GreenArt: A Tool for Non-Photorealistic Rendering of Plants and Trees

C. Campos[1], R. Quirós[2], J. Huerta[2], E. Camahort[3] , J. Lluch[3] and R. Vivó[3]

1 Departamento de Informática, Universidad de Vigo, Ourense, Spain
2 Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I, Castellón, Spain
3 Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia, Spain

Abstract

Plant and tree rendering is critical to many computer graphics applications, like interactive walkthroughs, architectural presentations, entertaintment and computer games. Current research focuses on realistic modeling and rendering of plants and trees. In this paper, however, we focus on non-photorealistic rendering of these models. We introduce GreenArt, a tool for a modeling and rendering non-photorealistic plants and trees. We present GreenArt's functionality and capabilities and we show some examples of scenes generated using the system.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

For several decades, modeling and rendering of synthetic plants and trees has received much attention among computer graphics practitioners. There are serveral application areas of this field, particularly, modeling of forests, gardens and other plant environments. Plant and tree models are known to be expensive to generate, process and render. This is due to the huge amount of geometry required to model this type of scene. A tree model can be made of hundreds of thousands of polygons, and a forest model may contain hundreds or thousands of trees. Therefore, interactive rendering of these scenes requires special custom-made tools.

Such tools support both modeling and rendering of plants and trees. Traditionally, more effort has been devoted modeling techniques and tools. Nowadays, there are applications that allow using different combinations of techniques to generate very complex structures. These structures can then be edited and realistically rendered using the same applications. Most of these applications are based on L-systems [8, 9, 10, 11].

Currently, research in plant and tree rendering focuses on achieving interactive rates. For these purpose, we have developed a tool that allows tree modeling using Random Parametric L-Systems (RL-Systems) [17]. This tool, called Green, generates a tree's geometry by deriving and interpreting an RL-System. It also includes conventional rendering based on OpenGL, and a more realistic rendering based on POV-Ray.

Recently, a new breed of rendering techniques has appeared. Non-photorealistic rendering wants to achieve a less realistic and more expressive look in computer generated imagery [3, 4]. Such techniques include methods to obtain more artistic looking images [1, 7], like painterly rendering, toon shading, and pen-and-ink illustration [5, 6]. In all these applications, plant and tree models are commonplace. Therefore, we need specific techniques to expressively render these models.

In this paper we introduce GreenArt, a tool for non-photorealistic rendering of plants and trees modeled using Green. The paper is organized as follows. Section 2 summarizes Green's features for modeling plants and trees. Section 3 describes GreenArt, including both its underlying structure and its graphical user interface. In Section 4 we show some results obtained with our system. Finally, Section 5 presents some conclusions and directions for future work.

2. Plant and Tree Modeling: Green

Green is a plant and tree modeler based on Random L-Systems (RL-Systems) [2]. These systems are an extension of Parametric L-Systems [8], widely used to generate selfrepeating natural structures. RL-Systems include random variables so that multiple unique individuals may be generated starting from the same system. Green contains context-sensitivity, pruning, logical and arithmetic expressions, interrogation modules and the possibility of adding new modules to improve the derivation process [17].

The modeling process generates an RL-System, which contains the declaration of the random variables to be used, the initial string (axiom) and the production rules used in the derivation. This system defines the geometry of the resulting plant and tree. Using Green we have produced data amplification models such as fractals, trees, corals and other geometric substitution models [12]. Figure 1 shows an example rendering that was generated with Green.

Figure 1: *An image generated with Green.*

Some of Green's features focus on speeding up the rendering of scenes containing many trees. We have developed different approaches for branches and leaves. For example, we generate a single polygonal mesh representing a branched structure from a tree modeled with Green. The final model shows smooth transitions in the tree areas joining the branches. That way we avoid the overlapping problem, as well as geometry discontinuities that appear when using multiple meshes. The mesh supports both homogenous texturing and simplification algorithms [14]. On the other hand, we have proposed procedural multiresolution to reduce the number of branches used to render the tree. We build procedural models that reflect a tree's visual structure at different resolution levels. We generate a new chain of modules with embedded multiresolution information. The algorithm is based on a metric that quantifies the relevance of the branches of a tree. The representation supports efficient geometry extraction and produces good visual results. [15].

Finally, we have developed a new method for realistic interactive rendering of tree foliage. Our methods starts by pre-processing an RL-system and storing a hierarchy of texture images. For each frame, we traverse the texture tree and retrieve an appropriate set of images. Those images are blended with other images to obtain the rendered tree. We substantially reduce the number of polygons, enabling interactive rendering and smooth transition between levels of detail. This technique can be easily applied to computer games and other interactive applications that require plant and tree rendering.

3. Non-Photorealistic Rendering: GreenArt

Given an RL-system we generate an image in two steps: (i) we derive the RL-system and obtain a chain representing a tree, and (ii) we interpret the chain using a 3D turtle and obtain the geometric model of the tree. Green contains an interpreter that generates low-quality OpenGL-based renderings of an RL-system. It also supports the generation of a geometry file suitable for rendering using POV-Ray, a free ray tracer.

GreenArt is a new alternative interpreter for Green. When GreenArt interprets a chain, it generates geometry that can be rendered using one of several non-photorealistic techniques, like painterly, hatching and sketching. The interpreter uses a 3D turtle to build a tree data structure. The data structure is complemented with information on what rendering technique to use and how. Figure 2 shows two example images of two different non-photorealistic rendering techniques obtained with GreenArt.

Figure 2: *Two examples of non-photorealistic renderings produced with GreenArt.*

We describe GreenArt's main features: the data structure that supports non-photorealistic rendering, and the graphical user interface used to configure and display the renderings. Our goal in this paper is to describe the tool. For more details about the computer graphics techniques underlying the tool, the reader is referred to our other work in this area [16, 17].

3.1. Data Structure

GreenArt's functionality is based on the data structure used to store a plant or tree. Given a chain generated by deriving an RL-system, we use a 3D turtle to interpret the symbols contained in the chain. Symbols have different meanings. There are: (i) turtle control commands, like rotate and move, (ii) commands that generate graphics objects, like cylinder and instance, and (iii) branching control commands, like push and pop. The 3D turtle can add to the scene the following geometric elements: branches, leaves and instances of other objects, like flowers and fruits. A branch is represented by its endpoints and its thickness at each endpoint. Leaves and object instances are located and oriented with respect to a canonical system. They are always associated to a branch.

The turtle starts from an initial position and generates a tree data structure containing all the information pertaining to its branches, leaves and other object instances. Each geometric element has two representations, a set of parameters and a set of brush strokes. These representations can be used to render a realistic and a non-photorealistic image of the element, respectively [16]. Here is the information stored for the nodes of the hierarchical data structure.

Node

- Branch parameters: p1, p2, w1, w2
- List of brush strokes
- List of leaves
- Level of detail selection criteria
- Simplified leaf model
- List of object instances
- Convex Hull
- List of children nodes

Leaf

- Leaf parameters: p, H, L, U
- List of brush strokes

Instance

- List of brush strokes
- Level of detail selection criteria
- Simplified object models

Stroke

- Associated polygon
- Brush stroke index
- Color

The **Node** data structure contains all the information associated to a branch, including its geometry, set of brush strokes and leaves and object instances. The structure supports multiresolution for branches, leaves and object instances [17]. Storing the convex hull associated to each branch supports using a texture image instead of geometry for rendering. Convex hulls are also useful for sketching and hatching.

3.2. Graphical User Interface

Once the tree data structure has been built, GreenArt allows the selection of different rendering parameters for the tree. The GUI that makes this possible also supports:

- illumination control
- camera control
- selection of a brush stroke family
- randomness of stroke placement
- background and canvas configuration

During normal use of the GUI changes in the rendering parameters immediately affect the image currently in display. After modeling and rendering, GreenArt allows the optimization of the tree data structure for future rendering using either OpenGL or a game engine.

4. Results

In this Section we present some results obtained with GreenArt. The scenes contain plants and trees that were modeled using RL-systems. Figure 3 shows the same tree species rendered in two different ways: only with branches and leaves, and with branches, leaves and object instances. The RL-system used to model this tree is based on a ternary tree model like those described in [8, 16].

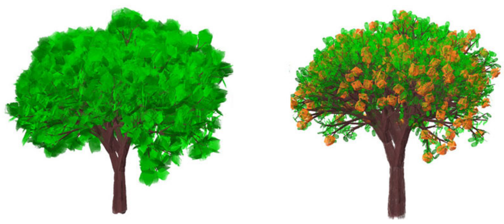

Figure 3: *A tree species rendered only with branches and leaves, and with branches, leaves and object instances.*

Some tree species, like cone tree and cypress, contain so many, often tiny, leaves that each can not be represented by a brush stroke. In those situations we use object instances to represent sets of neighboring leaves (see Figure 4 for an example). The RL-systems shown in Figure 4 are based on a monopodial tree model [8].

Figure 4: *Two trees whose leaves have been represented using object instances.*

Finally, we have used RL-systems to generate the simple herbaceous plants shown in Figure 5 [8]. This figure was composed using Impressionist, GreenArt and several brush stroke families supported by both tools.

Figure 5: *Two plants rendered using GreenArt.*

5. Conclusions and Future Work

We have introduced GreenArt, a tool for non-photorealistic rendering of plants and trees. We have presented a system for modeling plants and trees using RL-Systems. We have shown how a chain generated by an RL-system can be interpreted by GreenArt. During interpretation GreenArt generates a hierarchical data structure that is further employed for rendering using painterly rendering and pen-and-ink illustration.

We are currently working on improving the speed of our rendering algorithms. We would like to apply our technique to areas like interactive rendering, simulation, and computer games. We are also planning on adding new functionality to GreenArt, like other non-photorealistic rendering algorithms, a better outdoor illumination model and better composition tools for scenes made of multiple species.

Acknowledgements

Partially supported by Spanish Government grants TIC2001-2416-C03-02 and TIC2002-04166-C03-02

References

1. B. Meier, *Painterly Rendering for Animation*, Proceedings of SIGGRAPH'96, New Orleans, pp 477-484 1
2. R. Quirós, J. Lluch, M. Chover, R. Vivó, *Geometric substitution using Random L-Systems*, Computers and Graphics, vol. 20, no. 5, 1996 1
3. B. Gooch, A. Gooch, *Non-Photorealistic Rendering*, A.K. Peters Ltd, 2001 1
4. T. Strothotte, S. Schlechtweg, *Non-Photorealistic Computer Graphics*, Morgan Kaufman Publishers, Elsevier Science, 2002 1
5. M. Webb, E. Praun, A. Finkelstein, H. Hoppe, *Fine tone control in hardware hatching*, Proceedings of NPAR 2002, pag. 53-58. 1
6. R.D. Kalnins, L. Markosian, B. Meier, M. Kowalski, J.C. Lee, P. Davidson, M. Webb, J.F. Hughes, A. Finkelstein. *WYSIWYG NPR: Drawing Strokes Directly on 3D Models*, Proceedings of SIGGRAPH'2002, New Orleans 1
7. C. Curtis, S. Anderson, J. Seims, K. Fleischer, . D.H. Salesin, *Computer Generated Watercolor*, Proceedings of SIGGRAPH 97, Los Angeles, 1
8. P. Prusinkiewicz and A. Lindenmayer, *The algorithmic beauty of plants*, Springer-Verlag, 1990 1, 3
9. P. Prusinkiewicz, M. James, R. Mech, *Synthetic Topiary*, Computer Graphics, 351-358, 1994 1
10. P. Prusinkiewicz, *Modeling of plants and plant ecosystem*, Communications of the ACM, 43(7), 2000 1
11. B. Lane and P. Prusinkiewicz, *Generating spatial distributions for multilevel models of plant communities*, Proceedings of Graphics Interface 2002,Calgary, pag. 69-80. 1
12. M. Chover, R. Vivó , R. Quirós, J. Lluch, *Texture, displacement and immersion: A model for tree rendering*, Proceedings of WSCG'95, Plzen, 1995. 2
13. J. LLuch, M.J. Vicent, R. Vivó, R. Quirós, *GREEN: A new tool for modelling natural elements*, Proceedings of WSCG'2000, Plzen, 2000.
14. J. Lluch, M.J. Vicent, C. Monserrat, S. Fernández, *A single polygonal mesh representing a botanical tree surface*, Proceedings of ICAV3D, Mykonos, 2001 2
15. J.Lluch, E.Camahort, R.Vivó, *Procedural Multiresolution for Plant and Tree Rendering*, Proceedings of AFRIGRAPH 2003, Cape Town, 2003. 2
16. C. Campos, R. Quirós, J. Huerta, M. Chover, J. Lluch, R. Vivó *Non-Photorealistic Rendering of Plants and Trees*, Proceedings of ICAV3D, Mykonos, 2001. 2, 3
17. C. Campos, E. Camahort, R. Quirós, J. Huerta, I. Remolar *Acceleration Techniques for Non-Photorealistic Rendering of Trees*, Proceedings of SIACG 2002, Guimaraes, Portugal. 1, 2, 3