Capturing and Animating the Morphogenesis of Polygonal Tree Models

Sören Pirk[1*] Till Niese[1*] Oliver Deussen[1*] Boris Neubert[2*]

[1]University of Konstanz, Germany
[2]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

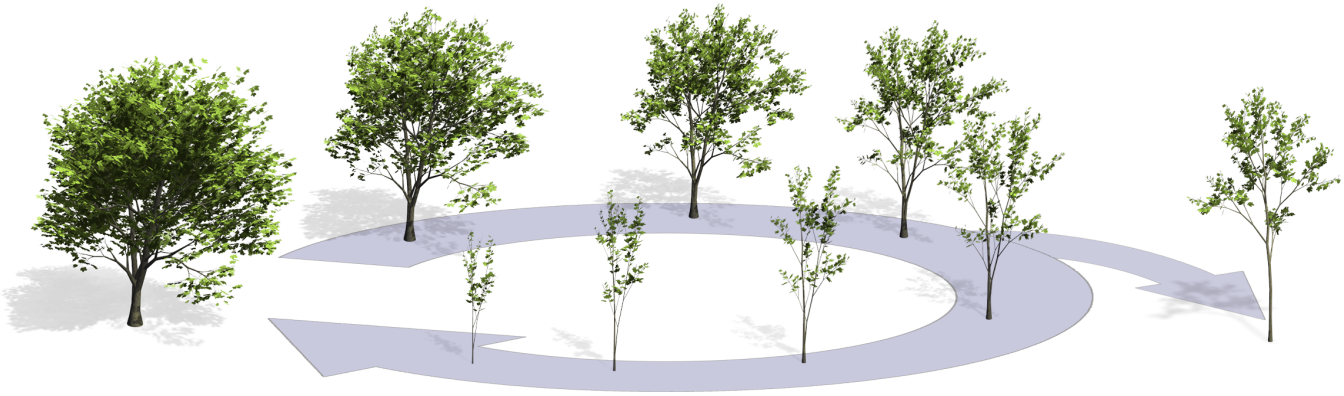

Figure 1: *The static tree model on the left is converted into a developmental model (middle part) that encompasses the ability to create arbitrary intermediate stages between a very young model and the given geometry. We define a "growth space" that allows the user to edit the model in an enhanced way. A corresponding model is shown on the right.*

Abstract

Given a static tree model we present a method to compute developmental stages that approximate the tree's natural growth. The tree model is analyzed and a graph-based description its skeleton is determined. Based on structural similarity, branches are added where pruning has been applied or branches have died off over time. Botanic growth models and allometric rules enable us to produce convincing animations from a young tree that converge to the given model. Furthermore, the user can explore all intermediate stages. By selectively applying the process to parts of the tree even complex models can be edited easily. This form of reverse engineering enables users to create rich natural scenes from a small number of static tree models.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques 1.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.6.8 [Simulation and Modeling]: Types of Simulation—Visual

Keywords: Generative Tree Modeling, Interactive Procedural Modeling, Plant Growth, Animation, Visual Models of Trees

Links: DL PDF WEB

*email: boris.neubert@gmail.com, {soeren.pirk, till.niese, od}@uni.kn

1 Introduction

Modeling complex natural scenes and objects was actively investigated for many years; trees and plants belong to almost every animated movie or computer game. Various methods have been proposed that successfully ease the modeling process while still creating realistic models. Today, thousands of such tree models can be found on the internet, large libraries exist that encompass all sorts of plants. While their visual fidelity has steadily improved over time, most of these models are still stored in the form of static geometries that cannot be altered easily, if needed. Specialized programs have to be used and sets of quite specific parameters have to be known to create variations or even animations –a task typically much too tedious for most content creators.

In this paper we present an automatic method for analyzing and re-modeling such tree models. The method allows developmental stages to be generated from a single input and supports animating growth between these states. In addition, our method offers new forms of editing through selectively applying growth development to parts of the plants. This dramatically reduces the tedium of editing such complex models and opens up what we call a "growth space": a space for developmental edits that influences the model in a quite natural way.

Most polygonal tree models (generated by L-Systems, Laser-scanning, Xfrog or other systems) can serve as the input to our system. Using a mesh contraction algorithm we reduce them to a graph structure that represents the tree skeleton. The allometry – the geometric relations within the tree– and the branching statistics can be obtained from this data structure.

During tree growth the lower branches of a tree typically die off or are pruned by humans, so we have to artificially add back these branches to our developmental model. This is done by analyzing the main branching structure and by using structural similarity –the fact that trees typically repeat branching structures along their main axes. Furthermore, we adapt the geometric relations during growth since branches change their thickness and bend as the tree develops.

Once we have all the necessary information, we are able to simulate the growth of a tree as it develops from an initial young tree into the given input model. Figure 1(center) shows the steps of such a process. Different growth rates have to be applied to different branches in order to meet botanic rules whereby a branch that receives more light grows faster than others and if light is below a threshold, the branch eventually dies off. Seasons with falling leaves can be integrated, pruned branches can be removed by fading or dropping them, growth can be exaggerated to meet various artists' needs.

2 Related Work

With regard to the huge amount of research into modeling trees, we only discuss the most important methods and furthermore focus on those related works that incorporate growth.

Early work on tree modeling concentrated on describing the morphology of natural tree models. L-Systems [Prusinkiewicz et al. 1996; Měch and Prusinkiewicz 1996] were developed as a parallel mechanism representing growth processes in nature, Prusinkiewicz et al. applied the mechanism to geometric tree modeling and over the years showed that almost all natural forms can be described by these kinds of systems. While scientifically very helpful, L-Systems are not very intuitive and require a lot of knowledge about modeling, especially when the intension is to animate these systems (cf. [Prusinkiewicz et al. 1993]). Procedural models for example by Weber and Penn [1995] or by Lintermann and Deussen [1999] try to overcome such modeling limitations. These approaches, however, are more specialized and cannot create the same variety of shapes. A number of sketch- [Chen et al. 2008; Okabe et al. 2006] and image-based reconstructions [Shlyakhter et al. 2001; Tan et al. 2007] reduce the modeling effort by applying data-driven approaches. Only a few methods, such as [Prusinkiewicz et al. 1993; Lintermann and Deussen 1999] allow the creation of animated models. A survey of many tree modeling algorithms is found in Deussen and Lintermann [2005]

Self-organizing plant models allow the morphology of a plant to emerge from a complex dynamic system. Two early approaches for climbing plants used voxel automata to direct the growth of such (simple) structures [Arvo and Kirk 1988; Greene 1989]. These models are able to sense the environment and to adapt their growth accordingly; however, they cannot easily be applied to trees since a much more complex structure has to be developed in this case. More recently [Pirk et al. 2012] proposed a system that allows trees to dynamically react to their environment by computing the temporal light conditions and the inverse tropism of a tree model. In contrast to our method they consider the environmental conditions for a given tree but do not allow continuous animations of the developmental stages.

Particle-based simulations [Runions et al. 2007; Palubicki et al. 2009; Neubert et al. 2007] use large sets of interacting particles, trajectories are used to produce the branching skeleton and generalized cylinders for its geometric shape. Unfortunately, such approaches do not allow growth animations to be created automatically as the particles flow backwards (like in [Neubert et al. 2007]) or can behave in an unnatural way during flow. Hart et al. [Hart et al. 2003] present botanical methods to compute changes in the branching skeleton and wood production when a tree develops.

Another form of developmental system is implicitly controlled by modeling the statistics of botanical growth. DeReffye et al. [de Reffye et al. 1988] introduced these types of systems, and yet goal-directed animation is still not an easy task as it has to be controlled by statistical variables. Open L-Systems [Měch and Prusinkiewicz 1996] allow L-Systems to react to their environment and thus to actively self-organize, however they require developmental L-Systems as their input.

Animation of tree models. On the one hand, animation systems for tree models require the ability to let models grow easily in a natural way and on the other, they need to be editable in order to achieve the desired effects: Both requirements are typically in conflict. The Xfrog modeling system [Lintermann and Deussen 1999] allows the definition of key-frames that describe the developmental stages of a tree model in the form of parameter sets, which are interpolated in a later step. Unfortunately, various combinations of parameters can lead to strange animations when tree-specific constraints are missing. Animated L-systems [Prusinkiewicz 2004] can describe all forms of developing L-Systems, but are very hard to control.

As a result, most trees provided by today's libraries are represented in the form of a static polygonal models. After preprocessing the tree models, which is described in the next Section, we show how we compute important properties of the models such as growth rates and allometry. A number of models and growth simulations are shown and evaluated, limitations are discussed.

3 Processing Tree Models

Our developmental model relies on a graph-based data structure which is computed from the input tree models. Typically, these models are given as a grouped list of vertices with connectivity information. These groups represent the different organs of the plant such as branches, leaves or blossoms.

To generate the branch graph we use a mesh contraction algorithm as described in [Au et al. 2008] and assume that the input to this algorithm is such a grouped list of vertex data. The method allows us to process a wide variety of tree models ranging from reconstructed tress to models grown by L-Systems and models from commercial libraries.

When a tree develops, branching angles of bifurcation change over time. This is due to various kinds of tropisms, i.e. phototropism (growth towards a light source) or gravitropism, growth following gravity (or perpendicular to it) and as a result of forces that act on the branching structure and bend the branch. A tree develops additional wood at places with high tension, this also alters the branches.

To effectively model these changes, we use a branch representation in coordinates by two angles α,β which are *relative* to the parent edge. Please note that branches might be represented by more than one edge, since we allow consecutive sequences of edges without branching.

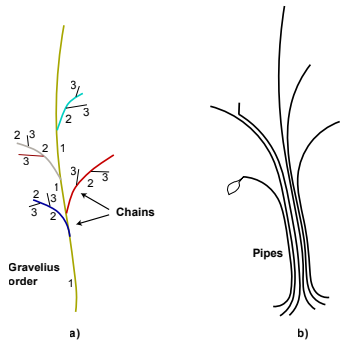

Figure 2: *a) Gravelius ordering in a branching system (main branch is assigned order $g = 1$) with marked chains; b) Pipe Model Theory [Shinozaki 1964]: the tree is seen as a system of pipes of uniform thickness that supports the leaves.*

Botanical branching structures are often described by marking the *terminal leader* (main axis) and the *Gravelius order* of branches; an ordering method originally developed for binary branching systems to describe rivers and streams [Gravelius 1914], but also successfully applied to tree modeling by Holton [1994].

Starting from the root edge of the graph, which is assigned a level $g_{root} = 1$, the child edge with the smallest deviation angle within each bifurcation is assigned the same level as the parent edge while the secondary edges are considered to be one level higher. To account for decurrent tree architectures (trees with weak apical dominance) we assign the same level to all child edges whenever their difference in angle and in radius is below a 5% range. Following the Gravelius order of the branches we are able to determine the main trunk and the side branches of the different orders. This scheme is our primary ordering within the graph.

We are now able to define chains in the branching graph: a chain is a sequence of consecutive edges without any level change. These chains are used to determine the growth rate and to insert missing geometry into the original model. They can reach from the root to the tip of a branch. Additional information is collected about the average branching angle α_{avg} for bifurcations with different child levels and also about the *internode* length I (average spacing between leaves) of the terminal branches.

The resulting data structure is a set of edges E with each edge having been assigned the following properties: rotation angles α, β, length l, order g, and radius r and a set of child edges E_c (all parameters see Table 1).

Table 1: *System variables used for the tree models.* $^{(*)}$*denotes time-dependent variables.*

Name	Description
$(\alpha_i^{(*)}, \beta_i, l_i^{(*)})$	relative coordinate frame (angle pair and length) of the edge e_i
g_i	level assigned to edge e_i according to the Gravelius order.
$r_i^{(*)}$	radius of edge/branch e_i. We limit our description to one radius per edge, although a precise description would have one radius for the start and another for the end.
b_i	radius coefficient of edge e_i (see Sec. 4.3).
$v^{(*)}$	growth rate
$v_\alpha^{(*)}$	angular velocity
r_{avg}	average radius of all terminal edges. Used as initial value for interpolation during growth.
$\alpha_{avg}^{(n)}$	average angle of branches with different assigned levels from n to $n+1$.
I	internode length defined as the average distance between leaves on terminal branches.
CR:	Crown ratio (crown height divided by tree height). Influences growth rate and time of removal of geometry added in growth space.

4 Reverse Growth Simulation

For the given graph we do a reverse growth simulation, i.e. parameters are computed for the intermediate stages of the tree that finally lead to the given graph. This involves computing individual growth rates for the branches, interpolation of the branch radii, updating the branch angles, adding pruned branches and also the pruning of existing branches.

4.1 Growth Rate

A tree with a constant and equal growth rate for all branches would result in a shape where all branches have the same distance from the root. This is neither the case for real trees nor for the models we obtain as our input. Growth at a uniform speed, for these models, would result in intermediate models with some branches stopping their growth at an early stage. Rather than having a constant growth rate across all branches, the growth rate in nature is dictated by a number of influences, but mainly by the amount of light received by a branch (see Fig. 3). Branches that receive less ressources produce less biomass and will eventually die off (for details please refer to [Leyser and Day 2003]).

The fact that we want the final model to be reproduced with all branches still growing, leads directly to a recursive formulation with different growth rates for individual branches. The growth rate v_g of a chain is defined as

$$v_g = \frac{\sum_i l_i}{t_{max} - t_{start}} \tag{1}$$

with l_i being the lengths of the edges that form the chain, and t_{max} the chosen duration of the growth process. We start with the chain that defines the main axis (edges with order $g = 1$) $t_{start} = 0$. For chains with $g > 1$ the computation of v_g depends on the time $t_{start} = \sum_p l_p / v_p$ (p index of parent edges) where all parent edges reach their full length.

The length of a chain at time t is computed by

$$l(t) = v_g \cdot (t - t_{start}). \tag{2}$$

This definition gives us a constant growth rate per edge with the merit of being fully automatic. However, often trees do not have such a constant growth rate. Modifying Eq. (2) allows us to define other biologically motivated functions for growth rate and angular speed such as the logistic function or linear acceleration (as discussed in [Prusinkiewicz et al. 1993]). The accompanying video, however, shows that constant growth already creates plausible animations for many tree models.

4.2 Angle Interpolation

During growth, branching angles change over time. We distinguish between two classes of branches: apical and lateral branches. For apical branches the initial direction is the one of the parent branch, for lateral branches of level g the initial angle is given by $\alpha_{avg}^{(n)}$. The angular velocity v_α of an edge is set to

$$v_\alpha = \frac{\alpha_i - \alpha_{init}}{\Delta t} \tag{3}$$

with Δt being the duration the angle changes. This duration of the interpolation depends on the time the edge starts to grow $t = \sum_p l_p / v_p$ (p: all parent edges) and the time the branch reaches a certain radius.

Following the pipe model theory of Shinozaki et al. [1964] (see Section 4.4) we stop the angle interpolation as soon as the number of terminal edges of the subgraph reaches an upper limit (a good value is a limit of 5). In this case the branch becomes "solid" and does not bend anymore.

4.3 Radius Interpolation

Another recursive formulation can be found for the branch radii, but this time from the tips of the twigs towards the root. According

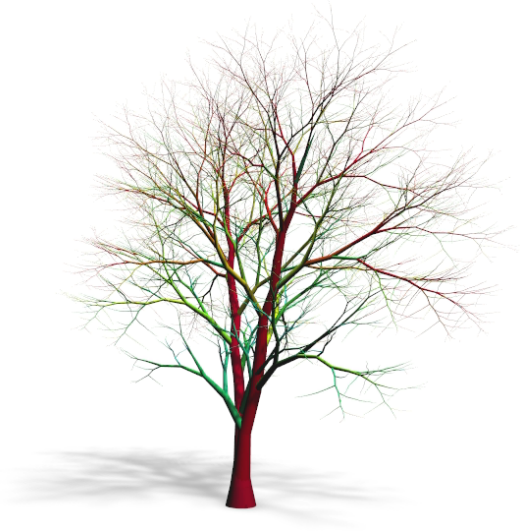

Figure 3: *Growth speed of a tree visualized by color. Red represents fastest speed, cyan slowest.*

to an observation of Leonardo da Vinci, the cross-section area of a branch is the sum of the cross-section areas of its child branches [Richter 1970]. Later Murray [1927] empirically found a relation of circumference c at bifurcations (c_1, c_2: circumferences of child branches)

$$c^{2.49} = c_1^{2.49} + c_2^{2.49} \tag{4}$$

This was based on measurements for branches of many different species. As a result, Murray proposed that trees statistically follow a 2.5 power law of branching.

We use this simple allometric rule to define branch radii during the growth simulation. However, since hand-modeled trees often ignore such rules (e.g. for efficiency, for artistic reasons or to emphasis some parts of the tree) and our main goal is to eventually match the final tree, we calculate a coefficient b of the original tree model for each edge using

$$b_p = \frac{1}{r_p^u} \sum_i r_i^u \quad u = 2.5 \ . \tag{5}$$

with r_i being the radii of child branches and r_p the associated radius of the edge. The coefficient b_p is used for computing the radii of intermediate stages using the following recursive description: For each branch e_p that currently does not have children (a terminal shot) the current radius r_p is initialized by r_{avg} while for non-terminal branches the radius r_p is defined by

$$r_p = \left(\frac{\sum r_i^u}{b_p} \right)^{1/u} \tag{6}$$

with r_i being the radii of the child branches.

4.4 Pipe Model Theory

A popular model in theoretical biology is the *Pipe Model Theory* by Shinozaki et al. [1964]. The theory postulates that tree forms emerge from vascular systems that have to distribute resources within the tree as well as having to mechanically support themselves. A tree structure is an assembly of leaf units of constant size

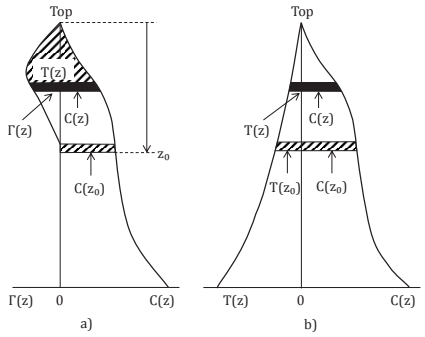

Figure 4: *a) Profile diagrams for trees (Figure 1 of [Chiba 1990]): (a) relation between $\Gamma(z)$ and $C(z)$; (b) relation between $T(z)$ and $C(z)$;*

and corresponding pipes of uniform thickness connecting the leaves to the root (see Figure 2(b)). While real trees have a much more diverse construction, the theory successfully explains many effects in natural trees.

One fundamental finding produced by the theory is the *profile diagram* (see Figure 4) that maintains its similarity among various plant communities despite their differences in species or morphology.

The profile diagram represents the vertical distribution of leaves $\Gamma(z)$ and of non-photosynthetic tissue $C(z)$ (contained in a horizontal layer of unit thickness Δz) downward from the top of the plant or community to the ground. $F(z)$ is the cumulated leaf quantity for the whole tree and $T(z)$ the cumulated tissue (leaves plus non-photosynthetic tissue).

These distributions are what we try to maintain when animating tree growth. If the distributions of the given model deviate too much from these values, we assume that branches were pruned, add such branches and let them die off during animation to finally reach the given model.

For empirical investigations Γ and C are determined by applying the so-called *Stratified Clipping Method* (STC) where the biomass for the leaves ($\Gamma(z)$) or the non-photosynthetic tissue ($C(z)$) is determined by selecting a vertical range $[z, z + \Delta z]$ within the tree. Another way of collecting the data for these distributions is the *Main-axis Cutting Method* (MAC) [Chiba 1990; Chiba 1991]. Here the values associated with a height z are those parts of the plant that emerge from the main-axis segment of length $[z, z + \Delta z]$.

Differences in the biomass distributions obtained by MAC and STC of the input model allow us to detect regions in which additional geometry is needed during earlier developmental stages of the model (we will explain this in more detail in the next section). Such regions are candidates for filling in additional geometry to account for pruned branches during our simulation.

4.5 Adding Missing Structures

Trees are affected by several environmental factors that influence their shape. During growth, parts of a tree get lost, e.g. through mechanical influences such as wind, intentionally by human pruning, or due to the lack of resources.

This information is not available by looking at the final tree, which is the basic problem for the input models we have. Our tree models have been developed for certain environments we do not have information about. A plant was potentially modeled to be part of

a tree stand or to be placed in an urban environment where lower branches have to be pruned. In both cases we need to fill in the missing geometry in order to obtain a plausible growth animation.

We do this by taking advantage of the principle of self-similarity within trees [Ferraro et al. 2005]. This principle has been heavily used for tree modeling in the past, grammar based methods mostly rely on it (see [Prusinkiewicz 1998, pp.121] for an overview). We exploit self-similarity and copy branches from the tree to the missing parts. This is done in the following way:

- copy the largest possible continuous part of branches in the tree canopy to the lower parts of the trunk that are empty.
- fill in the missing information with the fewest possible number of copies.

The regions to be filled with geometry are determined as follows: A well-known dimensionless measure used in forestry is the crown ratio (CR, see Table 1). Crown ratio values range from 0 (no crown, dead or defoliated) to 1 (crown extends over the entire tree bole). For trees where the crown ratio is smaller than one, the region below the crown needs to be filled with additional geometry (see Fig. 5). Additionally, predicting CR values for intermediate stages allows us to determine when additional geometry is pruned during growth (see next section).

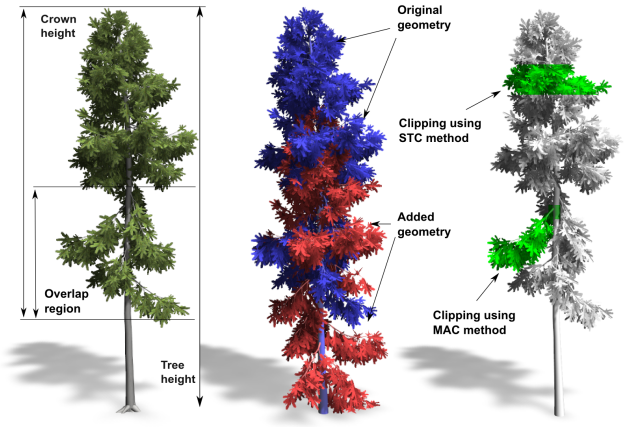

Figure 5: *Left to right: Tree model with marked crown region and overlap region; updated model with added geometry. Original branches are colored blue, added branches colored red; Different Clipping methods.*

While competing for space and resources such as sunlight individual branches eventually block out other lateral branches, finally obtaining a larger volume than their competitors that eventually diminished. Without adding additional geometry and branches during earlier growth stages these regions become apparent for intermediate model. The above-mentioned STC and MAC measures are taken to detect regions of reduced density within the tree. We call these kinds of regions "overlap regions", they are also filled with additional branches.

Following the first chain (g=1, the main trunk) we copy all attached sub-branches adding an offset along the chain to fill this region (see Fig. 6). We define the overlap region to be as large as the filled sub-crown region. We only add branches in this region that originate from parts of the axis that fullfill the following heuristic.

The difference between $T(z)$ obtained by MAC and STC is used to analyse the overlap region between copied and original branches. A significantly higher value of $T(z)^{(STC)}$ for a certain segment along the chain then $T(z)^{(MAC)}$ (obtained by MAC) suggests that the expected biomass for this region is concentrated along another lateral branch that originates from another part of the main axis. This analysis is done once for the original input model.

We want to reconstruct these candidate branches and keep branches for the overlap region that are associated with segments along the chain that exhibit larger $T(z)$ values generated by STC then by MAC or both values are low. In all other cases these candidate branches are neglected. The additional structures are added to the branching graph (together with the associated edge parameters) and used to generate intermediate stages. In the remainder of this section we show how the additional structures are removed during growth to eventually match the original model.

While pruning and self-thinning during early stages of development only affect the main branch, secondary level branches may be influenced in the same way during later stages. We fill in these missing parts in the same way as we add geometry to the main trunk.

The ability to add missing geometry and hallucinate branches that have died off during growth enables us to form what we call *growth space*, a space of possible extension (branches) obtained from the original model.

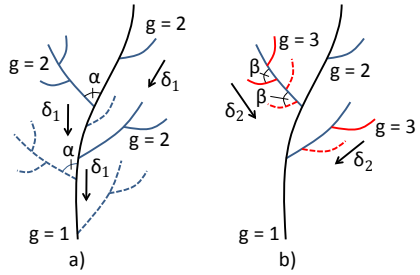

Figure 6: *Copying of branch geometry. a) Branches attached to the root chain (g=1) are copied and added with an offset δ_1 (dashed lines). b) We can apply this process to chains of any order (here g=3 with δ_2).*

4.6 Removing Added Geometry

The principle of self-similarity also guides us during animation. We try to maintain a plausible crown ratio of the model during growth while starting with a typical young tree. Since branches do not move upward during growth but are replaced by new ones, we use the added geometry to form the crown at early stages. When the tree grows, the early branches have to be pruned to maintain the crown ratio.

Empirical investigations indicate the following logistic function for the development of the crown ratio during growth (see [Hasenauer and Monserud 1996])

$$CR = \frac{1}{1+e^{-x}}, \qquad x = c_0 + c_1 * SZ + c_2 * CO + c_3 * ST, \quad (7)$$

where x is a linear combination of species-dependent and environment-specific input variables: SZ=Size of the model, CO=Competition about resources, ST=Site-specific parameters. We assume competition and site variables to remain constant during development of a tree and use tree height h for SZ. This leads to the simplified version of Eq. (7):

$$CR = \left(\frac{1}{1+e^{-(c_0'+c_1*h)}}\right)^{\gamma} \quad (8)$$

using CR_p and height h_p of the original model and $CR_0 = 0.99$ for $h = 0$ (discussion see Hasenauer and Monserud [1996]) we are able to compute the model specific coefficients c_0', c_1 for a standard value of $\gamma = 1$. Evaluating Eq. (8) for intermediate values of h gives us the time at when the additional geometry for regions below the original crown has to be removed.

This provides a fully automatic method to find plausible values for c_0' and c_1. However, an artists can choose different values and thus change the timing at which additional geometry is removed (see below). The above function with slightly different values for the target crown ratio at maximal plant height h_p is used to gradually remove additional geometry in the overlap area.

In Figure 7 the importance of adding and gradually removing geometry during growth is shown. Only using this mechanism a natural looking result can be achieved. It is important to note that branches that are removed during growth still contribute to the computation of the radii. If this were not the case, the deletion would alter other radii leading to visible artifacts in an animation.

4.7 Adding Leaves

Usually trees produce leaves only at terminal branches. However during growth, some edges turn into terminal edges, which were not terminal edges in the original model, and would therefore not have provided leaves for a realistic appearance of the developmental stages. To improve the intermediate appearance for each chain we use the leaf distribution of terminal branches in the original model as candiate set for tempoary leaf positions. During the growth interpolation we make these leaves visible if the edge is terminal and remove them again if the edge becomes a non-terminal edge. Since the number of branches slowly approaches the final number, the leaves also gradually approach the foliage of the input model.

5 Results

Figure 10 shows six models that were processed and animated using our system. The models were imported from different sources (reconstructed point scans, Xfrog, and L-System generated models), all trees show a visually plausible shape at the various stages of their growth.

5.1 Growth Based Editing

We presented a system that automatically generates intermediate models of a given input tree. However, different parameters can be introduced allowing artists to gain additional control over the appearance of intermediate models and influence the resulting animation. The proposed process allows for enhanced editing operations, such as scale, copy, and translate operations for tree models.

Changing Growth Speed and Timings: Changing the target crown range in Sec. 4.6 allows us to keep additional geometry for the final model (see Fig. 8 a)). Solving Eq. (8) for different values of γ automatically changes the time steps for which additional geometry is removed.

While the above changes are applied to the whole model, some enhanced editing can also be applied to parts of the tree. This allows for a natural manipulation of the model. The following operations for tree models can be seen as combinations of scale and structural growth where scale translates to growth (see Figure 8(b))

Advanced Scaling: Selecting individual branches and changing their age (while keeping the age of non-selected branches) corresponds to traditional scaling operations that can be accomplished with standard editing tools. The advantage of our method is that the resulting branch radii, sub-branch angles and lengths are properly transformed to match the new age of the branch.

Advanced Copy&Translate: Copying selected branches and moving them is combined with the growth process and bound to the time parameter. This means that branches that are translated to younger parts of the plant automatically reduce their age, angles, and radii, and vice versa. The radii of parent branches are changed according to Sec. 4.3.

This increases the number of variations of the original model and provides a powerful editing tool. Figure 8 shows the range of editing possibilities for a single input model. After the growth parameters have been computed the user is able to modify chains of the model and change growth rate and relative time. This allows us to take the input and produce diverse models from it, as can be seen on the right.

5.2 Approximating Seasonal Growth

Typically plants exhibit a rhythmic extension of their leafy axes during seasons. It is caused by an alternation of meristem inactivity and active shoot extension [Barthélémy and Caraglio 2007]. We emulate this growth pattern for animation purposes using a remapping function f of the time parameter t ($t \in [0, 1]$ with $t = 1$ for the full model)

$$t_{new} = f(t) = \frac{sin((t \cdot n - s) \cdot 2 \cdot \pi)}{(2 \cdot \pi \cdot n \cdot c)} + t \qquad (9)$$

with n being the number of years the growth simulation should approximate. The coefficient c defines the behavior during seasons with decreased growth rate ($c \in [1, 1.4]$, for $c = 1$ $f'(t)$ recurrently becomes 0 after a one year period). A shift parameter $s = 0.5$ is used to remap the angle interpolation parameter in order to account for the fact that the highest additional mechanical stress is caused by the biomass of leaves, while $s = 0$ is used for the growth remapping.

The derivative of the remapping function is used as a factor to change the leaf size of the model for deciduous trees. Figure 9 shows frames of a seasonal growth animation with typical leaf coloring.

5.3 Performance

We measured the performance of our system to demonstrate its efficiency. Table 2 shows the timing in milliseconds using an Intel Dualcore @ 2.66GHz with Nvidia GTX 580 graphics board. Even quite complex scenes can be animated at interactive rates using this kind of system (see Fig. 12).

Table 2: *Time required for modeling and rendering of tree models (in ms). Model size in number of triangles.*

Species	#Branches	#Leaves	Model	Render
Fagus Sylvatica	10,623	59,822	49.1	4.56
Ulmus Laevis	27,236	33,200	56.6	3.87
Aesculus Hippo.	43,466	47,148	76.4	4.05
Picea Abies	3,813	5,901	13.0	5,38
Quercus Petraea	9,585	58,796	41.6	3.84
Quercus Petraea (complex)	14,554	89,913	64.7	4.13

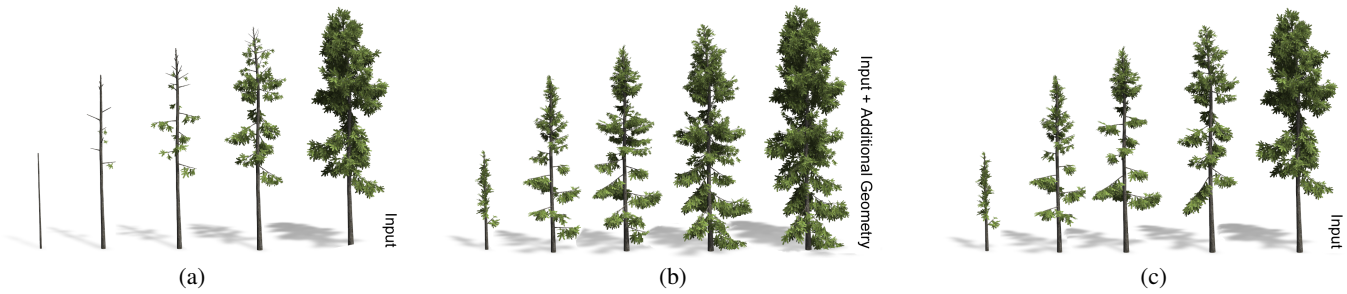

Figure 7: *Growth animations: a) without additional geometry the tree remains empty during growth; b) without the removal of additional geometry the crown ratio is not met; c) gradually removing additional geometry during growth creates a natural model in all steps.*

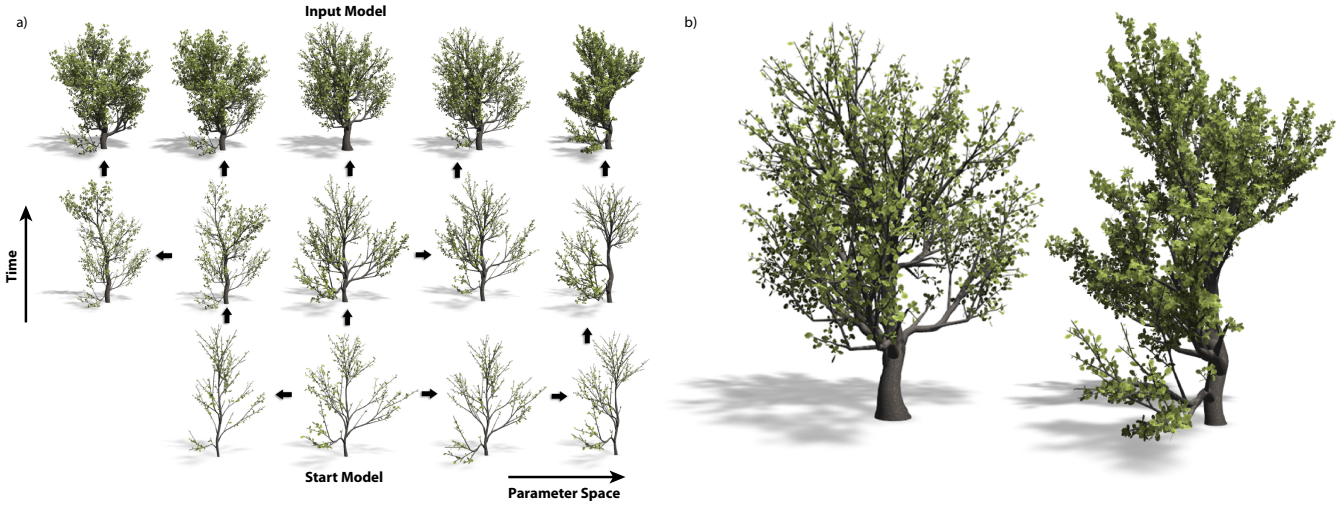

Figure 8: *a) Applying growth partially opens a growth space; b) two models produced from a single input tree.*

Figure 9: *Frames of a seasonal growth simulation with typical color change.*

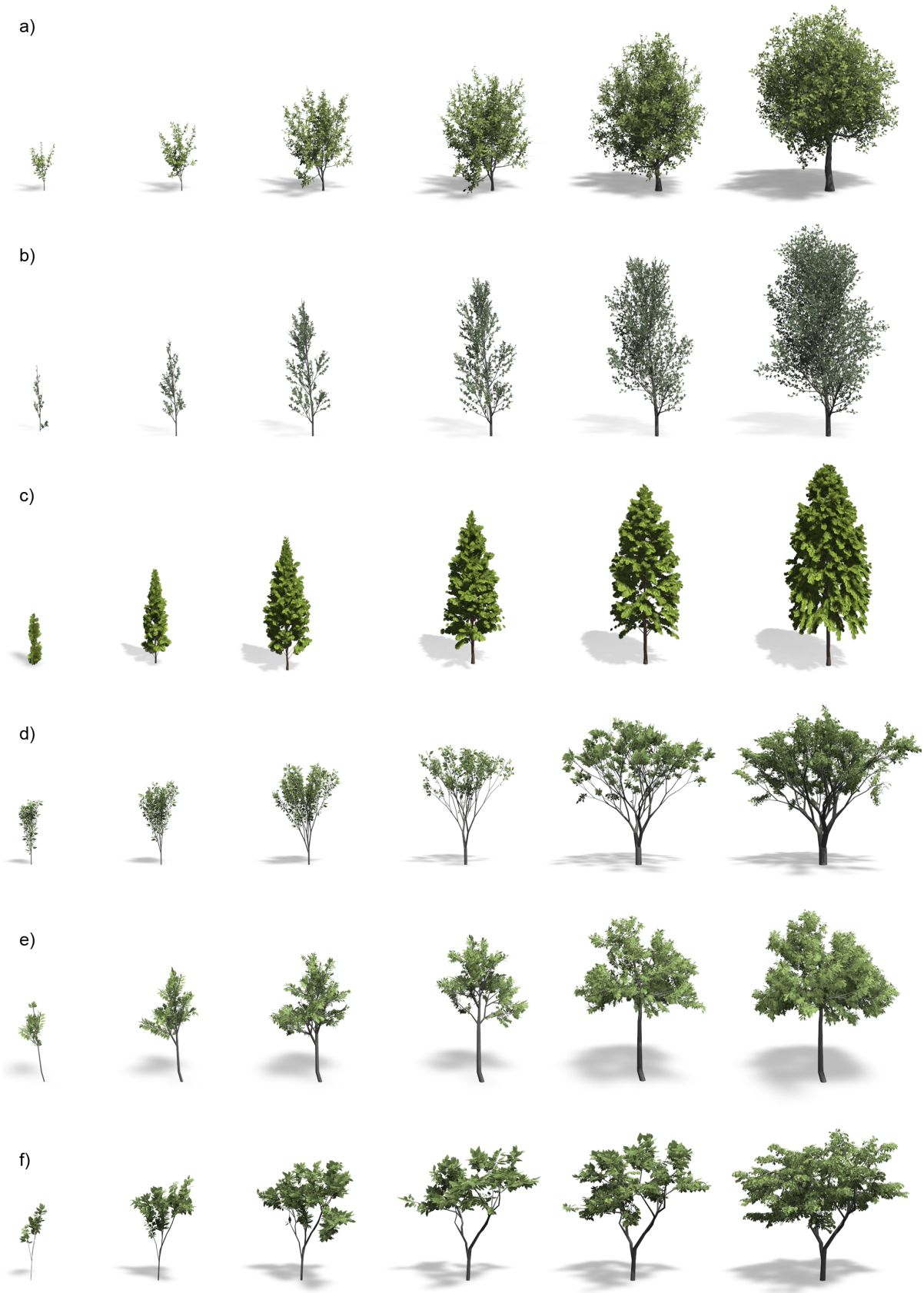

Figure 10: *Six models processed and animated with our system. The largest tree of each species (model on the right) served as input to our method. All other models were synthesized. We used three different model types to test our method; Xfrog: a), b), c); L-System generated model: d); Reconstructed LiDAR Scans: e), f).*

6 Limitations

We successfully used the proposed method to generate models of intermediate age for a number of different species from various sources. So far, our method is working with monopodial trees only; however, adding other branching structures is not a general problem and will be implemented in the future.

Another limitation is that during simulation we are bound to the environment for which the tree was modeled. This is what defines the crown ratio and specifies the environmental factors that are presumed to remain constant during growth. Changing this (e.g. relaxing the constraint to eventually meet the original model) might result in artifacts such as repetition. This effect is not usually seen since the original geometry and its copies are used with different local parameters, such as time, and therefore look different.

Figure 11: *Failure case: the Salix, a willow tree forms branches in a particular way that cannot be reconstructed by our system.*

Furthermore, there are species that do not develop their branches continuously. Every year, willow trees create hanging twigs that are not always converted into branches in the next year but fall off. It is therefore not possible to produce a continuous animation for this specific case. Figure 11 shows the result for a Weeping Willow. While intermediate states still look natural they do not show the typical bending of the small branches and twigs.

7 Conclusion

We presented a method for creating growth models from a static input tree. The tree is analyzed and self-similarity is used to create branches where others have been pruned during growth (growth space). Growth parameters are determined and interpolated to create convincing animations. Based on these parameters, the tree can be edited as a whole or in parts. The method allows the production of time lapse animations of tree development, providing plausible interpolation of branching angles and growth rates at the same time. Since our model is not a growth simulation, every step in the development of a tree can be created separately. This limits the method to a certain set of applications (e.g. games, movies) since it provides only a plausible approximation of the growth process. However, the method is efficient, and enables real time evaluation and rendering.

A promising extension of the proposed method would be the inclusion of environmental factors such as obstacles, shading, or competition with neighboring plants - all kinds of limits and influences for the natural growth process. However, since the objective is to finally match the original model, the freedom to change the plant during development is limited, especially since some of the environmental factors are implicitly encoded in the original model.

Acknowledgements

We thank the anonymous reviewers. This work was supported by the DFG Research Training Group GK-1042 "Explorative Analysis and Visualization of Large Information Spaces", University of Konstanz.

References

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

AU, O. K.-C., TAI, C.-L., CHU, H.-K., COHEN-OR, D., AND LEE, T.-Y. 2008. Skeleton extraction by mesh contraction. *ACM Trans. Graph. 27*, 3.

BARTHÉLÉMY, D., AND CARAGLIO, Y. 2007. Plant architecture: A dynamic, multilevel and comprehensive approach to plant form, structure and ontogeny. *Annals of Botany 99*, 3, 375–407.

CHEN, X., NEUBERT, B., XU, Y.-Q., DEUSSEN, O., AND KANG, S. B. 2008. Sketch-based tree modeling using markov random field. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia '08) 27* (December), 109:1–109:9.

CHIBA, Y. 1990. Plant form analysis based on the pipe model theory I. A statical model within the crown. *Ecological Research 5*, 207–220. 10.1007/BF02346992.

CHIBA, Y. 1991. Plant form based on the pipe model theory II. Quantitative analysis of ramification in morphology. *Ecological Research 6*, 1, 21–28.

DE REFFYE, P., EDELIN, C., FRANÇON, J., JAEGER, M., AND PUECH, C. 1988. Plant models faithful to botanical structure and development. *SIGGRAPH Comput. Graph. 22* (June), 151–158.

DEUSSEN, O., AND LINTERMANN, B. 2005. *Digital design of Nature - Computer Generated Plants and Organics*. Springer-Verlag.

FERRARO, P., GODIN, C., AND PRUSINKIEWICZ, P. 2005. Toward a quantification of self-similarity in plants. *Fractals 13*, 2, 91–109.

GRAVELIUS, H. 1914. *Flusskunde*. Grundriss der gesamten Gewässerkunde. G.J. Göschen.

GREENE, N. 1989. Voxel space automata: modeling with stochastic growth processes in voxel space. *SIGGRAPH Comput. Graph. 23* (July), 175–184.

HART, J. C., BAKER, B., AND MICHAELRAJ, J. 2003. Structural simulation of tree growth and response. *The Visual Computer 19*, 2-3, 151–163.

HASENAUER, H., AND MONSERUD, R. A. 1996. A crown ratio model for austrian forests. *Forest Ecology and Management 84*, 1-3, 49 – 60.

HOLTON, M. 1994. Strands, gravity and botanical tree imagery. *Computer Graphics Forum 13* (February), 57–67.

LEYSER, O., AND DAY, S. 2003. *Mechanisms in Plant Development*. Blackwell Publishing. 138–164,165–189.

LINTERMANN, B., AND DEUSSEN, O. 1999. Interactive modeling of plants. *IEEE Computer Graphics and Applications 19* (January), 56–65.

Figure 12: *A complex scene consisting of more than 20 trees interactivly rendered with our system.*

MURRAY, C. D. 1927. A relationship between circumference and weight in trees and its bearing on branching angles. *The Journal of General Physiology 10*, 5 (May), 725–729.

MĚCH, R., AND PRUSINKIEWICZ, P. 1996. Visual models of plants interacting with their environment. In *Proc. of SIGGRAPH '96*, ACM, New York, NY, USA, 397–410.

NEUBERT, B., FRANKEN, T., AND DEUSSEN, O. 2007. Approximate image-based tree-modeling using particle flows. *ACM Transactions on Graphics (Proc. of SIGGRAPH '07) 26* (July), 88:1 – 88:10.

OKABE, M., OWADA, S., AND IGARASHI, T. 2006. Interactive design of botanical trees using freehand sketches and example-based editing. In *ACM SIGGRAPH 2006 Courses*, ACM, New York, NY, USA, SIGGRAPH '06.

PALUBICKI, W., HOREL, K., LONGAY, S., RUNIONS, A., LANE, B., MĚCH, R., AND PRUSINKIEWICZ, P. 2009. Self-organizing tree models for image synthesis. *ACM Transactions on Graphics (Proc. of SIGGRAPH '09) 28* (July), 58:1–58:10.

PIRK, S., STAVA, O., KRATT, J., SAID, M. A. M., NEUBERT, B., MĚCH, R., BENES, B., AND DEUSSEN, O. 2012. Plastic trees: interactive self-adapting botanical tree models. *ACM Trans. Graph. 31*, 4 (July), 50:1–50:10.

PRUSINKIEWICZ, P., HAMMEL, M. S., AND MJOLSNESS, E. 1993. Animation of plant development. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '93, 351–360.

PRUSINKIEWICZ, P., HAMMEL, M., HANAN, J., AND MĚCH, R. 1996. L-systems: from the theory to visual models of plants. In *Proc. of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences*, CSIRO Publishing, vol. 3, 1–12.

PRUSINKIEWICZ, P. 1998. Modeling of spatial structure and development of plants: a review. *Scientia Horticulturae 74*, 1- 2, 113 – 149.

PRUSINKIEWICZ, P. 2004. Modeling plant growth and development. *Current Opinion in Plant Biology 7*, 1 (Feb.), 79–83.

RICHTER, J. P. 1970. *The Notebooks of Leonardo da Vinci*, vol. 1. Dover Publications Inc, New York. reprinted 1888.

RUNIONS, A., LANE, B., AND PRUSINKIEWICZ, P. 2007. Modeling trees with a space colonization algorithm. In *Proc. of the Eurographics Workshop on Natural Phenomena*, Eurographics Association, 63–70.

SHINOZAKI, K., YODA, K., HOZUMI, K., AND KIRA, T. 1964. A quantitative analysis of plant form - the pipe model theory, parts I and II. Basic analysis and further evidence of the theory and its application in forest ecology. *Japanese Journal of Ecology 14*, 97–104,133–139.

SHLYAKHTER, I., ROZENOER, M., DORSEY, J., AND TELLER, S. 2001. Reconstructing 3d tree models from instrumented photographs. *IEEE Computer Graphics and Applications 21* (May), 53–61.

TAN, P., ZENG, G., WANG, J., KANG, S. B., AND QUAN, L. 2007. Image-based tree modeling. *ACM Transactions on Graphics (Proc. of SIGGRAPH '07) 26* (July), 87:1 – 87:8.

WEBER, J., AND PENN, J. 1995. Creation and rendering of realistic trees. In *Proc. of SIGGRAPH '95*, 119–128.