Lworld: An Animation System Based on Rewriting

Hansrudi Noser
University of Zurich, Institute of Computer Technology, Multimedia Laboratory
Winterthurerstrasse 190, CH-8057 Zurich, Switzerland
noser@ifi.unizh.ch

Abstract

Lworld is a computer graphics animation system based on L-systems, a parallel rewriting technique used primarily in computer graphics for plant modeling. Because rule-based programming is a powerful technique, we use it as a basis for a general-purpose animation system. We describe the architecture, the features, and the programming language of the animation system. It is particularly well suited to model fractal curves, plants, fractal landscapes, group animation, visualization, and evolutionary optimizations. Lworld allows users to create real-time animations as well as raytraced image sequences for further movie production. It is freely available, and runs on PCs.

1. Introduction

Modeling and animation by rules is not wide spread. L-system based software are mostly research tools with a cryptic modeling language, or educational software serving to demonstrate the mechanism of production rules. But the work of Prusinkiewicz [4] or Parish [3] shows the inherent potential of rule-based programming by modeling with few rules complex plants and huge cities. Therefore, we developed a general-purpose animation system [1] based on parallel rewriting that enables designers of virtual 3D worlds to experience rule-based programming. The software contains many features that make it suitable for many tasks in visualization, interactive animation, research, education, and raytraced (Rayshade) movie generation.

2. Concept of the animation system

Rewriting is a powerful concept for modeling complex objects. We use this technique not only for modeling plants, but also as the basic principle for a general-purpose animation system. Parallel, timed rules can model not only hierarchical objects, but also their animation, complex behaviors, and interaction. Timed and parametric rules combine in one language the modeling of complete virtual worlds including structured objects, their piecewise-continuous development, their animation, and user interaction.

In computer graphics, an L-system describes a 3D object by an axiom and a set of production rules, also called its grammar. The axiom and the rules are composed of symbols of the L-system's alphabet. From such an L-system, the computer can derive formal symbol sequences from the axiom by a series of parallel rewriting steps where symbols are replaced according to the rules. Furthermore, the computer can visualize the derived symbolic objects by interpreting them as a kind of turtle-graphics language. This modeling principle is particularly advantageous for complex, structured objects for which we can find its grammar. Such a description is mostly compact and can provide immense data amplification factors.

The system we propose supports most features of advanced parallel rewriting techniques, such as timed, parametric, stochastic, bracketed, environmentally sensitive, networked, and behavioral [2] L-systems. At a higher level, we integrated fractal mountains, sound playback, networking, optimization and evolution by genetic algorithms, tropism forces, a particle system, synthetic vision for autonomous actors, as well as a design model for modeling physically correct static trees that visualizes the forces and moments at the tree joints. Figure 1 illustrates the architecture of the animation system. At runtime the application parses user defined L-systems and compiles them into an internal representation of the axiom and the rules. The iterator and interpreter modules are essential parts of the real-time animation loop. At each frame the loop executes first code for frame initialization. Then, the iterator applies all triggering rules to the symbolic object by replacing the corresponding symbols by the rules' right side. The interpreter module then executes the resulting symbolic objects that can be considered as a turtle program, and visualizes the resulting 3D world of the frame on the screen. If autonomous actors with synthetic vision have been declared in an animation, the frame is rendered for each actor into their vision window. Finally, the frame is terminated and the global animation time is incremented by the time step. The software written in C++ (OpenGL/GLUT) runs under Windows98/NT/2000/XP. It is freely available on the Internet [1]. Most existing languages for defining L-system

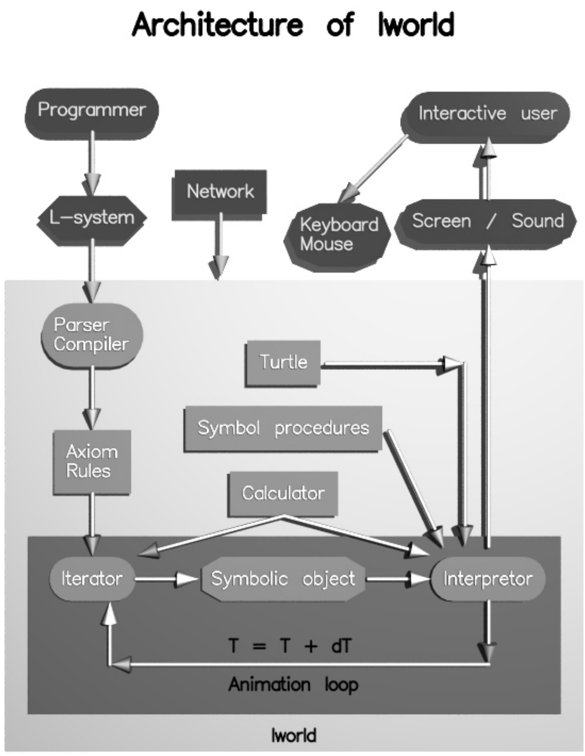

Figure 1. The architecture of the parallel rewriting-based animation system.

rules are somewhat cryptic and difficult to read for inexperienced designers. Therefore, we developed a new rule-language similar to C being familiar to most programmers. The execution model determined by the parallel rewriting paradigm, however, is not sequential and must be carefully studied by designers.

The animation system can be viewed as a virtual machine, defined by status variables that can be set by assignments or procedure calls within parameter and attribute blocks of symbols. The status variables define partially the viewing parameters, rendering features, animation control, and physical models. Actually, we implemented more than 80 symbols representing the alphabet of the L-systems. They can be grouped according to their semantic. Some of them manipulate the turtle state or the rendering mode. Others correspond to geometric primitives, sounds, or particular design models such as tropism forces or particles. Others control user interaction, networking, or the interpretation of the symbolic object. The actual set of implemented symbols, variables, and procedures allows programmers to design a vast variety of animated, interactive, and networked virtual 3D worlds.

3. Examples

Figure 1 showing lworld's architecture is a raytraced picture of an L-system that uses 3D text and arrow symbols. Figure 2 illustrates lworld's features in creating complex raytraced landscapes including fractal mountains with thousands of automatically placed growing plants and group animation realized with a force field based particle system. In the example, the butterflies fly around a flower. Differential equations that are defined by the designer in the corresponding animation file determine their behavior.

Figure 2. Raytraced landscape with fractal mountains, automatically placed plants, and butterflies animated by a particle system.

References

[1] H. Noser. Lworld: An animation system based on rewriting. http://www.ifi.unizh.ch/ noser/Lworld/lworld.html, July 2002.

[2] H. Noser and D. Thalmann. A rule-based interactive behavioral animation system for humanoids. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):281–307, October-December 1999.

[3] Y. I. H. Parish and P. Müller. Procedural modeling of cities. In *SIGGRAPH 2001, Conference Proceedings, August 12-17*, pages 301–308, 2001.

[4] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer Verlag, 1990.