Implicit Visualization and Inverse Modeling of Growing Trees

Callum Galbraith[†], Lars Mündermann and Brian Wyvill

Department of Computer Science, University of Calgary, Canada

Abstract

A method is proposed for photo-realistic modeling and visualization of a growing tree. Recent visualization methods have focused on producing smoothly blending branching structures, however, these methods fail to account for the inclusion of non-smooth features such as branch bark ridges and bud scale scars. These features constitute an important visual aspect of a living tree, and are also observed to vary over time. The proposed method incorporates these features by using an hierarchical implicit modeling system, which provides a variety of tools for combining surface components in both smooth and non smooth configurations. A procedural interface to this system supports the use of inverse modeling, which is a global-to-local methodology, where the local properties of plant organs are described in terms of their global position within the tree architecture. Inverse modeling is used to define both the tree structure at any time, and a continuous developmental sequence for the tree from a seedling. These techniques provide an intuitive paradigm for the definition of complex tree growth sequences and their subsequent visualization, based solely on observed phenomena. Thus, a key advantage is that they do not require any knowledge of, or simulation of, the underlying biological processes.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations I.3.7 [Computer Graphics]: Animation

1. Introduction

Branching structures with smoothly blending junctions are a key feature of many natural phenomena (herbaceous plants, trees, coral, shells, icicles, speleothems, animals). Several methods have been applied to model this phenomenon, however, a major failing of previous work in this area comes from the observation that many branching structures are not universally smoothly blending. Common features of trees such as the branch bark ridge shown in Figure 1, may combine both smooth and non smooth components in a single branching point. Another feature of trees, the bud scale scar shown in Figure 1, may vary from a non smooth to a smooth blend over time. These situations have not been modeled yet.

A generalized blending method based on the idea of bounded blending [PPIK02], referred to as generalized bounded blending (GBB), is proposed to model these phenomena using implicit surfaces. This method provides an intuitive way to model both the branch bark ridge and bud scale scars over time. The method has been incorporated into the BlobTree system [WGG99], which includes tools for precise contact modeling (PCM) [DCG98], constructive solid geometry (CSG) and various other blending methods. Only by combining PCM and GBB, is sufficient control achieved to model the appearance of the branch bark ridges and bud scale scars during the life of a tree.

Figure 1: *Photographs of poplar trees showing: the branch bark ridge (left), bud scale scars at age two years (center), and four years (right).*

[†] callum/lars/blob@cpsc.ucalgary.ca

A common approach to modeling developmental sequences of plant growth is to simulate the temporal development of plant structures based on biologically derived rules that control plant internal functionality using morphogenetic or physiologically based approaches [dREF*88, PL90]. The development and final structure of a plant model emerges from the developmental rules. It is usually found, however, that (a) visual realism requires a good understanding of the underlying process and (b) global characteristics of a plant's features, such as crown shape or branch distribution, cannot be explicitly specified. A method is also introduced to describe the developmental growth of a tree using a *global-to-local methodology* [PMKL01]. The advantage of this approach is that complex developmental sequences may be defined without knowledge of the underlying biological processes. Rather, the method allows specification of the model in a direct manner based solely on observed phenomena. Using these techniques, realistic visualizations of growing trees may be achieved. The method is demonstrated by its application to a growing model of *Populus deltoides* (Eastern Cottonwood).

2. Previous work

The simplest step in the quest for realism of branching structures was the representation of internodes as 3D cylinders [dREF*88, PL90]. A visually more advanced technique was the use of generalized cylinders [Blo85, LD99, Hol94, Opp86, PMKL01]. Unfortunately, these techniques do not properly capture the geometry of branching points.

Modeling of the geometry around branching points was first addressed by Bloomenthal, who proposed to solve this problem by crafting parametric patches [Blo85]. A more recent method is the use of subdivision surfaces, models of branching structures were created by Tobler et. al. [TMW02], who procedurally grew a mesh, then subdivided the resulting control mesh. Related techniques, using subdivision to model branches, were introduced by Weta Digital in the film "Lord of the Rings: The Two Towers"[AP] and by Akleman et. al. [ACS03]. However, all of these methods are globally smooth, and it is problematic to construct models of arbitrarily complex branching structures.

A limited implicit model of trees was proposed by Hart [HB96], Maritaud [Mar03] used implicit surfaces in combination with procedural bark texturing methods, and Jin et al. [JTFP01] used convolution surfaces to model branching structures, including trees. These methods all defined uniformly smooth blending at branching points.

2.1. Plant modeling

Many researchers have studied the creation of plant models and proposed a range of methods that may be used to model, simulate, and visualize plant structures. These methods
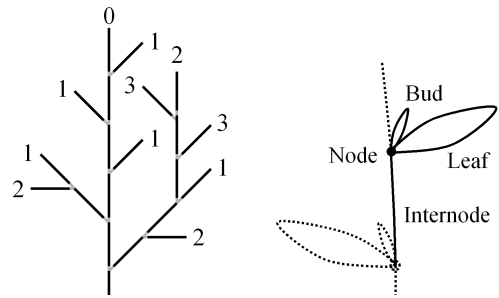

Figure 2: *Left: Plant structures are considered in terms of axes (branches) of different branching order. Right: A branch consists of a sequence of constructional units referred to as metamers, consisting of an internode and associated leaf and lateral bud, if present.*

include models based on parameters [Hon71, WP95], fractals [Man82, FFC82, Opp86], particle systems [RB85], spline skeletons [Blo85, Hol94], stochastic systems [FG86, VEJA89], graph structures [LD99], and rewriting systems [dREF*88, PL90].

In this research, an extension of the *global-to-local* modeling methodology [PMKL01, MÖ3] is used to describe the development of a tree. Prusinkiewicz et al. [PMKL01] presented a modeling approach for static plant structures that focused on intuitive and direct control of a plant's global appearance. Plant structures were explicitly described in terms of graphically defined, continuous functions that express plant components' characteristics as functions of their position within the plant structure. These functions were referred to as *positional information*.

The use of positional information for interactively modeling plant development was presented in [MÖ3], suitable for generating realistic plant development without detailed knowledge of biological processes of plant development. The key idea was to depart from the biological development paradigm, that the shape of an organ is implicitly contained in its growth history [Tho61]. The spatial distribution of plant components was isolated from their temporal formation and growth. The proposed modeling process consisted of three steps: using procedural algorithms to describe the architectural arrangement of plant components, using positional information to define the spatial distribution of plant components for the final structure, and using positional information to specify the formation and growth of plant components. This work was limited to modeling plants with a single season of growth, and as such is not directly applicable to modeling the growth of trees.

In the scope of this research, plants are considered as an assemblage of plant components with a structure described in terms of axes [JL87, GCC97]. The axis of order 0 is the main axis, axes of order 1 are the branches connected to the

main axis, and so on (Figure 2). A branch consists of a sequence of constructional units referred to as *metamers*, consisting of an internode and associated leaf and lateral bud, if present [Bel91] (Figure 2). A lateral bud may develop into a flower or give rise to a lateral branch.

2.2. Growth characteristics of *Populus deltoides*

Figure 3: *Size of lateral branches generally decreases with distance from the apical shoot.*

The reader is referred to [Hos90, Wil84] for a more thorough description of tree development and form. Relevant observations summarizing the phenomena modeled in this research are defined below.

Trees differ from other plants in that they grow over many years, and experience two types of growth, primary growth and secondary growth. Primary growth is the elongation of preformed metamers from a bud produced by the previous years growth. Internodes cease to elongate after their first year of growth, and experience secondary growth, characterized by the steady thickening of branches and production of bark. One years growth arising from a single bud is defined as a shoot, and typically consists of a sequence of metamers, each with associated leaf and lateral bud, terminated by an apical bud. Shoots may vary in length, and number of metamers, based on position in the global tree hierarchy.

In *Populus deltoides*, shoots arising from buds produced by a common shoot in the previous year, have a fairly regular pattern of lengths relative to each other. The apical bud produces the longest shoot, the lateral bud closest to the tip of the shoot produces the next longest shoot, and the lengths of lateral shoots decreases for lateral buds farther from the tip, shown in Figure 3. Lateral buds near the base of the shoot may not produce a branch at all. There are many exceptions to these guidelines, but we adhere to these rules when constructing our model. When a shoot grows, the bud scales are lost, and bud scale scars are observed (Figure 1) as a ring around the branch. Bud scale scars from the apical shoot slowly disappear as bark grows over them, and the scars from lateral shoots are gradually replaced by a branch bark ridge (Figure 1).

Branching angles vary based on length of shoot and time. Longer shoots have smaller branching angles than shorter

shoots. Observed angles range from approximately 80 degrees for short shoots to 35 degrees for long shoots. The variation in branching angle over time results from gravitational forces pulling larger branches down. This also affects the shape of branches over time. Branches typically curve gently upward toward available light. Longer branches may have an S curve, where they bend downward close to their base, and the tips gently curve upward again. The shape of the trunk is also variable. Typically this is a response to environmental conditions, resulting from variable branching density on different sides of the tree.

Crown shape varies over time. A young tree resembles a broom stick for its first few years, growing rapidly upward with short branches. As the tree ages it tends to a rounded crown, with long well developed branches. Branching density also varies with time. Branches which do not receive enough light will die, and eventually be lost. Typically these are smaller branches in the interior of the tree.

3. Implicit surfaces

An implicit surface [Blo97] *S* may be derived from a field function $F(x,y,z)$, and is defined as the set of points $P = (x,y,z)$ at which the value of F equals 0, as follows:

$$S = \{P = (x,y,z) \in \mathbb{R}^3, F(x,y,z) = 0\} \qquad (1)$$

Implicit surfaces are an intuitive means for modeling smoothly blending branching structures in computer graphics [Blo95, BS91]. A common approach is to first define the underlying skeletal structure, then represent each skeletal component using an implicit surface primitive, which have the inherent ability to blend smoothly with each other. In contrast to other methods, implicit surfaces use the same approach regardless of the complexity involved. Recently, implicit surfaces have been used as the basis for more complex modeling systems [WGG99, ACF*99] which incorporate techniques such as controlled blending, bounded blending, CSG, PCM and spacial warping in hierarchical structures.

PCM is a method of deforming implicit surfaces in contact situations to maintain a precise contact surface with C^1 continuity [Gas93, DCG98]. The method is only an approximation to a properly deformed surface, but is an attractive algorithm due to its simplicity. More recently it has been applied to convolution surfaces of an articulated skeleton in a branching situation, where distant parts of the skeleton could be defined not only to avoid blending, but to deform each other using PCM [AJC02]. One problem with this method is that it did not allow the surfaces generated by adjacent skeletal elements to be both in a blend and contact situation at the same time.
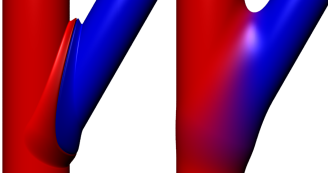
Figure 4: *Input fields defined by $B_1(p)$ (PCM - left) and $B_2(p)$ (Blend - right) to F_g.*
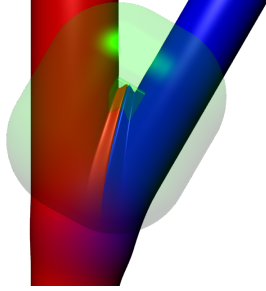

Figure 5: *Application of F_g, the bounding field $b = F_b(p)$ is visualized by two transparent surfaces, the inner one bounding a region where $F_b(p) = 1$ and the outer one representing the zero surface $F_b(p) = 0$.*

3.1. Generalized bounded blending

Pasko introduced the idea of bounded blending [PPIK02], in which two functional solids defined by field functions F_1 and F_2, could be locally blended based on a third bounding field function F_b. This method allowed an intuitive control over local blending between two solids, but was only defined to allow either a blending or non blending situation between F_1 and F_2.

Generalized bounded blending allows smooth interpolation between two arbitrary blend operators, and is defined at a point p by a function $F_g(p)$. F_g depends on a set of field functions, $F_1(p), F_2(p), ... F_n(p)$ defining field values $f_i = F_i(p)$, and a bounding field function $F_b(p)$. Two blending functions $B_1(f_1, f_2, ... f_n)$ and $B_2(f_1, f_2, ... f_n)$ are also required, which may use any definable blending method, such as summation, convolution, PCM, or CSG. F_b must meet the following criteria in order to produce smooth interpolation between B_1 and B_2:

$$
\begin{aligned}
F_b(p) &\in [0,1] \\
\nabla F_b(p) &= 0 \qquad when \; F_b(p) = 0 \\
\nabla F_b(p) &= 0 \qquad when \; F_b(p) = 1
\end{aligned}
$$

The field value of function $F_g(p)$ at point p, is given by:

$$b_1 = B_1(f_1, ..., f_n) \qquad (2)$$

$$b_2 = B_2(f_1, ..., f_n) \qquad (3)$$

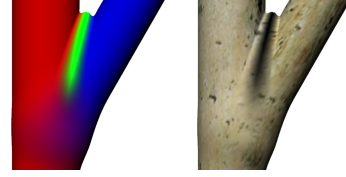$$F_g(p) = F_b(p) \cdot b_1 + (1.0 - F_b(p)) \cdot b_2 \qquad (4)$$

Figure 6: *Modifying the texture in the region of PCM deformation. Example image (left) and branch bark ridge model (right).*

Figure 7: *Evolution of bud scale scars and branch bark ridge over time. Left-right: 1 year, 4 years, 7 years, 10 years.*

Good results can be obtained by making F_b a sigmoid function, which decreases smoothly from 1 to 0 over $[0,1]$. In this research F_b was defined as a distance function computed by $F_b(p) = ((1-r)^2)^3$ where r is the distance from a skeletal primitive, however arbitrary fields could be used for F_b, including non-smooth fields such as produced by CSG operations. Application of F_g to combine two line primitives, with B_1 and B_2 defined as PCM and blend as shown in Figure 4, produces the result shown in Figure 5, where the bounding field F_b is visualized as two transparent surfaces. The inner one bounds a region where $F_b = 1$, and the outer one visualizes the zero surface of F_b. F_b is defined such that its value is 0 everywhere outside this region.

3.2. Visualization of tree features

The function F_b (see section 3.1) is applied to model the bark ridge ridges and bud scale scars in trees by interpolating between blend and PCM. This is appropriate as in nature the shape is a consequence of the collision between bark volumes and PCM was designed to model collision deformations. It can also be observed that the collision volume is bounded and there is a smooth change from the deformed part of the volume to the smoothly blended part. To capture the blackened colour of the branch bark ridge, the bounding field defined by $b = F_b(p)$ is used to determine texture attributes in the region of PCM as shown in Figure 6. On the left the same situation is shown as in Figure 5, except that where the field is modified by the PCM blend, the surface attributes are defined by the bounding field, illustrated by the green ridge. On the right we see the application of texture maps and a black colour defined by the bounding field to produce visually realistic results.

To model the variation of bud scale scars and the branch bark ridge over time, the field function F_b is modified over time. Results for a simple branching situation over the course of several years are shown in Figure 7. The bud scale scar on the main branch is gradually lost, and the bud scale scar of the lateral branch becomes a branch bark ridge.

4. Global-to-local modeling of a growing tree

In this section, the description of the model of tree growth is given for branches of order 0 (the trunk) and 1. It is then extended to higher order branches and animation. The following notation is used throughout this section; $y \in [0, y_{max}]$ denotes the current year after germination, where y_{max} is the maximum age of the tree. Length of a branch in year y is denoted as l_y, and the following inequality is enforced:

$$l_y + l_{min} < l_{y+1} \qquad (5)$$

where l_{min} is a user defined minimum length of shoot. The term x denotes the global position of a branch along its parent branch, and $x_y \in [0, 1]$ the relative position of a branch along its parent in year y, where $x_y = x/l_{p_y}$ and l_{p_y} is the length of the parent branch in year y. The relative position of a branch along its parent in year y_{max} is denoted by $x_{max} \in [0, 1]$ where $x_{max} = x/l_{p_{max}}$ and $l_{p_{max}}$ is the length of the parent branch in year y_{max}. Local parameters of tree components are determined interactively by the user with graphically defined functions G over the domain $[0, 1]$. Unless otherwise stated, these functions are defined in terms of x_y, x_{max} or time $t_y = y/y_{max}$. For example, the length of the trunk by year is defined by G_h as follows $l_y = G_h(t_y)$.

4.1. Structure of branches

The first step is to define the length of each branch for each year. A useful observation is that branches of every order may be considered as a series of growth increments produced annually. The difference in length from one year to the next determines the length of that years shoot.

The length of the trunk, or branch of order 0, is treated as a special case, and is determined by G_h as above. The length of order 1 branches is determined by the use of two functions $G_{s1}(x_y)$ defining the desired silhouette of the tree when it is young, and $G_{s2}(x_y)$ when it is mature. The length of a branch in each year is then determined as follows:

$$l_{d_y} = l_{p_y} \cdot ((1 - t_y) \cdot G_{s1}(x_y) + t_y \cdot G_{s2}(x_y)) \qquad (6)$$
$$l_y = max(l_{d_y}, l_{y-1} + l_{min}) \qquad (7)$$

where l_{d_y} is the determined length before enforcing equation 5. Resulting branch lengths are shown in Figure 8, where each years growth is coloured alternately in blue and red.

The new years growth is shown in green. The bands of colour represent a contour for each years growth. It can be seen that the contours become more rounded as the tree ages. Note that the input x_y to functions G_{s1} and G_{s2} varies for a given branch by year. This implies that it is not possible to individually control the appearance of a specific branch using G_{s1} and G_{s2}. The third step offers a solution to this problem.
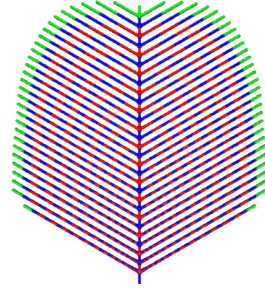

Figure 8: *Length of branches visualized by year. Each years growth is coloured alternately blue then red. New years growth in green.*

The second step is to define localized orientation for branch segments. This is done by interpolating between $G_{o1}(x_y)$, defining local orientations for a theoretical branch of length 0, and $G_{o2}(x_y)$, defining local orientations for a branch of maximum length l_{max}. For each growth increment along a given branch, a desired heading θ is specified by interpolating between these two curves as follows, where x_y represents the position of the growth increment along the current branch, and not the position of the branch along its parent:

$$s_\theta = l_y/l_{max} \qquad (8)$$
$$\theta = (1 - s_\theta) \cdot G_{o1}(x_y) + s_\theta \cdot G_{o2}(x_y) \qquad (9)$$

During construction of the branch, each segment is rotated by an appropriate amount relative to its predecessor such that it is oriented at an angle θ away from the original heading of the branch. A positive heading is defined to orient the branch more in the direction of its parent, and a negative heading has the reverse effect. By defining G_{o1} to be increasingly positive, shorter branches tend to grow more upward along their length. G_{o2} is defined to be initially negative, and then increasingly positive, which gives an S shaped curve to longer branches, and simulating the effect of gravity upon them. The result is shown in Figure 9.

The third step associates a scale factor s_b, defined by $G_{bs}(x_{max})$ with each branch as follows:

$$s_b = G_{bs}(x_{max}) \qquad (10)$$

The scale factor is used in several ways in the modeling process. Initially it provides a direct method for controlling the
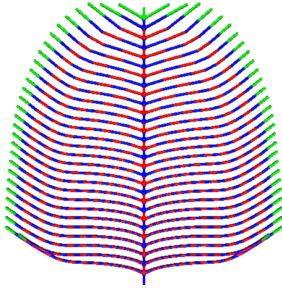
Figure 9: *Localized orientation of branch segments is defined by interpolating between two functions, one for short branches, and one for long branches.*

Figure 11: *Modeling loss of branches. Shorter branches are only observed near the top of the tree where they are younger.*

appearance of individual branches, which is not possible using the functions G_{s1} and G_{s2}. Once a branches length is determined by G_{s1} and G_{s2}, it is scaled by s_b. An exaggerated example is shown in Figure 10.
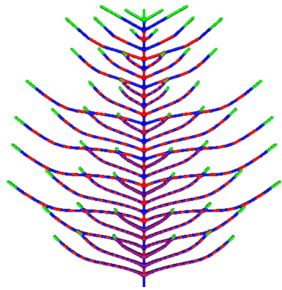

Figure 10: *Modifying branch length with the scale factor s_b.*

The fourth step defines how long a branch lives before it is lost. A function G_a representing age a of a branch when it is lost is defined as follows:

$$a = G_a(s_b) \qquad (11)$$

The result is that shorter branches will tend to be lost sooner than long ones of the same order as shown in Figure 11. Note that shorter branches remain near the top of the tree where they have a younger age. Short branches near the bottom from Figure 10 have been shed. Using the length of branches to determine their lifespan is avoided in order to allow short branches of high order to live longer than lower order branches of greater length, a phenomenon observed in real trees.

Finally, the radius of a branch is defined as a function of position along the branch, and stochastic variations are added to the branch orientation to create a more lifelike tree model.
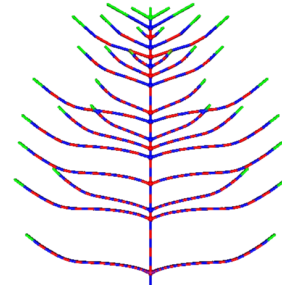
4.2. Higher order branches

Section 4.1 described the use of positional information to determine the structure of branches of order 0 and 1. To extend the method to higher order branches, paracladial relationships between the main and lateral branches are assumed, that is, the same information is recursively applied to branches of higher branching order. To account for variations in branch orders, some modifications are made as follows.

Length of higher order branches is defined as with order 1 branches, but are scaled based on the order of branching. Equation 7 is modified as follows:

$$l_y \; = \; \frac{max(l_{d_y}, l_{y-1} + l_{min})}{0.5 \cdot order + 1} \qquad (12)$$

The result is narrower crowns of branches relative to their length when compared to the trunk. Note that this does not affect the branch scale factor s_b of the branch. Rather we redefine equation 10 as follows:

$$s_b = s_{b_p} \cdot G_{bs}(x_{max}) \qquad (13)$$

where s_{b_p} is the parent branch's scale factor. Branch curvature is defined based on length of the branch and is not modified for higher orders. Similarly, the lifespan of higher order branches is determined as in equation 11 using s_b from equation 13 as input. Finally, lateral branches will not be created if s_b is below a user defined threshold s_{min}.

4.3. Structure and development of shoots

In each year, a new shoot will grow from the apical bud of each branch, and from some of the lateral buds. Each shoot is composed of several metamers, which may produce new lateral branches in the following year.

If a shoot grows in year y, then the length of the shoot

Figure 12: *Growth of a shoot over 1 year.*

Figure 13: *Foliage succession over one year.*

l_s is defined as $l_s = l_y - l_{y-1}$ where l_y and l_{y-1} are the defined lengths for the branch containing the shoot. Each shoot is composed of a number of metamers m defined by G_m as follows:

$$m = G_m(q) \ \ where \ \ q = l_s/l_{s_{max}} \qquad (14)$$

where $l_{s_{max}}$ is the maximum length of a shoot. Once the number of metamers has been defined, the length of internodes l_i is assumed to be equal for each metamer in the shoot where $l_i = l_s/m$. In the year a shoot is grown. Each metamer will produce a lateral leaf. The position of a lateral organ relative to its parent shoot is given by $x_s \in [0,1]$. A graphically defined function $G_l(x_s)$ defines the size of leaf for each metamer.

In the year following a shoots growth, it will produce lateral branches. Lateral branches for a given shoot are constructed as in the previous two sections, with the modification that all lateral branches use the position of the shoot along the branch to determine their lengths and scale values. In addition to this, the scale value s_b for each lateral branch is further modified by a graphically defined function $G_{ss}(x_s)$ which defines another scale value as a function of the lateral branch's position along the shoot. Equation 13 is thus modified as follows:

$$s_b = G_{ss}(x_s) \cdot s_{b_p} \cdot G_{bs}(x_{max}) \qquad (15)$$

where x_{max} is the position of the shoot along the branch. As noted in Section 2.2, the longest lateral shoot will be formed by the lateral bud closest to the shoot tip. G_{ss} is defined appropriately to model this phenomenon.

Lateral branches are arranged in a spiral phylotactic arrangement, where the lateral branch of each successive metamer is placed at an angle of 137.5 degrees from the previous one in the plane perpendicular to the axis of the branch. To avoid repetitive patterns in the final structure, caused by the longest branch from one years shoot being nearly coplanar with the longest branch from the previous

years shoot, the metamer count m for a given shoot is restricted to only allow internode counts such that the two longest lateral branches on the current shoot are placed at an angle at least 40 degrees away from the longest branch of the previous years shoot.

The development of a single shoot is specified as in [MÖ3]. The development of each metamer is considered to be identical, and is given by graphically defined growth curves for internode elongation, leaf growth, and bud growth. Growth curves of subsequent metamers are offset in time resulting in a continuous development of a single shoot. The result of applying this process is shown in Figure 12.

5. Results

Three animations illustrating various aspects of a growing *Populus deltoides* model were produced. Frames from *bbridge.mpg* are shown in Figure 7, which demonstrate the effectiveness of the proposed method in modeling the branch bark ridge and bud scale scars over time. The gross features are modeled although more work needs to be done on the bark texture and its placement. Figure 12 shows frames from *shoot.mpg*, illustrating the growth of a shoot over one year. Note the visible bud scale scar at the base of the shoot where the bud scales are lost. Finally, *populus.mpg* shows the growth of a *Populus deltoides* over 27 years, including succession of foliage in each year shown in Figure 13. Figure 14 shows (from left to right) a series of images of real trees at different stages of growth. Note the shape and distribution of the branches is matched quite closely by the generated images from *populus.mpg* shown in Figure 15.

While the use of implicit surfaces to visualize the resulting models is an advantage for producing more photo-realistic models of growing trees, they also vastly increase the computational complexity of the rendering pipeline. The high frequency nature of a tree model does not react well to standard polygonization techniques. Figure 16 shows a polygonized approximation based on a 1024^3 voxel grid and rendered in OpenGL on the left, and a ray traced image with

Figure 14: *Photographs of* Populus deltoides *(Eastern Cottonwood) at several developmental stages.*

Figure 15: *Implicit surface models of* Populus deltoides *(Eastern Cottonwood) at several developmental stages.*

Figure 16: *Polygonized rendering with* 1024^3 *voxel grid (left), ray traced image at* 768^2 *pixels (right).*

768^2 pixels and no anti-aliasing on the right. The polygonized model required 350 seconds to create the polygons, but even with a very fine voxel grid, the high frequency details are missing. In contrast, the ray traced image required only 430 second to produce an image which captures almost all of the high frequency detail.

Given the non-interactive goal of a photo-realistic animation of tree growth, ray tracing has been used to produce all three animations for its vastly superior image quality. Rendering time for *populus.mpg* varied from 2 seconds to 30 minutes per frame, with anti-aliasing, on a Pentium IV processor with 1Gb of RAM. Much of the time for the later frames was due to memory intensive acceleration algorithms which resulted in swapping memory to the hard drive. As an example, the final image in Figure 15 contains 181,692 primitives, 22220 operation nodes, and required 1.4 Gb of RAM to ray trace efficiently. It is estimated that 30% of the rendering time was spent swapping memory.

With the high rendering times per frame, construction of the model was correspondingly slow. It required 1 week to

fine tune the parameters once the algorithm was determined. Experience with another plant modeling system based on L-systems and parametrically defined surfaces [PMKL01], which was the inspiration for this work, leads to the conclusion that without the rendering delay for each iteration in the modeling process, the entire animation could be defined in a single day. The intuitive nature of the graphically defined functions used to control local properties of plant organs is a fundamental aspect of this process [PMKL01, MÖ3], drastically reducing the amount of guesswork involved in this process compared to other methods. The resulting structure could then be given to the ray tracer to produce high quality renderings of a corresponding implicit surface.

6. Conclusions

In this work we have proposed the use of a procedural system for modeling tree growth which includes a combination of smooth and non-smooth features not previously modeled. The system builds implicit surfaces as hierarchical Blob-Trees [WGG99], and contains many useful tools for representing growing trees, including generalized bounded blending as introduced in this work. The use of graphical interactively defined functions for specifying model parameters, combined with the inverse modeling paradigm and implicit surfaces, provides an intuitive and powerful approach to modeling growing trees. We demonstrate that a realistic procedural model of a growing tree can be built using this approach, which enables modeling of features which have not been represented using other methods. Specifically it was demonstrated that smooth visualizations (animations) of aging bud scale scars and branch bark ridges in developmental tree models can be obtained.

The main contributions of this work are summarized below:

- The concept of generalized bounded blending is introduced.
- A combination of blending, precise contact modeling, and generalized bounded blend is used to represent branch junctions (bud scale scars and branch bark ridges).
- A method is presented for producing smooth animations of growing trees using the inverse modeling paradigm.

There are many areas of future work. The proposed method has thus far been applied to only one species of tree. As it is an extension of the techniques described in [PMKL01, MÖ3], and given the wide ranging success of these methods, we believe that its application to many other tree species will prove successful. Application of known techniques for computing radius of branches [SYHK64], and orientation of plant organs such as leaves and lateral branches [WP95], should be included in our model of tree growth. Bifurcating structures in older trees have not been incorporated in the current methodology. Our texturing algorithm is a primitive one, and more work on realistic texturing around branch junctions is required.

References

[ACF*99]   ADZHIEV V., CARTWRIGHT R., FAUSETT E., OSSIPOV A., PASKO A., SAVCHENKO V.: HyperFun project: a framework for collaborative multidimensional F-rep modeling. In *Proceedings of Implicit Surfaces '99* (1999), pp. 59–69.

[ACS03]   AKLEMAN E., CHENB J., SRINIVASAN V.: A minimal and complete set of operators for the development of robust manifold mesh modelers. *Graphical Models 65*, 5 (September 2003), 286–304.

[AJC02]   ANGELIDIS A., JEPP P., CANI M.: Implicit modelling with skeleton curves: Controlled blending in contact situations. In *Proceedings of the International Conference on Shape Modeling and Applications (SMI 2002)* (May 2002), IEEE Computer Society, pp. 137–144.

[AP]   AITKEN M., PRESTON M.: Foliage generation and animation for "The Lord of the Rings: The Two Towers". SIGGRAPH 2003 DVD-ROM, ACM SIGGRAPH, New York, 2003.

[Bel91]   BELL A. D.: *Plant Form: An Illustrated Guide to Flowering Plant Morphology*. Oxford University Press, Oxford, 1991.

[Blo85]   BLOOMENTHAL J.: Modeling the Mighty Maple. In *Computer Graphics (SIGGRAPH 85 Conference Proceedings)* (1985), vol. 19, pp. 305–311.

[Blo95]   BLOOMENTHAL J.: *Skeletal Design of Natural Forms*. Ph.D. dissertation, University of Calgary, 1995.

[Blo97]   BLOOMENTHAL J. (Ed.): *Introduction to Implicit Surfaces*. Morgan Kaufmann, ISBN 1-55860-233-X, 1997.

[BS91]   BLOOMENTHAL J., SHOEMAKE K.: Convolution surfaces. In *Proceedings of SIGGRAPH 1991* (1991), ACM, pp. 251–256.

[DCG98]   DESBRUN M., CANI-GASCUEL M.-P.: Active implicit surface for animation. *Graphics Interface '98* (June 1998), 143–150. ISBN 0-9695338-6-1.

[dREF*88]   DE REFFYE P., EDELIN C., FRANCON J., JAEGER M., PUECH C.: Plant Models Faithful to Botanical Structure and Development. In *Computer Graphics (SIGGRAPH 88 Conference Proceedings)* (1988), vol. 22, pp. 151–158.

[FFC82]   FOURNIER A., FUSSEL D., CARPENTER

L.: Computer rendering of stochastic models. *Communications of the ACM 25*, 6 (1982), 371–384.

[FG86] FOURNIER A., GRINDAL D. A.: The Stochastic Modeling of Trees. In *Proceedings of Graphics Interface '86* (1986), pp. 164–172.

[Gas93] GASCUEL M.-P.: An Implicit Formulation for Precise Contact Modeling Between Flexible Solids. *Computer Graphics (Proc. SIGGRAPH 93)* (August 1993), 313–320.

[GCC97] GODIN C., COSTES E., CARAGLIO Y.: Exploring plant topological structure with the AMAPmod software: an outline. *Silva Fennica 31*, 3 (1997), 357–368.

[HB96] HART J. C., BAKER. B.: Implicit modeling of tree surfaces. In *Proceedings of Implicit Surfaces '96* (October 1996), pp. 143–152.

[Hol94] HOLTON M.: Strands, gravity, and botanical tree imagery. *Computer Graphics Forum 13*, 1 (1994), 57–67.

[Hon71] HONDA H.: Description of the form of trees by the parameters of the tree-like body. *Journal of Theoretical Biology 31* (1971), 331–338.

[Hos90] HOSIE R.: *Native Trees of Canada*. Fitzhenry and Whiteside Ltd., Markham, ON, Canada, 1990.

[JL87] JANSSEN J. M., LINDENMAYER A.: Models for the Control of Branch Positions and Flowering Sequences of Capitula in Mycelis muralis (L.) Dumont (Compositae). *New Phytologist 105*, 2 (1987), 191–220.

[JTFP01] JIN X., TAI C., FENG J., PENG Q.: Convolution surfaces for line skeletons with polynomial weight distributions. *Journal of Graphics Tools 6*, 3 (2001), 17–28.

[LD99] LINTERMANN B., DEUSSEN O.: Interactive Modeling of Plants. *IEEE Computer Graphics and Applications 19*, 1 (1999), 56–65.

[MÖ3] MÜNDERMANN L.: *Inverse modeling of plants*. PhD thesis, University of Calgary, 2003.

[Man82] MANDELBROT B. B.: *The Fractal Geometry of Nature*. W. H. Freeman and Co., San Francisco, 1982.

[Mar03] MARITAUD K.: *Rendu réaliste d'arbres vus de près en images de synthèse*. PhD thesis, University de Limoges, France, December 2003.

[Opp86] OPPENHEIMER P. E.: Real-Time Design and Animation of Fractal Plants and Trees. In *Computer Graphics (SIGGRAPH 86 Conference Proceedings)* (1986), vol. 20, pp. 55–64.

[PL90] PRUSINKIEWICZ P., LINDENMAYER A.: *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990.

[PMKL01] PRUSINKIEWICZ P., MÜNDERMANN L., KARWOWSKI R., LANE B.: The Use of Positional Information in the Modeling of Plants. In *SIGGRAPH 2001 Conference Proceedings* (2001), Fiume E., (Ed.), Annual Conference Series, ACM SIGGRAPH, pp. 289–300.

[PPIK02] PASKO G., PASKO A., IKEDA M., KUNNII T.: Bounded blending operations. In *Proceedings of the International Conference on Shape Modeling and Applications (SMI 2002)* (May 2002), IEEE Computer Society, pp. 95–103.

[RB85] REEVES W. T., BLAU R.: Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems. In *Computer Graphics (SIGGRAPH 85 Conference Proceedings)* (1985), vol. 19, pp. 313–322.

[SYHK64] SHINOZAKI K., YODA K., HOZUMI K., KIRA T.: A quantitative analysis of plant form - the pipe theory model, i. basic analysis. *Japanese Journal of Ecology 14*, 3 (1964), 97–105.

[Tho61] THOMPSON D. W.: *On Growth and Form*. Cambridge University Press, Cambridge, 1961.

[TMW02] TOBLER R. F., MAIERHOFER S., WILKIE A.: A Multiresolution Mesh Generation Approach for Procedural Definition of Complex Geometry. In *Proceedings of the International Conference on Shape Modeling and Applications (SMI 2002)* (May 2002), IEEE Computer Society, pp. 35–42.

[VEJA89] VIENNOT X. G., EYROLLES G., JANEY N., ARQUES D.: Combinatorial Analysis of Ramified Patterns and Computer Imagery of Trees. In *Computer Graphics (SIGGRAPH 89 Conference Proceedings)* (1989), vol. 23, pp. 31–40.

[WGG99] WYVILL B., GALIN E., GUY A.: Extending The CSG Tree. Warping, Blending and Boolean Operations in an Implicit Surface Modeling System. *Computer Graphics Forum 18*, 2 (June 1999), 149–158.

[Wil84] WILSON B.: *The Growing Tree*. The University of Massachuesetts Press, Amherst, USA, 1984.

[WP95] WEBER J., PENN J.: Creation and Rendering of Realistic Trees. In *SIGGRAPH 95 Conference Proceedings* (1995), Annual Conference Series, pp. 119–128.