Developmental Models of Herbaceous Plants for Computer Imagery Purposes

Przemyslaw Prusinkiewicz†, *Aristid Lindenmayer* ‡ and *James Hanan*†

† Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2

‡ Theoretical Biology Group
University of Utrecht
Padualaan 8, 3584 CH Utrecht, The Netherlands

ABSTRACT

In this paper we present a method for modeling herbaceous plants, suitable for generating realistic plant images and animating developmental processes. The idea is to achieve realism by simulating mechanisms which control plant growth in nature. The developmental approach to the modeling of plant architecture is extended to the modeling of leaves and flowers. The method is expressed using the formalism of L-systems.

CR Categories and Subject Descriptors: F.4.2 [Mathematical Logic and Formal Languages]: Grammars and Other Rewriting Systems: *Parallel rewriting systems.* I.3.5 **[Computer Graphics]**: Computational Geometry and Object Modeling: *Curve, surface, solid and object representation.* I.3.7 **[Computer Graphics]**: Three-Dimensional Graphics and Realism. J.3 **[Life and Medical Sciences]**: *Biology.*

Keywords: realistic image synthesis, L-system, parallel graph grammar, turtle geometry, developmental morphology and physiology of plants, scientific visualization.

1. INTRODUCTION.

In recent years, the modeling of plants has received considerable attention. The problem was approached from two directions. Kawaguchi [21], Aono and Kunii [2], Reeves and Blau [36], Bloomenthal [7] and Oppenheimer [31] defined branching structures primarily in geometrical terms, such as the lengths of branches and branching angles. Smith [39, 40], Prusinkiewicz [33, 34], Beyer and Friedel [6] and Eyrolles [10] concentrated on the specification of plant topology. In all cases, plants were defined by a small number of rules applied repetitively to produce complex structures. Some approaches made it possible to create forms which looked "younger" or "older", and even produce an impression of plant growth, as witnessed in the films of Aono and Kunii [3] and Smith [41]. However, the simulation of development was not a focal point of any of these methods.

We present a plant modeling method in which the simulation of development is the key to realism. Thus, in order to model a particular form, we attempt to capture the essence of the *developmental process* which leads to this form. The view that growth and form are interrelated has a long tradition in biology. D'Arcy Thompson [44] traces its origins to the late seventeenth century, and comments:

> The rate of growth deserves to be studied as a necessary preliminary to the theoretical study of form, and organic form itself is found, mathematically speaking, to be a function of time... We might call the form of an organism an *event in space-time*, and not merely a *configuration in space.*

This concept is echoed by Hallé, Oldeman and Tomlinson [16]:

> The idea of the form implicitly contains also the history of such a form.

The developmental approach to plant modeling has two distinctive features:

- **Emphasis on the space-time relation between plant parts.** In many plants, various developmental stages can be observed at the same time. For example, some flowers may still be in the bud stage, others may be fully developed, and still others may have been transformed into fruits. If the developmental technique is consistently used down to the level of individual organs, such *phase effects* are reproduced in a natural way.
- **Inherent capability of growth simulation.** The mathematical model can be used to generate biologically correct images of plants of different ages and to provide animated growth sequences.

We reenact plant development by simulating natural control mechanisms. Emphasis is put on the modeling and generation of growth sequences of *herbaceous* or *non-woody plants*, since the internal control mechanisms play a predominant role in their development. In contrast, the form of woody plants is determined to a large extent by the environment, competition between trees and tree branches, and accidents [47], which are unrelated to the mechanisms considered in this paper.

We express control mechanisms and simulate developmental processes using the formalism of L-systems [24]. In this sense, our approach to the modeling of plants has its origin in biological studies expressed in terms of L-systems [11-14, 20, 28]. Other approaches using L-systems for modeling purposes are also possible. For example, Hogeweg and Hesper [19] and Smith [40] searched a particular class of context-sensitive L-systems and selected those which generated interesting shapes.

2. BRANCHING STRUCTURES AND L-SYSTEMS.

2.1. Graph-theoretical and botanical trees.

In the context of plant modeling, the term "tree" must be carefully defined to avoid ambiguity. To this end, we introduce the notion of an axial tree (Fig. 1) which complements the graph-theoretic notion of a rooted tree [32] with the botanically motivated notion of branch axis.

A *rooted tree* has edges which are labeled and directed, and form paths from a distinguished node called the *root* or the *base* to the *terminal nodes*. In the biological context, these edges are referred to as *branch segments*. A segment followed by at least one more segment in some path is called an *internode*. A terminal segment (with no following edges) is called an *apex*.

An *axial tree* is a special type of rooted tree. At each of its nodes we distinguish at most one outgoing *straight* segment. All remaining edges are called *lateral* or *side* segments. Within an axial tree, a sequence of segments is called an *axis* if: (a) the first segment in the sequence originates at the root of the tree or as a lateral segment at some node, (b) each subsequent segment is a straight segment, and (c) the last segment is not followed by any straight segment in the tree. Together with all its descendants, an axis constitutes a *branch*. A branch is itself an axial tree.

Axes and branches are ordered. The axis originating at the root of the entire plant has order zero. An axis originating as a lateral segment of an *n*-order parent axis has order *n*+1. The order of a branch is equal to the order of its lowest-order or *main* axis. The terminal node of this axis is called the branch *top*.

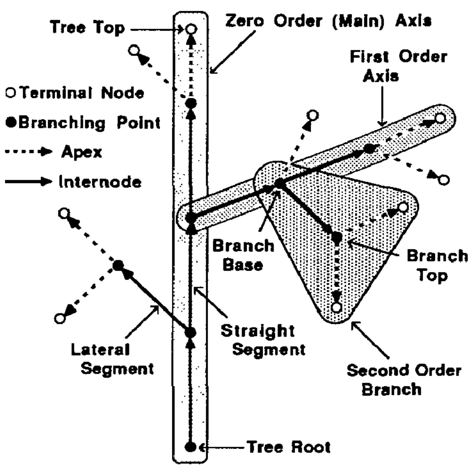

Figure 1. An axial tree.

Axial trees are purely topological objects. The geometric connotation of such terms as straight segment, lateral segment and axis should be viewed at this point as an intuitive link between the graph-theoretic formalism and real plant structures.

2.2. Definition of tree L-systems.

An essential aspect of plant development is the process in which some segments (usually the apices) are transformed into more complex structures. We model this process by a graph-rewriting mechanism which operates on axial trees. From the viewpoint of graph grammar theory, this is a special case of edge rewriting [15]. A rewriting rule, or *tree production*, replaces an edge, specified as the production *predecessor*, by an axial tree called the *successor*, in such a way that the starting node of the predecessor is identified with the successor's base and the ending node is identified with the successor's top (Fig. 2).

In the case of *context-free* rewriting the label of the replaced edge determines the production to be applied. In contrast, a *context-sensitive* production requires context, or the neighbour edges of the replaced edge, to be tested as well. Thus, a predecessor of a context-sensitive production p consists of three components: a path l called the *left context*, an edge S called the *strict predecessor*, and an axial tree r called the *right context* (Fig. 3). The asymmetry between the left context and the right context reflects the fact that there is only one path from the root of a tree to a given edge, while there can be many paths from this edge to various terminal nodes. Production p matches a given occurrence of the edge S in a tree T if l is a path in T terminating at the starting node of S, and r is a subtree of T originating at the ending node of S. The production can then be applied by replacing S with the axial tree specified as the production successor.

A rewriting system can operate either in a sequential or in a parallel manner. The former type of rewriting is found in Chomsky grammars. However, parallel rewriting is more appropriate for the modeling of biological development, since development takes place concurrently in all parts of the organism.

Parallel rewriting systems are commonly referred to as L-systems. Specifically, a *tree L-system G* is specified by three components: a set of edge labels called the *alphabet* and denoted by V, an axial tree ω with labels from V called the *axiom*, and a set of *tree productions P*. If for any edge label A and any context (l, r) there exists exactly one applicable production in P, the L-system is *deterministic*; otherwise it is *nondeterministic*. Nondeterministic L-systems provide a convenient tool for representing general features of a developmental process without considering mechanisms which control production selection (Section 4.3).

Given an L-system G, an axial tree T_2 is *directly derived* from (or *generated* by) a tree T_1, $T_1 \Rightarrow T_2$, if T_2 is obtained from T_1 by simultaneously replacing each edge in T_1 by its successor according to the production set P. A tree T is generated by an L-system G in a *derivation of length n* if there exists a *developmental sequence* of trees $T_0, T_1, \ldots, T_n$ such that $T_0 = \omega$, $T_n = T$ and $T_0 \Rightarrow T_1 \Rightarrow \cdots \Rightarrow T_n$ (see Section 2.4 for examples).

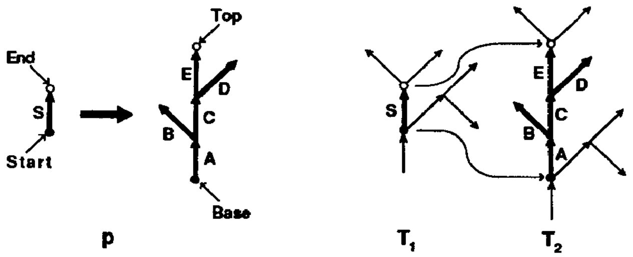

Figure 2. A tree production p and its application to the edge S in a tree T_1.

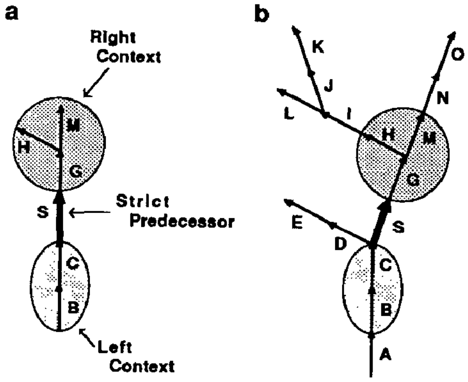

Figure 3. The predecessor of a context-sensitive production (a) matches the edge S in a tree T (b).

2.3. Representation of tree L-systems.

The definition of a tree L-system does not specify the data structure for representing axial trees. One possibility is to use a list representation with a tree topology. A different representation makes use of *bracketed strings* as introduced by Lindenmayer [24]. In this case, a tree with edge labels from alphabet V is represented by a string over alphabet $V \cup \{[,]\}$, where the bracket symbols [and] enclose branches. For example, the tree shown in Fig. 3b is represented by the bracketed string: .

$$ABC[DE][SG[HI[JK]L]MNO] \qquad (*)$$

A context-free production is denoted $A \rightarrow w$, where A belongs to V and w is a (possibly empty) bracketed string over V. A derivation step from string $x = a_1 a_2 \cdots a_n$ to string $y = w_1 w_2 \cdots w_n$ is performed by concatenating terms $w_1, w_2, \ldots, w_n$ obtained from productions with predecessors $a_1, a_2, \ldots, a_n$. The brackets are rewritten into themselves. In the case of a context-sensitive production, symbols $<$ and $>$ separate the strict predecessor from the left and right context, respectively. Since the string representation of axial trees does not preserve segment neighbourhood, the context matching procedure must skip over branches or branch portions when necessary. For example, a production with the predecessor $BC < S > G[H]M$ can be applied to symbol S in the string (*) (compare with Fig. 3).

2.4. L-systems and control mechanisms in plants.

The mechanisms which control plant development in nature can be divided into two classes, called *lineage* and *interactive* mechanisms. The term *lineage* refers to the transfer of genetic information from an ancestor cell to its descendants. *Interaction* is a mechanism in which information is exchanged between neighbouring cells (for example, in the form of nutrients or hormones). Within the formalism of L-systems, lineage mechanisms are represented by context-free productions, while interactive mechanisms correspond to context-sensitive productions. Two simple L-systems which simulate development controlled by lineage mechanisms are given below.

	L-system (a)		**L-system (b)**
ω:	S	ω:	A
p:	$S \rightarrow S[S]S[S]S$	p_1:	$A \rightarrow S[A]S[A]A$
		p_2:	$S \rightarrow SS$

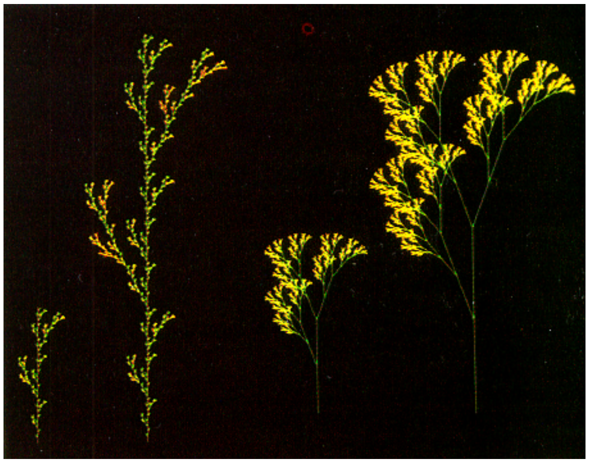

Figure 4. Structures which branch everywhere (left) and branching structures with a subapical growth pattern (right).

In case (a) all segments S branch. Only primitive organisms (for example, some bacteria and algae) develop this way. Herbaceous plants employ *subapical* growth mechanisms, in which new branches are created exclusively by apices. L-system (b) provides a simple example of such development. Production p_1 simulates the creation of new branches by apices A. Production p_2 simulates the gradual elongation of internodes, represented by sequences of symbols S. The resulting structures are shown in Fig. 4.

In the simulation of interaction between cells, the left context represents control signals which propagate *acropetally*, i.e. from the root or the basal leaves towards the apices of the modeled plant, while the right context represents signals which propagate *basipetally*, i.e. from the apices towards the root. The following L-systems simulate signal propagation in non-growing branching structures as illustrated in Fig. 5.

L-system (c)		**L-system (d)**	
ω:	$J[I]I[I]I[I]I$	ω:	$I[I]I[I]I[I]J$
p:	$J < I \rightarrow J$	p:	$I > J \rightarrow J$

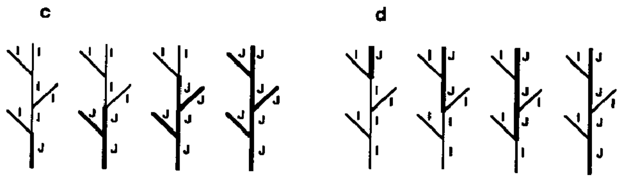

Figure 5. Acropetal (c) and basipetal (d) signal propagation.

The symbol J represents an internode already reached by the signal, while I represents an internode which has not yet been reached. In order to keep the specification of these (and subsequent) L-systems short, the following two conventions are observed: (1) if no production applies to a given symbol, this symbol is replaced by itself, and (2) if a context-free production and a context-sensitive production both apply to a given symbol, the context-sensitive production is chosen.

3. GEOMETRICAL INTERPRETATION OF AXIAL TREES.

The L-systems (a)-(d) considered above specify branching structures on a topological level. For the purpose of image synthesis, it is also necessary to specify geometric and graphical aspects of the modeled objects. Some previous approaches to the geometrical interpretation of L-systems are presented in [5, 17, 19]. Our approach was originally introduced to generate geometric patterns and fractals [43, 33] and was extended to describe three-dimensional plant structures in [34]. The method is as follows. After a string has been generated by an L-system, it is scanned from left to right and the consecutive symbols are interpreted as commands which maneuver a LOGO-like turtle in three dimensions [1]. The turtle is represented by its *state* which consists of turtle *position* and *orientation* in the Cartesian coordinate system, as well as other attribute values, such as current color and line width. The orientation is defined by three vectors $\vec{H}, \vec{L}, \vec{U}$, indicating the turtle's *heading* and the directions to the *left* and *up*. These vectors have unit length, are perpendicular to each other, and satisfy the equation $\vec{H} \times \vec{L} = \vec{U}$. Rotations of the turtle can then be expressed by the equation $[\, \vec{H}' \, \vec{L}' \, \vec{U}' \,] = [\, \vec{H} \; \vec{L} \; \vec{U} \,]$ **R**, where **R** is a 3×3 rotation matrix.

Segment symbols such as S, A, I and J in L-systems (a)-(d) move the turtle forward by a distance d and cause a line to be drawn between the previous and the new position. Seven attribute symbols are used to control turtle orientation given an angle increment δ. Symbols + and − *turn* the turtle left and right around the vector $\vec{U}$, ^ and & *pitch* the turtle up and down around the vector $\vec{L}$, and / and \ *roll* the turtle left and right around its own axis, the vector $\vec{H}$ (Fig. 6). The symbol | is used to turn the turtle 180° around the vector $\vec{U}$ regardless of the value of δ. Branches are created using a stack; [pushes the current state on the stack, while] pops a state from the stack and makes it the current state of the turtle. No line is drawn in this case, although the position of the turtle usually changes.

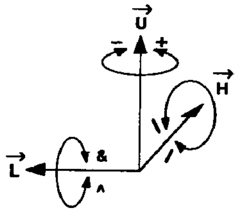

Figure 6. Turtle interpretation of geometric attribute symbols.

Figure 7. A bush.

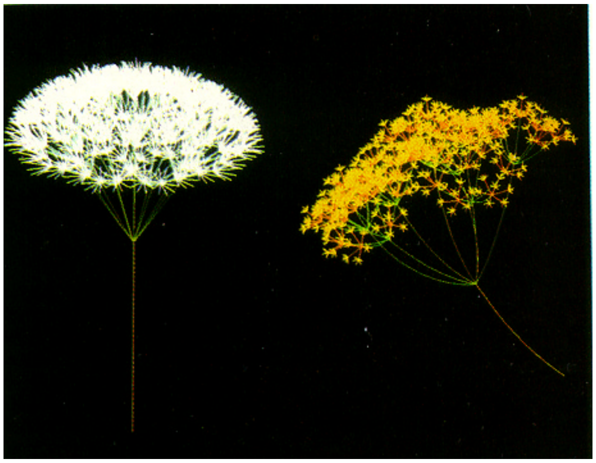

Figure 8. A comparison of branching structures modeled without tropism (left) and with tropism (right).

The list of attribute symbols can be augmented to control color, diameter and length of segments, incorporate predefined surfaces and objects in the model, and perform other functions as required. The extensions related to organ definition are discussed further in Section 6. Symbols without a specified interpretation are ignored by the turtle, which means that they can be used in the derivation process without affecting the interpretation of the resulting string.

Geometric extensions of L-systems (a) and (c) actually used to generate the left-hand structures in Figs. 4 and 5 are given below.

$$\begin{array}{llcll} & \textbf{L-system (a')} & & & \textbf{L-system (c')} \\ \omega: & S & & \omega: & J[+I]I[-I]I[+I]I \\ p: & S \rightarrow S[-'S]S[+'S]S & & p: & J < I \rightarrow J \end{array}$$

In case (a'), the edge length d is constant, the angle increment $\delta = 27.5°$, and the derivation lengths n are equal to 4 and 5. The attribute symbol $'$ increments the index to the color table. In case (c'), d is constant, $\delta = 45°$ and $n = 0–3$. The symbols + and – are ignored while context matching.

A more complex L-system generating the three-dimensional bush taken from [34] and shown in Fig. 7 is given below.

$$\begin{array}{ll} \omega: & //a \\ p_1: & a \rightarrow [[\&sl!a]/////'[\&sl!a]///////'[\&sl!a]] \\ p_2: & s \rightarrow Sl \\ p_3: & S \rightarrow S/////s \\ p_4: & l \rightarrow ['''\wedge\wedge\{-S+S+S-|-S+S+S\}] \end{array}$$

The attribute symbol ! decreases the diameter of segments S. The symbols a, s and l are not interpreted geometrically. The system operates as follows. Production p_1 creates three branches from an apex a. A branch consists of a stem s, a leaf l and an apex a which will subsequently create three new branches. Productions p_2 and p_3 specify the growth process of a stem; in subsequent derivation steps it gets longer by acquiring new segments S and produces new leaves l (in violation of the subapical growth rule, but with an acceptable visual effect in a still picture). Production p_4 describes the leaf as a filled polygon with six edges (see Section 6). More examples of completely specified L-systems which generate two-dimensional figures and three-dimensional objects are given in [33, 34, 35].

A characteristic feature of turtle interpretation is that directions are relative to the current orientation. However, absolute directions play an important role in the development of plants. For example, the axes may bend up towards the source of light, or down due to gravity. We simulate these effects by rotating the turtle slightly in the direction of a predefined *tropism vector* $\vec{T}$ after drawing each segment (Fig. 8). The angle α is calculated using the formula $\alpha = e\,\vec{H} \times \vec{T}$, where e is a parameter capturing axis susceptibility to bending. This heuristic formula has a physical motivation; if $\vec{T}$ is interpreted as a force applied to the endpoint of segment $\vec{H}$ and $\vec{H}$ can rotate around its starting point, the torque is equal to $\vec{H} \times \vec{T}$. A detailed analysis of tree dynamics for simulation purposes is presented in [4].

4. DEVELOPMENTAL MODELS OF PLANT ARCHITECTURE.

In this section we use the formalism of L-systems to present developmental models of herbaceous plants on the topological level. The geometric aspects are discussed in sections 5 and 6. We put particular emphasis on the modeling of compound flowering structures or *inflorescences*. As there is no commonly accepted terminology referring to inflorescence types, we chose to follow the terminology of Müller-Doblies [29], which in turn is based on extensive work by Troll [45]. Our presentation is organized by the control mechanisms which govern inflorescence development.

4.1. Racemes, or the phase beauty of sequential growth.

The simplest possible flowering structures with multiple flowers are those with a single stem on which an indefinite number of flowers are produced sequentially. Inflorescences of this type are called *racemes*. Their development can be described by the following L-system:

$$\begin{array}{lll} \omega: & A & \\ p_1: & A \rightarrow I_0[I_0F_0]A & \\ p_2: & I_i \rightarrow I_{i+1} & i \geq 0 \\ p_3: & F_i \rightarrow F_{i+1} & i \geq 0 \end{array}$$

The symbol A denotes the apex of the main (zero-order) axis, I_i denotes the i-th stage of internode elongation, and F_i is the i-th stage of flower development. The indexed notation, such as $F_i \rightarrow F_{i+1}$, stands for a set of productions $F_0 \rightarrow F_1, F_1 \rightarrow F_2, F_2 \rightarrow F_3, \cdots$. The developmental sequence begins as follows:

$$\begin{array}{l} A \\ I_0[I_0F_0]A \\ I_1[I_1F_1]I_0[I_0F_0]A \\ I_2[I_2F_2]I_1[I_1F_1]I_0[I_0F_0]A \\ I_3[I_3F_3]I_2[I_2F_2]I_1[I_1F_1]I_0[I_0F_0]A \\ \ldots \end{array}$$

At each developmental stage, the inflorescence contains a sequence of flowers of different ages. The flowers newly created by the apex are delayed in their development with respect to the older ones situated at the stem base. This effect is illustrated in Fig. 9, to which the following quotation from d'Arcy Thompson [44] applies:

> A flowering spray of lily-of-the-valley exemplifies a growth-gradient, after a simple fashion of its own. Along the stalk the growth-rate falls away; the florets are of descending age, from flower to bud; their graded differences of age lead to an exquisite gradation of size and form; the time-interval between one and another, or the "space-time relation" between them all, gives a peculiar quality - we may call it phase-beauty - to the whole.

A similar phase effect can be observed in other plants. For example, consider the fern-like structure shown in Fig. 10. In this case, nine zero-order branches grow subapically and produce new first-order branches, which also grow subapically and produce leaves. These processes are described by the following L-system:

$$\begin{array}{lll} \omega: & [A][A][A][A][A][A][A][A][A] & \\ p_1: & A \rightarrow I_0[B]A & \\ p_2: & B \rightarrow I_0[L_0][L_0]B & \\ p_3: & I_i \rightarrow I_{i+1} & i \geq 0 \\ p_4: & L_i \rightarrow L_{i+1} & i \geq 0 \end{array}$$

A and B denote apices of zero-order and first-order axes, $I_0,I_1,I_2, \cdots$ denote the internodes, and $L_0,L_1,L_2, \cdots$ denote the subsequent stages of leaf development.

4.2. Cymose inflorescences, or the use of delays.

In racemes the apex of the main axis produces lateral branches and continues to grow. In contrast, the apex of the main axis in *cymes* turns to a flower shortly after a few lateral branches have been initiated. Their apices turn into flowers as well and second-order branches take over. In time, branches of higher and higher order are produced. Thus, the basic structure of a cymose inflorescence is captured in the production

$$A \rightarrow I[A][A]IF$$

According to this description, the two branches are identical and grow in concert. In reality, this need not be the case, and one lateral branch may start growing before the other. This effect can be modeled by assuming that apices undergo a sequence of state changes which delay their further growth until a particular state is reached. For example, the development of the rose campion (*Lychnis coronaria*) shown in Fig. 11 is described by the following L-system:

$$\begin{array}{lll} \omega: & A_7 & \\ p_1: & A_7 \rightarrow I_0[L_0][L_0][A_0][A_4]I_0F_0 & \\ p_2: & A_i \rightarrow A_{i+1} & 0 \leq i < 7 \\ p_3: & X_i \rightarrow X_{i+1} & i \geq 0,\ X \in \{I, L, F\} \end{array}$$

Production p_1 specifies that, at their creation time, the lateral apices have different states A_0 and A_4. Production p_2 advances the apex states. Thus, the first apex requires eight derivation steps to produce a flower and new branches, while the second requires only four steps. Concurrently internodes elongate, leaves grow and each flower undergoes a sequence of changes, progressing from the bud stage to an open flower to a fruit. These processes are captured in production p_3. For a further analysis of the above model see [37].

4.3. Modeling qualitative changes of developmental processes.

The developmental sequences considered so far are homogeneous in the sense that the same structure is produced repeatedly at fixed time intervals. However, in many cases a qualitative change in the nature of development can be observed at some point in time. For example, consider the shepherd's purse (*Capsella bursa-pastoris*) shown in Fig. 12.

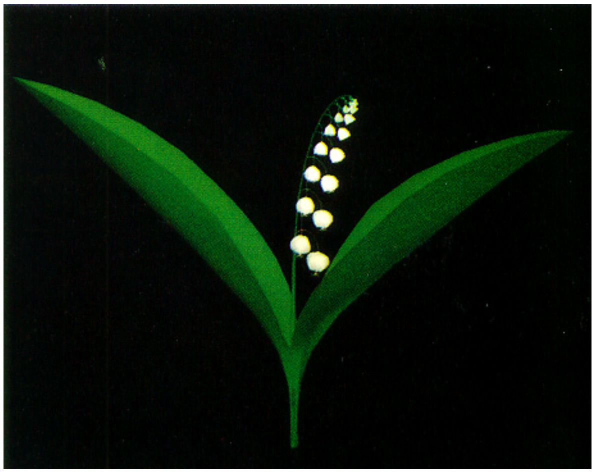

Figure 9. Lily-of-the-valley.

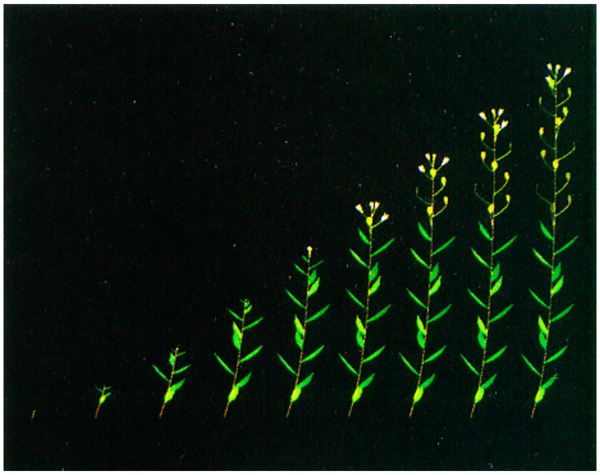

Figure 12. Development of a shepherd's purse.

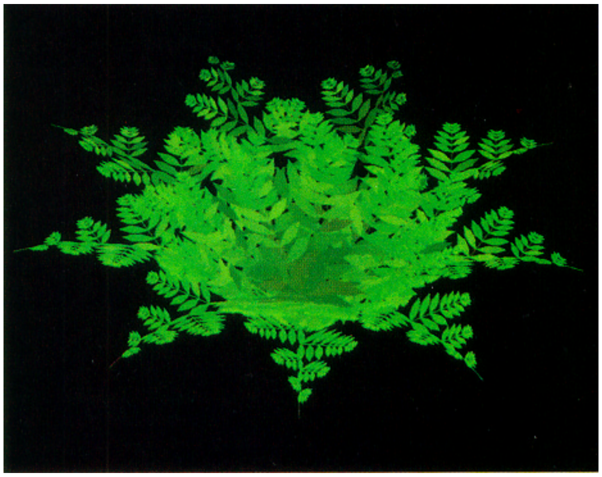

Figure 10. A fern.

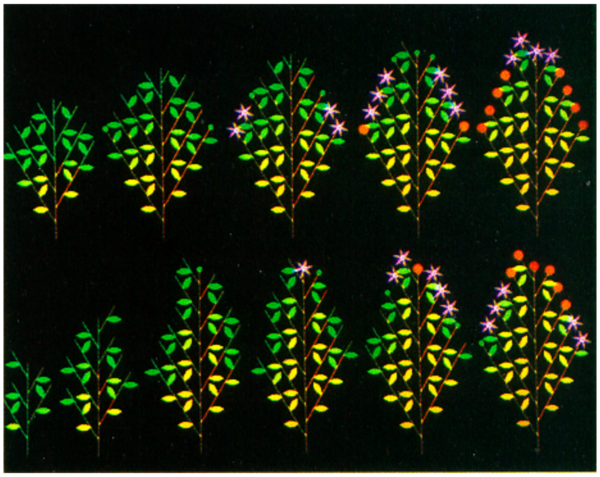

Figure 13. Acropetal (top) and basipetal (bottom) flowering sequences generated by the model with a single acropetal signal (shown as yellow-colored segments).

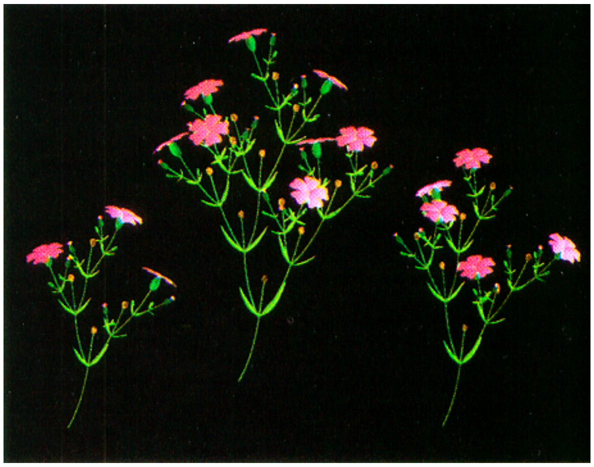

Figure 11. Development of a rose campion.

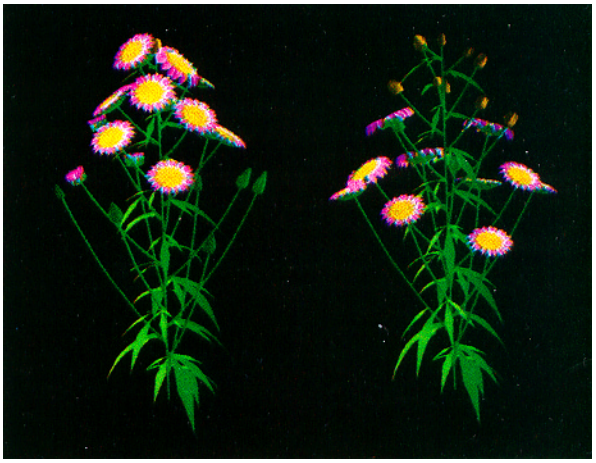

Figure 14. Two developmental stages of an aster.

In principle, its development can be described as follows:

$$\begin{array}{lll}
\omega: & A & \\
p_1: & A \rightarrow I_0[L_0]A & \\
p_2: & A \rightarrow I_0[L_0]B & \\
p_3: & B \rightarrow I_0[I_0F_0]B & \\
p_4: & X_i \rightarrow X_{i+1} & i \geq 0,\ X \in \{I, L, F\}
\end{array}$$

The initial vegetative growth is represented by production p_1 which describes creation of successive internodes and leaves by apex A. At some point in time, production p_2 changes the apex from the vegetative state A to the flowering state B. From then on, flowers are produced instead of leaves (production p_3), forming a raceme as discussed in Section 4.1. However, the moment in which this change occurs is not specified; the L-system is a nondeterministic one. Thus, for modeling purposes it must be complemented with an additional control mechanism which will determine the developmental switch time. Three applicable mechanisms are outlined below. Each of them is biologically motivated, and corresponds to a different class of L-systems.

4.3.1. A delay mechanism. The apex undergoes a series of state changes which delay the switch until a particular state is reached:

$$\begin{array}{lll}
\omega: & A_0 & \\
p_1: & A_i \rightarrow I_0[L_0]A_{i+1} & 0 \leq i < n \\
p_2: & A_n \rightarrow I_0[L_0]B & \\
p_3, p_4: & \textit{as before} &
\end{array}$$

According to this model, the apex *counts* the leaves it produces. While it may seem strange that a plant counts, it is known that some plant species produce a fixed number of leaves before they start flowering.

4.3.2. A stochastic mechanism. The vegetative apex has a probability π_1 of staying in the vegetative state, and π_2 of transforming itself into a flowering apex:

$$\begin{array}{ll}
\omega: & A \\
p_1: & A \overset{\pi_1}{\rightarrow} I_0[L_0]A \\
p_2: & A \overset{\pi_2}{\rightarrow} I_0[L_0]B \\
p_3, p_4: & \textit{as before}
\end{array}$$

For a formal definition of stochastic L-systems see [8, 46].

4.3.3. Environmental change. Many plants change from a vegetative to a flowering state in response to an environmental factor (such as the number of daylight hours or temperature). We can model this effect by using one set of productions (called a *table*) for some number of derivation steps before replacing it by another table.

Table 1

$$\begin{array}{lll}
\omega: & A & \\
p_1: & A \rightarrow I_0[L_0]A & \\
p_2: & X_i \rightarrow X_{i+1} & i \geq 0,\ X \in \{I, L\}
\end{array}$$

Table 2

$$\begin{array}{lll}
p_1: & A \rightarrow I_0[L_0]B & \\
p_2: & B \rightarrow I_0[I_0F_0]B & \\
p_3: & X_i \rightarrow X_{i+1} & i \geq 0,\ X \in \{I, L\}
\end{array}$$

The concept of table L-systems is formalized in [18, 38].

The developmental switch mechanism can also be applied to transform an apex from producing lateral flowers to producing a terminal flower which stops axis development. A raceme with a terminal flower is called a *closed* raceme, in contrast to the *open* racemes considered so far.

4.4. Inflorescence development with interactions.

Even in the presence of delays, the phase effects discussed so far reflect the sequential creation of branches, flowers and leaves by the subapical growth process. Consequently, organs near the plant roots develop earlier and more extensively than those situated near the axis ends. Such development results in *basitonic* plant structures (heavily developed near the base) with *acropetal* flowering sequences (the zone of blooming flowers progresses upwards along each branch). However, nature also creates *acrotonic* structures (heavily developed near the apex) and *basipetal* flowering sequences (progressing downwards). These structures and developmental patterns cannot be viewed as a simple consequence of subapical growth; for example, basipetal flowering sequences progress in the direction which is precisely opposite to that of plant growth. An intuitively straightforward and biologically well founded explanation can be given in terms of signals (Section 2.4) which propagate through the plant and control the timing of developmental switches. Below we consider two developmental models with signals. The first model employs a single acropetal signal, while the second one uses both acropetal and basipetal signals.

4.4.1. Developmental model with a single acropetal signal.

Let us assume that a flower-inducing signal (which represents the hormone *florigen*) stops axis development and causes production of a terminal flower upon reaching the apex. In this case, the overall phase effect results from an interplay between growth and control signal propagation [25, 20]. Assuming that only the first-order lateral branches are present, the development can be described by the following L-system:

$$\begin{array}{lll}
\omega: & D_0A_0 & \\
p_1: & A_i \rightarrow A_{i+1} & 0 \leq i < m-1 \\
p_2: & A_{m-1} \rightarrow I[L_0][B_0]A_0 & \\
p_3: & B_i \rightarrow B_{i+1} & 0 \leq i < n-1 \\
p_4: & B_{n-1} \rightarrow J[L_0]B_0 & \\
p_5: & D_i \rightarrow D_{i+1} & 0 \leq i < d-1 \\
p_6: & D_d \rightarrow S_0 & \\
p_7: & S_i \rightarrow S_{i+1} & 0 \leq i < \max\{u, v\} - 1 \\
p_8: & S_z \rightarrow \varepsilon & z = \max\{u, v\} - 1 \\
p_9: & S_{u-1} < I \rightarrow IS_0 & \\
p_{10}: & S_{v-1} < J \rightarrow JS_0 & \\
p_{11}: & S_0 < A_i \rightarrow F_0 & 0 \leq i \leq m-1 \\
p_{12}: & S_0 < B_i \rightarrow F_0 & 0 \leq i \leq n-1 \\
p_{13}: & X_i \rightarrow X_{i+1} & i \geq 0,\ X \in \{L, F\}
\end{array}$$

This L-system operates as follows (Fig. 13). The apex A produces segments of the main axis I, (optional) leaves L and the lateral apices B (p_1, p_2). The time between the production of two consecutive segments of the main axis, called its *plastochron*, is equal to m units (derivation steps). In a similar way, the first-order apices B produce segments J of the lateral axes and leaves L with plastochron n (p_3, p_4). After a delay of d time units a signal S is sent from the tree base towards the apices (p_6). The signal is transported along the main axis with a delay of u time units per internode I (p_7, p_9), and along the first-order axes with a delay of v units per internode J (p_7, p_{10}). Production p_8 removes the signal from a node after it has been transported further along the structure (ε stands for the empty string). When the signal reaches an apex (either A or B), the apex is transformed into a terminal flower F (p_{11}, p_{12}). Leaves and flowers undergo the usual developmental sequence (p_{13}).

In order to analyze the plant structure and flowering sequence resulting from the above development, let us denote by t_k the time at which the apex of the k-th first-order axis is transformed into a flower, and by l_k the length of this axis (expressed as the number of internodes) at the transformation time. Since it takes km time units to produce k internodes along the main axis and l_kn time units to produce l_k internodes on the first-order axis, we obtain $t_k = km+l_kn$. On the other hand, the transformation occurs when the signal S reaches the apex. The signal is sent d time units after the development starts, uses ku time units to travel through k zero-order internodes and l_kv time units to travel through l_k first-order internodes, resulting in $t_k = d+ku+l_kv$. Solving the above system of equations for l_k and t_k (and ignoring for simplicity some inaccuracy due to the fact that this system does not guarantee integer solutions), we obtain:

$$t_k = k\frac{un-vm}{n-v} + d\frac{n}{n-v}\ , \qquad l_k = -\frac{k}{n}\frac{m-u}{n-v} + \frac{d}{n-v}\ .$$

In order to analyze these solutions, let us first notice that the signal transportation delay v must be less than the plastochron of the first-order axes n (if this were not the case, the signal would never reach the apices). Under this assumption, the sign of the expression $\Delta = un-vm$ determines the flowering sequence, which is acropetal for $\Delta > 0$ and basipetal for $\Delta < 0$ (Fig. 13). If $\Delta = 0$, all flowers occur simultaneously. The sign of the expression $m-u$ determines whether the plant has a basitonic ($m-u < 0$) or acrotonic ($m-u > 0$) structure. Two stages of the development of an aster, modeled using the above L-system with $\Delta < 0$, are shown in Fig. 14.

4.4.3. Developmental model with several signals.

The development of some inflorescences is controlled by several signals, which may propagate with different delays and trigger each other. The use of more than one signal is instrumental in the modeling of a large class of inflorescences (found, for instance, in the family Compositae) characterized by terminal flowers on all apices, indefinite order of branching and basipetal flowering sequence. Figure 15 illustrates this type of development with an example of wall lettuce (*Mycelis muralis*). The underlying L-system operates as follows. First, the main axis is formed in a process of subapical growth which produces subsequent internodes and lateral apices. At this stage further development of lateral branches is suppressed (in botany, this effect is known as *apical dominance*). At some moment a flowering signal S_1 is sent from the bottom of the inflorescence up along the main axis. When this signal reaches its apex, the terminal flower is initiated and a basipetal signal S_2 enabling the growth of lateral axes is sent down the main axis. After a delay, a secondary basipetal signal S_3 is sent from the apex of the main axis. Its effect is to send the flowering signal S_1 along subsequent first-order axes as they are encountered on the way down. This entire process repeats recursively for each axis: its apex is transformed into a flower, the growth of lateral axes of the next order is successively enabled, and the secondary basipetal signal is sent to induce the flowering signal S_1 in these lateral axes. The resulting structure depends heavily on the values of plastochrons, delays, and signal propagation times. In the example under consideration, signal S_2 travels faster than S_3. Consequently, the time interval between the arrival of signals S_2 and S_3 increases while moving down the plant, potentially allowing the lower axes to grow longer then the upper ones. On the other hand, the lower branches start developing later, so in younger plants (in the middle of Fig. 15) they have not yet reached their full length. A detailed biological analysis of the above developmental pattern is given by Janssen and Lindenmayer [20].

4.5. Adding variation to models.

All plants generated by a deterministic L-system are identical. An attempt to include them in the same picture would produce a striking, artificial regularity. In order to prevent this effect it is necessary to introduce specimen-to-specimen variation which preserves the general aspects of a plant while modifying its details. We employ stochastic L-systems [8, 46] for this purpose. For example, Fig. 16 presents a field consisting of sixteen flowers generated by an L-system in which internode elongation is described by three stochastic rules:

$$\begin{array}{ll} \omega: & A \\ p_1: & I \rightarrow^{\pi_1} I \\ p_2: & I \rightarrow^{\pi_2} II \\ p_3: & I \rightarrow^{\pi_3} I[L][L]I \end{array}$$

where the probabilities π_1, π_2 and π_3 are equal to 1/3. The resulting field appears to consist of various specimens of the same (albeit fictitious) plant species. For more details on the use of stochastic L-systems for plant modeling purposes see [30, 34].

5. A NOTE ON PHYLLOTAXIS.

The longitudinal and angular displacement of consecutive branches or appendages with respect to each other is an important attribute of plant form, known as *phyllotaxis* [9, 42, 44]. In terms of the turtle interpretation of axial trees, these parameters represent the segment length and the *divergence angle* corresponding to the turtle's rotation about the heading vector $\vec{H}$. Abstracting from the mechanisms which govern the formation of phyllotactic patterns, two situations can be distinguished. In *alternating* patterns and *whorls* the angular positions of branches are repeated after a few nodes. In these cases, the divergence angle is equal to 360°/n, where n is a small integer. This type of arrangement occurs in lilac (Fig. 17), where consecutive pairs of (n+1)-order axes lie in the planes passing through the n-order axis and perpendicular to each other (Fig. 18). The divergence angle of 90° is also found in the rose campion (Fig. 11). On the other hand, in *spiral* patterns repetition occurs after a long period or cannot be detected at all. In these cases, the divergence angle is often close to the Fibonacci angle (approximately 137.5°). For examples, see shepherd's purse (Fig. 12), aster (Fig. 14) and wall lettuce (Fig. 15).

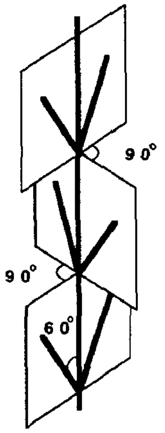

Figure 18. Branch arrangement in lilac inflorescences.

6. MODELING OF ORGANS.

So far we have discussed the modeling of "skeletal" trees with branches consisting of mathematical lines. In this section we extend the model to include surfaces and volumes.

Conceptually, the simplest approach is to incorporate predefined surfaces in the tree, with positions and orientations specified by the turtle. For example, leaves of the lily-of-the-valley (Fig. 9), buds, flowers and fruits of the rose campion (Fig. 11), buds, petals and fruits of the aster (Fig. 14) as well as leaves and flowers of the lilac (Fig. 17) were modeled using bicubic patches. Bicubic surfaces were also applied to model cylindrical stem segments in all these structures. Patches make it easy to manipulate and modify surface shapes interactively, but are incompatible with the developmental approach to modeling since they do not "grow". Consequently, each developmental stage of an organ must be modeled separately.

In order to fully simulate plant development and model phase effects present in plant structures, it is necessary to provide a mechanism for changing the size and shape of surfaces in time. A simple approach is to fill a polygon made of lines defined by an L-system. For example, leaves of the fern (Fig. 10) the shepherd's purse (Fig. 12) and the aster (Fig. 14) were modeled using the following L-system:

$$\begin{array}{ll} \omega: & L \\ p_1: & L \rightarrow \{-SX+X+SX-|-SX+X+SX\} \\ p_2: & X \rightarrow SX \end{array}$$

Production p_1 defines a leaf as a closed planar polygon. The parentheses { and } indicate that the polygon should be filled. Production p_2 linearly increases the lengths of the polygon edges.

The tracing of polygon boundaries leads to acceptable effects in the case of small, flat surfaces. In other cases it is more convenient to define surfaces using an underlying tree structure as the frame. The entire surface consists of polygons bounded by tree segments and extra edges inserted between appropriate terminal nodes of the tree to form closed contours. The three leaf shapes shown in Fig. 19 were obtained by modifying branching angles and growth rates of axes. Specifically, the blade of the *cordate* leaf (the leftmost one) was generated by the following L-system:

$$\begin{array}{ll} \omega: & [A][B] \\ p_1: & A \rightarrow [+A?]C\# \\ p_2: & B \rightarrow [-B?]C\# \\ p_3: & C \rightarrow IC \end{array}$$

The axiom contains symbols A and B which generate the left-hand side and the right-hand side of the blade. Each of the productions p_1 and p_2 creates a sequence of axes starting at the leaf base and gradually diverging from the midrib. Production p_3 increases the axis lengths. The axes close to the midrib are the longest since they were created first (thus, the leaf shape is yet another manifestation of the phase effect). The symbols ? and # indicate the endpoints of edges to be inserted while forming closed polygons. The following string represents the left-hand side of the leaf after four derivation steps:

$$[+[+[+[+A\underset{\curvearrowright}{?]C\#}\underset{\curvearrowright}{?]IC\#}\underset{\curvearrowright}{?]IIC\#}\underset{\curvearrowright}{?]IIIC\#}$$

The arrows indicate the inserted edges (the first one has zero length, the second is collinear with an axis, and the subsequent ones bound triangles). The developmental sequence is shown in Fig. 20. Leaves generated in a similar way were incorporated in the model of the rose campion (Fig. 11).

The frame-based approach can be extended to three-dimensional organs. The right-hand images in Fig. 19 illustrate construction of the flowers for the lily-of-the-valley in Fig. 9. The L-system generates a supporting framework composed of five curved lines which spread radially from the flower base and are connected by a web of inserted edges. In this case each polygon is a trapezoid bounded by two "regular" and two inserted edges.

Another developmental approach to leaf modeling was recently proposed by Lienhardt and Françon [23] and Lienhardt [22].

7. IMPLEMENTATION.

The concepts described in this paper were implemented using a modeling program called *pfg* designed for an IRIS 3130 workstation. The input to the program consists of an L-system specified in the bracketed string notation and approximately 30 parameters, most of which control rendering and viewing. Additionally, an arbitrary number of

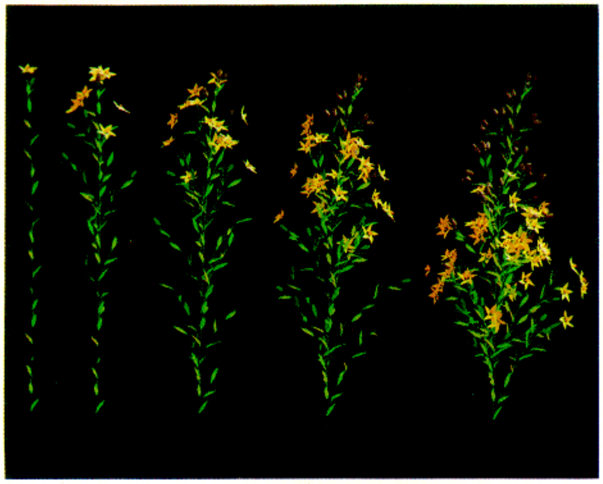

Figure 15. Development of a wall lettuce.

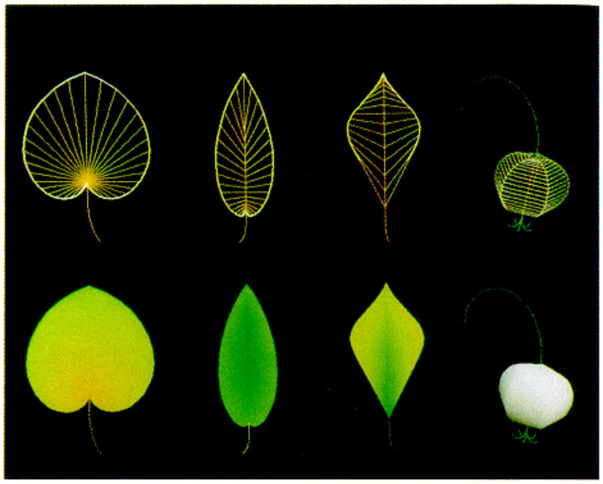

Figure 19. Developmental models of leaves and a flower. The top row shows the underlying tree structures (yellow lines) and the edges inserted to form closed polygons (white lines). The bottom row shows the same structures with filled polygons.

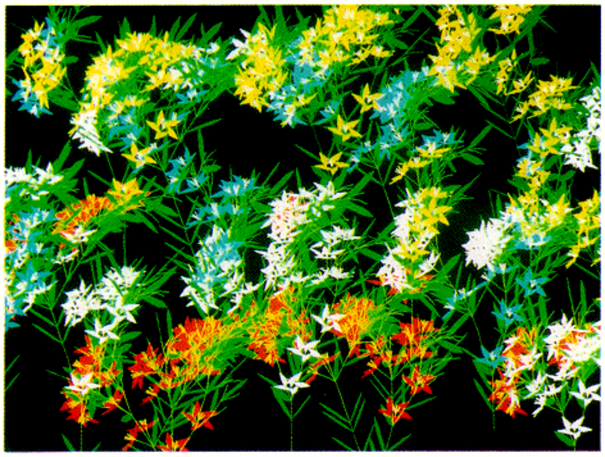

Figure 16. A flower field.

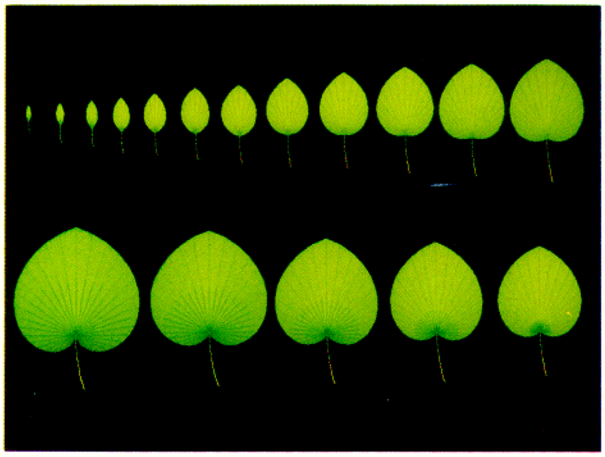

Figure 20. Developmental sequence of a cordate leaf.

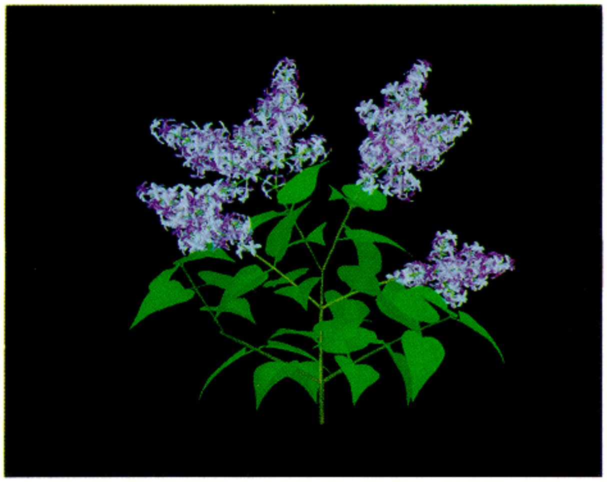

Figure 17. A lilac twig.

files containing patch descriptions can be read in (patches are edited outside of *pfg*). The animation of developmental processes is controlled interactively. The total simulation and rendering time for plants images shown in this paper ranges from one to five minutes. The consecutive frames of schematic developmental sequences (such as shown in Fig. 13) are generated a few seconds apart, which is sufficient for analysis of development using animation.

8. CONCLUSIONS.

In this paper we presented guidelines for modeling herbaceous plants and simulating their development. Plant structures have been described in terms of developmental processes controlled by lineage and interactive mechanisms. The developmental approach was extended to model plant organs.

In computer imagery applications, construction of a developmental model is an intermediate step leading to the final goal, a realistic image of a synthetic plant. To a biologist the model itself can be of primary interest as a formal description of a developmental process. The notion of L-systems makes it easy to specify a model in terms consistent with those used in developmental morphology and physiology, and to experiment with a wide range of processes and structures. Thus, the modeling methods presented in this paper can be used as a research tool for visualizing scientific hypotheses related to development in nature.

A number of problems are open for further research.

- **Addition of texture.** The surfaces shown in this paper lack texture. Specifically, a major component of leaf texture is its venation. For consistency with the developmental approach to modeling, the venation itself should be generated by a developmental algorithm. The problem is that veins may form closed cycles and therefore cannot be described in terms of axial trees. An extension of tree L-systems to graphs with cycles (map L-systems) was proposed by Lindenmayer [26, 27] but has not been applied yet to model venation.
- **Improved surface models.** The described model of surface development is difficult to apply to complex three-dimensional surfaces, such as snap-dragon flowers or wrinkled petals of petunias. A difficult situation also occurs when organs composing a larger structure are crowded, for example cabbage leaves, or the petals in rose and peonia flowers. More flexible developmental surface models would be very useful in these cases.
- **Time step control.** The formalism of L-systems is discrete in nature. A developmental model can be constructed assuming longer or shorter time intervals, but once the choice has been made, the time step is a part of the model and cannot be changed easily. From the viewpoint of computer animation it would be preferable if the time step were controlled by a single parameter, decoupled from the underlying L-system.
- **Analysis of simulation complexity.** Various data structures can be used to represent axial trees and carry out the derivation process (Section 2.3). Although bracketed strings appear to be more memory-efficient than list representations, no formal analysis of time and space trade-offs related to the choice of data structure has been made. Such analysis could lead to optimal algorithms.
- **Addition of a graphical interface.** In the present implementation of the *pfg* program, input L-systems are specified in the bracketed string notation. In some applications, such as computer-assisted instruction of developmental morphology, it may be preferable to avoid the textual interface and define productions graphically, as shown in Fig. 2a. The formalism of tree L-systems, which dissociates the graph-theoretic concept from the string implementation, could lend itself to such an interface.

ACKNOWLEDGMENTS

The aster flowers were modeled by Debbie Fowler. The generous support from the Department of Computer Science, University of Regina, and the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

REFERENCES

1. Abelson, H., and diSessa, A. A. *Turtle geometry.* M.I.T. Press, Cambridge (1982).
2. Aono, M., and Kunii, T. L. Botanical tree image generation. *IEEE Computer Graphics and Applications* **4**, 5 (1984), 10-34.
3. Aono, M., and Kunii, T. L. *Botanical tree image generation.* [Video tape], IBM, Tokyo (1985).
4. Armstrong, W. W. The dynamics of tree linkages with a fixed root link and limited range of rotation. *Actes du Colloque Internationale l'Imaginaire Numérique '86* (1986), 16-21.
5. Baker, R., and Herman, G. T. Simulation of organisms using a developmental model, Parts I and II. *Int. J. of Bio-Medical Computing* 3 (1972), 201-215 and 251-267.
6. Beyer, T., and Friedell, M. Generative scene modelling. *Proceedings of EUROGRAPHICS '87* (1987), 151-158 and 571.
7. Bloomenthal, J. Modeling the Mighty Maple. Proceedings of SIGGRAPH '85 (San Francisco, CA, July 22-26, 1985). In *Computer Graphics* **19**, 3 (1985), 305-311.
8. Eichhorst, P., and Savitch, W. J. Growth functions of stochastic Lindenmayer systems. *Inf. and Control* **45** (1980), 217-228.
9. Erickson, R. O. The geometry of phyllotaxis. In J. E. Dale and F. L. Milthrope (Eds.): *The growth and functioning of leaves,* Cambridge University Press (1983), 53-88.
10. Eyrolles, G. *Synthèse d'images figuratives d'arbres par des méthodes combinatoires.* Ph.D. Thesis, Université de Bordeaux I (1986).
11. 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 (Eds.): *L Systems,* Lecture Notes in Computer Science **15**, Springer-Verlag, Berlin (1974), 24-52.
12. Frijters, D., and Lindenmayer, A. Developmental descriptions of branching patterns with paracladial relationships. In A. Lindenmayer and G. Rozenberg (Eds.): *Automata, languages, development,* North-Holland, Amsterdam (1976), 57-73.
13. Frijters, D. Principles of simulation of inflorescence development. *Annals of Botany* **42** (1978), 549-560.
14. Frijters, D. Mechanisms of developmental integration of *Aster novae-angliae* L., and *Hieracium murorum* L. *Annals of Botany* **42** (1978), 561-575.
15. Habel, A., and Kreowski, H.-J. On context-free graph languages generated by edge replacement. In H. Ehrig, *et al.* (Eds.): *Graph grammars and their application to computer science; Second Int. Workshop,* Lecture Notes in Computer Science **153**, Springer-Verlag, Berlin (1983), 143-158.
16. Hallé, F., Oldeman, R., and Tomlinson, P. *Tropical trees and forests: an architectural analysis.* Springer-Verlag, Berlin (1978).
17. Herman, G. T., and Liu, W. H. The daughter of CELIA, the French flag, and the firing squad. *Simulation* **21** (1973), 33-41.
18. Herman, G. T., and Rozenberg, G. *Developmental systems and languages.* North-Holland, Amsterdam (1975).
19. Hogeweg, P., and Hesper, B. A model study on biomorphological description. *Pattern Recognition* **6** (1974), 165-179.
20. Janssen, J. M., and Lindenmayer, A. Models for the control of branch positions and flowering sequences of capitula in *Mycelis muralis* (L.) Dumont (Compositae). *New Phytologist* **105** (1987), 191-220.
21. Kawaguchi, Y. A morphological study of the form of nature. Proceedings of SIGGRAPH '82 (July 1982). In *Computer Graphics* **16**, 3 (1982), 223-232.
22. Lienhardt, P. *Modélisation et évolution de surfaces libres.* Ph.D. Thesis, Université Louis Pasteur, Strasbourg (1987).
23. Lienhardt, P., and Françon, J. *Synthèse d'images de feuilles végétales.* Technical Report R-87-1, Département d'informatique, Université Louis Pasteur, Strasbourg (1987).
24. Lindenmayer, A. Mathematical models for cellular interaction in development, Parts I and II. *J. Theor. Biol.* **18** (1968), 280-315.
25. Lindenmayer, A. Positional and temporal control mechanisms in inflorescence development. In P. W. Barlow and D. J. Carr (Eds.): *Positional controls in plant development,* Cambridge University Press (1984).
26. Lindenmayer, A. Models for multicellular development: characterization, inference and complexity of L-systems. In A. Kelmenová and J. Kelmen (Eds.): *Trends, techniques and problems in theoretical computer science.* Lecture Notes in Computer Science **281**, Springer-Verlag, Berlin (1987), 138-168.
27. Lindenmayer, A. An introduction to parallel map generating systems. In H. Ehrig, *et al.* (Eds.): *Graph grammars and their application to computer science; Third Int. Workshop,* Lecture Notes in Computer Science **291**, Springer-Verlag, Berlin (1987), 27-40.
28. Lindenmayer, A., and Prusinkiewicz, P. Developmental models of multi-cellular organisms: A computer graphics perspective. Paper submitted to the *Proceedings of the Artificial Life Workshop* held in Los Alamos, NM, September 1987.
29. Müller-Doblies D., and U. Cautious improvement of a descriptive terminology of inflorescences. *Monocot newsletter* 4, Institut für Biologie, Technical University of Berlin (West), 13 (1987).
30. Nishida, T. KOL-systems simulating almost but not exactly the same development - the case of Japanese cypress. *Memoirs Fac. Sci., Kyoto University, Ser. Bio.* **8** (1980), 97-122.
31. Oppenheimer, P. Real time design and animation of fractal plants and trees. Proceedings of SIGGRAPH '86 (Dallas, Texas, August 18-22, 1985). In *Computer Graphics* **20**, 4 (1986), 55-64.
32. Preparata F. P., and Yeh, R. T. *Introduction to discrete structures.* Addison-Wesley, Reading (1973).
33. Prusinkiewicz, P. Graphical applications of L-systems. *Proc. of Graphics Interface '86 - Vision Interface '86* (1986), 247-253.

34. Prusinkiewicz, P. Applications of L-systems to computer imagery. In H. Ehrig, *et al.* (Eds.): *Graph grammars and their application to computer science; Third Int. Workshop*, Lecture Notes in Computer Science **291**, Springer-Verlag, Berlin (1987), 534-548.

35. Prusinkiewicz, P., and Hanan, J. Lindenmayer systems, fractals, and plants. In D. Saupe (Ed.): *Fractals: Introduction, basics and applications.* [Course notes] SIGGRAPH '88 (Atlanta, Georgia, August 1-5, 1988).

36. Reeves, W. T., and Blau, R. Approximate and probabilistic algorithms for shading and rendering structured particle systems. Proceedings of SIGGRAPH '85 (San Francisco, CA, July 22-26, 1985). In *Computer Graphics* **19**, 3 (1985), 313-322.

37. Robinson, D. F. A symbolic notation for the growth of inflorescences. *New Phytologist* **103** (1986), 587-596.

38. Rozenberg, G., and Salomaa, A. *The mathematical theory of L-systems.* Academic Press, New York (1980).

39. Smith, A. R. About the cover: "Reconfigurable machines". *Computer* **11**, 7 (1978), 3-4.

40. Smith, A. R. Plants, fractals, and formal languages. Proceedings of SIGGRAPH '84 (Minneapolis, Minnesota, July 23-27, 1984). *Computer Graphics* **18**, 3 (1984), 1-10.

41. Smith, A. R. *Grammars for generating the complexity of reality.* [Video tape], Lucasfilm/PIXAR, San Rafael (1985).

42. Stevens, P. S. *Patterns in nature.* Little, Brown and Co., Boston (1974).

43. Szilard, A. L., and Quinton, R. E. An interpretation for DOL systems by computer graphics. *The Science Terrapin* **4** (1979), 8-13.

44. Thompson, d'Arcy. *On growth and form.* University Press, Cambridge (1952).

45. Troll, W. *Die Infloreszenzen*, Vol. I. Gustav Fischer Verlag, Stuttgart (1964).

46. Yokomori, T. Stochastic characterizations of E0L languages. *Information and Control* **45** (1980), 26-33.

47. Zimmerman, M. H., and Brown, C. L. *Trees - structure and function.* Springer-Verlag, Berlin (1971).