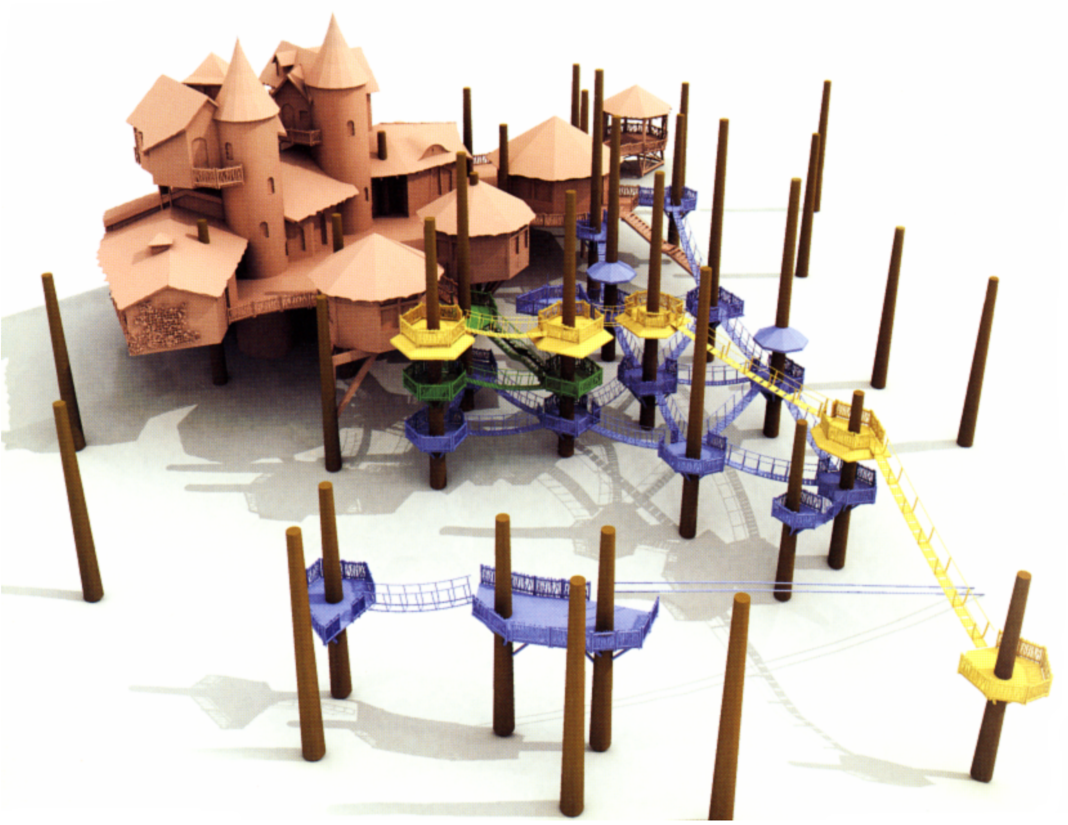
Universiteit Utrecht

Planning Procedural Architecture in Tree-like Geometry
non-realistic procedural game worlds

by Ruud op den Kelder



Abstract

Contents

Chapter 1

Introduction

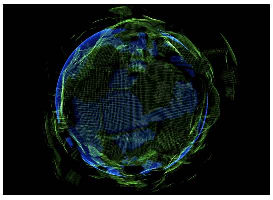
Procedural modeling is an exciting research area with trumendous potential for virtual worlds. Virtual worlds range from games, to simulations and virtual communities. These worlds are often very large and with advances in technology and user demand they rapidily grow larger for each new application. The problem is that space in these vast virtual worlds needs to be filled with interesting content. Creating such high amounts of content by hand does provide nice job opportunities, but if we could generate the same content with a set of parameters and the push of a button the choice is rather obvious. However, time and money efficiency are not the only arguments for the development of procedural methods. By studying real world objects or sets of real world objects in order to dynamically synthesise them in virtual worlds we gain a deeper knowledge of the characteristics of these objects and relationships between different objects.

A couple of succesful existing procedural modeling implementations are interesting to discuss. In games, procedural methods have been employed for the generation of worlds with restricted sets of environmental elements, such as dungeons and mazes. A good example is the game "Diablo 2". The game "Spore" was one of the first games that heavily utilized procedural modeling techniques for a considerable part of content in the game. A very interesting project that is in development at the time of writing, is the game "Love" (figure 1) developed by a single man. Procedural generation is used extensively for every part of this game.

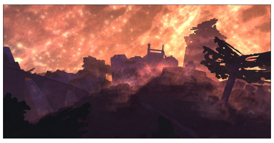
This thesis adresses the problem of creating procedural architecture in the special case of restrictions and possibilities enforced by geometric properties of tree-like support-structures. A side goal for this research was to develop methods for intuitive interactive control for this specific procedural modeling method.

Visual modeling of plant development is a field which started in 1962, when Ulam applied cellular automata to simulate the development of branching patterns [6]. A formalism for modeling plants was proposed by Lindenmayer in 1968, this formalism was called L-systems since. The following definition of an L-system is given by Przemyslaw [6]:

> An L-system is a parallel rewriting system operating on branching



(a) World



(b) Ingame Environment

Figure 1.1: Screenshots of the game Love

> structures represented as bracketed strings of symbols with associated parameters, called modules. Matching pairs of square brackets enclose branches. Simulation begins with an initial string called the axiom, and proceeds in a sequence of discrete deriviation steps. In each step, rewriting rules or productions replace all modules on the predeccesor string by succesor modules.

Przemyslaw [6] has employed and extended the l-system formalism for realistic visualisation of entire plant ecosystems. Within a ecosystem organisms interact with each other and this interaction determines many properties for individual organisms; such as growth rate. Since the original L-system formalism does not account for communication between two processes, Przemyslaw proposed *open l-systems* which incorporates *communication modules*.

L-systems also proved to be useful in the field of urban procedural generation [9]. Muller showed that the L-system formalism could be succesfully used for the generation of road networks and in lesser extent building generation.

Techniques for tree generation in this thesis are based on L-systems, however these techniques are simplified to a certain degree since the modeling of plants is not the maintopic of this thesis. We have used the l-system formalism to generate simple branching tree structures. We did not strive to generate visually realistic models of trees since this has already been achieved by many people before me with impressive results. Instead our simplified method generates the main structure of a tree which is then used as input in our method to generate configurations of architectural elements.

It is the case that traditional general purpose modeling software is very time consuming for the creation of complex scenes as a result of the lowlevel tools they feature. Special purpose modeling tools such as cityEngine [5] (in the case of urban modeling) and speedTree (in the case of modeling plants and trees) provide procedural methods to generate objects within a certain class with very high time effiency. However the human touch still remains a very important part of the modeling procedure. Eastatics are very hard to turn into a set of formal rules on which an algorithm can operate.

The special purpose modeling tool for the generation of organic geometry and treehouse architecture that was developed for this thesis provides the user with intuitive interaction tools to allow easy manipulation.

1.1 Overview

This thesis is structured as follows. In the next section I will discuss related work. In the related work section I will bring my thesis into context with regard to procedural modeling of trees, architecture and generation of levelmaps. A precise statement of the problem and why it's an interesting problem to solve will follow in section 3.

In section 4 I will present the conceptual system model. The method for the generation of the forest layout and the generation of the tree geometry is presented in section 5.

With the forest geometry in place we are ready to review the planning method which is responsible for the construction of a connected graph representing a tree community. I will conclude the method description with a look at the final stage of the pipeline, translating the symbols from the nodes and edges of the graph to the geometrical architectural elements. Section 7 discusses user interactivity and usability of the proposed system. The last two sections present results and conclusions respectively.

Chapter 2

State of the Art in Procedural Modeling

2.1 Plants and Trees

Algorithmic Beauty of Plants by Lindenmayer and Przemyslaw [7] is the first complete work discussing the generation of plant geometry using procedural methods such as L-systems and fractals. To model and visualise realistic ecosystems, Przemyslaw extended the l-systems concept to a system which allowed communication between systems (open l-systems [6] [3]). Visual editing of procedural plants models is discussed in **(author?)** [1].

2.2 Road Networks

The foundation for procedural city and building modeling was provided by Parish and Muller **(author?)** [10] in their paper *"Procedural Modeling of Cities"*. The main contribution of this paper is the use of extended L-systems for the generation of city roadmaps. They also propose a method for the texturing of facades. An intuitive editing approach for road networks with the use of tensor fields and bush techniques is presented by Chen et al. **(author?)** [2].

2.3 Architecture

An attempt was made to use L-systems for the creation of buildings **(author?)** [10], however this did not prove to be effective. L-systems are designed to handle growth-like processes, it has been acknowledged that the construction process of a building is not a growth like process. Instead, building construction is better expressed by series of partitioning steps. These partitioning steps can be described by another kind of rewriting grammar called *set grammar*. In **(author?)** [9] Wonka presents a method for the automatic creation of building using such grammar systems. In this work Wonka introduces the idea of a specialized type of set grammar called *split grammar* which operates on shapes. In **(author?)** [5] the split rules from the split grammar concept are defined in a grammar system called *CGA Shape*, which was the first procedural system for the creation of detailed buildings with consistent mass models. The process of creating a ruleset in CGA shape for a specific type of building is not straightforward and requires a trained expert. Lipp et al. **(author?)** [4] introduce a visual method for the editing of the CGA Shape grammar for procedural architecture to simplify the rule building process.

Chapter 3

Problem Statement

The purpose of this thesis is to tackle a specific instance of the problem of planning connected architectural geometric elements into an environment with existing geometry which is to be used as the supporting structure for the architectural elements. We handle the specific case in which the pre-existing geometry is a virtual forest, using the trunks and branches of the trees as supporting structures for architectural elements such as platforms, bridges, stairs and buildings. We propose several heuristics for the generation of a connected 'tree house community' and using these heuristics we propose a planning algorithm. The architectural elements need a large degree of adjustibility to be able to incorporate them into the irregular environment posed by the tree geometry, therefore we present procedural methods for the generation of these elements. Another goal for this thesis was the design of intuitive procedural modeling user tools.

This work has been inspired by advancements in the fields of procedural plant generation and procedural generation of urban structures like buildings, roads and entire cities. These two fields have been explored to some extent now. Although the generation of urban structures is a relatively new research area compared to procedural plant generation. Never before have these fields been combined for our idea: the generation of architectural geometry controlled and supported by tree-like geometry. With our research we attempt to generate interesting meaningful *'tree-house community'* scenes.

I believe this is an interesting problem to be solved for the following reason. Previous solutions for procedural virtual worlds focussed on realistic synthesis of the real world around us. In games, virtual worlds are designed with high user-entertainment value as top priority. This also means that the configuration of geometry in virtual game worlds must enable the user to enjoy the playing experience. Determining the leveldesign principles that make a virtual world enjoyable is hard on its own. Transforming these principles into formal rulesets is even harder. To make the problem even more complex, the principles for good leveldesign highly depend on the game play rules the virtual world dictates. We present two scenarios that incorporate a set of gameplay rules. From these scenarios and known pricipals of good leveldesign we create heuristics. Using these heuristics we construct a planning algorithm that takes forest geometry, a set of architectural elements and parameters we extracted from the scenarios as input and outputs a geometric configuration of the architectural elements.

An important inspiration for our research is work done on the procedural generation of dungeons.

Chapter 4

Concept

This section discusses the concept for the proposed procedural modeling method with main focus to the planning algorithm for architectural objects.

Chapter 5

Procedural Forest Generation

5.1 Simple Tree Structures

Procedurally generating trees by means of l-systems has had a great amount of succes since the original proposal by Lindenmayer. The L-system formalism is widely applied in academic work and is also succesfully used within commercial applications. This thesis does not focus on the l-system formalism in particular, since it is already a well established theory **(author?)** [7]. However, the following section will describe the basics of l-systems to establish some basic understanding wich will be needed in following sections.

I am interested in the structure of trees and the possibilities and restrictions it poses for placement of architectural shapes. For the purpose of this thesis the generated trees do not have to be visually convincing, however the basic shape should still be identified as a tree. The geometric properties of a tree model that are to interest of us are those that have effect on the possibilities with respect to the incorporation of architectural man-made structures. The tree geometry functions as the support structure for the building blocks I define in detail in section 6.

In this section we will identify geometric configurations within tree geometry that allow for the construction of the proposed set of architectural elements.

5.2 Scattering Techniques for Tree Positions

(author?) [8]

5.3 Ecosystem Modeling

forest: a spatial configuration of a set of trees.

5.4 Multilayer TreeNode Generation

For our method we have to be able to strategically place geometric architectural elements within the generated tree structures. Finding the positions at which these architectural elements can be placed could be performed as a postprocess type process by analyzing the generated geometry, however we have the oppertunity to incorporate positional semantic information to the trees during the generation phase which simplifies the problem.

L-system tree generation methods are mainly focused at producing convincing visual representations of trees [**?**]. However, we do believe that these methods can be extended quite easily to enable the addition of structural semantics. This thesis does not pursue the introduction of such an extension. For our problem we mainly have to deal with structure of a forest which we will abstract to a set of nodes, representing trees, for the moment. Finding useful connections between nodes and incorporation of structual elements within these nodes is what we are ultimately interested in.

We propose a multilayer graphbased approach for the generation of forest structure. We will describe the structure of tree in this forest with a graph. The height (z coordinate) of a node is determined by the height property of the containing layer, while the position of a node in the plane defined by the layer represents the x, y coordinate of the branch.

Our forest generation method uses the following parameters:

1. Area A
2. Density $D(0.0 < D < 1.0)$
3. Layers $L(L > 0)$
4. Tree parameters defining min bounds T_min
5. Tree parameters defining max bounds T_max

The positions of the root nodes of the trees, which are contained by the base layer, are determined by a random scattering algorithm [8]. For each root node the method calculates a semi-random parameterset P_t ($T_{min} < P_t < T_{max}$). The second phase of the algorithm creates the upper $L - 1$ layers. For each layer l_i we determine $l_i.height$ by adding an interval height to $l_{i-1}.height$. This interval is determined by function **??**.

??$interval_i = l_i.height + 10$

Once all layers are in place we iterate through the root node list. For every node n_i contained by the base layer the method performs the folowing steps:

1. Step1:
2. Step2:
3. Step3:

4. Step4:

After construction of the multilayer graph representation of the forest, we feed it to the architecture planning algorithm that is discussed in section **??**.

Chapter 6

Architecture

6.1 Background

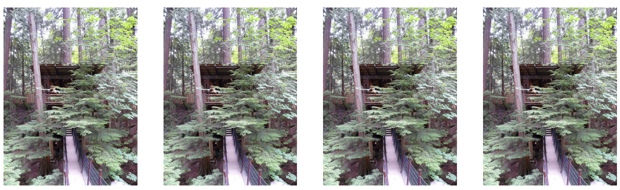
6.2 Elements for Tree-Based Architecture

In this section we present the set of elements we use for the construction of our treebased architecture. We will first introduce the elements in a broad sense, followed by descriptions of methods to construct geometry for each of the elements procedurally.

6.2.1 The Element Vocabulary

We established a set of architectural elements we believe are the most basic tree-based architectural structures:

1. platform
2. housing
3. bridge
4. stairs



(a) Platform (b) House (c) Bridge (d) Stairs

Figure 6.1: Caption of subfigures (a), (b) and (d)

6.2.2 Platforms

6.2.3 Housing

6.2.4 Bridges

Figure 6.2: A typical suspension bridge

The curved surface of a suspension bridge can easily be described with a spline.

6.2.5 Stairs

We define two types of stairs. The first is the simple vertical linear stairs-type. The second is the revolving staircase, which revolves around a tree trunk like a corkskrew.

6.2.6 Connecting Elements

We construct an architectural structure by combining the elements we have in our element vocabulary. In order to construct logical combinations we need to define for each elements how it can be joined with other elements in our vocabulary.

6.3 Scenarios

Here we define one or more scenarios for meaningful architectural planning within a forest scene. Each of the scenarios we propose here is based on a popular videogame genre:

1. Puzzle / Adventure
2. 3rd Person Strategy

3. 1st Person Shooter

6.3.1 Scenario 1: Puzzle / Adventure

6.3.2 Scenario 2: 3rd Person Strategy

6.3.3 Scenario 3: 1st Person Shooter

6.4 Formalizing the Scenarios

From the scenarios I discussed in the previous section we can extract a set of parameters and goals for our algorithms.

6.5 Structure Planning

Using the scenarios and architectural elements we have defined in the previous section we will formulate our planning algorithm now. I will first introduce the planning algorithm informally followed by a more formal approach.

6.5.1 Alternative approaches

6.6 Transforming Structure to Geometry

Chapter 7

User Interaction Tools

7.1 Growing Surfaces

Chapter 8

Results

Chapter 9

Conclusion

Bibliography

[1] F. Boudon, P. Prusinkiewicz, P. Federl, C. Godin, and R. Karwowski. Interactive design of bonsai tree models. *Computer Graphics Forum*, 22(3):591–599, 2003.

[2] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang. Interactive procedural street modeling. *ACM Trans. Graph.*, 27(3):1–10, 2008.

[3] O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 275–286, New York, NY, USA, 1998. ACM.

[4] M. Lipp, P. Wonka, and M. Wimmer. Interactive visual editing of grammars for procedural architecture. *ACM TRANSACTIONS ON GRAPHICS*, 27(3), AUG 2008.

[5] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. Procedural modeling of buildings. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 614–623, New York, NY, USA, 2006. ACM.

[6] R. Měch and P. Prusinkiewicz. Visual models of plants interacting with their environment. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 397–410, New York, NY, USA, 1996. ACM.

[7] A. L. Przemyslaw Prusinkiewicz. *The Algorithmic Beauty of Plants*. Springer Verlag, 1990.

[8] M. West. Random scattering: Creating realistic landscapes. Webpage, 2006.

[9] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky. Instant architecture. *ACM Trans. Graph.*, 22(3):669–677, 2003.

[10] P. M. Yoav I H Parish. Procedural modeling of cities. *ACM Trans. Graph.*, 2001.