

# Interactive Procedural Modeling: Architecture in Organic Scenes interactivity in procedural modeling tools

Ruud op den Kelder

May 1, 2009

## **Abstract**

Interactive Procedural Modeling: Architecture in Organic Scenes

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Related Work: Review of State of the Art in Procedural Modeling</b>	<b>6</b>
2.1	Procedural Modeling . . . . .	6
2.2	Map generation . . . . .	6
2.2.1	spatial growth based . . . . .	6
2.2.2	Interactivity . . . . .	6
2.3	Required Input . . . . .	7
2.4	The growth model map generator tool . . . . .	7
<b>3</b>	<b>Overview</b>	<b>7</b>
<b>4</b>	<b>Procedural Forest Generation</b>	<b>7</b>
4.1	Scattering Techniques for Tree Tositions . . . . .	7
4.2	Procedural Generation of Tree Geometry . . . . .	7
<b>5</b>	<b>Tree House Architecture</b>	<b>7</b>
5.1	Architectural Elements . . . . .	8
5.2	Semantic Rules . . . . .	8
<b>6</b>	<b>Method</b>	<b>8</b>
6.1	dynamic structures: polygonal surfaces . . . . .	8
6.2	Surfaces . . . . .	8
6.3	( . . . . .	8
6.4	( . . . . .	9
6.4.1	Water . . . . .	10
6.4.2	Afforestation . . . . .	10
6.4.3	Caves . . . . .	10
<b>7</b>	<b>User interaction tools</b>	<b>10</b>
7.1	Growing Surfaces . . . . .	10
7.2	Modifiers . . . . .	10

7.2.1	Spatial modifiers . . . . .	10
7.3	User controlled map generation . . . . .	10
7.4	automatic map generation . . . . .	10
<b>8</b>	<b>Implementation</b>	<b>10</b>
8.1	Tools used . . . . .	10
8.2	Application Class Structure . . . . .	11
8.3	User Interface . . . . .	11
8.4	Communication system between models . . . . .	11
<b>9</b>	<b>Results</b>	<b>11</b>
<b>10</b>	<b>Conclusion</b>	<b>11</b>

# 1 Introduction

**description:** *This is a general introduction to what the thesis is all about – it is not just a description of the contents of each section. Briefly summarize the question (you will be stating the question in detail later), some of the reasons why it is a worthwhile question, and perhaps give an overview of your main results. This is a birds-eye view of the answers to the main questions answered in the thesis.*

This thesis addresses the problem of procedural architecture in the special case of restrictions and possibilities enforced by the geometric properties of pre-existing objects, trees, in the scene. A side goal for this research was to develop methods for intuitive interactive control over the objects which were to be generated. Regarding user control and procedural techniques it is important to strike a balance which allows for a fast and intuitive building process, and meanwhile the user should have just the right amount of control to make the scene to its liking.

This thesis combines research from the fields of procedural architecture and procedural plant generation.

Visual modeling of plant development is a field which started in 1962, when Ulam applied cellular automata to simulate the development of branching patterns [?]. 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 [?]:

An L-system is a parallel rewriting system operating on branching 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 derivation steps. In each step, rewriting rules or productions replace all modules on the predecessor string by successor modules.

Przemyslaw [?] has employed and extended the l-system formalism for realistic visualisation of entire plant ecosystems. Within an 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 incorporate *communication modules*.

Since 2001 L-systems also proved to be useful in the field of urban procedural generation [?]. It was then that Muller proposed the use of L-systems for the generation of road networks and building generation.

hier wat over building, generation, city generatoin, texture generation. shape grammars

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 for the generation of treehouse architecture.

hier resultaat van treehouse generatie module

It is often the case that traditional general purpose modeling software is hard to master as a result of the freedom which is given to the user. Special purpose modeling tools such as cityEngine [?] (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 efficiency. 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 treehouse architecture that was developed for this thesis provide the user with intuitive interaction tools such that its easy to make alterations to the scene.

In the next section I will give an overview of this thesis followed by a discussion of related work.

**Old:** *The research which preceded this document strived to construct an intuitive organic modelling method for the creation of 2D maps. The central focus of this research has been on methods for construction of dynamic structures using real-time growth models. To enable a large degree of control over these dynamic structures the is a need for skeletons. A large part of this thesis is about constructing skeleton representations for dynamic structures. There obviously is a realtime demand in order to enable real-time interactivity, which means building such skeletons should above all be an efficient process.*

**description: 2. A brief review of treehouse architecture** *A brief section giving background information may be necessary, especially if your work spans two or more traditional fields. That means that your readers may not have any experience with some of the material needed to follow your thesis, so you need to give it to them. A different title than that given above is usually better; e.g., "A Brief Review of Frammis Algebra."*

## 2 Related Work: Review of State of the Art in Procedural Modeling

[?] [?] [?] [?] [?] [?]

**description:** *3. Review of the State of the Art Here you review the state of the art relevant to your thesis. Again, a different title is probably appropriate; e.g., "State of the Art in Zylon Algorithms." The idea is to present (critical analysis comes a little bit later) the major ideas in the state of the art right up to, but not including, your own personal brilliant ideas. You organize this section by idea, and not by author or by publication. For example if there have been three important main approaches to Zylon Algorithms to date, you might organize subsections around these three approaches, if necessary: 3.1 Iterative Approximation of Zylons 3.2 Statistical Weighting of Zylons 3.3 Graph-Theoretic Approaches to Zylon Manipulation*

procedural techniques: \* l-systems \* Extended l-systems \* Shape Grammar \*  
Scattering techniques

### 2.1 Procedural Modeling

**Summary:** *beschrijf kort geschiedenis van procedural modeling en werk naar map generation*

Procedural modeling

### 2.2 Map generation

**Summary:** *beschrijf huidige games en tools die gebruik maken van procedural generation voor geometrie van virtuele werelden.*

#### 2.2.1 spatial growth based

**Summary:** *verhaal over het feit dat spatial growth based-user controlled bouwen van maps relatief nieuw is.*

#### 2.2.2 Interactivity

**Summary:** *beschrijf tools voor het manipuleren van modellen die afhankelijk*

*zijn van l-systems en tools die worden gebruikt voor level design, zoals tools voor terrain editors.*

## 2.3 Required Input

**TODO:** *Belangrijk om snel vast te stellen hoeveel en welke input er nodig is.*

## 2.4 The growth model map generator tool

**TODO:** *discussieren dat gebruiker gelimiteerde controle heeft over het proces: nadelen en voordelen*

# 3 Overview

# 4 Procedural Forest Generation

## 4.1 Scattering Techniques for Tree Positions

## 4.2 Procedural Generation of Tree Geometry

# 5 Tree House Architecture

**description:** *Engineering theses tend to refer to a "problem" to be solved where other disciplines talk in terms of a "question" to be answered. In either case, this section has three main parts: 1. a concise statement of the question that your thesis tackles 2. justification, by direct reference to section 3, that your question is previously unanswered 3. discussion of why it is worthwhile to answer this question. Item 2 above is where you analyze the information which you presented in Section 3. For example, maybe your problem is to "develop a Zylon algorithm capable of handling very large scale problems in reasonable time" (you would further describe what you mean by "large scale" and "reasonable time" in the problem statement). Now in your analysis of the state of the art you would show how each class of current approaches fails (i.e. can handle only small problems, or takes too much time). In the last part of this section you would explain why having a large-scale fast Zylon algorithm is useful; e.g., by describing applications where it can be used. Since this is one of the sections that*

*the readers are definitely looking for, highlight it by using the word "problem" or "question" in the title: e.g. "Research Question" or "Problem Statement", or maybe something more specific such as "The Large-Scale Zylon Algorithm Problem."*

## 5.1 Architectural Elements

## 5.2 Semantic Rules

# 6 Method

## 6.1 dynamic structures: polygonal surfaces

**Summary:** *2D polygonale expansie van een ondergrond model als land of water die wordt gestuurd door eigenschappen van het type maar ook door ruimtelijke manipulatie methodes zoals vectorfields. representatie(onder voorbehoud):*

- *Cellular automata*
- *Cell systems*
- *iets wat niet met cellen werkt*

*realtime hertriangulatie zal moeten worden toegepast. polygonal meshes:*

- *representation for irregular 2D volumes.*
- *suitable for water, land, cave like structures*
- *real-time demand, so need for efficient algorithms*
- *houdt rekening met triangulatie*
- *houdt rekening met smoothing*
- *datastructure*

*citeer: On Vertex-Vertex Systems and Their Use in Geometric and Biological Modelling*

**TODO:** *dit kan al snel worden geïmplementeerd dus logisch om dit als eerste te beschrijven en methode uit te werken.*



## 6.2 Surfaces

### 6.3 (

Forest Generation Scattering techniques)

### 6.4 (

Forest Generation Scattering techniques)

**Summary:** *Per type spatial growth model bespreken wat de context is binnen outdoor areas en dus met welke andere spatial growth modellen het communiceert en welke effecten het spatial growth model heeft op de andere en vice versa. Ook bespreken op wat voor manier een vector field (en andere tools die invloed hebben op de ruimtelijke indeling) effect heeft op de uitdijning van dit type model.*

A definition of land: "The part of Earth which is not covered by oceans or other bodies of water".(bron: <http://en.wiktionary.org/wiki/land> ...jaja ik vind nog wel een betere bron) In reality land obviously does not grow, so first we must ask ourselves whether growing land in our virtual world can be made useful and intuitive to use. There are no real rules for the bounding shape of a piece of land, the bounds can be configured in any way. However due to this fact almost any random configuration of the shape of land looks realistic. The question is, as with other growth models with a minimum amount of rules, whether generating land by means of a growth model, thereby having less control over the result compared to conventional methods, is something which we want.

When you want maximum control over the resulting geometry of a piece of land, using a growth model is not a very attractive method. However there are some situations in which this method is quite helpful.

**Summary:** *wanneer is het groeien van land handig:*

- *Obviously: In situaties waarin de gebruiker niet veel waarde hecht aan de precieze vorm van de resulterende geometrie.*
- *stimuleert de creativiteit: elaborate.*

Land provides the underlying surface of many of the growth models discussed in this paper.

**Summary:** *dus het bestaan van land op de seedpositie van een afhankelijk groeiproces is een eis. Wellicht moet het genereren van een map bestaan uit verschillende fases:*

- *fase 1: generatie van land en water.*
- *fase 2: plaatsen en activeren van groei processen die afhankelijk zijn van land.*

The field of biological modeling provides some interesting modeling techniques for the process of cell division. Cell division is interesting with respect to the concept of growing land because it is an expanding process with some nice properties. **-Insert:** *benoem properties of revise deze zin.*

#### 6.4.1 Water

#### 6.4.2 Afforestation

#### 6.4.3 Caves

## 7 User interaction tools

### 7.1 Growing Surfaces

### 7.2 Modifiers

#### 7.2.1 Spatial modifiers

#### Summary:

- *vector fields: dit moet je wellicht helemaal aan het begin doen omdat alle spatial growth models gemanipuleerd worden door deze vector fields.*
- *obstruction with solid objects*
- *smudge-like tool*

### 7.3 User controlled map generation

### 7.4 automatic map generation

## 8 Implementation

### 8.1 Tools used

#### Summary:

- *programming language: c++ (for now)*
- *visualization library: OpenGL or Ogre3D (if it features easy vertex manipulation)*
- *physics engine: ODE (onder voorbehoud)*
- *collision detection: D-Collide (onder voorbehoud)*

## 8.2 Application Class Structure

## 8.3 User Interface

## 8.4 Communication system between models

# 9 Results

# 10 Conclusion

## References

- [1] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang. Interactive procedural street modeling. *ACM Trans. Graph.*, 27(3):1–10, 2008.
- [2] O. Deussen, P. Hanrahan, B. Lintermann, R. Mëch, 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.
- [3] H. M. George Kelly. A survey of procedural techniques for city generation. *ITB Journal*.
- [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. Mëch 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] C. Smith. *On Vertex-Vertex Systems and Their Use in Geometric and Biological Modelling*. PhD thesis, University of Calgary, 2006.
- [8] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky. Instant architecture. *ACM Trans. Graph.*, 22(3):669–677, 2003.