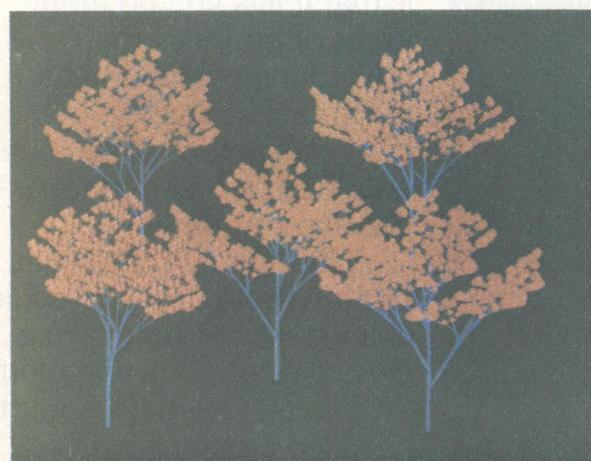


Figure 26b. A simulation of trees with tinted autumnal leaves.



Figure 26e. A simulation of weeping willows.

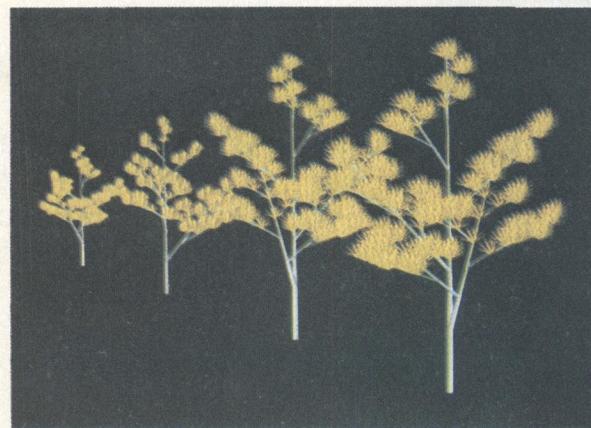


Figure 26c. A simulation of street plants.

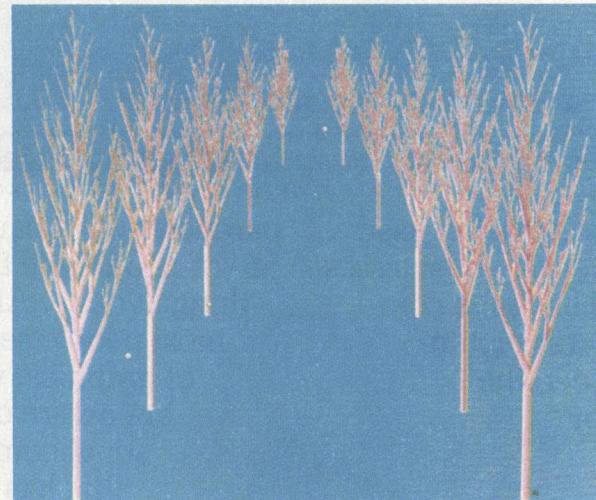


Figure 26f. A simulation of poplars along a street.

Figure 26. The generated tree images can be applied to graphic design, landscaping, and city planning.

Nevertheless, it is a handy, flexible, and extensible model that can generate natural 3-D tree images for many applications. ■

Acknowledgments

This work was partially supported by the Software Research Laboratory of Ricoh Company, Ltd. Hisao

Honda of the Kanebo Institute for Cancer Research supplied us with the reference on an optimal branching pattern for the tropical tree *Terminalia* on the basis of his geometric model. The reference on statistical data for GMT4 was provided by Shungo Kariyama of the Kurashiki Museum. Our thanks also go to the members of the Kunii Laboratory of Computer Science.

Appendix: A-system syntax

The syntax of the A-system is given below in the BNF notation. It has seven basic operations, as shown in Figure 22.

```

<OPERATIONS> ::= <TREE_OPERATION> | <CLUSTER_OPERATION> |
                  <VIEWING_OPERATION> | <TREE_MODE_SET_OPERATION> |
                  <TREE_DECORATION_OPERATION> | <WORKSTATION_SET_OPERATION> |
                  <MISCELLANEOUS_OPERATION>

<TREE_OPERATION> ::= <CREATE_TREE> | <DELETE_TREE> | <COPY_TREE> |
                     <TRANSFORM_TREE> | <ANIMATE_TREE> | <QUERY_TREE> |
                     <MERGE_TREE> | <COMBINE_TREE>

<CLUSTER_OPERATION> ::= <CLUSTER_TREE> | <REMOVE_CLUSTER> |
                         <APPEND_TREE> | <EXPTEL_TREE> | <FLUSH_CLUSTER> |
                         <QUERY_CLUSTER> | <MERGE_CLUSTER>

<VIEWING_OPERATION> ::= <COMPOSE_TREE_VIEW> | <VIEW_CLUSTER>

<TREE_MODE_SET_OPERATION> ::= <TREE_COLOR_SET> | <TREE_VISIBILITY_SET> |
                               <TREE_GIRTH_SET> | <CONTINUOUS_TRANSFORM_SET> |
                               <CUT_MAIN_AXIS_SET> | <CUT_LATERAL_BRANCH_SET> |
                               <DIVERGENCE_SET>

<TREE_DECORATING_OPERATION> ::= <SHAPE_DEFORMATION_STATEMENT> |
                                 <ATTACH_LEAF_STATEMENT> |
                                 <SHADOW_TREE_STATEMENT>

<WORKSTATION_SET_OPERATION> ::= <OPEN_WORKSTATION_STATEMENT> |
                                 <CLOSE_WORKSTATION_STATEMENT> |
                                 <WORKSTATION_VIEWPORT_STATEMENT> |
                                 <ENABLE_WORKSTATION_STATEMENT> |
                                 <DISABLE_WORKSTATION_STATEMENT> |
                                 <CLEAR_WORKSTATION_STATEMENT> |
                                 <WORKSTATION_WINDOW_STATEMENT>

<MISCELLANEOUS_OPERATION> ::= <BACKGROUND_OPERATION> | <EXIT_STATEMENT> |
                               <LOCAL_INTERACTION_AND_EXIT_STATEMENT> |
                               <DEFER_TIME_STATEMENT> | <HELP_STATEMENT>

<TRANSFORM_TREE> ::= <TRANSLATE_TREE_ABSOLUTE> | <TRANSLATE_TREE_RELATIVE> |
                      <ZOOM_TREE_ABSOLUTE> | <ZOOM_TREE_RELATIVE> |
                      <ROTATE_TREE_ABSOLUTE> | <ROTATE_TREE_RELATIVE> |
                      <RELEASE_TREE_TRANSFORM>

<CREATE_TREE> ::= create <tree_type> <tree_segment_name> <growth_level>
                  <tree_descriptor>;

<COMBINE_TREE> ::= combine <tree_segment_name> <growth_level> { { <tree_type>
                  <level_range> <tree_descriptor> } . . . };

<CUT_MAIN_AXIS_SET> ::= cut_axis <cut_flag> { <cut_level> };

<CUT_LATERAL_BRANCH_SET> ::= cut_lateral <cut_flag> { <cut_level>
                  { { <level_max> } . . . } };

<DIVERGENCE_SET> ::= divergence <divergence_angle>;

<SHAPE_DEFORMATION_STATEMENT> ::= deform <tree_segment_name> <uniform_deviator_
                                    _mask> <controller_mask> <starting_locator_
                                    _mask> { <deviator> } { <controller> . . .
                                    { <starting_locator> <additive_initial_length> } };

<ATTACH_LEAF_STATEMENT> ::= leaf <tree_segment_name> <tree_type> <leaf_type>
                            <leaf_level> <leaf_color> { <leaf_type_dependent_specification> };

<SHADOW_LEAF_STATEMENT> ::= shadow <tree_segment_name> <tree_type>
                            <emitter_x> <emitter_y> <emitter_z> <shadow_color>
                            { <merge_flag> };

<MERGE_TREE> ::= glue <tree_segment_name1> <tree_segment_name2> <tree_segment_name3>;
<MERGE_CLUSTER> ::= merge <cluster_name1> <cluster_name2> <cluster_name3>;
<BACKGROUND_COLOR_SET> ::= back_color <back_color>;
<EXIT_STATEMENT> ::= exit;
<LOCAL_INTERACTION_AND_EXIT_STATEMENT> ::= local;
<DEFER_TIME_STATEMENT> ::= defer <time_lag>;
<HELP_STATEMENT> ::= help { <command_name> };
<OPEN_WORKSTATION_STATEMENT> ::= open <workstation_name> <workstation_type>;
<CLOSE_WORKSTATION_STATEMENT> ::= close <workstation_name>;
<WORKSTATION_VIEWPORT_STATEMENT> ::= port <workstation_name> <view_type>
                                         <port_left> <port_right> <port_top>
                                         <port_bottom>;
<WORKSTATION_WINDOW_STATEMENT> ::= window <workstation_name> <window_left>
                                         <window_right> <window_top> <window_bottom>;

```

```

<ENABLE_WORKSTATION_STATEMENT> ::= activate <workstation_name> <logical_device_name>;
<DISABLE_WORKSTATION_STATEMENT> ::= deactivate <workstation_name> <logical_device_name>;
<ANIMATE_TREE> ::= animate <tree_type> <tree_segment_name> <maximum_growth_level> { <time_lag> };
<CLUSTER TREES> ::= cluster <cluster_name> <tree_segment_name> { <tree_segment_name> } . . . ;
<DELETE_TREE> ::= delete <tree_segment_name>;
<REMOVE_CLUSTER> ::= remove <cluster_name>;
<APPEND_TREE> ::= append <cluster_name> <tree_segment_name>;
<EXPEL_TREE> ::= expel <cluster_name> <tree_segment_name>;
<COPY_TREE> ::= copy <tree_segment_name1> <tree_segment_name2>;
<VIEW_CLUSTER> ::= view <cluster_name> { <viewing_type> } { <viewing_point> };
<COMPOSE_TREE_VIEW> ::= compose <tree_segment_name> { <viewing_type> }
                           { <viewing_point> };
<QUERY_TREE> ::= inform <tree_segment_name>;
<QUERY_CLUSTER> ::= inquire <cluster_name>;
<TRANSLATE_TREE_ABSOLUTE> ::= abs_translate <tree_segment_name> <translate_x>
                               <translate_y> <translate_z>;
<TRANSLATE_TREE_RELATIVE> ::= rel_translate <tree_segment_name> <translate_x>
                               <translate_y> <translate_z>;
<ZOOM_TREE_ABSOLUTE> ::= abs_zoom <tree_segment_name> <zoom_factor>;
<ZOOM_TREE_RELATIVE> ::= rel_zoom <tree_segment_name> <zoom_factor>;
<ROTATE_TREE_ABSOLUTE> ::= abs_rotate <tree_segment_name> <rotate_angle>;
<ROTATE_TREE_RELATIVE> ::= rel_rotate <tree_segment_name> <rotate_angle>;
<FLUSH_CLUSTER> ::= flush <cluster_name> { <time_lag> };
<TREE_COLOR_SET> ::= color { <tree_segment_name> } <color_specification>;
<TREE_VISIBILITY_SET> ::= <TREE_APPEARANCE> | <CLUSTER_APPEARANCE>;
<TREE_APPEARANCE> ::= appear <tree_segment_name> <visibility_flag>;
<CLUSTER_APPEARANCE> ::= visible <cluster_name> <visibility_flag>;
<TREE_GIRTH_SET> ::= girth { <tree_segment_name> } <girth_flag>
                      { <init_girth> <girth_ratio> <girth_dense> <shade_flag>
                        <main_axis_flag> <girth_circ> };
<CONTINUOUS_TRANSFORM_SET> ::= cont <cont_flag> { <step_divider> };
<CLEAR_WORKSTATION> ::= clear <workstation_name>;

```

References

1. A. Fournier, D. Fussel, and L. Carpenter, "Computer Rendering of Stochastic Models," *Comm. ACM*, Vol. 25, No. 6, June 1982, pp. 371-384.
2. B. B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman and Company, 1982.
3. C. Csuri, R. Hackathorn, R. Parent, W. Carlson, and M. Howard, "Towards an Interactive High Visual Complexity Animation System," *Computer Graphics* (Proc. Siggraph '79), Vol. 13, No. 2, Aug. 1979, pp. 289-299.
4. J. F. Blinn, "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces," *Computer Graphics* (Proc. Siggraph '82), Vol. 16, No. 3, July 1982, pp. 21-29.
5. W. T. Reeves, "Particle Systems—A Technique for Modeling a Class of Fuzzy Objects," *ACM Trans. Graphics*, Vol. 2, No. 2, Apr. 1983, pp. 91-108.
6. A. Lindenmayer, "Mathematical Models for Cellular Interactions in Development, I: Filaments with One-sided Inputs," *J. Theoretical Biology*, Vol. 18, 1968, pp. 280-299.
7. A. Lindenmayer, "Mathematical Models for Cellular Interactions in Development, II: Simple and Branching Filaments with Two-sided Inputs," *ibid.*, pp. 300-315.
8. N. Rashevsky, *Mathematical Biophysics*, Vol. 2, Dover Publications, Inc., 1960.
9. H. Honda, "Description of the Form of Trees by the Parameters of the Tree-like Body: Effects of the Branching Angle and the Branch Length on the Shape of the Tree-like Body," *J. Theoretical Biology*, Vol. 31, 1971, pp. 331-338.
10. T. A. McMahon, "The Mechanical Design of Trees," *Sci. Am.*, Vol. 233, 1975, pp. 93-102.
11. R. Marshall, R. Wilson, and W. Carlson, "Procedure Models for Generating Three-Dimensional Terrain," *Computer Graphics* (Proc. Siggraph '80), Vol. 14, No. 3, July 1980, pp. 154 -162.
12. E. Izuhara, "A Logical Notation for Natural Patterns," *Annals of Hokkaido Tokai University*, Vol. 1, 1980, pp. 1-10.
13. Y. Kawaguchi, "A Morphological Study of the Form of Nature," *Computer Graphics* (Proc. Siggraph '82), Vol. 16, No. 3, July 1982, pp. 223-232.
14. N. Chomsky, *Syntactic Structures*, Mouton, The Hague, 1957.
15. G. Rozenberg, "Theory of L Systems: From the Point of View of Formal Language Theory," in *L Systems*, Lecture Notes in Computer Science 15, Springer-Verlag, New York, 1974, pp. 1-23.
16. R. V. Jean, "Growth and Entropy: Phylogenesis in Phyllotaxis," *J. Theoretical Biology*, Vol. 71, 1978, pp. 639-660.
17. D'Arcy Thompson, *On Growth and Form*, abridged edition, Cambridge University Press, New York, 1961.
18. H. Meinhardt, *Models of Biological Pattern Formation*, Academic Press, New York, 1982.

19. B. B. Mandelbrot, "The Fractal Geometry of Trees and Other Natural Phenomena," in *Geometrical Probability and Biological Structures: Buffon's 200th Anniversary*, Lecture Notes in Biomathematics 23, Springer-Verlag, New York, 1978, pp. 235-249.
20. J. B. Fisher and H. Honda, "Branch Geometry and Effective Leaf Area: A Study of Terminalia Branching Pattern, I: Theoretical Trees," *American J. Botany*, Vol. 66, No. 6, 1979, pp. 633-644.
21. H. Honda, "Description of Cellular Patterns by Dirichlet Domains: The Two-Dimensional Case," *J. Theoretical Biology*, Vol. 72, 1978, pp. 523-543.
22. H. Honda and J. B. Fisher, "Tree Branch Angle: Maximizing Effective Leaf Area," *Science*, Vol. 199, Feb. 1978, pp. 888-890.
23. H. Hamano et al., "Preliminary Study on the Shape of Trees," *Proc. Autumnal Conf. Japanese Gardening Society*, 1982, pp. 38-39.
24. F. R. A. Hopgood, D. A. Duce, J. R. Gallop, and D. C. Sutcliffe, *Introduction to the Graphical Kernel System (GKS)*, Academic Press, New York, 1983.
25. G. Enderle, K. Kansy, and G. Pfaff, *Computer Graphics Programming: GKS—The Graphics Standard*, Springer-Verlag, New York, 1984.
26. C. M. Eastman, "Preliminary Report on a System for General Space Planning," *Comm. ACM*, Vol. 15, No. 2, Feb. 1972, pp. 76-87.



Masaki Aono is a member of the IBM Japan Science Institute. He has finished graduate research in computer graphics at the Department of Information Science, the University of Tokyo, and has received the BSc and MSc degrees in information science from the University of Tokyo, 1981 and 1984, respectively. He is a member of the ACM and the IEEE.

Toshiyasu L. Kunii is the guest editor of this special issue on computer graphics in Japan. His biography and photo appear on page 7.

The authors' address is Department of Information Science, Faculty of Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113.

Computer Graphics Market Research Reports

Frost & Sullivan has recently published analyses and forecasts of the following computer graphics industry segments:

- | | |
|---|---------------|
| <input type="checkbox"/> 1028 Computer Graphics in Manufacturing | Price \$1,300 |
| <input type="checkbox"/> 1030 Computer Graphics in the Graphic Arts Industry | Price \$1,350 |
| <input type="checkbox"/> 1029 Computer Graphics Workstation Market | Price \$1,275 |

For free descriptive literature, including a detailed table of contents, check the above reports of interest.

- Please call me. I have questions about these reports.**

Name & Title: _____

Company: _____

Address: _____

City: _____ State: _____ Zip: _____ Phone #: _____



Frost & Sullivan 106 Fulton Street New York, New York 10038 (212) 233-1080