# Using procedural methods to generate realistic virtual rural worlds

University of Cape Town • iDyunivesithi yaseKapa • Universiteit van Kaapstad

*Minor Dissertation presented in partial fulfilment of the requirements for the degree of Master of Science in Computer Science*

by

**Harry Long**

Supervised by:

James Gain and Marie-Paul Cani

February 2016

*I know the meaning of plagiarism and declare that all of the work in this document, save for that which is properly acknowledged, is my own.*

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Creating detailed virtual worlds can be a tedious task for artists. Indeed, modelling terrain, vegetation, water streams, rivers, water reserves, soil, rocks, buildings and road networks for large virtual worlds "by hand" can be extremely repetitive and tiresome. This is especially true when realism is a key requirement. The increase in size and complexity of these virtual worlds mirror that of the processing capabilities of computing hardware. As consequence, the task is only getting worse.

A popular technique to overcome the burden of repetitive tasks is to have them automated. In computer graphics, this involves generating algorithms which, given a set of input parameters, generate the required content automatically. This is called *procedural content generation* and has already been successfully applied in different areas of computer graphics including: the generation of non-repetitive textures [2, 39, 67], modelling plants [7, 20, 26, 37], generating terrains [60, 22, 16], generating river networks [11, 18] and generating city landscapes [23, 33] (figure 1)
A common difficulty with these methods, however, is finding the appropriate input parameters for the procedural algorithms. The correlation between the parameters and the resulting content is often unintuitive and, as a consequence, often comes down to iterative trial-and-errors until a "close enough" result is found. To overcome this, interactive techniques are often used in an attempt to make generating the input parameters more intuitive. These range from simple paint tools such as lassos and brushes [18] to sketch-based recognition algorithms [22].

The intent of this thesis is to develop procedural algorithms to automate the generation of virtual rural worlds. The input parameters for the procedural algorithms must be interactive and/or self-explanatory.

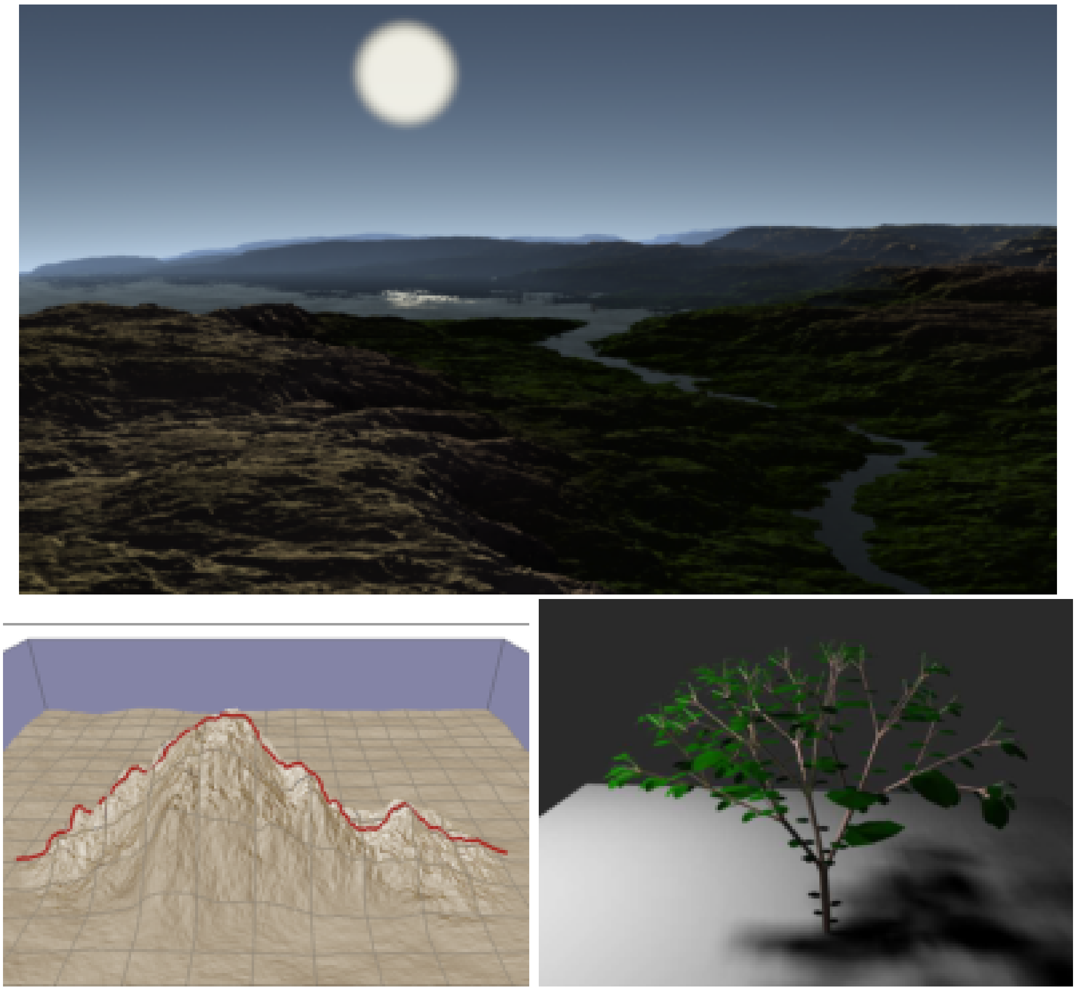

Figure 1.1: Example of procedurally generated content. From top to bottom, left to right: Procedurally generated river stream [11], procedurally generated terrain through sketching [22], procedurally generated plant [63]

## 1.1 Research Goals

The research goals for this project are as follows:

- Develop procedural methods to automate the generation of realistic virtual rural worlds.
- Provide intuitive and smart controls.
- When possible, make interactions real-time.

One of the most important aspect of rural landscapes is vegetation. As such, our *first goal* must strongly focus on the insertion of plants. The automation provided should not limit user control and the flexibility of the system. For example, it must be possible to generate worlds with varying elevations, river networks, water sources and vegetation.

For the *second goal*, lots of thought must be put into making all user oriented controls intuitive. To do so, it will be important to research the pros and cons of other graphical applications in terms of control. If need be, multiple prototype controls should be developed in an attempt to find the best suited.

Maintaining a continuous feedback loop between user action and corresponding reaction is extremely important for both user-friendliness and to optimize usage. In an attempt to meet our *third goal* therefore, efficient algorithms must be developed in order to keep there time complexity to a minimum. When suited, these algorithms should be developed to run on the GPU.

## 1.2 Contributions

State contributions of this thesis

## 1.3 Structure

Outline structure of the thesis

# Chapter 2

# Background

This chapter gives an overview of previous work related to our topic. Procedural methods applied to computer graphics is a wide area of research with an exhaustive number of publications. As a consequence, we cannot pretend to review all this work. Instead, we will focus on reviewing work which is closely linked to generating virtual *rural* worlds.

We first present research which deals with the procedural generation of terrains. This will be followed by a review of methods to generate water flows on terrains. To conclude, an overview of techniques to generate vegetation will be presented.

## 2.1 Terrains

Terrain.

## 2.2 Rivers & Streams

Rivers and Streams

## 2.3 Vegetation

Vegetation is core to rural landscapes. The species present along with their associated densities create a relationship between ecotopes and areas on earth on which resources are adequate. To ensure realism in virtual rural worlds, much emphasize must be put on efficiently modelling these underlying ecosystems.

This section will review different methods to generate suitable vegetation for virtual worlds. These methods will be split into three main categories: *Explicit instancing, probabilistic instancing* and *ecosystem simulators*.
*Explicit instancing* use user input to either directly or indirectly pinpoint exact locations for individual plant instances.
*probabilistic instancing* methods use statistical models to generate suitable vegetation.
*ecosystem simulators* attempt to reproduce plants battling for available resources algorithmically.

### 2.3.1 Explicit Instancing

Explicit instancing methods require input from the user to explicitly outline the location of individual plant instances.
Arnaud et al. [18] permit users to insert individual plants manually by simply clicking the appropriate location on the terrain. To overcome the tedious task of manually placing individual plant instances on large terrains, the system is able to analyse existing distributions for reproduction. For example, to generate a large forest, the user is only required to generate a small subsection which can then be used to reproduce it on any scale (figure 2.3.1)

Similarly, Deussen et al. [15] allow users to use grayscale raster images as input to specify terrain vegetation. The location of individual plants is determined by pixel location whereas plant properties are correlated to pixel intensity.

During their work focused on improving the realism of roadside landscapes, C. Andujar et al. [4] use orthophotos as input to determine the location and properties of individual plants. Unlike ordinary aerial photographs, aerial orthophotos use normalisation techniques to take into account terrain relief and camera tilt to produce the image. The result is an image with uniform scale throughout, which, similarly to a map, can be used to accurately measure distances between points. Here, the orthophotos are analysed to determine the center point of individual plants.

Explicit instancing methods provide extensive user control and freedom for the resulting virtual world. However, although some automation is provided, it is often limited. Therefore, albeit simplified, generating large virtual worlds can still be a lengthy process.

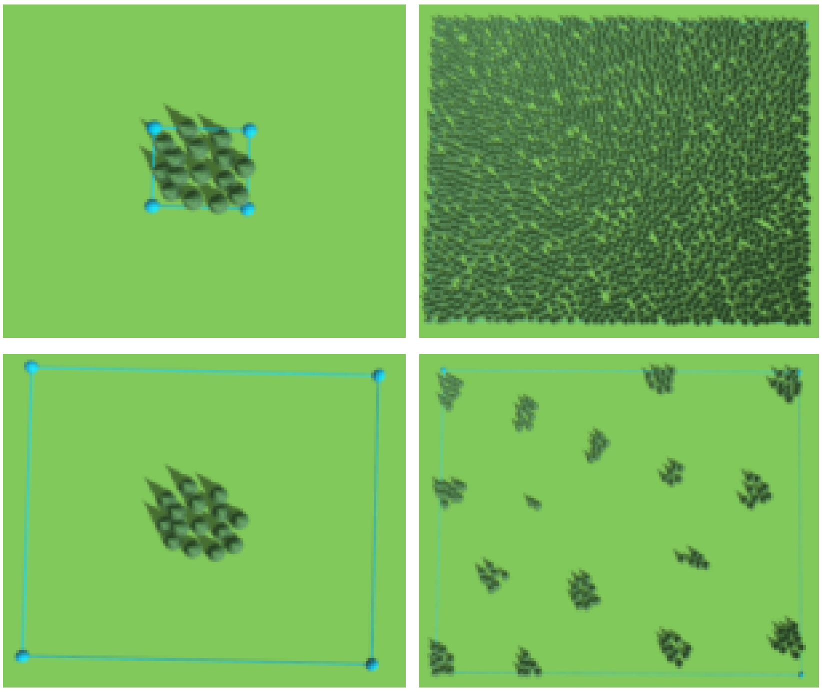

Figure 2.1: Using explicit instancing as input examplars for reproduction [18]

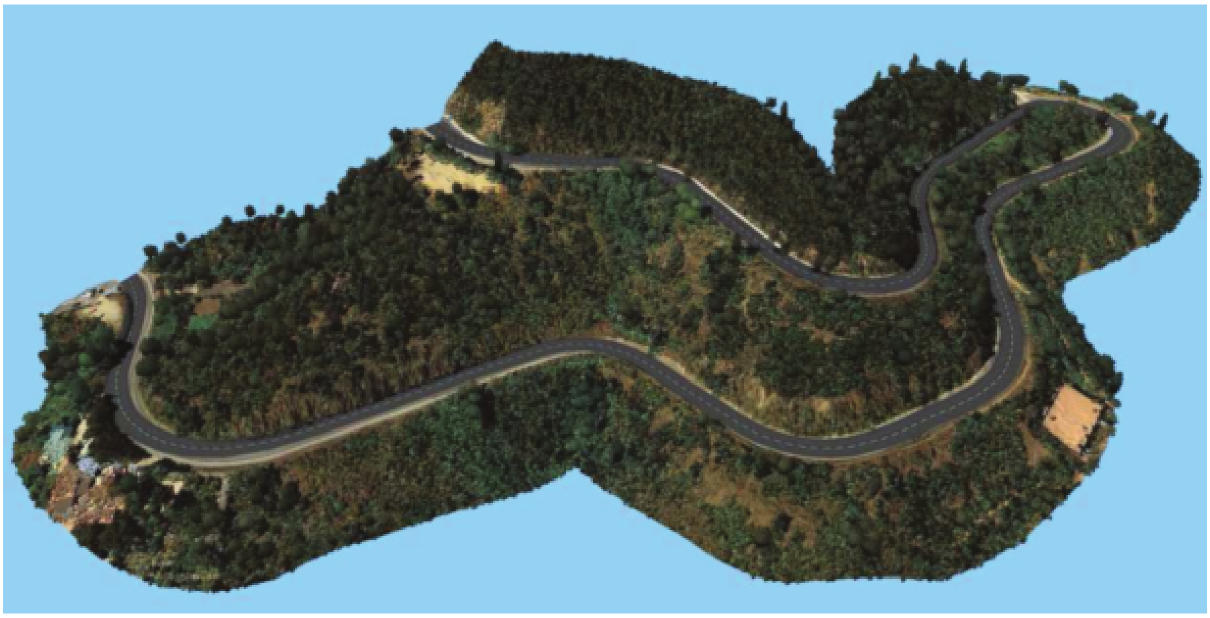

Figure 2.2: Reconstructed roadside vegetation using orthophotos [4]

### 2.3.2 Probabilistic Instancing

Probabilistic instancing methods use statistical models in an attempt to produce adequate vegetation. These methods can be further split into two sub-categories which are discussed in further detail below: *Radial distribution analysis* and *Predefined ecotopes*.

**RADIAL DISTRIBUTION ANALYSIS**

Work by Emilien et al. [18], Boudon et al. [6] and Lane et al. [35] use radial distribution analysis to grasp the underlying layout of given input examplars. The data generated by the analysis stage can later be used to synthesise, at any scale, new point distributions which respect the characteristics of the input exemplar [46].
For example, by analysing the positions of individual plants in a small subset of a forest and using it as the input exemplar, it is possible to reproduce it at a much larger scale in order to model the full-size forest.

**Analysis** Generating the analytical data involves measuring the distances between individual points of different categories from the input examplar. For plant distribution analysis, the points represent individual plant instances and the categories represent the different species.

Before performing the analysis, the following parameters need to be configured:

- $\mathbf{R_{min}}$: The minimum distance from which point distances need to be analysed.

- $\mathbf{R_{max}}$: The maximum distance after which point distances don't need to be analysed.

- **Bin size**: When analysing the distances of given points, it is necessary to aggregate the points which reside at similar distances into bins. The bin size is the distance represented by a single bin.

A core part of radial distribution analysis is generating pair correlation histograms for each category pair combination. A pair correlation histogram $H_{AB}$ represents the variation in the distance between points of of category $C_A$ and $C_B$ ranging from $R_{min}$ to $R_{max}$ in *bin size* increments (figure 2.3.2)

To generate the pair correlation histogram $H_{AB}$, the algorithm iterates through each reference point of category $C_A$ and for each destination point of category $C_B$ at a distance between $R_{min}$ and $R_{max}$ increments the relevant bin in the histogram.
In figure 2.3.2, for example, are being measured the points that lie within the annular shell of radius $r$ with bin size $d_r$ (area $d_A$).
The coverage area of annular shells are larger for bins being analysed at further distances. In other words, $A_r$ ¡ $A_{r+1}$ where $A_r$ is the area covered by the annular shell starting at distance $r$. To counter for this, and the fact that there will naturally be more points in these larger annular shells, normalisation is performed.

The radial distribution analysis function $h_{rdf}$ is as follows:

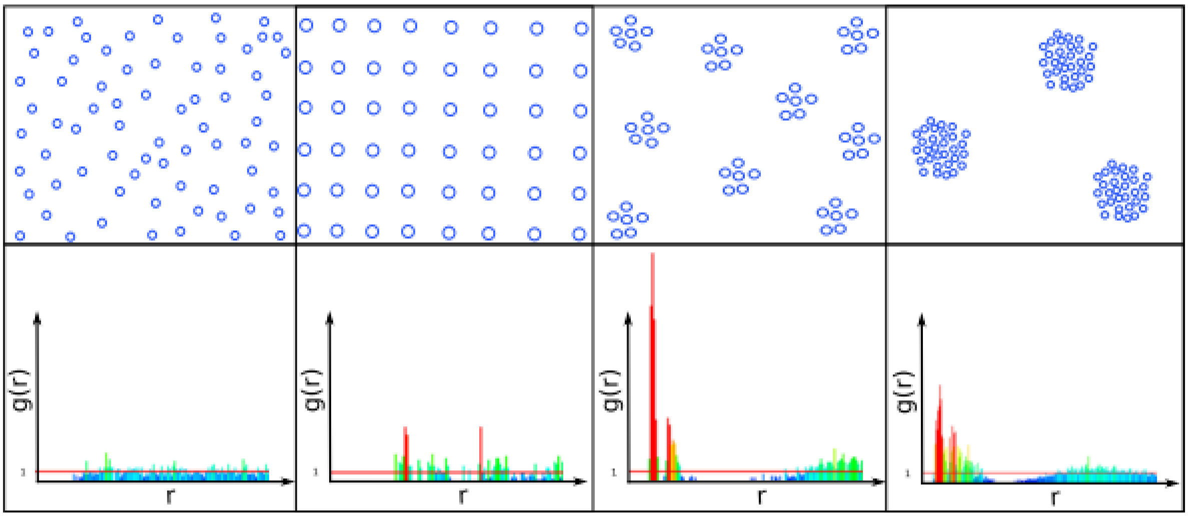


Figure 2.3: Point distributions with associated pair correlation histogram [17]

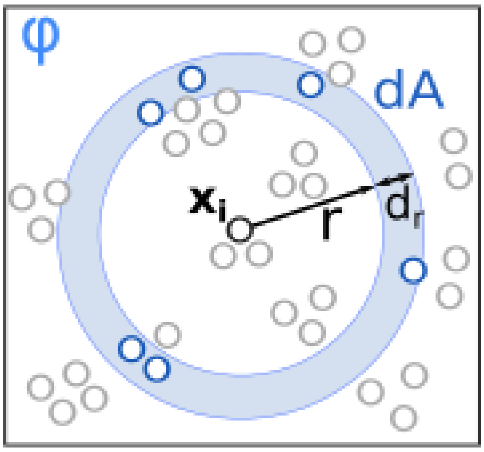


Figure 2.4: Radial distribution analysis]

$$h_{rdf}(k) = \sum_{x_i \in X} \sum_{y_j \in Y \& k d_r \leq d(x_i, y_j) < (k+1) d_r} \frac{A}{d_A n_x n_y}$$

Where:

- *hrdf(k)* is the k-th value of the pair wise histogram.
- $X$ are the reference points.
- $Y$ are the target points.
- $d_r$ is the annular shell width.
- $A$ is the total analysed area.
- $n_x$ and $n_y$ are the number of points of categories $x$ and $y$ respectively.

- $d_A$ is the area of the annular shell being analysed.

Conceptually, this formula calculates the variance from the average density of the target category at incremental distances from points of the reference category.

**Reproduction** In order to reproduce the distribution of the input exemplar, points must be added iteratively whilst matching as closely as possible the categorical point separation data calculated during the analysis stage. Arnaud et al. [18] use Metropolis-Hastings sampling [28]. This technique involves performing a fixed number of birth-and-death perturbations. A change from the initial arrangement $X$ to the new arrangement *X'* is accepted with probability $R$, where:

$$R = \frac{f(X')}{f(X}$$

*f(X)* is the probability density function (PDF) and is expressed as:

$$f(X) = \prod_{C_{Y_K} \leq C_X} \prod_{x_i \in X} \prod_{y_i \in Y_k} h_{X,Y_k}(d(x_i, y_j))$$

Where:

- $C_y$ and $C_x$ represent categories $Y$ and $X$ respectively
- $X$ are all points of category $X$
- $Y$ are all points of category $Y$
- $\mathrm{h_{X,Y_k}(d(x_i,\ y_j))}$ is the value retrieved from the pairwise histogram of categories $X$ and $Y$ given the distance between points $x_i$ and $y_i$.

Intuitively, the PDF defines, given a set of points, the aggregate strength of the current distribution.

The main advantage of the probabilistic approach is computational efficiency. In the work by Arnaud et al. [18], both analysis and reproduction is performed in near real-time. Another advantage is that specific specie properties are not needed.
The primary disadvantage of this approach is scalability. Given an input exemplar, the system will only be able to reproduce the given plant distribution characteristics. If a new plant is to be added to the forest, for example, a new input exemplar must be created.

**Predefined ecotopes**

# Bibliography


[1] No Title. 8:55–62.

[2] a.a. Efros and T K Leung. Texture synthesis by non-parametric sampling. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2:1033 – 1038, 1999.

[3] Ratish Agarwal and Dr. Mahesh Motwani. Survey of clustering algorithms for MANET. 16(3):645–678, 2009. URL `http://arxiv.org/abs/0912.2303`.

[4] C. Andújar, A. Chica, M. a. Vico, S. Moya, and P. Brunet. Inexpensive Reconstruction and Rendering of Realistic Roadside Landscapes. *Computer Graphics Forum*, 33(6):101–117, February 2014. ISSN 01677055. doi: 10.1111/cgf.12281. URL `http://doi.wiley.com/10.1111/cgf.12281`.

[5] C. Andújar, a. Chica, M. a. Vico, S. Moya, and P. Brunet. Inexpensive Reconstruction and Rendering of Realistic Roadside Landscapes. *Computer Graphics Forum*, 00(0):1–18, 2014. ISSN 01677055. doi: 10.1111/cgf.12281. URL `http://doi.wiley.com/10.1111/cgf.12281`.

[6] F Boudon and G Le Moguédec. Déformation asymétrique de houppiers pour la génération de représentations paysageres réalistes. *Revue Electronique Francophone d'Informatique*, 1(1):9–19, 2007. URL `http://www.irit.fr/REFIG/index.php/refig/article/viewArticle/11`.

[7] Frédéric Boudon, Christophe Pradal, Thomas Cokelaer, Przemyslaw Prusinkiewicz, and Christophe Godin. L-py: an L-system simulation framework for modeling plant architecture development based on a dynamic language. *Frontiers in plant science*, 3(May):76, January 2012. ISSN 1664-462X. doi: 10.3389/fpls.2012.00076. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3362793&tool=pmcentrez&`

[8] Doug A Bowman, David Koller, and Larry F Hodges. Travel in Immersive Virtual Environments: An Evaluation Motion Control Techniques of Viewpoint. pages 45–52, 1997.

[9] Jonathan M Cohen, John F Hughes, and Robert C Zeleznik. Harold: A World Made of Drawings. *NPAR 2000*, pages 83–90, 2000.

[10] Rudy P Darken, John L Sibert, and Computer Science. A Toolset for Navigation in Virtual Environments. pages 157–165, 1993.

[11] E. Derzapf, Björn Ganster, M. Guthe, and Reinhard Klein. River Networks for Instant Procedural Planets. *Computer Graphics Forum*, 30(7):2031–2040, 2011. ISSN 01677055. doi: 10.1111/j.1467-8659.2011.02052.x.

[12] Brett Desbenoit, Eric Galin, and Samir Akkouche. Simulating and modeling lichen growth. 23(3), 2004.

[13] Brett Desbenoit, Eric Galin, and Samir Akkouche. Modeling Autumn Sceneries. 2006.

[14] Oliver Deussen and Carsten Colditz. Interactive visualization of complex plant ecosystems. *Proceedings of the conference on Visualization '02 (VIS '02)*, pages 219–226, 2002. URL `http://dl.acm.org/citation.cfm?id=602133`.

[15] Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomir Měch, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic Modeling and Rendering of Plant Ecosystems. *Conference on Computer Graphics and Interactive Techniques*, pages 275—-286, 1998. URL `http://portal.acm.org/citation.cfm?id=280814.280898`.

[16] Jonathon Doran and Ian Parberry. Controlled Procedural Terrain Generation Using Software Agents. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(2):111–119, June 2010. ISSN 1943-068X. doi: 10.1109/TCIAIG.2010.2049020. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5454273`.

[17] Arnaud Emilien. Création interactive de monde virtuels. 2014.

[18] Arnaud Emilien and Marie-Paule Cani. WorldBrush: Interactive Example-based Synthesis of Procedural Virtual Worlds. *Siggraph '15, TO BE PUBLISHED*, 2015.

[19] Plant Models Faithful. ˜l˜ Computer Graphics, Volume 22, Number 4, August 1988. 22(4):151–158, 1988.

[20] Thierry Fourcaud, Xiaopeng Zhang, Alexia Stokes, Hans Lambers, and Christian Körner. Plant growth modelling and applications: the increasing importance of plant architecture in growth models. *Annals of botany*, 101 (8):1053–63, May 2008. ISSN 1095-8290. doi: 10.1093/aob/mcn050. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2710283&tool=pmcentrez&`

[21] Sven Fuhrmann, Alan M Maceachren, and Westfaelische Wilhelms-universitaet. Testing on Usability: Navigation in Desktop GeoVirtual Environments. 2000.

[22] James Gain, Patrick Marais, and Wolfgang Straß er. Terrain sketching. *Proceedings of the 2009 symposium on Interactive 3D graphics and games - I3D '09*, 1(212):1–8, 2009. URL `http://dl.acm.org/citation.cfm?id=1507149.1507155`.

[23] James Gain, Patrick Marais, and Rudolph Neeser. City Sketching. *WSCG 2014*, pages 1–10, 2014.

[24] E Galin. Real-time Rendering of Realistic-looking Grass. 2005.

[25] Eric Galin, Adrien Peytavie, Eric Guérin, and Bedich Beneš. Authoring Hierarchical Road Networks. *Computer Graphics Forum*, 30(7):2021–2030, September 2011. ISSN 01677055. doi: 10.1111/j.1467-8659.2011.02055.x. URL http://doi.wiley.com/10.1111/j.1467-8659.2011.02055.x.

[26] Y. Guo, T. Fourcaud, M. Jaeger, X. Zhang, and B. Li. Plant growth and architectural modelling and its applications. *Annals of Botany*, 107(5): 723–727, April 2011. ISSN 0305-7364. doi: 10.1093/aob/mcr073. URL http://aob.oxfordjournals.org/cgi/doi/10.1093/aob/mcr073.

[27] Johan Hammes. Modeling of Ecosystems as a Data Source for Real-Time Terrain Rendering. *Framework*, pages 98–111, 2001. doi: 10.1007/3-540-44818-7_14.

[28] T Hurtut, PE Landes, and J Thollot. Appearance-guided synthesis of element arrangements by example. *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, pages 51–60, 2009. URL http://dl.acm.org/citation.cfm?id=1572623.

[29] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: A Sketching Interface for 3D Freeform Design. *SIGGRAPH*, pages 1–8, 2004.

[30] M Jaeger and P H D E Reffye. Basic concepts of computer simulation of plant growth. 17(3):275–291, 1992.

[31] Marc Jaeger. Philippe de Reffye (Cirad) & Marc Jaeger (Cirad).

[32] N M Kapolka and D J Dollhopf. Effect of slope gradient and plant growth on soil loss on reconstructed steep slopes. *International Journal of Surface Mining*, 15(2):86–89, 2001. ISSN 13895265. doi: 10.1076/ijsm.15.2.86.3416. URL http://www.szp.swets.nl/szp/journals/sm152086.htm.

[33] George Kelly and Hugh McCabe. Citygen: An interactive system for procedural city generation. *Fifth International Conference on Game Design and Technology*, pages 8–16, 2007. URL http://www.citygen.net/files/citygen_gdtw07.pdf.

[34] Brendan Lane and Przemyslaw Prusinkiewicz. Generating Spatial Distributions for Multilevel Models of Plant Communities.

[35] Brendan Lane and Przemyslaw Prusinkiewicz. Generating Spatial Distributions for Multilevel Models of Plant Communities. *Interface*, 2002:69–80, 2002. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.98.9523.

[36] Eddie Lau. Visually appealing water flow over a terrain. 2010.

[37] Philip Lewis. Three-dimensional plant modelling for remote sensing simulation studies using the Botanical Plant Modelling System. 1999.

[38] Philip Lewis. Three-dimensional plant modelling for remote sensings studies using Botanical Plant Modeling System, 1999.

[39] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 20(3):127–150, July 2001. ISSN 07300301. doi: 10.1145/501786.501787. URL `http://portal.acm.org/citation.cfm?doid=501786.501787`.

[40] Marcelo M Maes, Tadahiro Fujimoto, and Norishige Chiba. Efficient animation of water flow on irregular terrains. *GRAPHITE - International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, 1(212):107–115, 2006. doi: 10.1145/1174429.1174447. URL `http://portal.acm.org/citation.cfm?doid=1174429.1174447`.

[41] N. Maréchal, E. Guérin, E. Galin, S. Mérillou, and N. Mérillou. Heat Transfer Simulation for Modeling Realistic Winter Sceneries. *Computer Graphics Forum*, 29(2):449–458, May 2010. ISSN 01677055. doi: 10.1111/j.1467-8659.2009.01614.x. URL `http://doi.wiley.com/10.1111/j.1467-8659.2009.01614.x`.

[42] Xing Mei and Philippe Decaudin. Fast Hydraulic Erosion Simulation and Visualization on GPU. *Pacific Graphics*, 2007.

[43] Andreas Muhar. Three-dimensional modelling and visualisation of vegetation for landscape simulation. *Landscape and Urban Planning*, 54(1-4):5–17, 2001. ISSN 01692046. doi: 10.1016/S0169-2046(01)00122-0.

[44] Karl J Niklas. Maximum plant height and the biophysical factors that limit it. pages 433–440, 2007.

[45] Communications Of and T H E Acm. July 2000/Vol. 43, No. 7 COMMUNICATIONS OF THE ACM. 43(7), 2000.

[46] AC Öztireli and Markus Gross. Analysis and synthesis of point distributions based on pair correlation. *ACM Transactions on Graphics (TOG)*, 31(6):170:1–170:10, 2012. URL `http://dl.acm.org/citation.cfm?id=2366189`.

[47] C. E. Timothy Paine, Toby R. Marthews, Deborah R. Vogt, Drew Purves, Mark Rees, Andy Hector, and Lindsay a. Turnbull. How to fit nonlinear plant growth models and calculate growth rates: an update for ecologists. *Methods in Ecology and Evolution*, 3(2):245–256, April 2012. ISSN 2041210X. doi: 10.1111/j.2041-210X.2011.00155.x. URL `http://doi.wiley.com/10.1111/j.2041-210X.2011.00155.x`.

[48] Randy Pausch, Tommy Burnette, Dan Brockway, and Michael E Weiblen. Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures. pages 399–400, 1995.

[49] C. Pradal, F. Boudon, C. Nouguier, J. Chopard, and C. Godin. PlantGL: A Python-based geometric library for 3D plant modelling at different scales. *Graphical Models*, 71(1):1–21, January 2009. ISSN 15240703. doi: 10.1016/j.gmod.2008.10.001. URL `http://linkinghub.elsevier.com/retrieve/pii/S1524070308000143`.

[50] Christophe Pradal, Samuel Dufour-kowalski, Christian Fournier, and Christophe Godin. OpenAlea: a visual programming and component-based software platform for plant modelling. pages 751–760, 2008.

[51] ALP Prusinkiewicz and A Lindenmayer. *The Algorithmic Beauty of Plants.* 1990. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.353.4321`.

[52] Przemyslaw Prusinkiewicz. Visual Models of Plants Interacting with Their Environment. *SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1:397–410, 1996.

[53] Przemyslaw Prusinkiewicz. Modeling of spatial structure and development of plants. *Scientia Horticulturae*, 74(1-2):113–149, 1998.

[54] Przemyslaw Prusinkiewicz, Mark Hammel, and Eric Mjolsness. Animation of Plant Development. *SIGGRAPH '93 Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 93:351–360, 1993.

[55] Stefan Roettger and Computer Graphics Group. Ndvi-based vegetation rendering.

[56] Karan Singh and James Mccrae. Sketch-Based Path Design. *Proceedings of the Graphics Interface Canadian Inform. Process. Soc.*, pages 95–102, 2008.

[57] R M Smelik, T Tutenel, K J De Kraker, and R Bidarra. Interactive Creation of Virtual Worlds Using Procedural Sketching. *Proceedings of Eurographics 2010: Short Papers. Eurographics Association*, pages 1–4, 2010.

[58] R. M. Smelik, T. Tutenel, K. J. De Kraker, and R. Bidarra. A declarative approach to procedural modeling of virtual worlds. *Computers and Graphics (Pergamon)*, 35(2):352–363, April 2011. ISSN 00978493. doi: 10.1016/j.cag.2010.11.011. URL `http://linkinghub.elsevier.com/retrieve/pii/S0097849310001809`.

[59] Ruben Smelik and Tim Tutenel. Integrating procedural generation and manual editing of virtual worlds. *Proceedings of the 2010 ...*, 2:1–8, 2010. URL `http://dl.acm.org/citation.cfm?id=1814258`.

[60] Ruben M Smelik, Klaas Jan De Kraker, Saskia A Groenewegen, The Hague, Tim Tutenel, and Rafael Bidarra. A Survey of Procedural Methods for Terrain Modelling . 2009.

[61] Ruben M. Smelik, Tim Tutenel, Klaas Jan de Kraker, and Rafael Bidarra. Declarative Terrain Modeling for Military Training Games. *International Journal of Computer Games Technology*, 2010:1–11, 2010. ISSN 1687-7047. doi: 10.1155/2010/360458. URL `http://www.hindawi.com/journals/ijcgt/2010/360458/`.

[62] M Smith, R Allen, and L Pereira. Revised FAO methodology for crop water requirements. pages 51–58.

[63] Cyril Soler, FX Sillion, F Blaise, and Philippe De Reffye. A physiological plant growth simulation engine based on accurate radiant energy transfer. Technical report, 2001. URL `https://hal.inria.fr/inria-00072514/`.

[64] Dietrich Stoyan and Helga Stoyan. Non-Homogeneous Gibbs Process Models for Forestry A Case Study. 40:521–532, 1998.

[65] Wei Sun, Junhui Wang, and Yixin Fang. Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics*, 6 (April 2011):148–167, 2012. ISSN 19357524. doi: 10.1214/12-EJS668.

[66] Kenneth Vanhoey, Basile Sauvage, Frédéric Larue, and Jean-Michel Dischler. On-the-fly multi-scale infinite texturing from example. *ACM Transactions on Graphics*, 32(6):1–10, November 2013. ISSN 07300301. doi: 10.1145/2508363.2508383. URL `http://dl.acm.org/citation.cfm?doid=2508363.2508383`.

[67] Li-yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. State of the Art in Example-based Texture Synthesis. *In proceedings of Eurographics 09*, (2):1–25, 2009.

[68] H.-P. Yan. A Dynamic, Architectural Plant Model Simulating Resource-dependent Growth. *Annals of Botany*, 93(5):591–602, March 2004. ISSN 0305-7364. doi: 10.1093/aob/mch078. URL `http://aob.oupjournals.org/cgi/doi/10.1093/aob/mch078`.

[69] Robert C Zeleznik, Kenneth P Herndon, John F Hughes, and Scientific Visualization. SKETCH: An Interface for Sketching 3D Scenes. 1910.

[70] Guo Xin Zhang, Song Pei Du, Yu Kun Lai, Tianyun Ni, and Shi Min Hu. Sketch guided solid texturing. *Graphical Models*, 73(3):59–73, May 2011. ISSN 15240703. doi: 10.1016/j.gmod.2010.10.006. URL `http://linkinghub.elsevier.com/retrieve/pii/S1524070310000226`.