Self-organizing tree models for image synthesis

Article · July 2009

DOI: 10.1145/1576246.1531364 · Source: DBLP

7 authors, including:

Wojtek Palubicki
Adam Mickiewicz University
29 PUBLICATIONS **655** CITATIONS
SEE PROFILE

Adam Runions
The University of Calgary
54 PUBLICATIONS **2,460** CITATIONS
SEE PROFILE

Radomir Mech
Adobe Inc.
87 PUBLICATIONS **5,928** CITATIONS
SEE PROFILE

Przemyslaw Prusinkiewicz
The University of Calgary
245 PUBLICATIONS **16,431** CITATIONS
SEE PROFILE

Self-organizing tree models for image synthesis

Wojciech Pałubicki[1] Kipp Horel[1] Steven Longay[1] Adam Runions[1] Brendan Lane[1]
Radomír Měch[2] Przemyslaw Prusinkiewicz[1]

[1]University of Calgary [2]Adobe Systems Incorporated

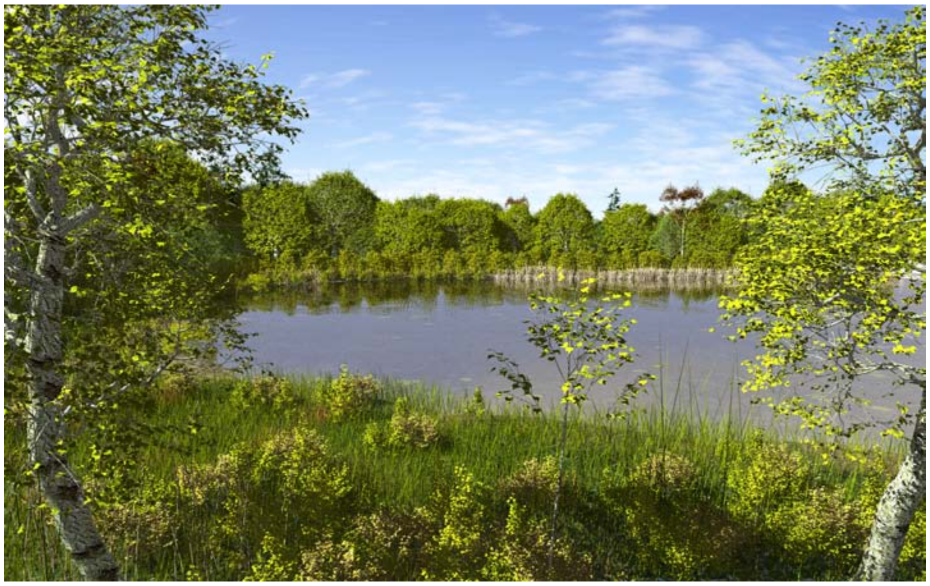


Figure 1: *Synthetic landscape with self-organizing trees.*

Abstract

WE PRESENT a method for generating realistic models of temperate-climate trees and shrubs. This method is based on the biological hypothesis that the form of a developing tree emerges from a self-organizing process dominated by the competition of buds and branches for light or space, and regulated by internal signaling mechanisms. Simulations of this process robustly generate a wide range of realistic trees and bushes. The generated forms can be controlled with a variety of interactive techniques, including procedural brushes, sketching, and editing operations such as pruning and bending of branches. We illustrate the usefulness and versatility of the proposed method with diverse tree models, forest scenes, animations of tree development, and examples of combined interactive-procedural tree modeling.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.6.8 [Simulation and Modeling]: Types of simulation—Visual; J.3 [Life and Medical Sciences]: Biology.

Keywords: generative tree model, tree development, apical control, bud fate, emergence, interactive-procedural modeling.

[1]email: {wppalubi|kjhorel|slongay|runionsa|laneb|pwp}@cpsc.ucalgary.ca
[2]email: rmech@adobe.com

1 Introduction

Most methods for modeling trees are based on the assumption that trees have a repetitive, recursive structure. This assumption is consistent with the established notion of architectural tree models [Hallé et al. 1978], according to which the form of a tree is largely determined by its branching pattern. However, the genetically-controlled distribution of buds from which branches arise is "remarkably similar in all trees, be they oaks, columnar cacti, or palms" [Sachs and Novoplansky 1995]. Consequently, branching patterns provide "an important, but partial, picture" (*ibid.*). This is particularly evident in temperate-climate trees and shrubs. They commonly produce many buds, most of which do not develop into lasting branches. Those that do are arranged in a locally irregular, relatively nondescript manner, which nevertheless yields well-balanced, recognizable whole-tree forms. A clear concept of what controls the fate of buds and branches is thus necessary to provide a fundamental understanding of the development and form of trees, and a firm basis for tree modeling.

A view of tree form complementing architectural tree models was proposed by Sachs and Novoplansky [1995]. They emphasized the self-organizing character of tree development, in which "every bud and branch is constantly compared with alternatives that could have the same role in overall tree structure" [Sachs 2004]. In this paper, we propose a modeling method that exploits this concept of self-organization. The method integrates three elements of tree development: local control of branching geometry, competition of buds and branches for space or light, and regulation of this competition through an internal signaling mechanism. We show that:

- The reliance on self-organization simplifies the modeling process, because well-balanced branch distributions emerge automatically from the generative algorithm;
- The integration of architectural and self-organizing components improves the visual realism of tree models, compared to previous methods;

- The incorporation of signaling makes it possible to simulate apical control of development, and consequently capture a wide range of tree forms, both *excurrent* (with a conspicuous tree trunk) and *decurrent* (without a pronounced trunk);
- This range of modeled forms can be further extended with the use of procedural brushes, which allow the user to control the form of generated trees by interactively manipulating their environment.

Overall, the proposed method makes it possible to generate a wide range of highly realistic trees (Fig. 1), and control their form using a small number of parameters or interactive manipulations.

The paper is organized as follows. We begin with a review of previous work on tree modeling (Section 2) and the relevant terminology (Section 3). We then describe the key components of our modeling method (Section 4), extend it with interactive control (Section 5) and outline our implementation (Section 6). Finally, we discuss the main contributions of the paper, and present topics for further research (Section 7).

2 Previous work

The origins of computer modeling of trees can be traced to the seminal papers of Ulam [1962] and Honda [1971]. Honda considered a tree as an explicitly-defined, recursive structure, characterized by parameters such as branching angles and the ratio of module sizes at consecutive recursion levels. This view lent itself well to recursive generative algorithms [Aono and Kunii 1984; Bloomenthal 1985; Reeves and Blau 1985; Oppenheimer 1986; Weber and Penn 1995; Lintermann and Deussen 1999; Prusinkiewicz et al. 2001]. In contrast, Ulam considered trees as self-organizing structures, in which branching patterns emerge from the competition of individual units for space. This idea, originally expressed in terms of 2D cellular automata, was extended to 3D voxel spaces and augmented with constructs needed for realistic image synthesis by Arvo and Kirk [1988], Greene [1989] and, more recently, Beneš and Millan [2002], Pałubicki [2007], and Bornhofen and Lattaud [2008]. All of these authors emphasized the ability of their models to adapt to the surrounding space, be it in the form of obstacles or support for growth. Algorithms using continuous representations of space were proposed by Cohen [1967], and in computer graphics by Rodkaew [2003] and Runions et al. [2007].

The theoretical analysis of branching patterns by Borchert and Slade [1981] provides an insight into the key differences between patterns produced with recursive and self-organizing techniques. Borchert and Slade observed that in a repetitive branching pattern with internodes of constant size, the number of internodes grows exponentially with age (recursion depth), but the total volume they may occupy grows only as the third power of age. Consequently, the density of elements increases indefinitely over time. Honda addressed this problem by decreasing the size of modules with the level of recursion while preserving the repetitive topology at all recursion levels. The mathematical essence of this approach is captured by fractal trees [Mandelbrot 1982] (Fig. 2, left). In contrast, Ulam assumed modules of constant size, but allowed terminal segments to have different fates, some giving rise to new modules, others not, depending on the results of the competition for space (Fig. 2, right). In the light of observations by Sachs and Novoplansky, the concept of self-organization inherent in Ulam's approach better captures the essence of tree development. We have adopted it as the conceptual cornerstone of our method.

The fate of different segments in Ulam's model can be viewed as an abstraction of the fate of buds in trees: some may become active and produce new branches, while others produce flowers and fruits, remain dormant, or abort [de Reffye et al. 1988; Bell 1991].

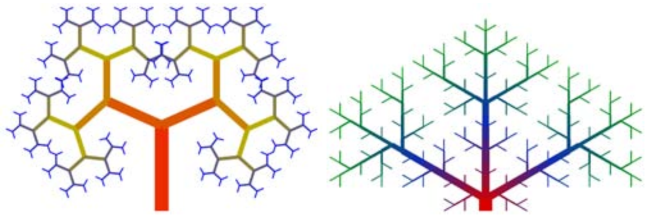


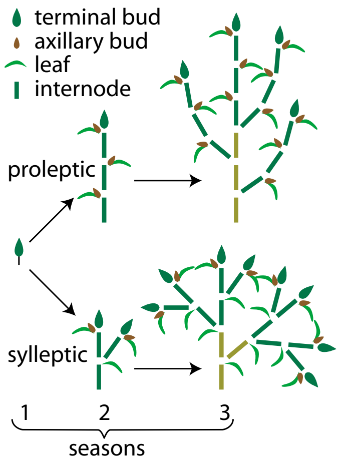
Figure 2: *Contrasting approaches to tree modeling. Left: Fractal tree generation with a recursive procedure. Right: Self-organizing tree development. In the latter case, each terminal segment attempts to create new branches growing forward, 60° to the left, or 60° to the right. A segment is added to the structure iff it does not collide with another existing or attempted segment. Colors indicate generation steps.*

Buds with different fates have previously been incorporated into computer models of trees using stochastic estimators [de Reffye et al. 1988; Prusinkiewicz et al. 1994; Costes et al. 2008] or environmental input [Borchert and Honda 1984; Chiba et al. 1994; Měch and Prusinkiewicz 1996; Beneš and Millan 2002]. Our paper advances this class of models beyond the visually rudimentary models described previously.

In order to be useful in image synthesis applications, tree modeling methods must strike a balance between procedural generation and user control of the modeled structures. In early recursive models (*e.g.* [Honda 1971; Oppenheimer 1986]), this control was limited to local aspects of the branching structure, such as branching angles and ratios of tree branch lengths, from which the overall form emerged. Global-to-local techniques, pioneered by Reeves and Blau [1985] and further developed by Weber and Penn [1995], Lintermann and Deussen [1999], and Prusinkiewicz et al. [2001], allow the modeler to explicitly specify the silhouette and density of branches. This makes it possible to create highly realistic models in capable hands. However, the user has to control a large number of parameters, and it is difficult to model older trees with irregular yet harmoniously balanced crowns. We show that self-organization addresses this problem, by allowing tree branches to autonomously adapt to the available space.

Power et al. [1999] and Boudon et al. [2003] pioneered direct manipulation of plant models, by allowing the user to prune and bend branches between steps of the generative process. Even tighter control was achieved in sketch-based models of herbaceous plants and small trees [Ijiri et al. 2005; Ijiri et al. 2006a; Okabe et al. 2005; Anastacio et al. 2006]. Extensions of sketch-based methods to trees with large, procedurally generated branch systems were proposed by Ijiri et al. [2006b] and Zakaria and Shukri [2007], and brought to a higher level of realism by Wither et al. [2009]. The latter work is complementary to ours in that it provides sketch-based control of plausible tree models, but uses recursive rather than self-organizing generative procedures. We also note that the rapidly growing body of work on the modeling of trees from scanned data ([Côté et al. 2009]) or photographs (*e.g.* [Neubert et al. 2007]) is related to the control of procedural models, since a sketch made by the user may complement or replace the input data.

Models of trees interacting with their environment can be further controlled by modifying the environment itself. In the seminal work of Arvo and Kirk [1988] and Greene [1989], the form of climbing plants and vines was guided by their supporting structures. A later model of root development [Měch and Prusinkiewicz 1996] was guided by the distribution of water in the soil specified with a paint program. In L-system models of topiary trees [Prusinkiewicz et al. 1994], branches that grew outside a predefined pruning volume were cut off, promoting outgrowth of nearby lateral branches that filled this volume. Likewise, the space colonization algorithm [Runions



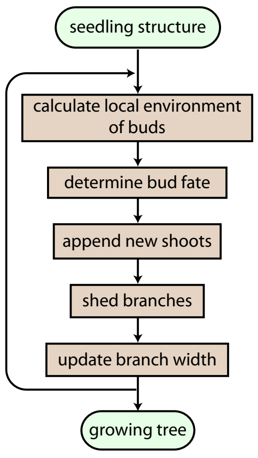


Figure 3: *Left: Proleptic vs. sylleptic tree development. Dark green indicates metamers produced in the current season. Right: Organization of computation.*

et al. 2007] allowed the user to control the form of trees by constraining the space in which they grew. We employ both direct manipulation and environmental input to control the form of self-organizing trees.

3 Terminology

We describe a tree as a hierarchically organized modular structure [de Reffye et al. 1988; Bell 1991; Barthélémy and Caraglio 2007]. The point at which one or more leaves are or were attached to a stem is termed a *node*. The part of the stem between two nodes is an *internode*. A *lateral bud* is created in the axil of each leaf, *i.e.*, the angular space between the stem and the petiole of the supporting leaf. An internode with attached leaf and bud forms a *metamer*.

A bud may have four different *fates*: produce a new metamer, produce a flower, remain dormant (retaining the possibility of growing in the future), or abort. Metamers can be produced *continuously* or *rhythmically*. The growth of temperate-climate trees is rhythmic, regulated by the cycle of seasons. A sequence of metamers created in a single spurt (spring in temperate-climate trees) forms a *shoot*. The shoot *axis* is produced by the *terminal bud*, situated at the shoot's end. The main trunk develops from the terminal bud in the seedling and has order 0. A lateral bud on an axis of order n becomes the terminal bud of the axis of order $n + 1$ it produces.

Shoots produced in consecutive years may be arranged in a linear sequence, forming a *monopodial* branching structure. Alternatively, the main thrust of development can be transferred to lateral branches, with the terminal bud producing a flower or aborting. This results in *sympodial* branching. New branches may be produced *sylleptically*, *i.e.*, in the same season in which the stem supporting them has been formed, or *proleptically*, which means that the bud formed in year m enters a resting period and cannot produce a new branch until year $m + 1$ (Fig. 3, left). Typical branches of temperate-climate trees are produced proleptically. The distinction between prolepsis/syllepsis is important from the modeling perspective, as it affects the timing of decisions regarding new shoots.

Trees employ different forms of information flow to develop in a coordinated manner. We use the term *endogenous* flow or *signaling* to denote information propagation within the tree structure, and *exogenous* flow to denote information exchange through the space in which the tree grows [Měch and Prusinkiewicz 1996]. Endogenous flow may be *acropetal* (from the base of the tree towards its extremities), *basipetal* (from the extremities to the base), or bidirectional.

4 The modeling method

We create a tree structure by simulating its development. This is accomplished in a cycle of interactions between the tree and its environment (Fig. 3, right).

4.1 Calculation of the environmental input

In each iteration, the environmental process estimates the availability or quality of the space surrounding each bud (a number Q) and the optimal direction of shoot growth (a vector $\vec{V}$). To calculate these values, we implemented two simple methods that are faster than previous, more accurate methods (*e.g.* [Soler et al. 2003; Cieslak et al. 2008]).

Space colonization. This method is related to earlier space colonization algorithms for generating leaf venation patterns [Runions et al. 2005] and trees [Runions et al. 2007]. The key modification is the introduction of buds as the only locations that can produce new branches. We assume that each bud is surrounded by a spherical occupancy zone of radius ρ and has a conical perception volume characterized by the perception angle θ and distance r (Fig. 4, left). Typical values for these parameters are: $\theta \approx 90°$, $\rho = 2$ internode lengths, and $r = 4$ to 6 internode lengths. The space available for tree growth is represented by a set S of marker points M. In the simplest case, these points are generated algorithmically, with uniform distribution, at the beginning of the simulation. In each iteration of the simulation, markers within the occupancy zone of any bud are deleted from the set S. The buds then compete for the remaining points (Fig. 4, right). A marker M within the perception volume of a single bud A is associated with A. A marker M within the perception volume of several buds is associated with the closest of these buds. A bud A has space for growth ($Q = 1$) if the set $S(A)$ of markers associated with A is nonempty; otherwise bud A has no available space ($Q = 0$). The optimal growth direction $\vec{V}$ is calculated as the normalized sum of the normalized vectors towards all markers in $S(A)$ [Runions et al. 2007].

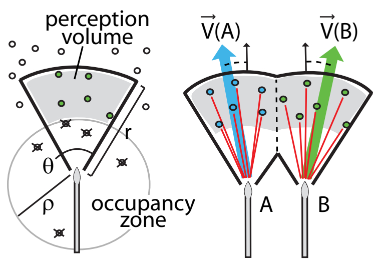


Figure 4: *Evaluation of space available for colonization. Left: Perception volume and occupancy zone associated with a bud. Right: Competition for space between two buds.*

Shadow propagation. As an alternative to space colonization, we use a method proposed by Pałubicki [2007] (see also [Bornhofen and Lattaud 2008]) to compute a coarse estimate of the exposure of each bud to light. The space is divided into a grid of voxels. Associated with each voxel (i, j, k) is a "shadow value" s, initially set to zero. A bud A located in voxel (I, J, K) creates a pyramidal penumbra that propagates to the voxels underneath (Fig. 5). The affected voxels have indices $(i, j, k) = (I \pm p, J - q, K \pm p)$, where $q = 0, 1, \ldots, q_{max}$ and $p = 0, 1, \ldots, q$. Due to the presence of bud A, the shadow value s in each affected voxel is increased by $\Delta s = ab^{-q}$, where $a > 0$ and $b > 1$ are user-defined parameters. The light exposure Q of a sample bud B in voxel (i, j, k) is then calculated as $Q = \max(C - s + a, 0)$, where C is

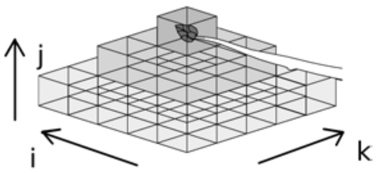


Figure 5: *Shadow propagation model.*

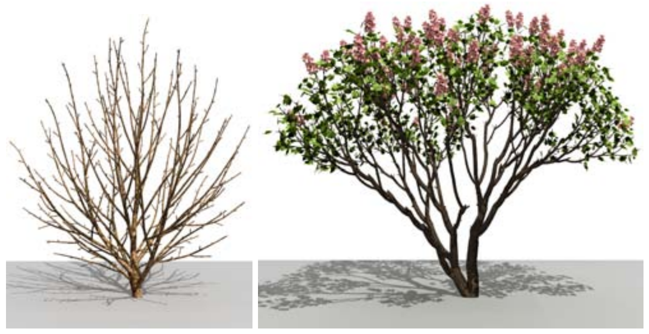


Figure 6: *Decurrent self-organizing shrubs resulting from sylleptic development (left) and proleptic development with sympodial branching (right).*

a constant representing full exposure. The addition of the term a corresponds to the assumption that a bud does not cast a shadow on itself. The optimal growth direction $\vec{V}$ is calculated either as the negative gradient of the shadow value, or by considering all voxels neighboring (i, j, k) within the perception volume of the bud and selecting the voxel with the lowest shadow value.

4.2 Calculation of bud fate

In this phase we use the environmental input — the availability of free space or light — to determine which buds will produce new shoots and how large they will be. The simplest case is sylleptic development, in which all buds are equivalent. This process produces decurrent bushes (Fig. 6, left). Decurrent forms also result from proleptic development with sympodial branching, in which only lateral buds can give raise to new shoots in the next season, while terminal buds produce flowering structures or die (Fig. 6, right).

Development of excurrent forms requires *apical control*, *i.e.*, suppression of lateral branch growth by vigorously developing shoots above [Bell 1991]. It is not known whether apical control in nature is exerted through competition for resources, hormonal control, or both [Bangerth 1989]. We implemented two simple resource allocation models, described below.

Extended Borchert-Honda (BH) model. The BH model was originally proposed as a purely endogenous mechanism that regulates the extent of branching by controlling the distribution of a growth-inducing resource to buds [Borchert and Honda 1984]. We have adapted the BH model to self-organizing trees by using the amount of light received by the buds to guide the distribution of the resource. The algorithm operates in two passes. In the first pass, information about the amount of light Q that reaches the buds flows basipetally, and its cumulative values are stored within the internodes (Q_m, Q_l and Q in Fig. 8a, left). The cumulative value at the

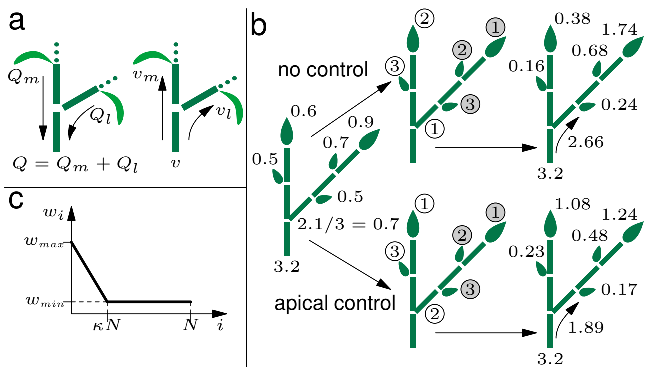


Figure 8: *(a) Basipetal and acropetal information flow through a branching point in the extended BH model. (b) Example of resource allocation by the priority model of bud fate control. Left: Propagation of light exposure and numbers of buds. Center: Ordering of branch axes. Right: The resulting resource allocation for weights* $w_1 = 2$, $w_2 = 1$, $w_3 = 0.5$. *(c) The form of the weight function.*

base determines the amount of resource that will be distributed in the second, acropetal pass: $v_{base} = \alpha Q_{base}$, where α is a coefficient of proportionality. The amount of resource v reaching a branching point is distributed between the continuing main axis (v_m) and the lateral branch (v_l, see Fig. 8a, right) using the equations

$$v_m = v\frac{\lambda Q_m}{\lambda Q_m + (1-\lambda)Q_l} \quad \text{and} \quad v_l = v\frac{(1-\lambda)Q_l}{\lambda Q_m + (1-\lambda)Q_l}.$$

Parameter $\lambda \in [0, 1]$ controls whether resource allocation is biased towards the main axis ($\lambda > 0.5$), not biased ($\lambda = 0.5$), or biased toward the lateral branch ($\lambda < 0.5$). The integer part of the amount v of the resource reaching a bud determines the number of metamers produced by this bud: $n = \lfloor v \rfloor$. The length l of these internodes is calculated as $l = v/n$. Parameter λ provides a useful control of the decurrent vs. excurrent habit of the generated trees (Fig. 7).

Priority model. The BH model allocates an acropetally flowing resource by considering branching points one at a time. In contrast, the priority model, described below, operates at the level of entire axes. First, information representing the amount of light received by the buds, and the number of buds in the branches, propagates basipetally and is stored at the base of each axis (Fig. 8b, left). The

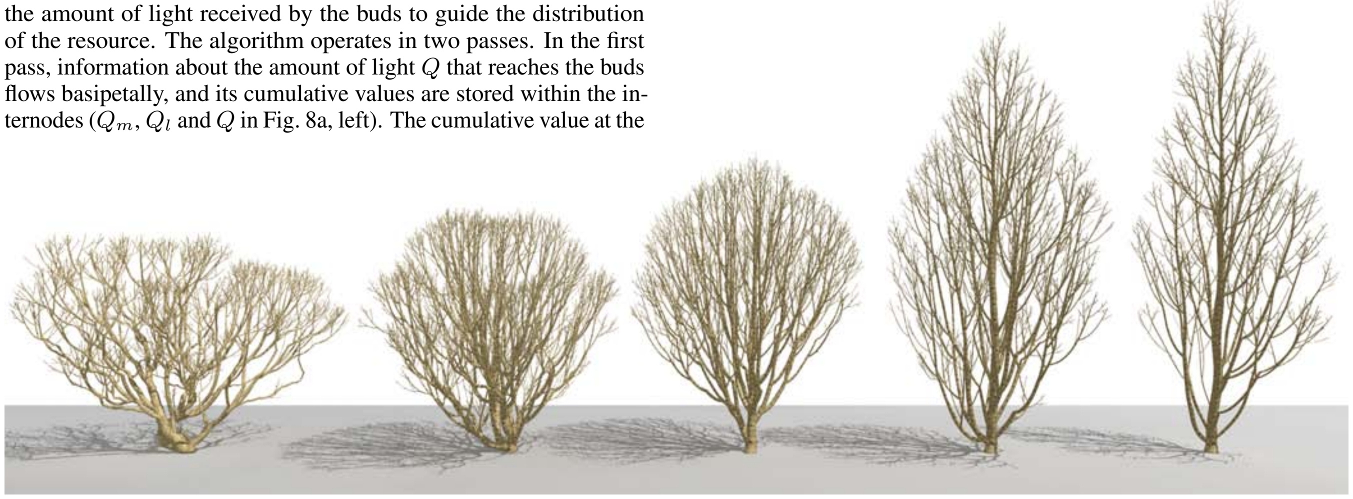


Figure 7: *Progression of tree forms created with the extended Borchert-Honda model of bud fate control. From left to right, the values of parameter* λ *are* $0.46, 0.48, 0.50, 0.52, 0.54$. *All trees modeled with* $\alpha = 2$ *and the same light conditions.*

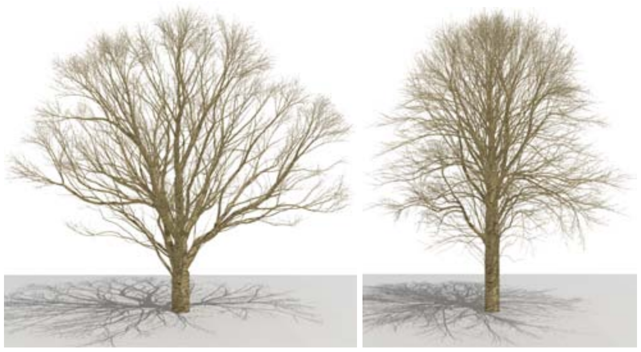


Figure 9: *Sample trees generated with the priority model of bud fate control using weight function shown in Fig. 8c with parameters* $w_{max} = 1$*,* $w_{min} = 0.006$*, and* $\kappa = 0.5$ *(left) or 0.35 (right).*

average amounts of light received by the supported branches (total light divided by the number of buds in a branch, *e.g.* $2.1/3 = 0.7$ in Fig. 8b, left) are then sorted, yielding ordered priority lists associated with each axis (Fig. 8b, center). Individual buds attached to the axis are considered as single-metamer branches. Apical control is simulated by placing the terminal bud at the beginning of this list, irrespective of its light exposure. Finally, the resource is distributed between the supported branches and buds using weights that depend on their position in the priority list (Fig. 8b, right). Specifically,

$$v_i = v \frac{Q_i w_i}{\sum_{j=1}^{N} Q_j w_j}, \quad i = 1, 2, ..., N,$$

where v is the amount of resource flowing into the axis under consideration, N is the number of buds or branches supported by this axis, v_i is the amount of resource allocated to branch i, and $w_1, \ldots, w_N$ are the weights. We have used the piecewise linear function shown in Fig. 8c to select the weights. As in the BH model, the total amount of distributed resource is assumed to be proportional to the cumulative amount of light reaching the base, $v_{base} = \alpha Q_{base}$. The amount of resources reaching each bud determines the number of metamers this bud will produce in the next simulation step.

Assigning large weights to a smaller number of most productive branches (at the beginning of the priority list) results in a more excurrent tree form (Fig. 9). Furthermore, a wide diversity of forms can be obtained by regulating the time at which the apical control is removed in the main stem or the lateral branches (Fig. 10). Removal of apical control is also important in animations of development, since many types of temperate-climate trees have a well defined main axis when young, but progress to more a decurrent form when

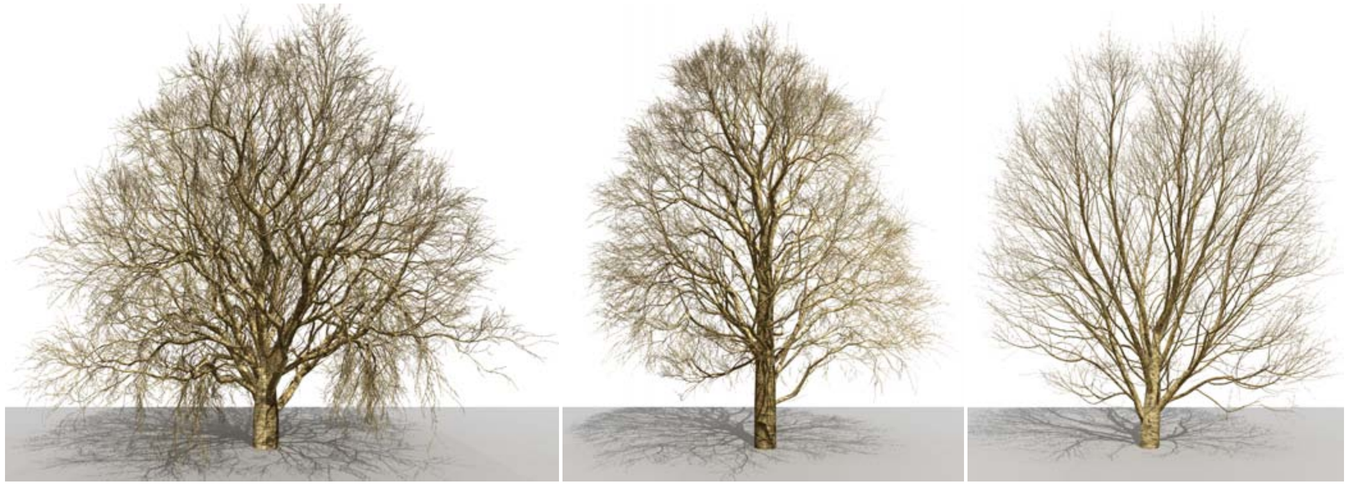


Figure 10: *Sample tree forms created by using the priority model of bud fate control. Left: apical control removed from main stem and lateral branches early in development. Center: apical control limited to the main stem, removed late in development. Right: apical control removed early in the main stem, persistent in first-order lateral branches.*

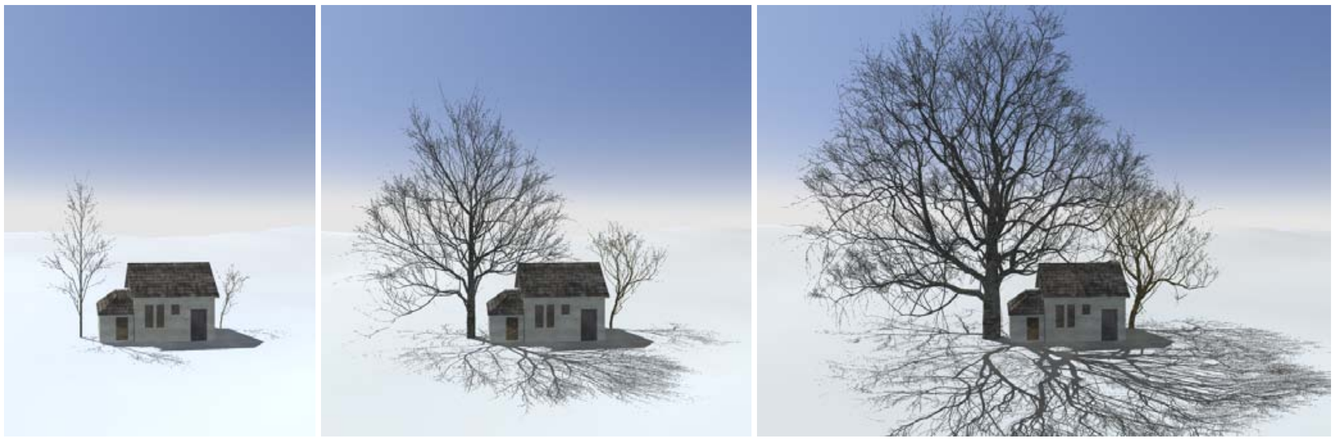


Figure 11: *Snapshots from an animation of tree development simulated with the priority model of bud fate. Apical control was initially present in the tree on the left, then removed in the course of development, resulting in a progression from the excurrent form of the young tree to the decurrent form of the old tree. This simulation also shows the adaptation of self-organizing trees to the presence of other trees and obstacles.*

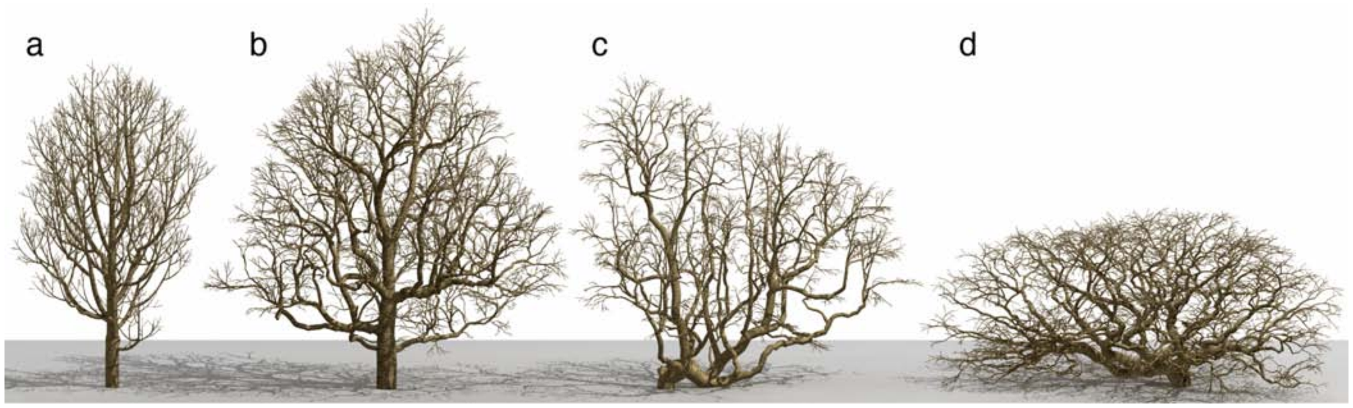


Figure 12: *The impact of tropism η and the preference of growing towards light ξ. (a) Small upward η, small ξ. (b) Small downward η, small ξ. (c) Large downward η, large ξ. (d) Large downward η, small ξ. The fate of buds was controlled using the extended BH model.*

old [Barthélémy and Caraglio 2007]. A sample developmental sequence illustrating this progression is shown in Fig. 11.

4.3 Addition of new shoots

By default, new shoots are issued in the direction pointed to by the buds. Terminal buds have the orientation of their supporting internodes. Lateral buds have orientation defined by phyllotaxis and the branching angle between the bud and its parent internode.

The orientation of consecutive metamers forming a shoot is modified by two factors: the optimal growth direction determined by the environment (vector $\vec{V}$ in Section 4.1) and the tendency to maintain a preferred orientation with respect to gravity (tropism). The actual orientation of a metamer is thus calculated as a weighted sum of three vectors: the default orientation, the optimal growth direction (weight ξ), and a tropism vector (weight η). A change in the tropism direction from pointing upward (Fig. 12a) to downward (Fig. 12b) results in a predictable change in the shape and orientation of branches, which spread more widely in the second case. More interestingly, a strong downward tropism combined with a strong tendency to grow towards light results in decurrent forms with highly gnarled branches (Fig. 12c, d). In all cases, the density of branches is automatically maintained due to the self-organization of the crown. A further variety of forms can be obtained by changing tropism over time (Fig. 13) or with the order of branches (Fig. 20, center).

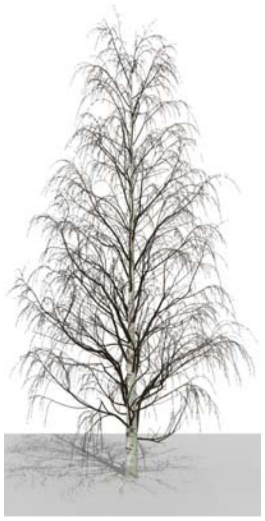


Figure 13: *Tree simulated by increasing the downward tropism near the end of the simulation.*

4.4 Shedding of branches

Shedding of branches is an important component of crown self-organization: for instance, it is the key to the formation of tall boles in developmental tree models (Fig. 14). We simulate branch shedding as proposed by Takenaka [1994] and brought to computer graphics by Měch and Prusinkiewicz [1996]. The total amount of light gathered by a branch is compared with the branch size measured in the number of internodes. If this ratio falls below a specified threshold, the branch is considered a liability for the tree and is shed. The above method is more suitable for models that rely on shadow propagation rather than space colonization. In the latter case, the binary nature of the environmental input ($Q = 1$ or $Q = 0$) would cause branches to be shed immediately after they stop growing.

4.5 Calculation of branch diameter

Branch diameter is a key factor affecting the appearance of trees. We calculate it in a basipetal pass of information flow. Each leaf contributes an initial diameter value. These values are accumulated along tree axes using the formula $d^n = d_1^n + d_2^n$, where d is the diameter of the internode below the branching point, d_1 and d_2 are diameters of the internodes above, and n is a user-defined parameter, usually between 2 and 3 [Macdonald 1983]. This formula is a version of the pipe model [Shinozaki et al. 1964], according to which individual leaves contribute vascular strands, or pipes, that extend towards the base of the tree. Importantly, branch width is not decreased when leaves and branches are shed or pruned. The model thus requires a memory of past leaves and branches.

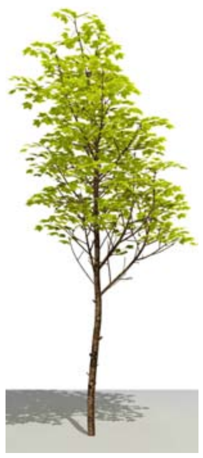


Figure 14: *Tree with a tall bole.*

5 Interactive control of the models

Self-organizing trees lend themselves well to different forms of user input, which provide additional control of tree development and form. We implemented operations for bending and pruning branches, which can be applied between simulation steps, as described by Power et al. [1999]. In addition, we found of particular use a technique for guiding tree development that is based on the manipulation of the environment in which the tree grows, rather than the tree itself.

Guided growth extends the space-colonization model of the environment (Section 4.1) with a 3D procedural brush that dynamically creates markers of free space. The brush is controlled by a pressure-sensitive tablet pen. If a stroke begins at an existing component of the tree (initially, a predefined seedling), its depth determines the depth of the working plane, parallel to the screen, in which the brush is assumed to move [Cohen et al. 1999; Ijiri et al. 2005]. This plane can be changed by rotating the tree and selecting a previously created component. At each iteration of the modeling algorithm (Fig. 3, right), the brush position defines the center of a sphere within which marker points are generated. Light pressure results in focused positioning of the markers and makes it possible to sketch individual

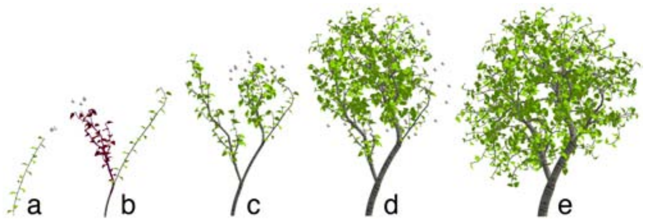


Figure 15: *Example of interactive procedural tree modeling. (a) The first stem is sketched with a single light stroke of the tablet pen. (b) A branch is brushed using a single medium-pressure stroke, starting at an existing bud. Growth is constrained to this branch, which is shown in purple. (c,d) The overall tree crown is defined by clouds of marker points generated with broad heavy-pressure strokes. (e) The final tree. Gray spheres represent marker points.*

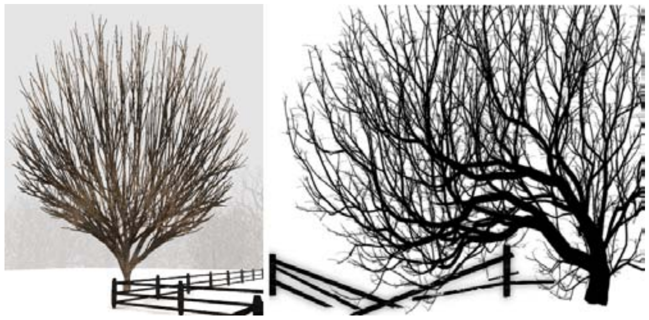


Figure 16: *Trees generated autonomously (left) and with a procedural brush to capture leaning over the fence (right).*

limbs of a tree (Fig. 15a). Heavier pressure yields clouds of markers that define the space in which entire branches, or even the whole tree, may grow (Figs. 15b-e). If a stroke begins at an existing bud, growth is constrained to this bud and its descendants (other buds remain dormant, Fig. 15b). This constraint is useful when guiding a new branch in proximity of existing ones, the form of which we wish to preserve. By controlling pen pressure and the starting points for the strokes the user may thus seamlessly transition between sketching of limbs, constrained brushing of branches, and broad brushing of the whole tree form (Fig. 15). A comparison of trees grown autonomously and using a procedural brush is shown in Figs. 16 and 18. The use of procedural brushes makes it easy to generate highly irregular trees fitting the needs of composed scenes.

Guided growth can be used not only to create natural-looking trees, but also ornamental tree forms and patterns. To facilitate this process, we implemented symmetric brushes, in which brush positions are multiplied in space using an assumed symmetry type. Examples created using bilateral symmetry and a tiling pattern are shown in Fig. 17. Due to the self-organization of tree form and the use of the pipe model to calculate branch width at the global scale of the entire pattern, the resulting forms are not fully symmetric, which adds to their visual appeal.

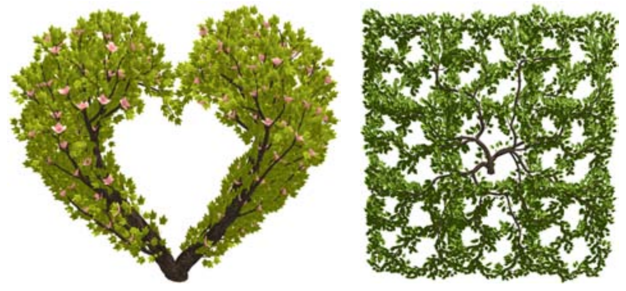


Figure 17: *Ornamental tree forms created with symmetric brushes.*

6 Implementation and performance

The models described in this paper were implemented using the L-system–based modeling environment L-studio [Karwowski and Lane 2004]. To support user interaction, the modeling language L+C [Karwowski and Prusinkiewicz 2003] was extended with constructs for selecting tree components and locating points in space. The implementation of the models was simple: for example, the basic model employing the extended BH procedure for calculating bud fate consists of about 300 lines of L+C code (in addition to 200 lines of C++ for the shadow propagation model).

Scenes were designed interactively, using a terrain editor extended with features for placing imported objects. Artifacts (houses and fences) were modeled with Maya. All trees and scenes were rendered using POV-Ray. The complexity of trees shown varies from approximately 1000 metamers in young trees (Fig. 14) to 700,000 in old ones (Fig. 10, left). We have typically used a $200 \times 200 \times 200$ grid for shadow propagation, with each cell edge approximately the same length as an internode of the modeled trees. The shadow depth q_{max} typically ranges from 4 to 8 cell layers. Conceptually, colonization operates in continuous space, but we use space partitioning (typically $10 \times 10 \times 10$ cells) to speed up computation. The number of space markers ranges from 1 to approximately 1,000 in interactive modeling with the procedural brush, but may reach 1,000,000 in non-interactive models. Sample tree generation times using the shadow propagation light model are shown in Table 1. With the space colonization algorithm, generation times for trees of similar size are about three times slower. Nevertheless, space colonization is fully adequate to model smaller trees at interactive speeds using procedural brushes. For example, the total modeling time for the tree shown in Figure 15 was 9 seconds. The generation times are similar for the extended BH and the priority model of bud fate control.

Figure	Steps	Internodes	Generation time
10 left	106	700,000	82 sec
10 center	90	642,000	60 sec
10 right	68	225,000	21 sec

Table 1: *Sample tree generation times on a 2.4 GHz Pentium 4.*

7 Conclusions

7.1 Summary of the results

We have presented a tree modeling method that integrates three elements of tree development: local control of branching geometry, competition of buds and branches for space or light, and regulation of this competition through an internal signaling mechanism. We have shown that this method makes it possible to generate visually realistic trees, and that their form can be controlled using a small number of intuitive parameters. In particular, apical control, *i.e.* the competitive advantage of the main stem *vs.* lateral branches, determines whether the tree has decurrent or excurrent character. A reduction of apical control during tree development yields mature tree forms. The development of trees can be animated, and the modeled trees are inherently responsive to their environment. The modeling method is robust, producing visually plausible results for various combinations of environmental models, branching patterns, and internal regulatory mechanisms. We achieved short generation times through the use of fast approximate algorithms for simulating plant environment and growth regulation. We introduced a seamless integration of sketching and procedural brushes to specify the layout of individual axes, branches, and the overall tree silhouette. Finally, we implemented our method within an L-system-based modeling environment, demonstrating that L-systems are useful in modeling not only herbaceous plants, but also trees, and can support a highly interactive modeling process.

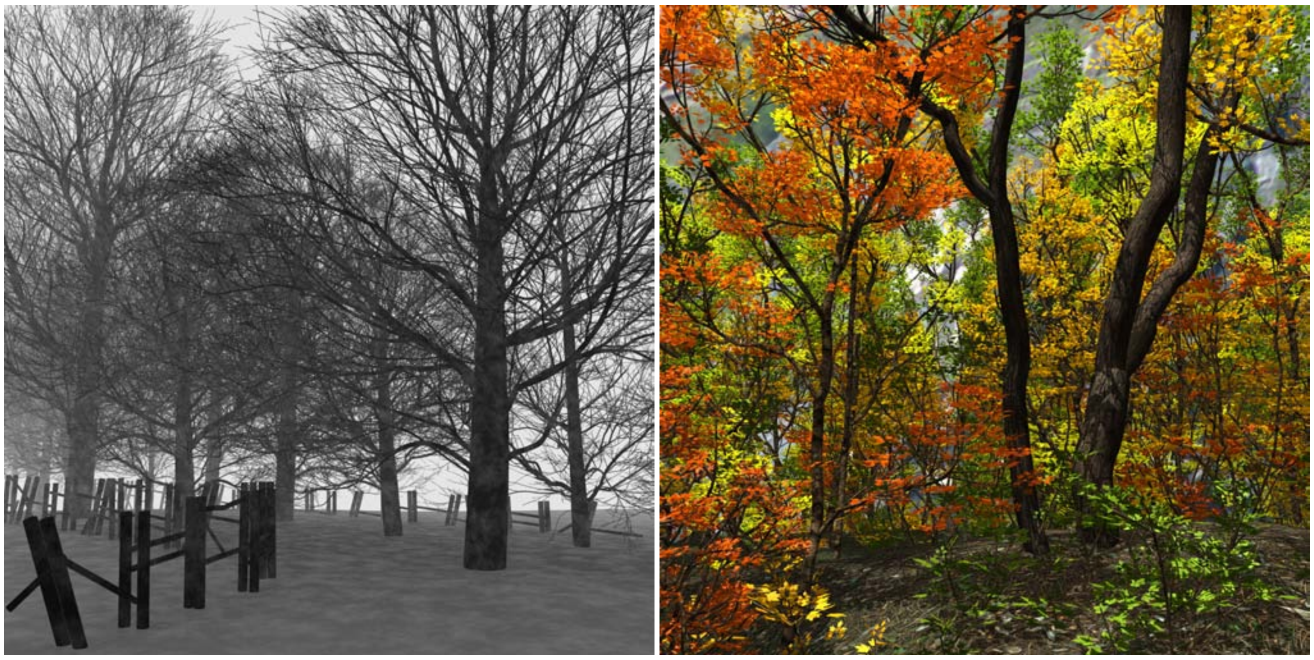


Figure 18: *Complex scenes with trees generated autonomously (left) and with a procedural brush (right).*

7.2 Contributions

While many individual elements of our method have been described before, their integration offers a combination of visual quality of tree models, possibility of animating tree development, and model control that has not been achieved in previous systems. Below we place our contributions in the context of previous work.

Self-organization simplifies the modeling process. Recursive models rely on the user defining the distribution of branches along their supporting axes. This requires a large number of user-tunable parameters [Weber and Penn 1995; Lintermann and Deussen 1999; Prusinkiewicz et al. 2001], which have to be manually adjusted when other aspects of the model are changed (Fig. 19a–c). In contrast, our method provides automatic control of the density and distribution of branches (Fig. 20; see also Fig. 12), which result from the self-organizing nature of competition for space or light. A wide range of forms can be obtained by manipulating a small number of key parameters (Figs. 7, 10, 12, 13), with the density and distribution of branches automatically adjusted by the generative algorithm.

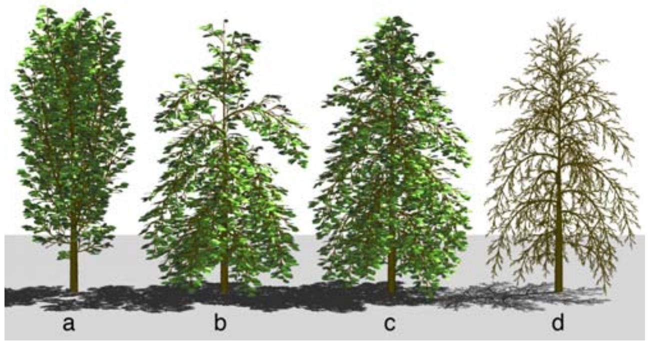


Figure 19: *Analysis of a recursive tree model. (a) Sample tree defined using the global-to-local method [Prusinkiewicz et al. 2001]. (b) A change in the tropism results in uneven branch density: too dense at the bottom, too sparse at the top. (c) Uniform branch density was restored by the user, but this operation required manual adjustment of several user-defined functions. (d) Removing leaves highlights the highly repetitive structure of the tree.*

Integration of architecture and self-organization improves the visual realism of tree models. Recursively defined trees tend to inherit the regular structure (Fig. 19d) of their architectural models [Hallé et al. 1978]. In temperate climate trees, however, such regularity can only be found in young trees. On the other hand, trees generated exclusively using spatial competition [Rodkaew et al. 2003; Runions et al. 2007] tend to be quite disorganized (Fig. 21a), because the pattern of bud distribution is not considered, and a new branch can be attached to any existing branch at any position and orientation. The close integration of architectural and self-organizing components captures well the balance between the regularity and variability of temperate-climate trees (Fig. 20).

Apical control yields a wide range of tree forms. Our method is most closely related to previous models of trees interacting with their environment [Chiba et al. 1994; Měch and Prusinkiewicz 1996], and could be implemented within the modeling framework of open L-systems proposed in the latter paper. However, the actual model

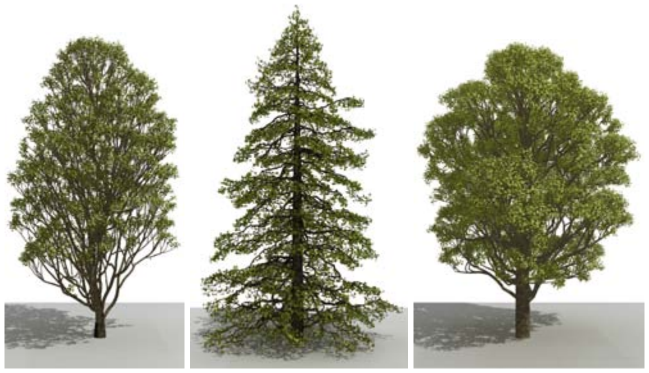


Figure 20: *Sample trees generated using our method. A well-balanced branch distribution is automatically maintained when other attributes of form are changed. Compare with the recursive model in Fig. 19b. The distribution of branches represents a plausible compromise between regularity of recursive models (Fig.19d) and disorganization of models based on pure competition (Fig. 21a). Apical control makes it possible to model excurrent, elongated tree forms (compare with Fig. 21b).*

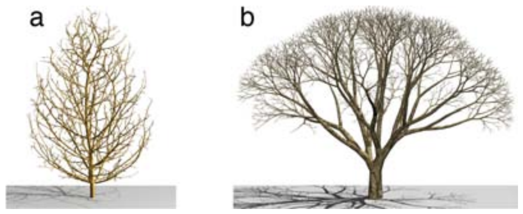


Figure 21: *(a) A typical tree generated using the original space colonization algorithm [Runions et al. 2007]. (b) Spherical form of a deciduous tree generated using an open L-system without apical control [Měch and Prusinkiewicz 1996].*

of deciduous trees described in that paper was highly limited in the visual realism and range of generated structures by oversimplified assumptions regarding branching patterns (each shoot was limited to support at most one lateral branch) and the lack of apical control, resulting in a hemispherical form of tree crown (Fig. 21b). This can be contrasted with the variety of forms obtained with apical control, exemplified by Fig. 20 and further illustrated by the variety of images included in this paper. The more sophisticated functional-structural tree models devised subsequently for biological purposes require significant computation times needed to simulate specific physiological or environmental processes, and represent tree structures with limited realism. Their different design objectives and tradeoffs make them less suitable for image synthesis.

Procedural brushes provide additional control of tree form. We introduced an interactive technique that allows the user to control the development of tree stems, branches, and the entire tree silhouettes using a combination of sketching and brushing. This technique relies on the manipulation of tree environment, rather than the tree itself, and thus is fully compatible with the concept of tree self-organization. Our technique provides a counterpart of the sketch-based design of recursive trees proposed by Wither *et. al.* [2009].

7.3 Future work

While the obtained results are immediately useful in image synthesis applications, a number of problems remain open.

- We found it remarkably easy to control the form of self-organizing trees by manipulating parameter values and using the interactive interface. A formal usability study is needed, however, to verify whether our perception of controllability and ease of modeling is objectively justified.
- Our method generates visually realistic tree forms, but it would be useful to also validate it in objective terms. A possible step in this direction would be a comparison of bud fates, and the distribution of branches in the models, with measurements of real trees.
- In our model, we ignored changes in branch position and orientation over time. Such changes, due to active reorientation or passive bending of branches under their weight, play an essential role in the development of some tree forms (for example, those adhering to Champagnat and Troll architectural tree models [Hallé et al. 1978]). Biomechanical mechanisms of branch reorientation have been modeled before (*e.g.* [Costes et al. 2008]) and could be used to extend the range of developmental processes and structures modeled using self-organizing trees.

Acknowledgments. We thank Fred Boudon, Christophe Godin and the referees for insightful, constructive comments that led to significant improvements in the final paper. The support of this research by grants from the Natural Sciences and Engineering Research Council of Canada and Adobe Systems Incorporated, and graduate scholarships from NSERC and iCORE, is greatly appreciated.

References

ANASTACIO, F., COSTA SOUSA, M., SAMAVATI, F., AND JORGE, J. 2006. Modeling plant structures using concept sketches. *Proceedings of NPAR 2006*, 105–113.

AONO, M., AND KUNII, T. L. 1984. Botanical tree image generation. *IEEE Computer Graphics and Applications 4*, 5, 10–34.

ARVO, J., AND KIRK, D. 1988. Modeling plants with environment-sensitive automata. *Proceedings of Ausgraph 1988*, 27–33.

BANGERTH, F. 1989. Dominance among fruits/sinks and the search for a correlative signal. *Physiologia Plantarum 76*, 608–614.

BARTHÉLÉMY, D., AND CARAGLIO, Y. 2007. Plant architecture: A dynamic, multilevel and comprehensive approach to plant form, structure, and ontology. *Annals of Botany 99*, 375–407.

BELL, A. 1991. *Plant form: An illustrated guide to flowering plants.* Oxford University Press, Oxford.

BENEŠ, B., AND MILLAN, E. 2002. Virtual climbing plants competing for space. *IEEE Computer Animation 2002*, 33–42.

BLOOMENTHAL, J. 1985. Modeling the Mighty Maple. *Computer Graphics 19*, 3, 305–311. Proceedings of SIGGRAPH 1985.

BORCHERT, R., AND HONDA, H. 1984. Control of development in the bifurcating branch system of *Tabebuia rosea*: A computer simulation. *Botanical Gazette 145*, 2, 184–195.

BORCHERT, R., AND SLADE, N. 1981. Bifurcation ratios and the adaptive geometry of trees. *Botanical Gazette 142*, 3, 394–401.

BORNHOFEN, S., AND LATTAUD, C. 2008. Competition and evolution in virtual plant communities: a new modeling approach. *Natural Computing*. In press.

BOUDON, F., PRUSINKIEWICZ, P., FEDERL, P., GODIN, C., AND KARWOWSKI, R. 2003. Interactive design of bonsai tree models. *Computer Graphics Forum 22*, 3, 591–599. Proceedings of Eurographics 2003.

CHIBA, N., OHKAWA, S., MURAOKA, K., AND MIURA, M. 1994. Visual simulation of botanical trees based on virtual heliotropism and dormancy break. *The Journal of Visualization and Computer Animation 5*, 1, 3–15.

CIESLAK, M., LEMIEUX, C., HANAN, J., AND PRUSINKIEWICZ, P. 2008. Quasi-Monte-Carlo simulation of the light environment of plants. *Functional Plant Biology 35*, 9/10, 837–849.

COHEN, J., MARKOSIAN, L., ZELEZNIK, R., HUGHES, J., AND BARZEL, R. 1999. An interface for sketching 3D curves. *Proceedings of the 1999 ACM Symposium on Interactive 3D Graphics*, 17–21.

COHEN, D. 1967. Computer simulation of biological pattern generation processes. *Nature 216*, 246–248.

COSTES, E., SMITH, C., RENTON, M., GUÉDON, Y., PRUSINKIEWICZ, P., AND GODIN, C. 2008. MAppleT: Simulation of apple tree development using mixed stochastic and biomechanical models. *Functional Plant Biology 35*, 9/10, 936–950.

CÔTÉ, J.-F., WIDLOWSKI, J.-L., FOURNIER, R., AND VERSTRAETE, M. 2009. The structural and radiative consistency of three-dimensional tree reconstructions from terrestrial lidar. *Remote Sensing of Environment 113*, 1067–1081.

DE REFFYE, P., EDELIN, C., FRANÇON, J., JAEGER, M., AND PUECH, C. 1988. Plant models faithful to botanical structure and development. *Computer Graphics 22*, 4, 151–158. Proceedings of SIGGRAPH 1988.

Greene, N. 1989. Voxel space automata: modeling with stochastic growth processes in voxel space. *Computer Graphics 23*, 4, 175–184. Proceedings of SIGGRAPH 1989.

Hallé, F., Oldeman, R. A. A., and Tomlinson, P. B. 1978. *Tropical trees and forests: An architectural analysis.* Springer, Berlin.

Honda, H. 1971. Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of Theoretical Biology 31*, 331–338.

Ijiri, T., Owada, S., Okabe, M., and Igarashi, T. 2005. Floral diagrams and inflorescences: Interactive flower modeling using botanical structural constraints. *ACM Transactions on Graphics 24*, 3, 720–726. Proceedings of SIGGRAPH 2005.

Ijiri, T., Owada, S., and Igarashi, T. 2006. Seamless integration of initial sketching and subsequent detail editing in flower modeling. *Computer Graphics Forum 25*, 3, 138–146. Proceedings of Eurographics 2006.

Ijiri, T., Owada, S., and Igarashi, T. 2006. The sketch L-system: Global control of tree modeling using free-form strokes. *Proceedings of Smart Graphics 2006*, 138–146.

Karwowski, R., and Lane, B., 2004. L-studio 4.0 User's Guide. http: //algorithmicbotany.org/lstudio.

Karwowski, R., and Prusinkiewicz, P. 2003. Design and implementation of the L+C modeling language. *Electronic Notes in Theoretical Computer Science 86*, 2, 134–152.

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

Macdonald, N. 1983. *Trees and networks in biological models.* J. Wiley & Sons, New York.

Mandelbrot, B. B. 1982. *The fractal geometry of nature*. W. H. Freeman, San Francisco.

Měch, R., and Prusinkiewicz, P. 1996. Visual models of plants interacting with their environment. *Proceedings of SIGGRAPH 1996*, 397–410.

Neubert, B., Franken, T., and Deussen, O. 2007. Approximate image-based tree modeling using particle flows. *ACM Transactions on Graphics 26*, 3, 88–1–88–8. Proceedings of SIGGRAPH 2007.

Okabe, M., Owada, S., and Igarashi, T. 2005. Interactive design of botanical trees using freehand sketches and example-based editing. *Computer Graphics Forum 24*, 3, 487–496. Proceedings of Eurographics 2005.

Oppenheimer, P. 1986. Real time design and animation of fractal plants and trees. *Computer Graphics 20*, 4, 55–64. Proceedings of SIGGRAPH 1986.

Pałubicki, W. 2007. *Fuzzy plant modeling with OpenGL.* VDM Verlag, Saarbrucken.

Power, J., Bernheim-Brush, A. J., Prusinkiewicz, P., and Salesin, D. 1999. Interactive arrangement of botanical L-system models. *Proceedings of the 1999 ACM Symposium on Interactive 3D Graphics*, 175–182.

Prusinkiewicz, P., James, M., and Měch, R. 1994. Synthetic topiary. *Proceedings of SIGGRAPH 1994*, 351–358.

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

Reeves, W. T., and Blau, R. 1985. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Computer Graphics 19*, 3, 313–322. Proceedings of SIGGRAPH 1985.

Rodkaew, Y., Chongstitvatana, P., Siripant, S., and Lursinsap, C. 2003. Particle systems for plant modeling. In *Plant growth modeling and applications. Proceedings of PMA03*, B.-G. Hu and M. Jaeger, Eds. Tsinghua University Press and Springer, Beijing, 210–217.

Runions, A., Fuhrer, M., Lane, B., Federl, P., Rolland-Lagan, A.-G., and Prusinkiewicz, P. 2005. Modeling and visualization of leaf venation patterns. *ACM Transactions on Graphics 24*, 3, 702–711. Proceedings of SIGGRAPH 2005.

Runions, A., Lane, B., and Prusinkiewicz, P. 2007. Modeling trees with a space colonization algorithm. *Proceedings of the 2007 Eurographics Workshop on Natural Phenomena*, 63–70.

Sachs, T., and Novoplansky, A. 1995. Tree from: Architectural models do not suffice. *Israel Journal of Plant Sciences 43*, 203–212.

Sachs, T. 2004. Self-organization of tree form: A model for complex social systems. *Journal of Theoretical Biology 230*, 197–202.

Shinozaki, K., Yoda, K., Hozumi, K., and Kira, T. 1964. A quantitative analysis of plant form — the pipe model theory. I. Basic analyses. *Japanese Journal of Ecology 14*, 3, 97–104.

Soler, C., Sillion, F., Blaise, F., and de Reffye, P. 2003. An efficient instantiation algorithm for simulating radiant energy transfer in plant models. *ACM Transactions on Graphics 22*, 2, 204–233.

Takenaka, A. 1994. A simulation model of tree architecture development based on growth response to local light environment. *Journal of Plant Research 107*, 321–330.

Ulam, S. 1962. On some mathematical properties connected with patterns of growth of figures. *Proceedings of Symposia on Applied Mathematics 14*, 215–224.

Weber, J., and Penn, J. 1995. Creation and rendering of realistic trees. *Proceedings of SIGGRAPH 1995*, 119–128.

Wither, J., Boudon, F., Cani, M.-P., and Godin, C. 2009. Structure from silhouettes: a new pradigm for fast sketch-based design of trees. *Computer Graphics Forum 28*, 2, 541–550. Proceedings of Eurographics 2009.

Zakaria, M. N., and Shukri, S. R. M. 2007. A sketch-and-spray interface for modeling trees. *Proceedings of Smart Graphics 2007*, 23–35.