The Visual Computer

Structural simulation of tree growth and response

John C. Hart[1], Brent Baker[2], Jeyprakash Michaelraj[3]

[1] Department of Computer Science, University of Illinois, Urbana-Champaign, 1304 W. Springfield Ave., Urbana, IL 61801, USA
E-mail: jch@cs.uiuc.edu
[2] Ogden Profession Services, U.S. Environmental Protection Agency, Corvallis, OR 97333, USA
E-mail: brent@heart.cor.epa.gov
[3] Alias | Wavefront, 210 King St., Toronto, ON, M5A 1J7, Canada
E-mail: jmichael@aw.sgi.com

Published online: 5 February 2003

Each tree is unique because of the physical environment it experiences over the course of its life. Environmental factors shape a tree within the bounds of its genotype. Only by modeling the environmental influences can we create realistic models of trees. To this end, we constructed a structural simulation that calculates the mass of each branch of the tree to emulate the mechanisms the tree uses to balance its weight, and that estimates the photosynthesis return of the leaves to simulate phototropism. Our effort is motivated by a desire to construct a predictive tool that can be used by both those in computer graphics and forest management, with applications in image synthesis, dendrochronology, mensuration and the simulation of forest succession.

Key words: Natural phenomena – Trees – L-systems – Statics

1 Introduction

Tree development is a continuous compromise between the maximization of available light and the minimization of structural stress. Computer graphics research has produced many models of trees and simulations of tree development based on genetic models of the branching process. The most prominent of these models has been the L-system. In an effort to make these models produce more realistic results, some have encoded the environmental influences, such as geotropism (sagging branches), directly into the L-system [23]. While this allows the genetic model to include more detailed information about its reaction to external forces, it makes the model much more complex and difficult to decypher.

This work separates the environmental information from the genetic information in tree development. While a tree grown in space might exhibit many of the self-similar traits predicted by their genetic growth models, real trees are exceptionally non-self-similar. Figure 1 illustrates how the external influences of gravity and sunlight affect tree form.

Trees (and most other natural forms) derive their strength from surface tension [27]. The strength of a tree's trunk is therefore proportional to its cross-sectional area. As the tree grows, its mass increases cubicly, but its cross-sectional area (and therefore its support strength) increases quadratically. While a young tree need not concern itself with such mechanical constraints, a mature tree must not grow too tall lest its trunk collapse under its own weight. Trees have developed quite clever methods for managing mechanical stress in their development, and many of these methods have been categorized and illustrated by Mattheck [20].

In an effort to simulate these effects, we created an empirical model of the tree's reaction to mechanical stress, to produce visually realistic images of tree adaptations. This model resulted in an approximation of the proportions of the tree's limbs by regulating growth to ensure the limbs were sufficiently strong to support their extremities. This growth regulation also strived to reduce torque in the tree's trunk by balancing the tree's center of mass over its trunk.

A simple L-system without this growth-inhibition model yields the purely "genetic" tree in Fig. 1. The left side of this tree is heavier and threatens to topple the entire tree because the main left branch is older than the other main branches. Unlike the "balanced" tree in this same figure, purely genetic devel-

Digital Object Identifier (DOI) 10.1007/s00371-002-0189-4

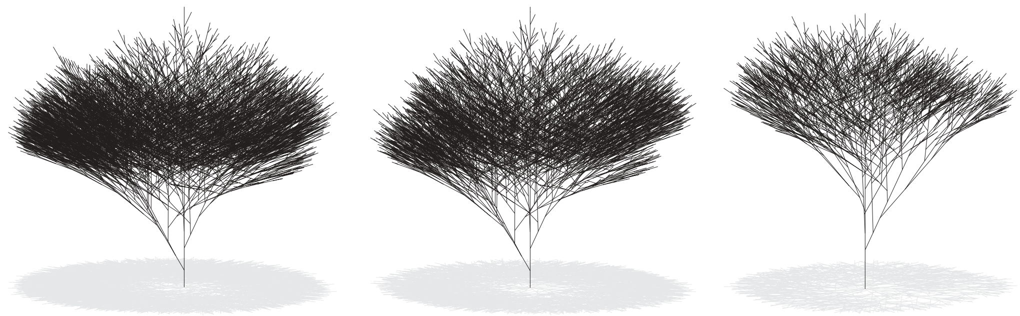


Fig. 1. A purely genetic tree, a tree with its center of mass positioned over its trunk and a tree influenced by sunlight

opment ignores the torque created by this lopsided development.

Our model of mechanical stresses is limited to trees that have a single, non-hollow trunk, that exhibit an annual growth cycle. The branches in this model do not bend, and once they have sprouted, their orientations remain fixed. Each branch is represented geometrically with a cone clipped by two planes perpendicular to its major axis.

Nature has arrived at very diverse solutions for some common problems, and the extremes of these solutions are outside the scope of this paper. Structural changes caused by disease or damage by insects are not modeled; nor are the effects of typical genetic anomalies such as spiral grain or burls. Although the living sapwood at the perimeter of each branch is somewhat weaker than the dead heartwood in its center, formal tests are not often conducted on the sapwood. Hence, the heartwood measurements are used [9] and the simulation treats the tree's volume uniformly.

This paper also describes a model for photosynthesis, producing phototropic (light seeking) effects in the tree throughout the development process. By killing shaded limbs, the "phototropic" tree in Fig. 1 is able to collect approximately the same amount of light energy as the genetic tree, but with only 25% of the branches (and mass).[1]

[1] Figure 1 represents only the structure of the trees (using line segments) in winter (without leaves). Hence, the shadow of the "photosynthesis" tree is misleadingly sparse. With leaves, the shadows of all three trees are similarly dense.

2 Relationship to existing tree models in computer graphics

Trees possess an immense amount of detail, and some have worked on non-developmental models of mature trees that organize their geometry for best efficiency. Oppenheimer modeled trees based on the affine transformation of primitives for interactive design [22]. Hart and DeFanti also used affine transformations to ray-trace forests of highly detailed tree models [13].

Trees pose difficult geometric and texturing problems. Bloomenthal simulated a Maple tree using generalized cylinders, and developed a parametric blending model to maintain continuity in geometry and texture around branching points [5]. A bark bump-map was made from an X-ray of real bark. Hart formulated an implicit blending model for interpolating a procedural bark displacement-map texture through a branching point [14]. We published a related paper on generating the branching shapes resulting from the structural information created by the model described in this paper [12].

Many have used observations documented in the botanical research community to construct non-developmental tree models. De Reffye et al. simulated trees branching and growth based on rules derived from the classifications and observations documented in the botanical research community [6]. They mention the possibility of simulating external forces such as wind and gravity, but as a post-process to make the tree appear more realistic.

Weber and Penn continued this observation-based trend with a highly detailed non-developmental tree model with a tremendous number of parameters to represent a wide variety of tree species [29]. These parameters are based on various physical properties of the tree, and can be easily observed and recorded. They simulate several structural properties of trees, such as stem bending, but these actions were controlled by static input parameters.

While non-developmental models are more time and space efficient, often developmental simulations are necessary to capture the form of natural shapes such as trees [27].

Prusinkiewicz et al. used developmental L-systems to encode biological observations to simulate the structure and growth of plants [24]. They focused on non-woody plants, which better adhere to the "internal control mechanisms" as opposed to trees whose shape is determined by "the environment, competition between trees and tree branches, and accidents." Prusinkiewicz and Lindenmayer summarized a variety of previous L-systems controlling the basic shape of trees where the interaction with the environment was limited to rotating branches to face incoming light (planartropism) and adjusting branching directions to simulate the effect of gravity (geotropism) [23].

There have been a large number of efforts to incorporate the external environment into the developmental model of the tree to produce more realistic results. These efforts have synthesized the desired environmental impact on the resulting plant shape, but often at the expense of plants that might not be capable of supporting the shape they have formed.

Arvo and Kirk simulated the environmentally sensitive growth of vines using ray casting to determine proximity [1]. Greene discretized space into a voxel array to more efficiently sample the environment at each step of the growth process of such vines [11]. These vines actively grew toward light sources, avoiding shadowing obstacles when necessary, by sensing direct light from a moving sun and indirect light from the sky. Two striking figures illustrate the vine approximation without the supporting model, but there lacked any calculation of the resulting structure's stability.

An environmental query operator was later added to the L-system paradigm that allowed shrubs to be pruned to a predefined shape to simulate topiary [25]. Context sensitivity in the L-system was used to pass a signal along the stem to emulate the plant's method of reacting to its environment. These environmental queries were later used to simulate the interactions of neighboring trees as they competed for resources [21], and eventually resulted in the modeling of entire ecosystems [7]. The mechanical and growth regulation factor provided by this paper could be integrated as environmental controls by such L-systems.

Some have incorporated the analysis of structural support in the development of plant forms.

Viennot et al. examined the *ramification matrix*, used in the study of river networks, as a model for binary tree growth [28]. They also simulated the reproduction of trees by an interpolation of their ramification matrices.

Holton investigated the *strand model* (based on the pipe model used to study river networks and bronchi of the lungs) for simulating the vascular structure of trees [15]. The strand model simulated many properties of a tree limb, such as diameter, length and branching angle, based on the number of strands in the limb. External forces were also simulated. The direction of trunk growth was adjusted in the direction of the central axis of the branching structure. Branches were also affected by the pull of gravity, and could be directed to grow upward, horizontal and/or planarly. Phototropism was elegantly approximated by having branches grow in directions away from the center of the tree.

Jirasek and Prusinkiewicz developed a plant model that accounts for structural support in the growth of curved branches based on Young's modulus and other measured parameters [17]. The resulting shapes more closely resemble the curved leaves of some ferns. This model was later integrated into the L-system model [18].

3 The physics of trees

Whereas much of the physics used in computer graphics is optics and dynamics, trees are best understood through the use of *statics*. Such analysis reveals that the shape of a tree is significantly influenced by its response to the forces affecting its structure.

3.1 Statics

Dynamics (kinematics and kinetics) simulates an object in motion. Typically, objects are rigid bodies connected by springs, and forces alter position and orientation [3]. Statics simulates objects at rest. Such objects are typically beams connected by pins into a rigid structure, and external forces that would ordinarily alter position and orientation are countered by equal and opposite internal forces. Statics focuses on the computation of these internal forces.

The following principles of statics [4] use the notation $\boldsymbol{F}(\boldsymbol{x})$ to denote a force $\boldsymbol{F}$ acting on a point $\boldsymbol{x}$ of some rigid body. The result of two forces $\boldsymbol{F}_1$, $\boldsymbol{F}_2$ acting on the same point $\boldsymbol{x}$ of a rigid body is equivalent to that of a single force equal to the sum of the original forces:

$$(\boldsymbol{F}_1+\boldsymbol{F}_2)(\boldsymbol{x}) \equiv \boldsymbol{F}_1(\boldsymbol{x})+\boldsymbol{F}_2(\boldsymbol{x}). \tag{1}$$

The result of two parallel forces $\boldsymbol{F}_1$, $\boldsymbol{F}_2$ acting on a rigid body along the same line of action is equivalent to that of a single force equal to the sum of the original forces:

$$\begin{aligned}&(\boldsymbol{F}_1+\boldsymbol{F}_2)(\boldsymbol{x}_1) \equiv \boldsymbol{F}_1(\boldsymbol{x}_1)+\boldsymbol{F}_2(\boldsymbol{x}_2)\\ &\quad \text{iff} \quad \boldsymbol{F}_1 \times \boldsymbol{F}_2 = \boldsymbol{0}\\ &\quad \text{and} \ (\boldsymbol{x}_2-\boldsymbol{x}_1)\times \boldsymbol{F}_1 = \boldsymbol{0}.\end{aligned} \tag{2}$$

The *principle of transmissibility* therefore states that a force at a point on a rigid body can be replaced by a force of equal magnitude and direction acting on a different point along the same direction of action:

$$\boldsymbol{F}(\boldsymbol{x}) \equiv \boldsymbol{F}(\boldsymbol{x}+\alpha\boldsymbol{F}) \ \forall\alpha. \tag{3}$$

The result of two forces acting on different points of the rigid body along different lines of action is equivalent to a force and a *couple*. A couple is a pair of opposite forces $\boldsymbol{F}_1 = -\boldsymbol{F}_2$ acting on different points $\boldsymbol{x}_1, \boldsymbol{x}_2$ of a rigid body. A couple causes a rigid body to rotate, and is typically represented by the *moment of the couple* $\boldsymbol{M}$:

$$\boldsymbol{M} = \boldsymbol{F}_1 \times (\boldsymbol{x}_2-\boldsymbol{x}_1). \tag{4}$$

The force of gravity acts simultaneously on every point in a rigid body. Using the above rules this gravitational force can be simplified, replaced by a single force acting on the *center of mass* of the rigid body. The center of mass $\bar{\boldsymbol{x}}$ of an object of uniform density is given by

$$\bar{\boldsymbol{x}} = \frac{1}{v}\int \boldsymbol{x}\,\mathrm{d}V \tag{5}$$

where $v = \int \mathrm{d}V$ is the volume of the object.

3.2 Reaction wood

There are two general classes of trees: gymnosperms and angiosperms. *Gymnosperms* are so named because their seeds are not enclosed, but are external. These include ginkos, pine, spruce, fir and other conifers whose seeds are carried in an open cone. *Angiosperms* have seeds enclosed in a mature ovary, a fruit. These species include those called deciduous and most tropical species. The classification is important because the two groups have developed different mechanisms to resist mechanical stress.

Several factors can create an unbalanced load on a structural member of a tree. The most common is the tree's own growth. Branches are occasionally lost, due to external trauma or shade from higher branches. Trees counteract these forces by growing *reaction wood* [20].

Reaction wood either pushes a branch (*compression wood*) or pulls a branch (*tension wood*) to counteract rotational force. Compression wood is formed by gymnosperms whereas tension wood is formed by angiosperms [8].[2] Both can be identified as a thickening, or eccentricity, of the annual ring on the side of the branch where it forms.

Our model simulates reaction wood as an increase in basal diameter, but does not simulate the compression/tension eccentricity of the growth. Hence the proportions of the branches are modeled appropriately, though the shape of the tree at the branching points will be uniform and does not currently differ between angiosperms and gymnosperms. A companion paper has investigated the local shape of reaction wood at branching points that accounts for the difference between angiosperms and gymnosperms [12].

3.3 Structural analysis of branching points

Statics provides the necessary tools to compute the internal forces of rigid bodies with connections involving various degrees of freedom, such as trusses, frames and machines. The tree is an example of a frame.

Figure 2 illustrates structural models of both kinds of trees and their appropriate reaction wood. Per-

[2] Some authorities maintain that angiosperms form both tension wood and compression wood scattered around the perimeter of the branch, and that reaction wood in angiosperms does not always create a stem eccentricity [10].

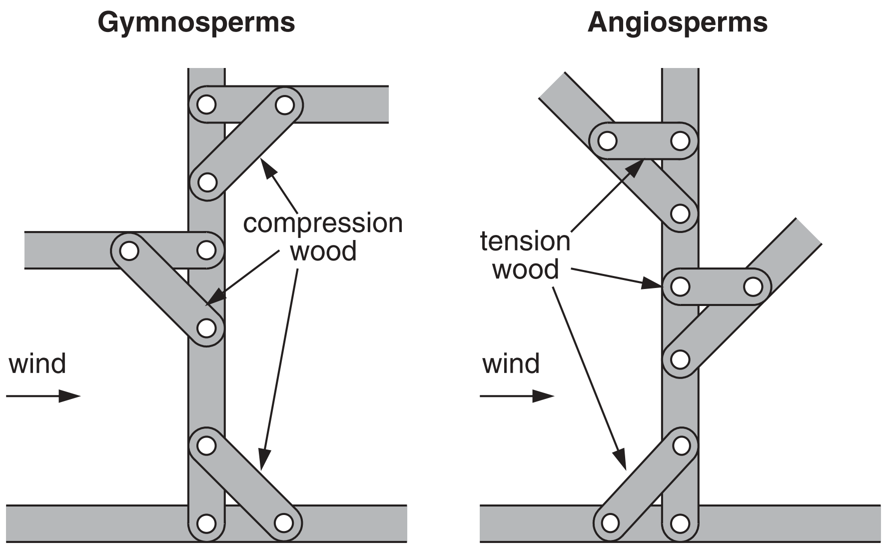


Fig. 2. Structural analysis of reaction wood

forming a static analysis at the branch points indicates how much counteracting reaction wood the tree needs to grow.

3.4 Mechanical properties of wood

Natural structures derive their strength from surface tension [27]. Hence, the mass supported by a tree's trunk is proportional to its cross-sectional area [2]. Let

$$\mathrm{supports}(R, d) = kdR^2 \tag{6}$$

be the maximum mass that a limb of basal radius R and density d can support. Let its inverse

$$\mathrm{needstosupport}(m, d) = \sqrt{\frac{m}{kd}} \tag{7}$$

return the minimum radius necessary to support a mass m.

The constant of conversion k can be derived from wood strength tables, such as from the Forest Products Laboratory [9]. A simpler definition comes from the observation that trees structurally cannot exceed 300 ft in altitude [27]. This defines k such that it solves

$$\mathrm{supports}(R, d) = dv(R, 0, 300\,\mathrm{ft}), \tag{8}$$

specifically $k = \pi \cdot 100\,\mathrm{ft}$.

4 Simulating tree growth and response

4.1 Growth and response algorithm

Each iteration of tree growth follows the following algorithm steps.

1. Compute mass
2. Compute center of mass
3. Compute photosynthesis
4. Compute growth rates
5. Grow branches

The tree is simulated as a collection of limbs, organized in a hierarchical tree structure (of course). Table 1 lists the limb variables used in this process. The *trunk* is defined as the limb with index $i = 0$.

The limb's spatial variables are computed with respect to a canonical limb coordinate system. The branch's base is at the origin of this coordinate system, and the branch grows along the positive y axis. The instancing transformation T_i consists only of a rotation and a translation, and maps branch coordinate onto world coordinates.[3]

[3] Since the branches' orientation and base position remain fixed with respect to world coordinates in this simulation, the instancing transformation is not necessary. It is included to support future simulations that would allow branches (and their children) to change orientation.

Table 1. Parameters and variables used during tree growth and response

Constant across tree	
d	density of wood
Variable for each limb i	
T_i	instancing transformation
l_i	length of branch
R_i, r_i	branch base (tip) radius
t_i	age of branch
v_i	volume of branch
m_i, M_i	mass of limb (everything supported)
e_i	limit on mass of new growth
$\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{X}}_i$	center of mass of limb (everything supported)
o_i	adjustment to balance tree
g_i, G_i	rate of growth for limb (and children)
p_i, P_i	photosynthesis of limb (everything supported)
J_i	indices of children

Compute mass: The mass of a branch $m_i = dv_i$ is the product of the density of its wood d with its volume v_i, where

$$v_i = v(l_i, R_i, r_i) = \frac{1}{3}\pi l_i \frac{R_i^3 - r_i^3}{R_i - r_i} \tag{9}$$

is the volume of a truncated cone. The mass a branch supports M_i is recursively the mass of the branch plus the mass its children support:

$$M_i = m_i + \sum_{j \in J_i} M_j. \tag{10}$$

Compute center of mass: The center of mass for a branch is the point

$$\bar{\boldsymbol{x}}_i = T_i \left(0, \frac{1}{4} l_i \frac{R^4 - r^3(4R - 3r)}{(R - r)(R^3 - r^3)}, 0, 1\right)^T. \tag{11}$$

For a cone ($r = 0$), the center of mass is simply $(0, \frac{1}{4} l_i, 0)$.
The center of mass for everything a branch supports is computed recursively:

$$\bar{\boldsymbol{X}}_i = \frac{1}{M_i}\Big(\bar{\boldsymbol{x}}_i m_i + \sum_{j \in J_i} \bar{\boldsymbol{X}}_j M_j\Big). \tag{12}$$

Compute offset: The offset o_i indicates the effect of branch growth on the tree's center of mass:

$$o_i = T_i \boldsymbol{y} \cdot \bar{\boldsymbol{X}}_0, \tag{13}$$

where $\boldsymbol{y} = (0, 1, 0, 0)^T$ indicates the limb's growth direction in limb coordinates. This value is then clamped to $[-1, 1]$.

Compute photosynthesis: Photosynthesis is simulated by rendering the tree from the light source's point of view. Instead of shading the leaves their usual color, they are instead colored with the index i of the limb they are attached to. The limbs themselves gather no light and so are rendered in the background color (e.g. a limb index of -1). The amount of photosynthetic nutrition returned to each limb by its leaves p_i is proportional to the percentage of pixels in the rendered image whose index matches the limb.
Trees receive light directly from a moving sun, and indirectly from the sky.[4] Hence, for each growth iteration, we take a single stochastic sample of photosynthesis, taking samples from the sun's path more than other locations proportional to the ratio of the sun's intensity to the sky's intensity.[5]
The total photosynthetic nutrition returned to each limb by its leaves and its children is accumulated recursively:

$$P_i = p_i + \sum_{j \in J_i} P_j. \tag{14}$$

Phototropism, the desire of the tree to grow toward light, occurs in real trees from a variation in growth rate of the sides of a branch that cause it to bend toward the light. Our model does not allow the growth rate to vary along a single branch, so this effect is accounted for in our model by enhancing two related processes. First, the simulation kills limbs that do not return sufficient photosynthetic nutrition. Second, it increases the growth rate of branches whose leaves return more nutrients.
The shape and size of leaves affects the shape and size of tree crowns [16]. Leaves are represented by polygons that sprout from limbs at fixed intervals where their radius is sufficiently small. The leaves are oriented about their stem axis to face up.

Compute growth rate: The growth rate g_i indicates how much the tree should increase the length of limb i. The growth rate is initialized with some minimum growth (such as $g_i = 0.1$). The rate is then incremented by the center-of-mass offset value

[4] This model neglects the indirect light received from diffuse interreflection and transmission.
[5] Greene used an 80%/20% sun-to-sky weighting [11]. However, Horn shows that the nutrients from photosynthesis remains fairly constant from 20% to 100% of full sunlight intensity [16]. While trees may grow toward the brightest point in the sky, direct sunlight is not necessary for effective photosynthesis.

(such as $g_i = g_i + 0.45 + 0.45o_i$) for a maximum total of 1. The rate is also incremented slightly by the branch photosynthesis nutrition $g_i = g_i + p_i$, which can push it above 1.

The total growth rate for the branch and its children is computed recursively as

$$G_i = g_i + \sum_{j \in J_i} G_j. \tag{15}$$

The ratio g_i/G_i dictates the limb's allocation of new growth resources. The rest it must pass on to its children.

Grow branches: Growth begins at the trunk. Each increase in the trunk radius allows it to support more weight. The first priority is that the trunk be allowed to support its own weight. The remaining residual mass is passed on to the branches to limit their growth.

If the current limb is the trunk, then the radius of its base increases by an amount proportional to the ratio of nutrition returned by photosynthesis to the mass of the tree:

$$R_0' = R_0 + \alpha \text{photosynthesis} \frac{m_0}{M_0}. \tag{16}$$

(The prime indicates the post-growth variable.) The extra mass supported by the trunk's growth is given by

$$e_0 = \text{supports}(R_0') - M_0. \tag{17}$$

If the current limb is not the trunk, then the radius of the current limb increases to support the extra mass made available by the trunk's growth

$$R_i' = R_i + \text{needstosupport}(M_i + e_i). \tag{18}$$

Enough of the extra mass needs to be allocated to expanding the current branch to its new radius. Hence the extra mass is reduced as

$$e_i' = e_i - \Delta m_i, \tag{19}$$

where $\Delta m_i = m_i' - m_i$ is the change in the mass of the branch due to the increased base radius. This increase in mass only accounts for the change in base radius and does not lengthen the branch.

The mass of the branch is further increased by extending the branch's length proportionally to its growth rate:

$$m_i' = m_i' + \frac{g_i}{G_i} e_i'. \tag{20}$$

Given the new branch mass m_i', the new branch length l_i' is

$$l_i' = \frac{3m_i'}{\pi R_i'^2}. \tag{21}$$

The rest of the mass is pass along to the child branches:

$$e_j = \frac{G_j}{G_i} e_i', \tag{22}$$

for each child branch $j \in J_i$.

5 Results

The tree simulation was implemented in an interactive multiwindow system, using the polygon-rendering and user-interaction tools available in the OpenGL, GLUT and XForms libraries.

Figure 3 demonstrates the tree growing system on a 15-year-old apple tree. The tree can be displayed with (summer) or without (winter) leaves. The branching structure can be drawn with cones or lines, and may be shaded. The centroid in other figures is a red sphere whose volume is proportionate to the volume of the tree. The hand button allows the viewer to rotate the view. The scissors allows the user to prune a selected branch, and the arrow allows the user to investigate the structural properties of a given branch.

The menu consists of a Grow button that can cause the tree to grow indefinitely or incrementally in steps of one year. The Genotype button activates the Genotype window that allows the user to set the various genetic settings for the tree, including the branching structure and angles. One can also set the structural properties of the wood.

The Photosynthesis button opens a window that allows the user to change when photosynthesis starts to affect the tree shape and the size of the leaves. The Photosynthesis graphical window displays an item buffer [30] of the leaves. Each leaf is rendered with a color corresponding to the twig index. This allows the simulation to determine which leaves are visible from the light source. The simulation increments the photosynthesis data for each twig for every pixel of its color in the photosynthesis graphical window. We have found that this technique is faster than previous ray-casting techniques for measuring photosynthesis [21] because of its use of existing polygon-projection hardware. We have also found that a resolution of 256^2 sufficient for identifying illuminated leaves.

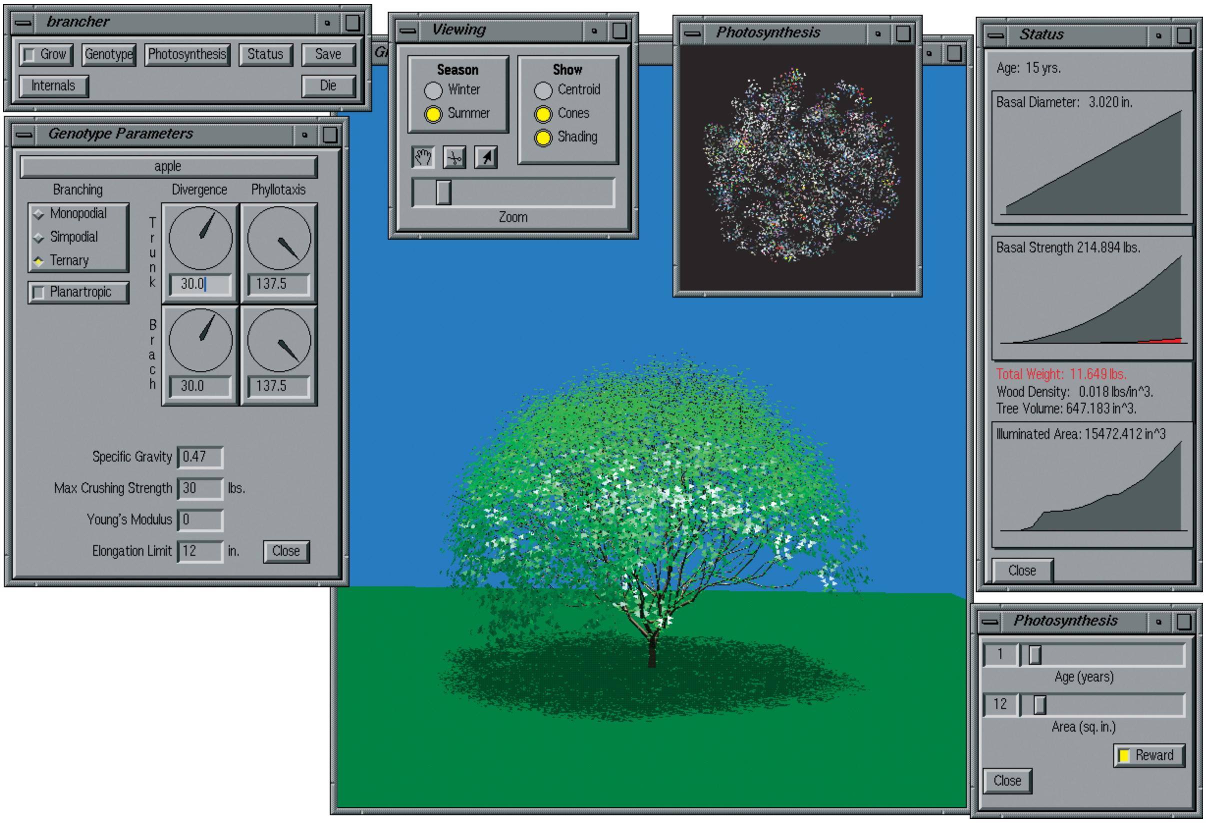


Fig. 3. Tree-growth simulation system

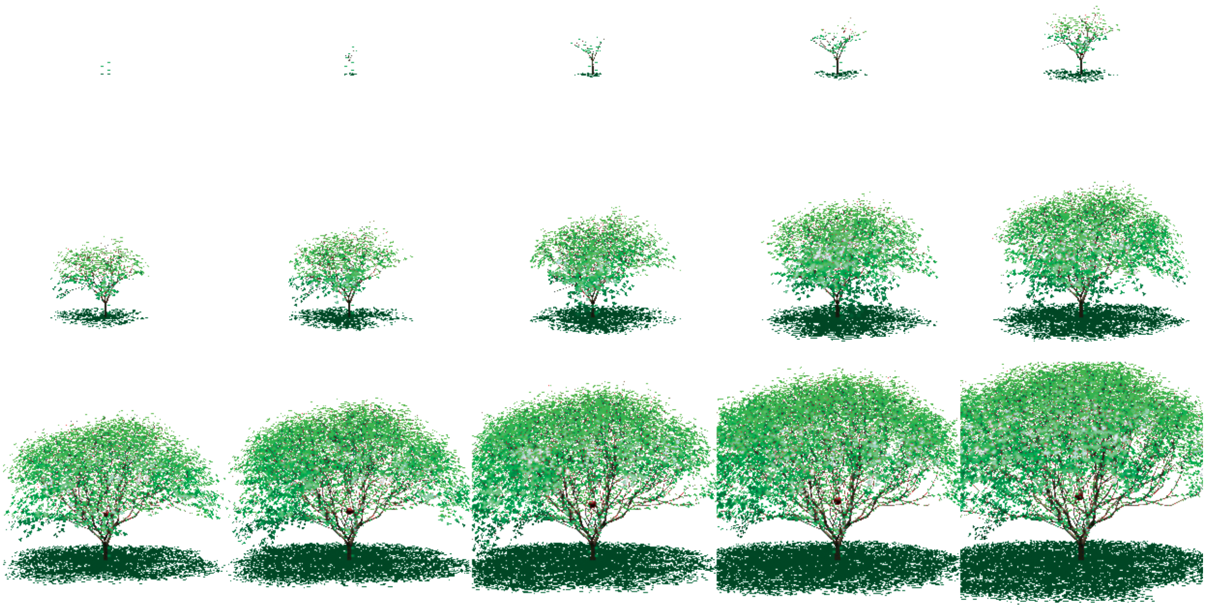
This "photosynthesis buffer" allows the simulation to quickly determine which twigs are most productive. The edges of this window form a natural aperture of available light such as a tree might receive if growing in a forest clearing. Obstacles can be placed in the scene that obscure this window's view, thereby simulating various phototropic effects that control tree shape.

Real leaves are often translucent and much of a tree's photosynthetic nutrition comes from secondary light. This could be approximated by replacing the Z-buffered rendering of opaque leaves with an A-buffered rendering of semi-transparent leaves. Such an implementation would need to modify the A-buffer to return sorted lists of the leaves' twig indices instead of just alpha-blended colors.

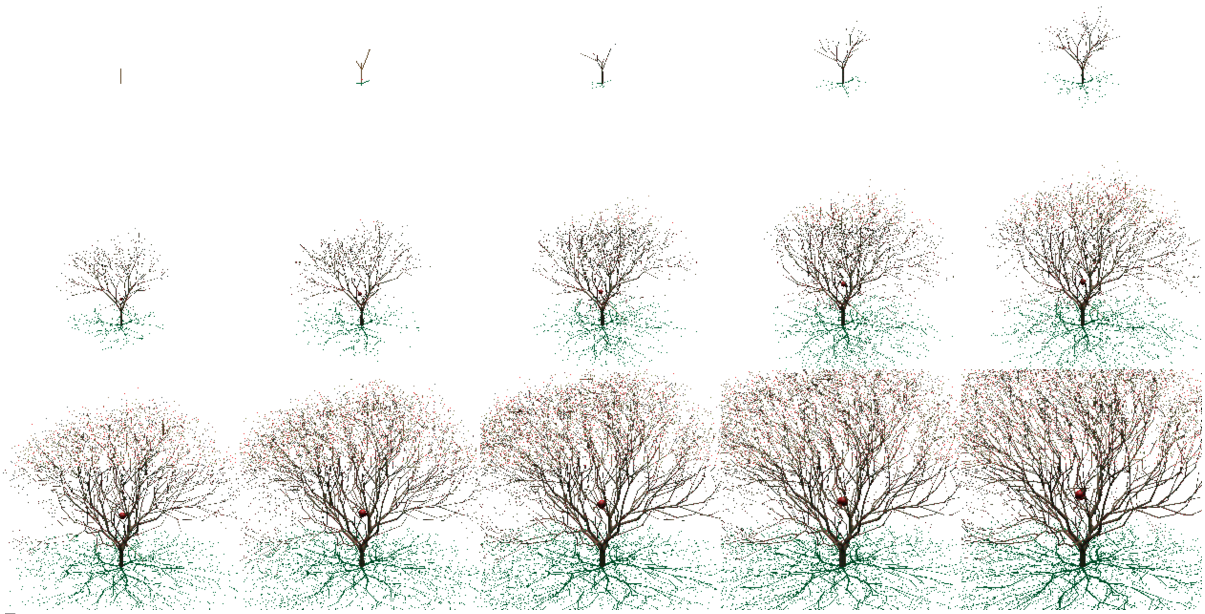
Leaves often orient themselves toward a light source. Our simulation approximates incoming light with a single light position and oriented the leaves to face this position to maximize potential photosynthesis return and accentuate the effects of phototropism. Future implementations might track the sun's arc across the sky to detect more subtle phototropic effects.

The status window tracks the structural properties of the tree from sprout. The diameter of the trunk increases linearly, which causes the strength of its trunk to increase quadratically. The volume of the tree increases cubically, and is displayed using the wood density as weight in red under the strength curve. If the cubic weight curve ever overtakes the quadratic strength curve, the tree collapses under its own weight. Phototropism increases non-linearly based on the visibility of the leaves.

The growth of this tree is shown in Fig. 4 with leaves and in Fig. 5 without. The centroid remains over the trunk because of the regulation of growth. The photosynthesis causes the tree to shed its shaded



4



5

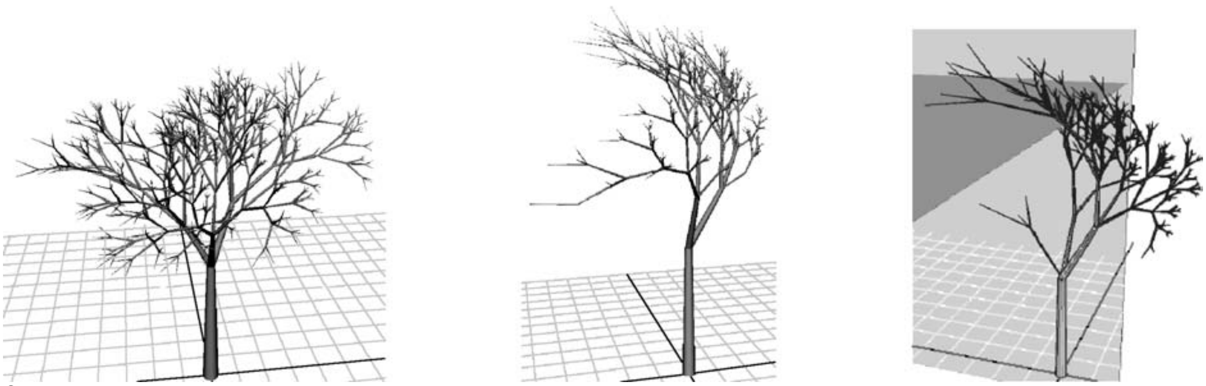
Fig. 4. Fifteen summers of an apple tree, showing the efficiency of its canopy structure

Fig. 5. Fifteen winters of an apple tree, showing its branching structure

limbs, so it can invest more resources to its sunnier limbs.

By making the growth interactive, the user can prune branches during the developmental process. Figure 6 shows a tree fighting to center its mass over the trunk after one of its main branches was either lopped or shaded by an obstruction.

Figure 7 demonstrates the tree-growth algorithm against a real-world situation. A tree in Washington State University's Glen Terrell Mall was selected and



6

7

Fig. 6. The same tree developing with its left branch intact (*left*) pruned (*center*) and shadowed (*right*)
Fig. 7. Real (*left*) and simulated (*right*) trees in the WSU Terrell Mall

photographed. This tree grew away from the building seeking sunlight, then once overtaking the building grew over it to balance its center of mass. Our simulation chose a similar stategy.

6 Conclusion

Analysis of the mechanical properties of tree growth has provided a simulation yielding reasonable results. The parameters are more physically meaningful than previous techniques, which makes the resulting tree proportions more accurate. The simulation also yielded expected results as trees coped with environmental anomalies such as pruned and shadowed branches.

While the simulation's cones do not photorealistically represent the bending nature of real trees, they do capture their general balanced form and composition, which has been a missing component in previous work. The work of Jirasek and Prusinkiewicz is a step toward modeling the bending of branches [17].

One serious roadblock to the interactive simulation of tree growth is that tree growth is fundamentally an exponential process, since the number of branches grows exponentially. Each growth iteration takes significantly longer to generate than the previous. Nature overcomes this problem with extremely fine-grained, highly scalable, massive parallelism. We estimate based on observation that trees contain about sixteen levels from trunk to twig. Hence, a parallel computer with 64K processors would be necessary to simulate tree growth interactively without experiencing time dilation.

We also implemented the simulation as a plug-in for the Alias | Wavefront Maya modeling system, which was used to integrate the tree with the architectural models in Fig. 7.

6.1 Applications

A structurally accurate tree model would find many applications, particularly in image synthesis, tree models for dendrochronology and mensuration, and in the simulation of forest succession.

Whereas the dinosaurs in *Jurassic Park*, its sequel, and *Dinosaur* were computer models, their heavily-forested environments were real. Physical interactions between the dinosaurs and their surroundings were necessary for the proper suspension of disbelief. The combination of synthetic dinosaurs with a real forest required carefully scripting and meticulously retouching. A highly realistic tree model would eliminate much of this work (and cost) by allowing designers to create realistic geometric models of forests that could interact more completely with computer generated characters.

Dendrochronology discerns past events, particularly weather, from the study of the tree's annual growth rings [26]. Saving all of the tree surfaces created during the growth of the tree yields a "solid" representation of the tree. Slicing such a solid representation yields a model of the tree rings resulting from the tree's annual growth cycle. Since the growth rate of the tree is affected by balance and available light, certain dendrochronological elements of the tree's simulated history are revealed by this growth-ring model.

Mensuration estimates various forestry properties, such as lumber output, through the analysis of easily measured tree dimensions, such as trunk diameter at breast height [2]. Currently these models are constructed through destructive sampling techniques. As tree models become more physically based and more accurately represent real tree structure, their dimensions also become more meaningful. This paper's growth model utilizes these techniques to increase its accuracy, and in turn, returns a less destructive geometric model of mature trees to forest science for more accurate mensuration.

Forest succession is the complex process controlling the composition of trees in a forest. As the number of national "old-growth" forests declines, some have considered the task of recreating old-growth effects in a new forest [19]. Current measurements from existing old-growth forests could be used to simulate an old-growth forest through selective human intervention in a new forest. The succession process of old-growth forests is a topic of current research. Detailed computer simulation provides new insight into the delicate equilibrium of old-growth forests. The developmental tree simulation described in this paper is a step toward such study.

6.2 Further research

This first step at mechanically simulating tree growth inspires an immense number of new directions. Many of the mechanical computations operated at the simplest level of statics analysis. More rigorous static analysis of specific tree structural members will lead to improved accuracy.

The most immediate next step is to simulate other structural forces, specifically wind. Trees at the edge of a forest grow shorter to reduce the moment of rotation on the trunk caused by wind resistance. Trees in the middle of a forest are shielded from the wind by the perimeter trees, and are free to grow much taller and fuller. When a road is cut through a forest, the force of wind resistance is immediately introduced to middle trees, which may then topple.

The current system supports single-tree growth. Simulating the simultaneous development of a forest of trees as they battle for resources should result in further insight into the forest-succession process.

Acknowledgements. This research was performed in the Image Research Laboratory in the School of Electrical Engineering and Computer Science at Washington State University. This project was supported in part by the Recurrent Modeling Project (NSF CCR-9529809) to determine what part of tree development is not self-similar.

References

1. Arvo J, Kirk D (1988) Modeling plants with environmentally-sensitive automata. In: Proceedings of Ausgraph '88, Melbourne, Australia, 4–8 July 1988. Australasian Computer Graphics Association, Melbourne
2. Avery TE, Burkhart HE (1994) Forest measurements. McGraw-Hill, New York
3. Barzel R, Barr AH (1988) A modeling system based on dynamic constraints. Comput Graph 22(4):179–188
4. Beer FP, Johnston Jr, ER (1984) Vector mechanics for engineers: statics and dynamics. McGraw-Hill, New York
5. Bloomenthal J (1985) Modeling the mighty maple. Comput Graph 19(3):305–311
6. de Reffye P, Edelin C, Francon J, Jaeger M, Puech C (1988) Plant models faithful to botanical structure and development. Comput Graph 22(4):151–158
7. Deussen O, Hanrahan PM, Lintermann B, Mech R, Pharr M, Prusinkiewicz P (1998) Realistic modeling and rendering of plant ecosystems. In: Proceedings of the 25th annual conference on computer graphics and interactive techniques. ACM Press, New York
8. Esau K (1960) Anatomy of seed plants. Wiley, New York
9. Forest Products Laboratory (1974) Wood handbook, wood as an engineering material. Forest Service, U.S. Department of Agriculture
10. Gartner B (1995) Personal communication, 1995. Oregon State University, Department of Forest Products
11. Greene N (1989) Voxel space automata: modeling with stochastic growth processes in voxel space. Comput Graph 23(3):175–184
12. Hart J, Baker B (1996) Implicit modeling of tree surfaces. In: Proceedings of the 1996 workshop on implicit surfaces, Eindhoven, The Netherlands, 7–8 October 1996
13. Hart JC, DeFanti TA (1991) Efficient antialiased rendering of 3-D linear fractals. In: Proceedings of the 18th annual conference on computer graphics and interactive techniques. ACM Press, New York
14. Hart JC (1995) Implicit representations of rough surfaces. In: First international workshop on implicit surfaces, Grenoble, France, 18–19 April 1995
15. Holton M (1994) Strands, gravity and botanical tree imagery. Comput Graph Forum 13(1):57–67
16. Horn HS (1971) The adaptive geometry of trees. Princeton University Press, Princeton, N.J.
17. Jirasek C, Prusinkiewicz P (1998) A biomechanical model of branch shape in plants. In: Lantin M (ed) Proceedings of the western computer graphics symposium, Whistler, Canada, 23–26 April 1998
18. Jirasek C, Prusinkiewicz P, Moulia B (1999) Integrating biomechanics into developmental plant models expressed using L-systems. In: Tigges M, Baranoski G (eds) Proceedings of western computer graphics symposium, Banff, Canada, 21–24 March 1999
19. Lorimer ND, Haight RG, Leary RA (1994) The fractal forest: fractal geometry and applications in forest science. Technical report NC-170, USDA Forest Service North Central Forest Experiment Station
20. Mattheck C (1991) Trees: the mechanical design. Springer, Berlin Heidelberg New York
21. Měch R, Prusinkiewicz P (1996) Visual models of plants interacting with their environment. In: Proceedings of the 23st annual conference on computer graphics and interactive techniques. ACM Press, New York
22. Oppenheimer PE (1986) Real time design and animation of fractal plants and trees. Comput Graph 20(4):55–64
23. Prusinkiewicz P, Lindenmayer A (1990) The algorithmic beauty of plants. Springer, Berlin Heidelberg New York
24. Prusinkiewicz P, Lindenmayer A, Hanan J (1988) Developmental models of herbaceous plants for computer imagery purposes. Comput Graph 22(4):141–150
25. Prusinkiewicz P, James M, Měch R (1994) Synthetic topiary. In: Proceedings of the 21st annual conference on computer graphics and interactive techniques. ACM Press, New York
26. Schweingruber FH (1993) Trees and wood in dendrochronology. Springer, Berlin Heidelberg New York, 1993
27. Thompson DW (1961) On growth and form, abridged ed. Cambridge University Press, Cambridge, U.K.
28. Viennot XG, Eyrolles G, Janey N, Arques D (1989) Combinatorial analysis of ramified patterns and computer imagery of trees. Comput Graph 23(3):31–40
29. Weber J, Penn J (1995) Creation and rendering of realistic trees. In: Proceedings of the 22nd annual conference on computer graphics and interactive techniques. ACM Press, New York
30. Weghorst H, Hooper G, Greenberg D (1984) Improved computational methods for ray tracing. ACM Trans Graph 3(1):52–69

Photographs of the authors and their biographies are given on the next page.

JOHN C. HART is an Associate Professor in the Department of Computer Science at the University of Illinois, and before that was an Associate Professor in the School of Electrical Engineering and Computer Science at Washington State University. His interests in computer graphics include procedural modeling and shading, implicit surfaces, ray tracing, computational topology and graphics hardware abuse. Hart is the Editor-in-Chief of ACM Transactions on Graphics, a co-author of "Real-Time Shading," a contributing author of "Texturing and Modeling: A Procedural Approach (Third Edition)" and an executive producer of the documentary "The Story of Computer Graphics."

BRENT BAKER graduated from the University of Oregon with a BS in Computer and Information Science in 1985, and an MS in 1989. He's currently working for Indus Corp. as a contractor for the EPA writing software to model the behavior of territorial species in a defined landscape. He also writes software for the University of Rhode Island Graduate School of Oceanography on the DODS project. In his spare time he's an avid woodworker and a mediocre skier.

JEYPRAKASH MICHAELRAJ works as a Senior SW Developer for Alias Wavefront, Toronto, Canada where he is mainly developing tools for the modeling component of Maya, the 3D SW package. He completed his Masters from WSU, Pullman, under the guidance of Prof. John C Hart, and has a B.S in Computer Science from Anna University, Madras, India.