

Modeling the Mighty Maple

*Jules Bloomenthal
Computer Graphics Laboratory
New York Institute of Technology
Old Westbury, New York*

Abstract

A method is presented for representing botanical trees, given three-dimensional points and connections. Limbs are modeled as generalized cylinders whose axes are space curves that interpolate the points. A free-form surface connects branching limbs. "Blobby" techniques are used to model the tree trunk as a series of non-circular cross sections. Bark is simulated with a bump map digitized from real world bark; leaves are textures mapped onto simple surfaces.

CR Categories and Subject Descriptors: 1.3.5 [Computer Graphics]: Computational Geometry and Object Modeling - *curve, surface, solid, and object representations.*

Keywords and Phrases: modeling, interpolation, space curve, spline, ramiform, blobby surface, generalized cylinder, texture map, bump map.

1. Introduction

Past efforts at computer generation of trees generally have focused on branching patterns [2, 7, 16, 21, 27, 28]; limbs and leaves were constructed from basic primitives [16, 21, 27] or ignored altogether by using mapping techniques [14]. The trees usually lacked detail when closely inspected.

The present work seeks to model trees with sufficient realism that they may be the subject of animation, rather than simple elements of the landscape. To accomplish this, the model should have a well-defined structure; beneath the bark the limb should be smooth; leaves should be properly attached to twigs. These requirements will be discussed in terms of a polygonal model, which is desired because of its generality. Bark and leaf detail are added using mapping techniques.

Beyond being an aesthetic object to model, a tree provides a useful test of a graphics system. The need for smooth surfaces, the interesting topology at a branch point, and various mapping constraints all place demands upon the modeling process. The polygonal resolution of the model is given as view dependent, and this places a demand on the animation process.

2. The Tree Skeleton

The limbs of a tree may be specified simply as a list of three-dimensional points and a list of connections ("limbs") between those points. The lists accommodate an arbitrary topology ("branching pattern") subject to these conditions: all points must be connected, a point may have at most one incoming limb, and one and only one point has no incoming limb. This format serves equally well when procedurally generating a branching pattern or when measuring an existing one.

The branching pattern seen in Figure 1 was generated recursively. Parameters such as number of branches, branching angles, and length, radius, and taper of a branch were assigned stochastically from ranges of values that changed according to the developing geometry of the tree.

Any representation of the branching pattern is a "tree skeleton." The simplest is a drawing with straight lines, but the resulting appearance is unnatural (Figure 1, left). Trees are perhaps the strongest structures in living nature; each limb is a cantilever beam for which the bending moment and, hence, the deflection, increase smoothly towards the support point [12]. Any sudden change in direction of the limb would result in a discontinuity of the rate of change in the bending moment, introducing stress.

Continuity of direction at fixed points is a well-known property of the interpolating spline (Figure 1, right). The cubic spline, in particular, serves well as a tree skeleton because it has a continuous second derivative ("C2 continuity"), which will be shown useful in constructing the limb surface.

Specifically, a limb consisting of n connections of $n+1$ data points will be interpolated by n spline segments. Not all cubic, interpolating splines exhibit C2 continuity at the interpolated points; the popular Catmull-Rom, for example, is C1 continuous at these points [3].

Rogers and Adams [24] provide a method for computing an interpolating spline that is C2 continuous internally as well as at the data points. Their method improves curve smoothness at the interpolated points by adjusting the parameterization of each segment according to the distance between the points. A tangent at each of the data points is computed.

Because the tree skeleton is defined by interpolated points, conventional control points are not used to manipulate the splines. Instead, each limb is represented by the coefficients of the cubic polynomials that define the spline. These coefficients are directly determined from the positional and tangential information at the data points [13].



Figure 1. Data points interpolated by straight lines (left) and by splines (right).

3. The Limb Surface

Agin [1] defines a generalized cylinder as a space curve (a class of curves including the spline) and a cross sectional contour perpendicular to the curve. The surface of a tree limb, then, may be considered a generalized cylinder with a circular cross section ("disk") of varying radii (Figure 2). To polygonize the

surface, a finite number (the "axial resolution") of cross sections are evaluated along the curve and connected together. Each cross section consists of a finite number (the "circumferential resolution") of points.

To prevent undesired twisting of consecutive disks, each one is oriented by two vectors, the "principal normal," \mathbf{N} , and the "binormal," \mathbf{B} . The principal normal is the unit length curvature vector, \mathbf{K} , which Barsky gives as:

$$\mathbf{K} = (\mathbf{T}\mathbf{X}\mathbf{A}\mathbf{X}\mathbf{T})/|\mathbf{T}|^4, \quad (1a)$$

where \mathbf{A} is the acceleration vector and \mathbf{T} is the tangent vector [4]. \mathbf{A} and \mathbf{T} are computed using conventional techniques [15], and \mathbf{B} is simply $\mathbf{N}\mathbf{X}\mathbf{T}$. \mathbf{K} may also be evaluated as:

$$\mathbf{K} = (\mu\mathbf{A} - (\mathbf{A}\cdot\mathbf{T})\mathbf{T}) / \mu^{3/2}, \text{ where } \mu = \mathbf{T}\cdot\mathbf{T}. \quad (1b)$$

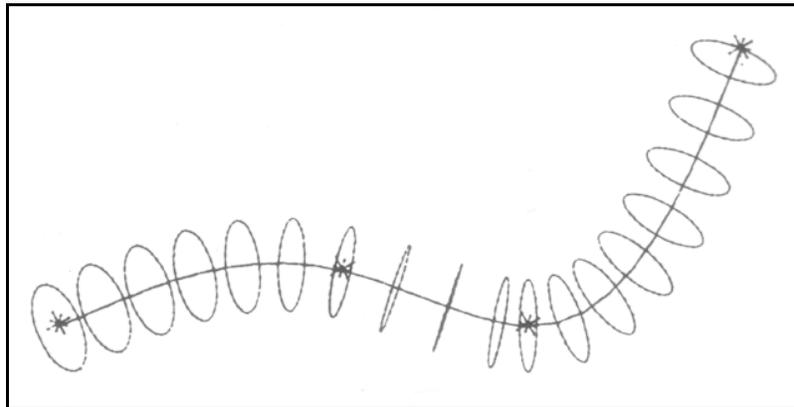


Figure 2. Several spline segments interpolate the data points (asterisks) with C^2 continuity. A strobe captures a disk as it passes along the curve.

The vectors \mathbf{T} , \mathbf{N} , and \mathbf{B} are known collectively as the "Frenet frame" [11]. Unfortunately, the Frenet frame cannot be computed at points where the acceleration is zero (such as at inflection points or along straight line segments). Various approximations, such as the one presented by Shani [26], overcome this limitation.

An efficient means for computing a point \mathbf{D} on the disk is to pre-compute a circle centered at the origin and, for any point \mathbf{P} on the curve, transform the circle such that it is centered on \mathbf{P} and lies in the plane defined by the principal normal and binormal. Thus,

$$\mathbf{D} = (\mathbf{N}_x\mathbf{X} + \mathbf{B}_x\mathbf{Y} + \mathbf{P}_x, \mathbf{N}_y\mathbf{X} + \mathbf{B}_y\mathbf{Y} + \mathbf{P}_y, \mathbf{N}_z\mathbf{X} + \mathbf{B}_z\mathbf{Y} + \mathbf{P}_z), \quad (2)$$

where (x,y) is a point on the pre-computed circle. \mathbf{P} is evaluated conventionally [15].

The Frenet frame and resulting polygonization are illustrated in Figure 3. Tangents, principal normals, and binormals are shown in black, green and red, respectively. The white dots represent the pre-computed circle. The axial and circumferential resolutions are 10 and 8, respectively.

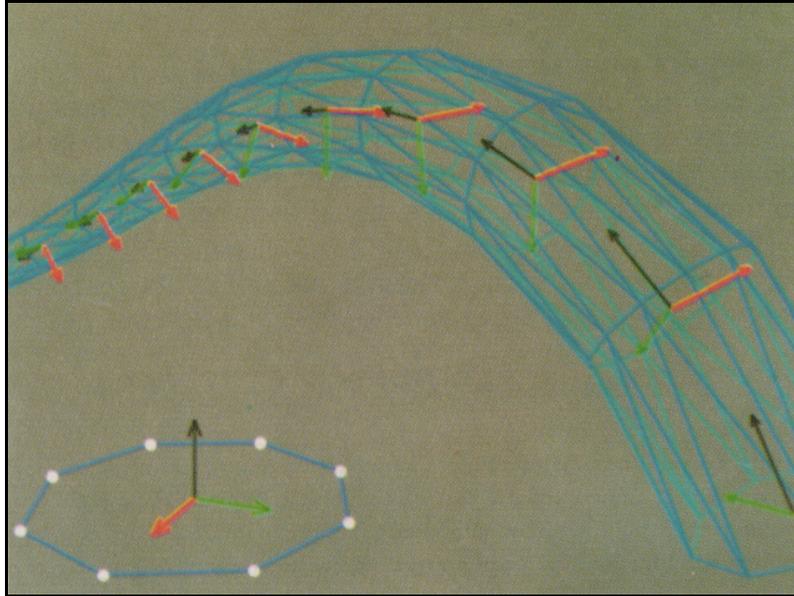


Figure 3. The Frenet frame at parametrically equal distances along a curve.

4. Tension

The tension of a spline may be adjusted by scaling the tangent at an interpolated point [4, 17]. This is useful to stiffen part or all of the tree skeleton, or to increase the angularity of branching for smaller, non-supporting limbs.

Scaling the tangent, however, creates a discontinuity in the second derivative at the interpolated point. Barsky distinguishes between derivative and geometric continuities, the latter consisting of continuity in position, tangency, and curvature, not acceleration. Continuity of curvature (and, thus, of the Frenet frame) is maintained upon scaling tangents, provided:

$$\beta_1^2 \mathbf{A}_{i-1}(1) + \beta_2 \mathbf{T}_{i-1}(1) = \mathbf{A}_i(0), \quad (3)$$

where β_1 and β_2 are beta parameters and i is an index to the interpolated points [4].

5. Branching

A "branch point" is the end of a limb that has two or more outgoing limbs. The outgoing limb of largest radius (or, if the radii are nearly equal, the outgoing limb that diverges least from the direction of the incoming limb) is considered "dominant." Initially, the incoming and dominant outgoing limbs are interpolated by a spline sequence that is C^2 continuous at the branch point. For each of the remaining, non-dominant outgoing limbs, an interpolating spline is begun at the branch point. Figure 4 depicts this ordering for a hypothetical tree.



Figure 4. Order of creation of limbs (red, then orange, yellow, green, blue, and white).

As mentioned in Section 2, determining the n spline segments interpolating $n+1$ data points requires the calculation of $n+1$ tangents. The solution to this system of equations permits the first and last tangents to be set freely ("clamped") [24]. If the initial tangent of a non-dominant outgoing limb is clamped to that of the incoming limb, the smoothness from limb to limb is increased, which is appropriate for large limbs. If the magnitude of the tangent is reduced, the limb becomes less curved (Figure 5), which suits smaller limbs and twigs.

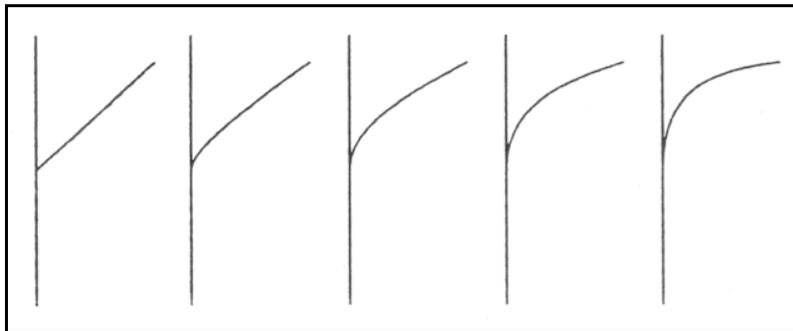


Figure 5. The non-dominant outgoing limb with tension from 1 (left) to 0 (right).

Clamping the initial acceleration would over-constrain the system and so, as suggested by Equation 3, a discontinuity results in the principal normal, and is manifested as a twist about the tangent at the branch point. To maintain continuity of the Frenet frame, then, it is necessary to add a rotation to Equation 2, and to propagate this rotation through the tree.

6. The Ramiform

The surface at a point of bifurcation is described topologically as a triangular prism with 322 symmetry [18]. In computational geometry, such a form is significant for its branching, and so will be referred to as a "ramiform."

There are a number of ways to parameterize a ramiform such that its surface is smooth, without self-intersections and without gaps. Figures 6 and 7 illustrate one technique although others have been published recently [9, 22].

Referring to Figure 6, disks (2a, 2b) are created far enough along the outgoing limbs so as not to interpenetrate. A spline (shown in yellow, 4), called the "saddle," may be constructed between the points of proximity of the disks, given the easily computed gradients at those points. The ramiform shape may be varied by changing the distances of the disks from the branch point **B** as well as by scaling the tangents at the saddle endpoints.

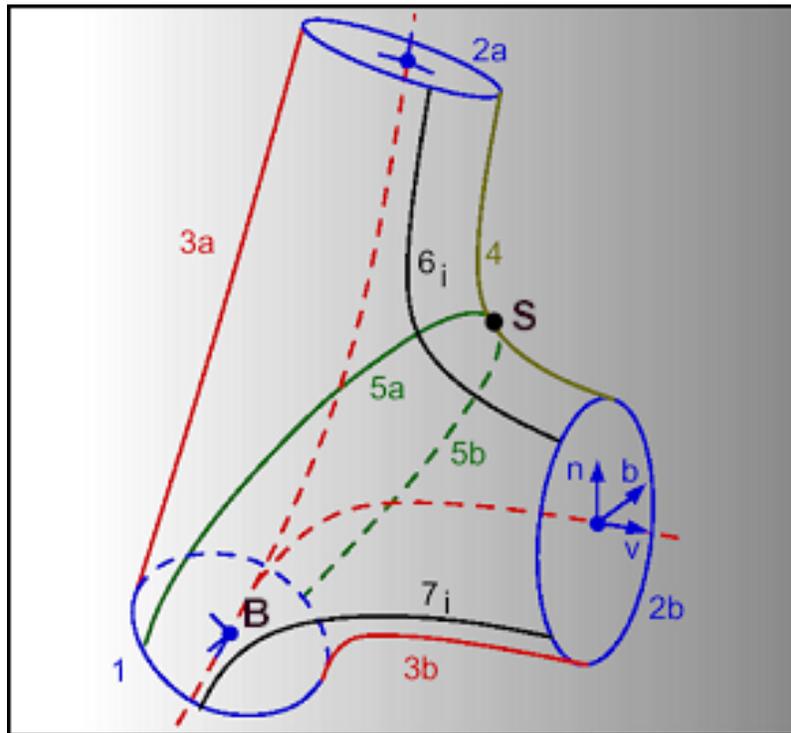


Figure 6. Curve and surface continuity at the branch point **B**.

A two-segment interpolating spline (shown in green, 5a, 5b) connects opposite points on the lower disk (1) to a the saddle point **S** (point on the saddle that is farthest from the straight line segment (not shown) connecting the saddle endpoints). This point is determined readily by recursive subdivision of the saddle. Additional splines (6i) are constructed between the upper disks, passing through the two-segment spline, and (7i) between upper and lower disks, as shown in Figure 7. Points along these splines are connected to polygonize the ramiform.

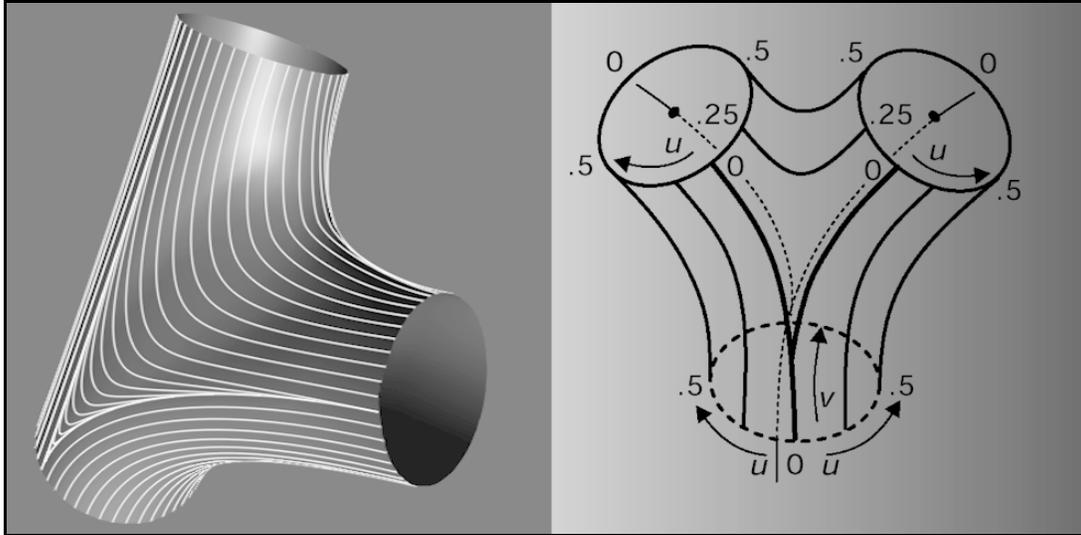


Figure 7. Ramiform and parameterization.

7. Resolution Issues

A tree tends to maximize its surface area to volume ratio. The limbs of a Norway maple, for example, occupy less than 20% of the convex volume that encloses the tree, as measured from a silhouette template [25]. This implies a large polygon to pixel ratio, especially for telephoto compositions; conversely, if the viewpoint is close to a limb, a large number of limbs will be off-screen. Thus, a method is desired for polygonizing limbs that varies the axial and circumferential resolutions according to the projection of the limb onto the screen and that culls off-screen limb sections.

Lane gives the maximum distance, M , that a straight line approximating a space curve will deviate from the curve as:

$$M = [\max|x^{(2)}(t)| + \max|y^{(2)}(t)| + \max|z^{(2)}(t)|] / [b-a]^2 / 8. \quad (4)$$

where a point on the curve is $(x(t), y(t), z(t))$, and $t \in (a, b)$ [19].

Thus, for each curve in the tree model, the acceleration vector is transformed to screen space and the axial resolution is determined with the aid of the above formula. After the transformation, the z term is dropped from the evaluation. Because acceleration of a cubic spline is linear, maximum absolute values need be evaluated for t only at a and b .

Circumferential resolution is made proportional to the projected size of a disk's radius. For extremely thin projected limbs, the model is rendered not as a polygonal solid but as a smoothly drawn curve.

Indexing polygonal resolution according to projected size (or volume) reduces the number of polygons generated, but requires the model be computed for any change in the camera to model relationship.

8. Verisimilar Bark

In 1978, Lance Williams digitized the surface of a plaster face [30] by x-raying the cast and using the digitized X ray as a depth map. In the case of tree bark, a piece of actual bark was cast (Figure 8) and its X ray was processed and used as a bump map (Figure 9).



Figure 8. Tree bark and plaster cast.



Figure 9. Bump map for bark.

In order that the map seamlessly repeat around a limb, the top and bottom rows must be identical, as must be the left and right columns. This was accomplished by overlapping opposite edges of the map. Artifacts of this process were reduced by separating the map into several band-limited images. For each separation, the merging width was adjusted according to the bandwidth, as suggested by Burt [8]. The resulting images were then summed together to produce the bump map in Figure 9.

The mapping itself must be a function of axial and circumferential distance along the surface of a limb, otherwise a tapered limb would exhibit an undesired compression of bark towards its narrow end. Accurate metrics exist for the generalized cylinders and for the curves shown on the ramiform (Figure 7); distances orthogonal to the curves shown on the ramiform are, however, approximated.

The mapping also must account for the progressive changes in bark depth and density as major limbs branch into minor limbs or twigs; discontinuities seem to occur at branch points of outgoing, non-dominant limbs.

9. Trunk-like Surfaces

Roots of a tree generally branch below ground; above ground they merge with each other to form a shape not easily polygonized as a ramiform. Instead, planar "blobby" techniques inspired by Blinn [5] may be used to compute the surface [6]. The blobby cross section and curved profile give the resulting surface an asymmetric, organic appearance.

Figure 10 shows a series of planar contours around a tree skeleton. Each contour is an equi-potential curve surrounding the points of intersection of the tree skeleton with the plane of the contour. The contours are evenly sampled and surfaces between adjacent planar curves are triangulated as described by Christiansen [10]. The shaded surface is displayed in Figure 11.

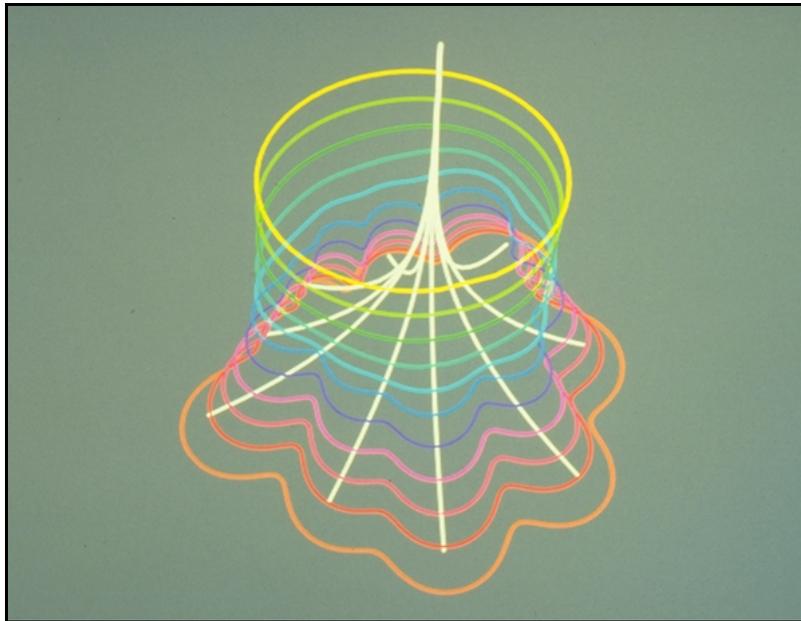


Figure 10. Contours simulating the irregularity at the base of a tree.



Figure 11. Trunk and ramiform with bark.

10. Leaves

A maple leaf was digitized from a photograph by a video camera; the veins were emphasized for dramatic effect, using a paint program. The resulting, non-black texture is mapped onto a three-polygon structure as shown in Figure 12. The structure is hinged along the dashed lines; the degree of hinging depends on the strength of a hypothetical wind as well as the extent to which the leaf faces the wind. The crease at the hinge is not visible if the structure is Phong shaded.

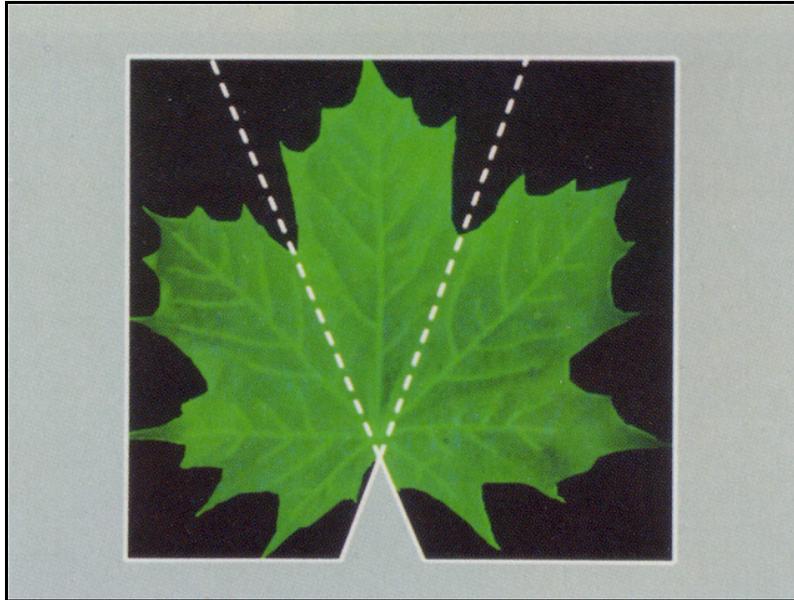


Figure 12. Three-polygon leaf structure.

Figure 13 shows six leaf structures differently transformed to make a leaf "configuration." This configuration was measured from an actual maple twig and is one of two configurations used in the generation of Figures 15 and 16.

For each limb that does not exceed a given diameter and that has no outgoing limbs, a leaf configuration is chosen randomly, scaled randomly, and placed at the tip of the limb. The

leaf configuration is oriented in the direction of the limb and rotated to an angle slightly below horizontal, conforming to observed maple leaves.

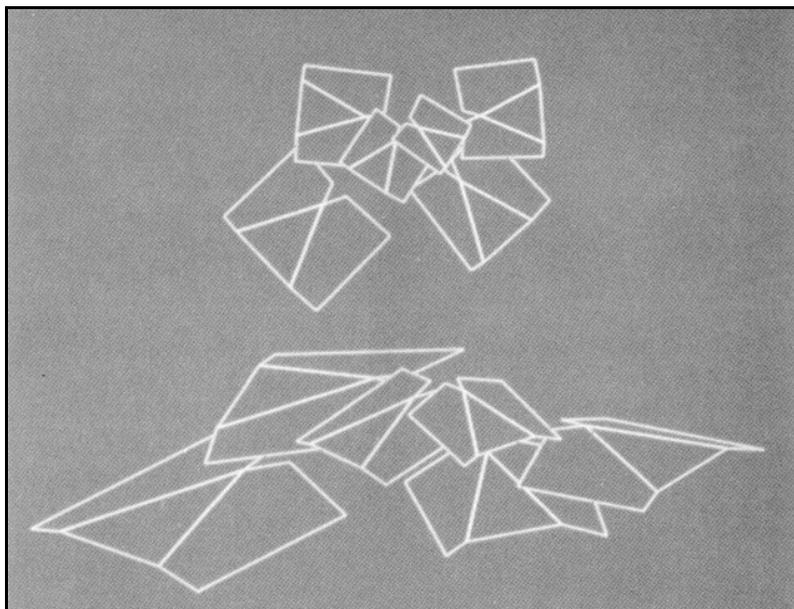


Figure 13. Leaf configuration: plan (top) and perspective (bottom) views.

Three twigs with leaves are shown in Figure 14. The leaves that face away from the viewer are shaded with a lighter color to simulate the lighter underside of the maple leaf. The stems are rendered as unshaded curves defined by the twig tip and direction, and the leaf position and direction.



Figure 14. A twig consisting of limbs, stems, and leaves.

11. Conclusions

From the methods described, a complex tree model may be generated given sparse data. Features such as trunk blobbiness, ramiform shape, and leaf angles are parameterized and do not require explicit modeling.

The basic structure of the model is the branching generalized cylinder. This structure shows promise as a general body form allowing limbs to move without cracks developing between them. The analysis of the structure demonstrates the value of geometric continuity.

The use of screen projection to determine the polygonal resolution of a model suggests its use in determining the model type as well; for example, an extreme close-up could require the bark be modeled as a corrugated structure rather than as perturbations mapped onto a smooth surface.

The techniques of model creation are varied, ranging from procedural generation to real-world digitization. The two extremes have strengths and weaknesses, but as with the use of real tree bark, digitization often yields superior realism. Regardless of the modeling method, the power of computer graphics remains in the manipulation of the models.

Rendered images of the test tree are presented in Figures 15 and 16. Shadows were added by a z-buffer based post-rendering process [29].



Figure 15. The Mighty Maple.



Figure 16. Acer Graphics.

Trees exist in nature, and a logical progression from the present work is their portrayal in different settings and conditions. William Reeves has created some convincing and romantic images of forests [23]; but cracked limbs, shriveled leaves, knot-holes, buds, and snow or moss covered branches would be new elements in the development of realism in tree images.

Mandelbrot has suggested that emulation of nature is its celebration [20]. Insofar as emulation of nature requires its study, one is inclined to agree.

12. Acknowledgments

To Kevin Hunter for the stochastic three-dimensional clouds that are the background in Figures 15 and 16; to Dick Lundin for suggesting the pre-computed disk; to Val Kupris for casting the bark and to James McDonnell for x-raying the cast; to Lance Williams for transparent texture mapping and for "wrapping" the bark; to Pat Hanrahan and Paul Heckbert for many hours of consultation; to Ariel Shaw, Brian Tramontana, and Sasha Shamszad for photographic assistance; to Rick Beach and Subhana Menis for help with the manuscript; and to the "Ops" for unending patience and service.

13. References

1. Agin, G.J., "Representation and Description of Curved Objects," Memo AIM-173, Stanford Artificial Intelligence Report, October 1972.
2. Aono, M., and Kunii, T.L., "Botanical Tree Image Generation," IEEE Computer Graphics and Applications, Vol. 4, No. 5, 1982.
3. Barnhill, R.E., and Riesenfeld, R.F., Computer Aided Geometric Design, Academic Press, 1974.
4. Barsky, B.A., and Beatty, J.C., "Local Control of Bias and Tension in Beta-splines," ACM Transactions on Graphics, Vol. 2, No. 2, April 1983.
5. Blinn, J.F., "A Generalization of Algebraic Surface Drawing," ACM Transactions on Graphics, Vol. 1, No. 3, July 1982.
6. Bloomenthal, J., "A Representation for Botanical Trees using Density Distributions," Proceedings, International Conference on Engineering and Computer Graphics, Beijing, China, September 1984.
7. Brooks, J., et al., "An Extension of the Combinatorial Geometry Technique for Modeling Vegetation and Terrain Features," Technical Report for the Department of Defense, Catalog No. AD782883.
8. Burt, P.J., and Adelson, E.H., "The Laplacian Pyramid as a Compact Image Code," IEEE Transactions on Communications, COM-31, 1983.
9. Charrot, P., and Gregory, J.A., "A Pentagonal Surface Patch for CAGD," Computer Aided Geometric Design, Vol. 1, No. 1, July 1984.
10. Christiansen, H.N. and Sederberg, T.W., "Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics," Computer Graphics, Vol. 12, No. 3, August 1978.
11. Faux, I.D., and Pratt, M.J., Computational Geometry for Design and Manufacture, John Wiley and Sons, 1979.
12. Fletcher, D.Q., Mechanics of Materials, CBS College Publishing, 1985.
13. Foley, J.D., and Van Dam, A., Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Company, 1982.
14. Gardner, G., "Computer Generated Texturing to Model Real World Features," Proceedings, First Interservice Industry Training Equipment Conference, Orlando, Florida, November 1979.
15. Greville, T., Theory and Applications of Spline Functions, Academic Press, 1969.
16. Kawaguchi, Y., "A Morphological Study of the Form of Nature," Computer Graphics, Vol. 16, No. 3, July 1982.

17. Kochanek, D.H.U., and Bartels, R.H., "Interpolating Splines with Local Tension, Continuity and Bias Control," *Computer Graphics*, Vol. 18, No. 3, July 1984.
18. Lalvani, H., "Generative Morphology of Transforming Space Structure," *Proceedings, Third International Conference on Space Structures*, Surrey, UK, 1984.
19. Lane, J., "Curve and Surface Display Techniques," *Siggraph Tutorial Notes*, 1982.
20. Mandelbrot, B., *Fractals: Form, Chance, and Dimension*, W.H. Freeman and Company, San Francisco, 1977.
21. Marshall, R., Wilson, R., and Carlson, W., "Procedure Models for Generating Three-Dimensional Terrain," *Computer Graphics*, Vol. 14, No. 3, July 1980.
22. Nasri, A.H., "Polyhedral Subdivision Methods for Free-Form Surfaces," *Doctoral Thesis, The School of Computing Studies and Accountancy, University of East Anglia*, 1984.
23. Reeves, W.T., "Andre's Forest," *Computer Graphics (back cover)*, Vol. 18, No. 3, July 1984.
24. Rogers, D., and Adams, J., *Mathematical Elements for Computer Graphics*, McGraw-Hill, New York, 1976.
25. Rogers, W.E., *Tree Flowers of Forest, Park and Street*, Dover Publications, New York, 1965.
26. Shani, U., and Ballard, D.H., "Splines as Embeddings for Generalized Cylinders," *Computer Vision, Graphics, and Image Processing*, Vol. 27, No. 2, August 1984.
27. Smith, A.R., "Plants, Fractals, and Formal Languages," *Computer Graphics*, Vol. 18, No. 3, July 1984.
28. Tomlinson, P.B., "Tree Architecture," *American Scientist*, Vol. 71, March 1983.
29. Williams, L.J., "Casting Curved Shadows on Curved Surfaces," *Computer Graphics*, Vol. 12, No. 3, August 1978.
30. Williams, L.J., private communication, 1981.

14. Addenda (post-publication)

Credit relative to resolution issues should have been given to Martin Newell for the issues raised (p. 38) in his doctoral dissertation.