Texture, Displacement and Immersion: A Model for Tree Rendering

Miguel Chover†, Roberto Vivó‡, Ricardo Quirós†, Xavier LLuch‡

† Departamento de Informática. Universidad Jaume I.
Campus de Penyeta Roja. 12071 Castellón. España
Tel: ++ 34 64 345771 Fax: ++ 34 64 344858
e-mail: chover@inf.uji.es

‡ Departamento de Sistemas Informáticos y Computación.
Universidad Politécnica. Camino de Vera s/n.
46071 Valencia. España

Abstract

This paper presents a new method to generate realistic details on tree surfaces and structures. The method combines texture, displacement and immersion techniques. We use an extension of L-Systems, namely random L-Systems, to construct the tree, and represent its structure of branches using parametric surfaces (cones). Adding displacement at the surface level by means of 2D and 3D functions, and deviation at the structure level by immersion in a 3D function, we perturb the surface definition. The parametric definition of the surface is adequate to apply textures and reduces the spatial cost of the previous tree representations. In addition, the proposed technique can even manage further detail than previous ones.

1. Introduction

In this work we introduce a useful method to represent with great detail tree shapes and structures. The interest of this work is not only the modeling of trunks and branches, but also the modeling of features like irregularities and attributes of tree surface (i.e.: deformations, deviations, bumps, coloring, etc.).

A great effort has been spent in recent years addressing the problems of producing realistic images of trees. These efforts include work on fractals [Smit84], geometric models [Aono84], particle systems [Reev85], textured surfaces [Gard84] [Bloo85], L-Systems [Prus90], combinatorial analysis [Vien89] and developmental botanical models [Reff88] [Holt94]. Regardless of the model used for the construction of trees, the principal elements of their representation are geometry and texture.

Being the texture methods satisfactory, the geometry is the major drawback of previous models regarding the realism of the final image. If we use generalized cylinders [Bloo85], Bezier curves [Holt94], etc. to render trees, they seem very smooth, and this is exactly opposite to the nature of the tree shapes that we want to model.

The basis of our method is a parametric representation of analytic surfaces to build the geometry, and the application at the surface and structure levels of some functions modifying the tree definition [Chov93]. These surfaces can be easily textured with the most common techniques developed as texture mapping [Heck86], bump mapping [Blin78], and solid texturing [Perl85] [Peac85]. With this approach we can model the complex nature of tree structures without using polygons in the modeling phase. The conversion from the parametric to the polygonal model can be carried out easily if required.

In the following sections, we will explain how from a description of the tree in terms of a random L-system, we generate tree structures formed by parametric surfaces including perturbation functions. Section 2 describes tree generation, section 3 analyzes the definition of the perturbed surface, section 4 discusses the implementation of the model, section 5 shows obtained results and section 6 is dedicated to future research possibilities.

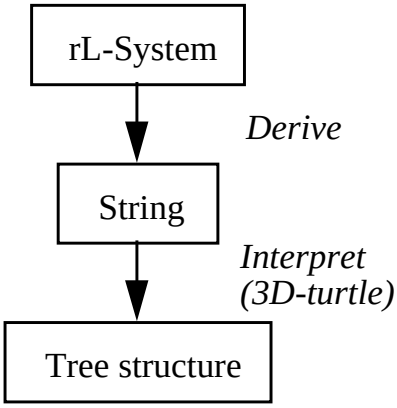
2. Tree Generation

Our method to generate tree shapes uses L-Systems, a modeling technique based in the concept of parallel rewriting grammars. These grammars are similar to those proposed by Chomsky [Chom56], but that productions are applied simultaneously, and there is no distinction between terminal and non-terminal symbols. An extensive description of L-Systems and their applications can be found in [Lind90].

Actually, we work with an extension of L-system, namely random L-System (rL-System for short). In the latter a set of random variables is added to the model, each one taking values in a probability space according to a distribution function. Then, we obtain different strings from the iteration of the same rL-System. Variations on the final

string are controlled by carefully choosing the sample spaces of the random variables. See [Quir92], [Lluch93] for details.

With rL-Systems we have a model to construct self similar natural structures as: cellular systems, plants, trees, etc. by generating strings of symbols in the system language. To generate the tree structure we use a 3D turtle which interpret a string of symbols, and moves in the 3D space generating the parameters of branch surfaces (see the work scheme in figure 2.1.).

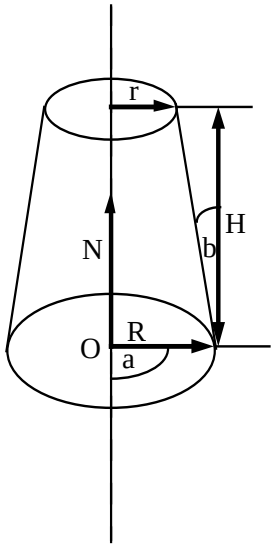


-Figure 2.1- work scheme

The branch surfaces are cones with the following parameters: branch origin (x,y,z), axis direction (nx,ny,nz), height (H), initial radius (R) and final radius (r) (see figure 2.2.). These cones can be represented in a wide variety of ways, but we have chosen parametric surfaces described analytically. Using this representation, a cone S(u,v) is defined by three functions X,Y,Z of two variables u,v:

-$X(u,v)=x+(R*(1-v)+r*v)*\sin(2\pi*u)$
-$Y(u,v)=y+H*v$
-$Z(u,v)=z+(R*(1-v)+r*v)*\cos(2\pi*u)$ $\quad u,v \in [0..1]$

In this way, a tree is a set of branches defined by parametric surfaces, to which we will apply deformations by adding displacement and immersion functions. Now in the next section we can see how the perturbation method is applied.



- Figure 2.2 - Truncated cone

3. Definition of the Perturbed Parametric Surface

The model to generate perturbations on the tree structure has two different levels of application. First, at the branches level we can apply displacement by means of 2D (displacement-mapping) or 3D functions on branch surfaces. Second, at the tree level we can apply distortion of the trunk and branches by structure immersion of the tree in a 3D function. In this way, and with a simple model of perturbation (applied directly on the parametric surface definition), we can represent a wide variety of trees, simply by changing the perturbation functions and their parameters.

Our proposal includes in the parametric surface definition S(u,v) a set of perturbation functions which are used to add the mentioned levels of details.

we will use the following terms:

-S(u,v): cone parametric function.
-NormalS(u,v): surface normal.
-DS(u,v): cone parametric functions with perturbation.
-Table(u,v): a matrix of displacement values (2D function).
-Turbulence(x,y,z): the typical function of solid textures (3D function)[Perl85]
-BD: branch displacement
-TD: tree displacement

At branch level:

We apply displacement based on 2D and 3D functions, on the normal direction of the tree. As result, we get a new parametric surface definition including the displacement.

a) 2D function:

$DS(u,v)=S(u,v)+NormalS(u,v)*Table(u,v)*BD$

b) 3D function:

$DS(u,v)=S(u,v)+NormalS(u,v)*Turbulence(S_X(u,v),S_Y(u,v),S_Z(u,v))*BD$

The Table function is a displacement map, that can be generated by hand (a paint program) or procedurally. We can generate a collection of displacement maps for tree bark and use them as a library.

We can also use a turbulent function or any 3D function to displace the branch surface according with the effect that we want to obtain (see results section and color plate 4).

At tree level:

To obtain a new surface definition, we submerge the tree in a 3D function and translate the points according with the function values.

$DS(u,v)=S(u,v)+Turbulence(S_X(u,v),S_Y(u,v),S_Z(u,v))*TD$

The turbulent function is used to randomly perturb the geometry of the tree structure. To obtain more defined effect we can use functions that generate tropisms (i.e.: gravity and wind tendencies).

We can combine all these simple functions to obtain a complex model that successfully represents the tree nature. With this representation, we can generate different tree structures and branch appearances by varying the perturbation functions of the model.

At this point, we always have a parametric definition of the surface being suitable to apply texture-mapping techniques easily, and of course to use other texture techniques as solid texturing. We have also developed some procedural textures to generate tree barks by using 2D and 3D functions.

Using this technique we define a tree with displacement by means of 2D and 3D functions at branch level, and with perturbation by immersion in a 3D function at the structure level, and combine this displacement with texture mapping, bump mapping and solid texture simultaneously. The combination of all these possibilities offers all the control required to effectively define a complex geometry and the surface detail.

The advantages of the model can be summarized in three points:

- compact model: The model is generated by adding a set of functions to the parametric surface definition.

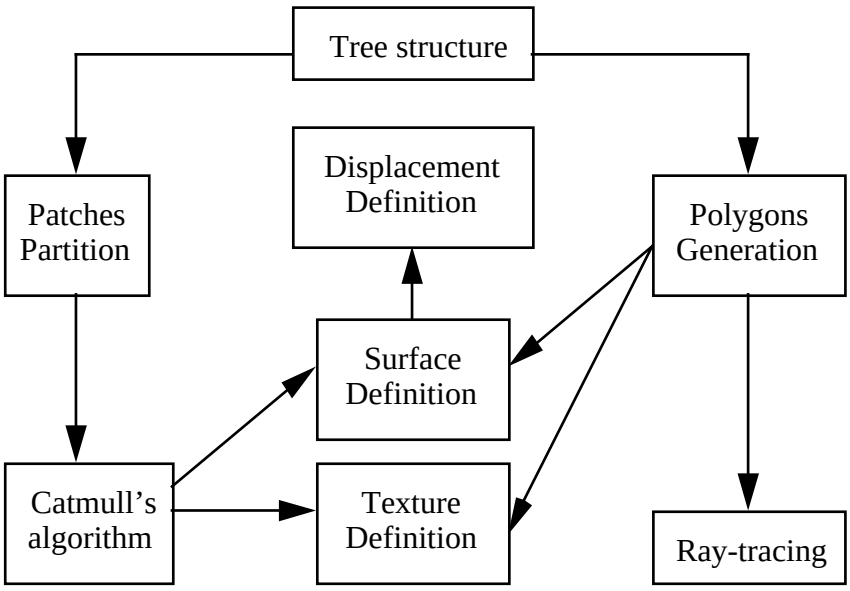
- simple utilization and more possibilities: The model can be easily used. By varying some parameters in the model, we can achieve very different results.

- reduced spatial cost: The reduced spatial cost is one of the major advantages. If we want to model tree irregularities by perturbing a polygonal model, we need to manage a great amount of data. Working with Bezier or B-Spline surfaces, we also have a lot of control points. In our method there is not a basic data structure. Instead, we define surfaces using functions, and some structure as displacement and texture maps, only to complete their definition.

4. Implementation

In this work, we present an initial approach in the rendering of tree shapes. By using the rL-System, we generate strings, which a 3D turtle interpret to generate a file. This file contains all the tree branches with the parameters used in the branch definition (parametric cones).

Now, we render all the branches using the perturbed parametric functions. We use Catmull's [Catm76] visualization algorithm and Phong's illumination model. In case we want more realism we can convert the parametric model to a polygonal one, and render it using ray-tracing (see figure 4.1.).



- *Figure 4.1 - Application modules*

In the application we have combined a set of displacement and texture parameters.

The displacement parameters are:

- The 3D displacement functions at branch level.
- The 2D displacement maps at branch level.
- The 3D displacement functions at tree level.
- The displacement limits at tree and branch level (TD, BD).

The texture parameters are:

- Texture methods.
- Texture functions.
- Texture maps.

5.- Results

We present some results showing the effects that we can achieve by applying this set of techniques. We start from a tree of simple structure, and we show how, by using the described method the images become more and more realistic.

The terrain appearing on the color plates has been generated by the same displacement method and textured with different maps . The background is a textured surface.

First, we show in color plate 1 a tree generated by a rL-System without displacement nor immersion techniques. As we can see, the tree branches have a smooth contour. In this case, only texturing is applied.

In color plate 2, we can see the tree structure perturbed at branch level. There are some bumps given by a 3D function applied in the direction of the surface normal.

In color plate 3, we show the tree structure perturbed at branch and structure level. Through the immersion of the tree structure in a 3D noise function, a deviation of branch paths is produced.

In color plate 4, we show the same tree cover with snow. This effect is obtained by a 3D texture function and a small displacement applied to points with normals going up.

6.- Conclusions

We have shown the possibilities of the displacement and immersion methods combined with different texturing techniques. With this approach we make a new effort in the production of realist images of trees and natural scenes. By using perturbed parametric surfaces we can model a wide variety of complex surfaces, difficult to model with other approaches.

The most valuable characteristic of this method is the accurate and simple control of the surface perturbation. It is not necessary to construct an initial structure and then perturb it. Instead, we include this perturbation in the surface definition. Other

advantage is the reduced spatial cost due to the small set of parameters that define non deformed parametric surfaces: initial radius, final radius, height, etc., which do not include perturbation.

The possibilities of this technique are very interesting and we can see how with a reduced set of parameters we can get very different results. In addition we can develop new textures for tree barks and new immersion functions, to combine them easily.

Our future research will be focused in applying the texture, displacement and immersion model to represent real tree species. We are interested in the development of functions that simulate different effects on tree surfaces, like the snow effect that we have shown. We want also to apply this method to other structures in the nature, like clouds, terrain, etc.

References

[Aono84] M. Aono and T.L. Kunii, *"Botanical Tree Image Generation"*, IEEE CG&A, 1984, vol. 4, pp. 16--34

[Blin78] James F. Blinn, *"Simulation of Wrinkled Surfaces"*, Computer Graphics (SIGGRAPH'78 Proceedings), 1978, pp. 286--292

[Bloo85] J. Bloomenthal, *"Modeling the Might Maple"*, Computer Graphics, 1985, vol. 19, no. 3, pp. 305--311

[Catm76] E. Catmull. "A Subdivision Algorithm for Computer Display of Curved Surfaces". Ph.D. Thesis, University of Utah, 1976

[Chom56] N. Chomsky, *"Three models for description of languages"*, IRE. Trans. on Information Theory, 1956, pp. 113--124

[Chov93] Miguel Chover, Roberto Vivó y Ricardo Quirós, *"Síntesis de Imagen con Hipertexturas"*, III Congreso Español de Informática Gráfica, Granada, 1993, pp. 276--287

[Gard84] G.Y. Gardner, *"Simulation of natural Scenes Using Textured Quadric Surfaces"*, Computer Graphics, 1984, vol. 18, no. 3, pp. 11--20

[Heck86] Paul S. Heckbert, *"Survey of Texture Mapping"*, IEEE Computer Graphics and Applications, 1986, November, vol. 6, no. 11, pp. 56--67

[Holt94] Matthew Holton, *"Strands, Gravity and Botanical Tree Imagery"*, Computer Graphics Forum, 1994, vol. 13, no. 1, pp. 57--67

[Lind90] A. Lindenmayer, P. Prusinkiewicz, *"The algorithmic beauty of plants"*, New York, Springer-Verlag, 1990

[Lluc93] J. Lluch, R. Quiros y B. Castellanos, *"Aplicaciones de los Sistemas Paramétricos Aleatorios a la generación de árboles y plantas herbáceas"*, III CEIG , Granada, 1993, June, pp. 195--207

[Peac85] Darwyn R. Peachey, *"Solid Texturing of Complex Surfaces"*, Computer Graphics (SIGGRAPH '85 Proceedings), San Francisco (CA), B. A. Barsky, 1985, July, vol. 19, pp. 279--286

[Perl85] Ken Perlin , *"An Image Synthesizer"*, Computer Graphics (SIGGRAPH'85 Proceedings), San Francisco (CA), B. A. Barsky, 1985, July, vol. 19, no. 3, pp. 287--296

[Perl89] Ken Perlin, *"Hypertexture"*, Computer Graphics, 1989, vol. 23, no. 3, pp. 253--262

[Prus90] P. Prusinkiewixz, J. Hanan, *"Visualization of botanical structures and processes using parametric L-systems"*, Scientific visualization and Graphics simulations, D. Thalmann, 1990, pp. 183--201

[Quir92] Ricardo Quiros, Javier Lluch y Roberto Vivó, *"Generación de Árboles y Plantas mediante Sistemas-L"*, II Congreso Español de Informática Gráfica (Actas del Congreso)., 1992, June

[Reff88] P. de Reffye, C. Edelin, J. Francon, M. Jaeger and C. Puech, *"Plant Models faithful to Botanical Structure and Development"*, Computer Graphics, 1988, vol. 22, no.4 , pp. 151--158

[Reev85] W.T. Reeves and R. Blau, *"Approximate and probabilistic algorithms for Shading and Rendering Structured Particle Systems"*, 1985, Computer Graphics, vol. 19, no. 3, pp. 313--322

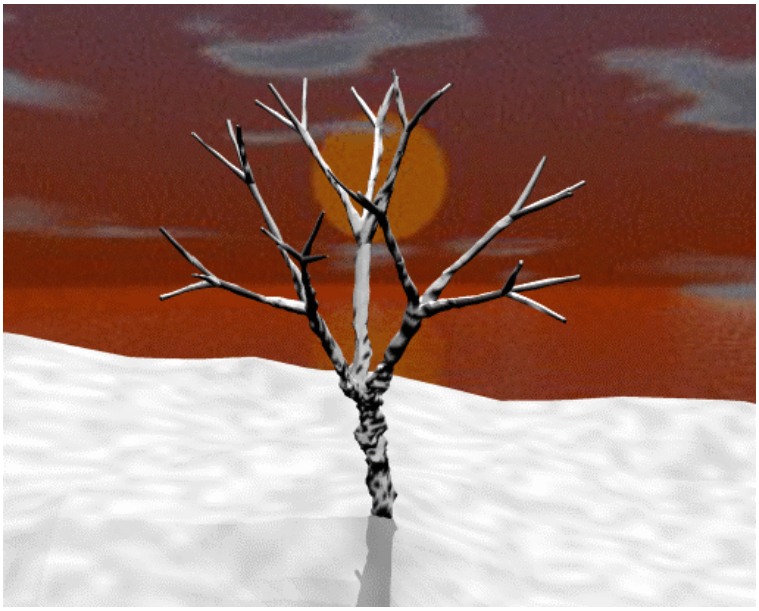
[Smit84] A.R. Smith, *"Plants, fractals and Formal Languages"*, Computer Graphics, 1984, vol. 18, no. 3, pp. 1--10

[Vien89] X.B. Viennot, G. Eyrolls and N. Janey, *"Combinatorial Analysis of Ramified Patterns and Computer Imagery of Trees"*, Computer Graphics, 1989, vol. 23, no.3, pp. 31--40

- Color Plate 1 -
a tree without displacement technique

- Color Plate 2 -
a tree with displacement at branch level

- Color Plate 3 -
a tree with displacement at branch level and immersion at structure level



- Color Plate 4 -
a tree with snow effect