PARAMETRIC MODELING OF TRUNK-ROOT JUNCTIONS USING ASTROIDAL EXPANSION GEOMETRY

Karthik Mahesh Varadarajan
ACIN, Technical University of Vienna, Vienna, Austria

Keywords: Geometric modeling, Rendering foliage, Trunk-root junctions, L-systems, Astroid, 3D parametric modeling.

Abstract: In the field of foliage or vegetation modeling for computer graphics, algorithms for parametric modeling of trees have largely focused on branching mechanisms with little emphasis on modeling the base of the trunk, especially the trunk-root junctions. Roots and trunk-root junctions that appear prominently above the ground in the case of many species of trees such as the Ficus macrophylla and the Picea sitchensis as a result of age and soil erosion are usually completely neglected by traditional trunk modeling methods. In this paper, we introduce a novel parametric modeling scheme to build such trunk-root junctions, while providing for an elegant framework to construct the bases of trunks and branches in order to provide better characterization for a variety of flora species. The paper also describes novel schemes for generating branch junctions and for rendering tree barks.

1 INTRODUCTION

There have been significant developments in the area of foliage or vegetation modeling in recent years (Federl and Prusinkiewicz' 04, Lefebvre and Neyret '02). The availability of multi-core GPU systems has widened the possibilities for realistic rendering of foliage. While template or model multi-resolution rendering systems enable fast visualization of a variety of plant and tree species, they are nevertheless restricted due to the lack of uniqueness of each rendered entity. Parametric models on the other hand, can create a variety of shapes and unique entities within a single tree sub-species or across a broad range of possible species. Parametric L-systems have long been used for modeling tree branches and trunks. However, little attention has been dedicated to the parametric modeling of roots of trees and more importantly trunk-root junctions that are visible above the ground.

Algorithms for parametric modeling of trees have largely focused on branching mechanisms with little emphasis on modeling the base of the trunk, especially the trunk-root junctions. Roots and trunk-root junctions that appear prominently above the ground in the case of many species of trees such as the Ficus macrophylla and the Picea sitchensis as a result of age and soil erosion are usually completely neglected by traditional trunk modeling methods. Figures 1 and 2 depict sample images of trunk-root junctions that are visible above the ground. In this paper, we introduce a novel parametric modeling scheme to build such trunk-root junctions, while providing for an elegant framework to construct the bases of trunks and branches in order to provide better characterization for a variety of flora species.

This work is largely focused on modeling structural components of a generic tree-like structure, with particular emphasis on root-trunk junctions. The algorithm is based on a number of user-modifiable parameters that can be used to generate a variety of plant species. Fundamentally, the structural components, angular limitations, and branching mechanisms may be varied to more accurately model different kinds of tree growth, soil and environmental effects.

Modeling of branches and root-trunk junctions, together with a bio-inspired algorithm for rendering barks and branches form specific contributions of this effort.

Varadarajan K..
PARAMETRIC MODELING OF TRUNK-ROOT JUNCTIONS USING ASTROIDAL EXPANSION GEOMETRY.
DOI: 10.5220/0003324900940099
In *Proceedings of the International Conference on Computer Graphics Theory and Applications* (GRAPP-2011), pages 94-99
ISBN: 978-989-8425-45-4

Figure 1: A Moreton Bay Fig (*Ficus macrophylla*) tree in Australia [Src: DO'Neil, Wikimedia].

Figure 2: A Sitka Spruce (*Picea sitchensis*) tree in Canada [Src: Cowichan Bay Journal].

2 RELATED WORK

Extensive work has been carried out on rendering algorithms for foliage, originating from ground breaking work by Przemyslaw Prusinkiewicz. The majority of these algorithms can be classified into (a) Mono-scale representation – with branching structure based on (i) cylinders (Prusinkiewicz and Lindenmayer '90) (ii) cone-spheres (iii) generalized cylinders (Bloomenthal '95) (iv) implicit surfaces (Hart and Baker '96) (iv) sub-division surfaces, bark modeling based on (i) bump mapping (ii) displacement mapping (iii) volumetric textures (Neyret '98) (Lefebvre and Neyret '02) (iv) polygons (iv) texture mapping, (b) Global representations such as billboards, slices (c) Structure based and Spatial based multi-scale representation systems using hierarchical billboards, volumetric texture slices (Federl and Prusinkiewicz '04), particle systems, volumetric point representation systems etc.

While the focus of the method presented in this paper caters to mono-scale representation, it specifically focuses on modeling of trunk-root junctions, which has hardly been modeled parametrically in the past. The only work similar to the one in this paper can be attributed to Jijoon Kim 2006. The method presented in this paper for modeling of barks is similar to methods such as that of volumetric textures and polygons. L-Systems based foliage rendering software *L-Studio* and *Xfrog* can use algorithms developed in this paper as plugins for realistic geometric modeling.

3 COMPONENT MODELING

The structural components of plants that are dealt with in this paper include stems (trunks, branches), barks and root-trunk junctions. The generative modeling presented in this paper is primarily focused at achieving realistic stems, as this poses sufficient unsolved challenges for modeling. This paper bases its algorithms for geometrically designing a tree by delving into the natural growth of foliage. Two main natural phenomena or courses of events in the growth of a tree, namely concentric trunk growth and root-trunk junction growth have been incorporated into the local geometry modeling. These are explained in sections 3.1.1 and 3.2.1.

3.1 Trunk-Root Junction Modeling

Trunk-Root junctions refer to the base of the tree where the trunk segment extends through the ground to form the roots of the tree. In the case of several common species of trees and plants such as the date palm (*Phoenix dactylifera*) and the bamboo (*Bambuseae*), the diameter of the trunk remains more or less uniform from the base to the top. More importantly, the trunk does not exhibit an expansion at the trunk-root junction. Traditional parametric branch and trunk modeling algorithms are successful in modeling such systems. However, in the case of trees such as in Figures 1 and 2, this is not true. The trunk expands into several diverging branches at the trunk-root junction, which cannot be modeled using a cylindrical geometric framework.

3.1.1 Phenomenon

The expansive trunk-root junction forms the first growth phenomenon considered in this paper. In the case of several species of trees, the roots rise up as they grow old. Since the trunk is cylindrical and the roots exhibit semi-random cylindrical explosion geometry, the base of the tree can be modeled with hypocycloids such as the deltoid, astroid etc. with random perturbations (Figure 3).

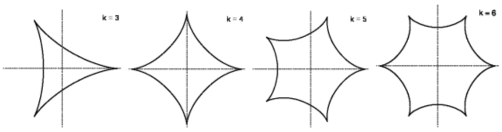

Figure 3: Family of hypocycloids – the three and four pronged geometries are called deltoid and astroid respectively.

3.1.2 Algorithm

Hypocycloids are very convenient to model trunk-root junctions. The number of limbs in the hypocycloid can be varied to produce a variety of branching systems at the trunk-root junction. This, when combined with the random perturbation of the surface contour can yield realistic trunk rendering. Since the trunk of the tree is best modeled as a cylindrical structure, it is necessary to model the continuity between the hypocycloidal trunk-root junction and the cylindrical trunk. This can be done by modeling the intermediate region as a folium with multiple limbs. For the case of the astroidal trunk-root junction, a quadrifolium is best suited for the transition to the cylindrical trunk. The quadrifolium is a rose or rhodonea, generated as a pedal of the astroid. Pedals can be obtained as the locus of arbitrary points, one each on the tangents of a curve, such that the line from a given vertex to the point and the tangent are perpendicular. An interpolation scheme such as the spline or bi-cubic method can be used to define intermediate points.

While the above scheme is well suited for the generation of a wide variety of trunk-root junctions, we emphasize the use of the astroid for modeling the trunk-root junction in this paper. This choice of hypocycloid for the modeling is the result of the unique property of the astroid, namely its status as the envelope of co-axial ellipses whose sum of major and minor axes is constant (Xahlee, Special Curves). Astroids are also the evolutes of ellipses. Astroids belong to a class of curves called the hypocycloids with cusps pointing away from the vertex. Hypocycloids are a special case of the Roulette family, the curves that roll upon other curves to give the locus of circles. While the fundamental method to create an astroid is as the trace of a point on a circle of radius '*r*' rolling inside a fixed circle of radius 4 '*r*' or 4/3 '*r*' (using single or double generation mechanisms respectively), the ellipse method enables one to construct an astroid analytically by locus generation. A simpler way to construct the astroid based on ellipses makes use of the Trammel of Archimedes - a mechanical devise where a fixed bar with endings sliding on two perpendicular tracks. The envelope of the moving bar is then the astroid. A fixed point on the bar traces out an ellipse. The axes of the astroid are defined to be the two perpendicular lines passing its cusps. The length of the tangent cut by the axes is constant. Thus the astroid can be constructed as the envelope of co-axial ellipses.

By reducing the difference between the lengths of the semi-major and semi-minor axes of the ellipse, characterized by the eccentricity, the ellipse can be made to converge to a circle when the eccentricity approaches one or the difference becomes zero. This gives a natural mechanism for defining and extrapolating the astroid to a circle, thus rendering a smooth transition from the astroidal trunk-root junction to the cylindrical trunk (with a roughly circular cross-section). The algorithm makes use of this approach to build the trunk of the tree. It creates an envelope of ellipses of equal semi-major and semi-minor axes length sums at every *z* level, with this sum varying with the level, thus producing a natural transition from astroid to ellipse (Stand Curves).

The algorithm also implements a semi-randomized transition to produce more organic irregularity in the trunk and branches. The family of asteroids and quadrifoliums are shown in Figure 4. The mathematical formulation of these geometric entities is depicted in Table 1.

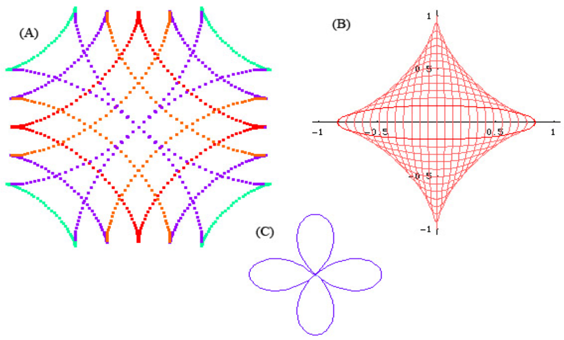

Figure 4: (A) Family of astroids; (B) Astroid as envelope of ellipses; (C) Quadrifolium.

Table 1: Mathematical modeling of the ellipse, astroid and the quadrifolium.

Ellipse	Parametric: {a*cos(t), b*sin(t)} $0 < t \leq 2\pi$ Cartesian: $x^2/a + y^2/b = 1$ $b=a*(1-e^2)^{1/2}$
Astroid	Parametric: $\{\cos(t)^3, \sin(t)^3\}$, $0 < t \leq 2 * \pi$ Cartesian: $(x^2 + y^2 - 1)^3 + 27 * x^2 * y^2 = 0$ Equivalent equation: $x^{2/3} + y^{2/3} = 1$
Rose (Rhodonea)	Polar equation: $r=\cos(p/q*\theta)$; p, q are integers (typical range: 1 to 11) Quadrifolium: p/q =2 Cartesian Equation for a 4-pedaled rose $r=\cos(2*\theta)$ rotated by $2*\pi/8$ is $(x^2+y^2)^3==4*x^2*y^2$

3.2 Trunk Bark Modeling

Traditional methods for modeling the trunk bark involve the use of bump maps. In this paper, an alternate scheme of geometry based parametric modeling of the bark using concentric layers is presented.

3.2.1 Phenomenon

Plants generally grow as cylindrical structures above the ground once their roots have developed. The local geometry of the trunk can be formulated as cylinders. As they grow, the bark expands in concentric circles. The most actively growing part of the trunk is the innermost ring, which gets pushes out the outer rings. The cylindrical rings at the outer layer are the oldest and this phenomenon is used in the dating of trees. Because the outer layers are the oldest, the bark eventually loses flexibility and becomes taut. The pressure from the inner cylinders causes the external bark to crack up. Crevice-like structures are formed. Since pressure is directed radially outward from an equidistant center, the cracks in the bark appear at points almost equidistant on the circumference of the bark. The radial pressure also causes the vertical cleavage lines. Furthermore, the effects of weathering create random fissures on the bark.

3.2.2 Algorithm

Incorporating this understanding into our algorithm, we generate the fissured geometry of the bark: a superimposition of two cylindrically varying structures. The face or planes of the tessellated polygons on the outer structure are shrunk to smaller triangles to simulate the cracking up of the bark. This is done by reducing the dimensions of the triangles along the vertical direction (simulating a radial bark expansion along the cross-section of the trunk) in a pseudo-random fashion. A randomized polygon removal scheme has been employed to produce the effect of weathered fissures. In other words, some polygonal along the vertical dimension of the trunk are arbitrarily removed. These effects result in the fissured bark geometry demonstrated in Figure 5. Application of different texture maps to the two cylinders yields the effect of a chiseled outer bark on a younger inner bark.

3.3 Branch Modeling

Branches are modeled as cylinders with diameters constrained by parameters of radii and tilt. Three specific issues were considered in the generation of the branches. Combining cylinders at junctions is a cumbersome process and involves a lot of projection geometry. In nature, branch junctions can be geometrically categorized as: T-junctions, Y-junctions and I-junctions.

T-junctions occur when a branch grows out of another, which extends much further. Such junctions are modeled as composing of cylindrical branches with semi-randomly varying radii and sinusoidally varying base, embedded at a certain distance into the parent branch, thereby producing smooth transition geometry.

For the I-sections or continuation branches, which chiefly occur when the branches have low radii of curvature or a high degree of curvature, smoothing effects are produced by extrapolating the parent cylinder and semi-randomly varying the gradient of inclination weighted by the parent cylinder radius to yield the child cylinder top radius, rather than create two separate cylinders.

For the Y-sections, the joint is more complex, as the top radius of the parent cylinder and the bottom radius of the child are different. Mounting the child cylinders on the parent would produce huge areas of discontinuity. This issue has been solved by using a sinusoidally varying base for the child cylinder, as in the case of the T-junction, and creating a spherical geometry atop the parent cylinder.

4 IMPLEMENTATION ISSUES –GEOMETRY AND TRIANGULATION

Parameterization and generalization of various aspects of the double-barked geometric tree model enables creation of different types of trees. Limits of feasible space and time complexity involved in the execution of the program required the development of optimization parameters. These include the *Triangle Reduction Factor*, the factor by which the number of triangles is reduced for rendering (by absolute or relative levels), the *Curve Speed-up Factor* that controls the number of points for the ellipse generator, the *Generator Ellipse Count*, that controls the number of ellipses that form the Astroid or Circle and the *Base Randomizer Factor* that determines the number of triangles in the complex astroidal base.

Various tree type characterizing parameters include the *Base Spread Factor* which determines the ratio of the base to the trunk of the tree, the *Height of Trunk* parameter, the *Chisel mode* that allows one to specify if the outer bark is to be shrunk, randomly chiseled out or both, two *Chisel factors* to control the chiseling in the two modes, the *Base* and the *Top Radii*, the *Bark Depth Parameter*, that determines the depth of the inner bark from the outer one and the *Tilt spread* to create a tilted tree.

After the generation of the inner and outer bark geometrical models as envelope of ellipses with varying major and minor axes and necessary tilting of both the inner and outer bark cylinders, the resulting geometry is Delaunay triangulated and converted to patches. The Delaunay triangulation of a point set is a collection of edges satisfying an "Empty Circle" property (i.e., for each edge we can find a circle containing the edge's endpoints but not containing any other points (UCI – Delaunay)). The outer bark triangles, after performing necessary limiting by count, are scaled, and some triangles are eliminated based on chisel ratios, thus rendering the natural bark effect.

Normals for the triangles are generated by simple cross-product of the edges. In order to produce an elegant look with Phong shading, the normal values at each vertex are calculated as the average of the normals of the different triangles that share the vertex.

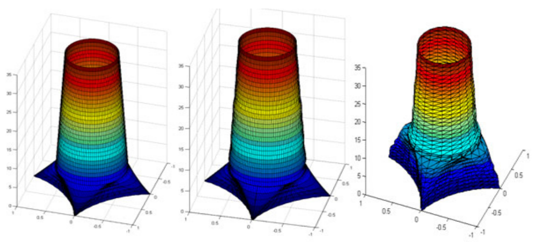

Figure 5: Trunk-Root Modeling.

Implementation of the local geometric modeling of the branches was done similar to that of the trunk, but with a single bark structure and the addition of functionality to solve the problem of the varying junctions. Additional parameters include *junction type*, *previous radii*, *initial* and *final center coordinates*.

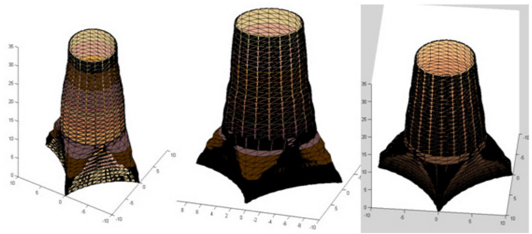

Figure 6: Double-Bark Modeling.

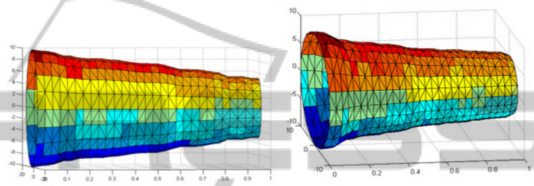

Figure 7: Branch Modeling.

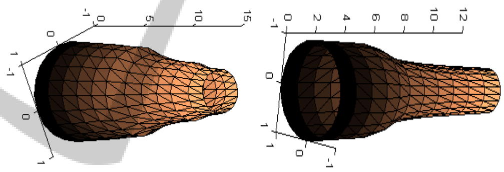

Figure 8: Branch Modeling with Color Mapping.

5 RESULTS

Figures 5 and 6 demonstrate the modeling of trunk using the trunk-root junction algorithm and the double bark algorithm for a variety of parameters. While the generated visualizations presented in this paper for the trunk-root geometry largely consists of convex boundaries, the system can also be used to generate junctions with concave boundaries. The concave boundaries can be obtained by reducing the bounds on the extreme values used for the ellipse axes lengths in the envelope generation process. This is shown in Figure 4B. Alternatively, quadrifoliums (Figure 4C) can be used, depending upon the tree species to be rendered. Figure 7 shows the modeling of branches using the randomized cylinder approach and Figure 8 after color mapping. Figure 9 demonstrates the final rendering of the entire tree structures using a basic tree rendering program taking input from the modeled trunk-root junctions and branches as templates for the rendering. It can be seen that the final rendering, though limited in terms of visual quality due to simplicity of the tree renderer, provides realistic visualization of the geometry of trees similar to fig and spruce, that posses trunk-root junctions above the ground.

Figure 9: Complete Tree Rendering.

6 FUTURE WORK

The next step involves the optimization of the code for real-time rendering of forests and other foliage using GPU systems. Development of plugins for usage with *Xfrog, Maya* and *L-Studio* will enable practical testing of the developed algorithms on a larger scale. The developed algorithms can also be extended to help render other types of trees with unique geometry characterization requirements for modeling.

REFERENCES

Jijoon Kim, A Growth Model for Root Systems of Virtual Plants with Soil and Moisture Control, IEICE Trans, 2006.

P. Prusinkiewicz and A. Lindenmayer. The algorithmic beauty of plants. Springer Verlag, 1990.

J. Bloomenthal. Skeletal Design of Natural Forms. PhD thesis, University of Calgary, 1995.

J. C. Hart and B. Baker. Implicit modeling of tree surfaces. In Implicit Surfaces, 1996.

P. Federl and P. Prusinkiewicz. Finite element model of fracture formation on growing surfaces, ICCS 2004.

S. Lefebvre and F. Neyret. Synthesizing bark. Eurographics Workshop on Rendering, 2002.

F. Neyret. Modeling, animating, and rendering complex scenes using volumetric textures. IEEE Transactions on Visualization and Computer Graphics, 1998.

Xahlee, Special Curves http://www.xahlee.org/SpecialPlaneCurves_dir/specialPlaneCurves.html

Stand Curves, http://www-groups.dcs.st-and.ac.uk/~history/Curves/Curves.html

UCI Delaunay, http://www.ics.uci.edu/~eppstein/gina/delaunay.html