

Modeling of RL - Cities

Aant van der Zee¹, Bauke de Vries²

Eindhoven University of Technology, the Netherlands

¹a.v.d.zee@tue.nl, ²b.d.vries@tue.nl

Abstract. *In this paper we present an outline of a newly started project to develop a city generator for use in urban planning. The aim of the project is to develop a rule-based system which is capable of generation lookalike cities. Lookalike cities are cities which resemble real life cities without being an exact copy of it. A city consists of several zones; each zone has its own identity. In order to generate lookalike cities, these zone-identities need to be captured into rules which the system can 'read'.*

Keywords. *Procedural modeling; urban development; L-systems; architecture; city generator.*

INTRODUCTION

With the rise of the gaming industry there was a demand for realistic or imaginary city models that accommodate game adventures. As a consequence there was a need for artists who 'build' in-world cities. With the increase of the speed of the personal computers, the game environments became larger and larger and so the demand for these artists. The gaming industry had to overcome the difficulty to hire more and more artists to design these large-scale cities. The answer was to develop methods, which use no art assets like: (building-) models and textures. Research was done to develop procedurally generated cityscapes (Parish 2001, Muller 2006). Most research has been done in the field of games (see figure 1). or ancient cities (see figure 2). Procedural generated buildings (Wonka 2003), temples and ancient cities like Rome and Pompeii (Haegler 2009) are used as an urban visualization tool.

Surprisingly these city generators also found their way also in the urban planning (Schirmer, 2011). There is already a commercial software package which generates cities.

It is our opinion that these tools lack some features which makes them less useful in the domain

of urban planning. The generated cities and buildings are 'abstract'; they don't resemble real life cities and buildings. In order to increase acceptance in the building industry, especially urban planning, research needs to focus on real life cities and buildings and trying to mimic their identity (see figure 3).

In the next paragraph's we will give an outline of our recently started research project. In this project we aim to develop a city generator.

RESEARCH QUESTIONS

In urban design we anticipate the following application areas of computer generated cities:

- Visual quality check
- Infill of sites in the city
- Test case for zero energy towns
- Virtual city model
- Military

The above list is not conclusive; there will be more, not foreseen, areas where generated cities can be of use.

Our research focuses on generating lookalike cities. Lookalike cities are generated city which resembles existing cities without being an exact copy.

In other words lookalike cities are cities with the same identity but different buildings en infrastructure.

We discriminate three levels of abstraction, namely: the city as a whole, the infrastructure and the buildings. We recognize that a city consists of several zones; each zone has it own identity. This identity of a zone depends mainly of:

- Type of streets (main road, secondary road, linear, curved etc)
- Street profile (cross section)
- Width of the streets
- Are pavements alongside the street
- Do the buildings have front gardens
- Green places (parks)
- Type of building (dwellings, shops etc)
- The architecture of those buildings

In order to generate lookalike cities we have to capture the identity of a city into a finite number of rules. The above mentioned 'items' need to become input for our system. The system must be capable of 'rewriting' those rules to come up with a generated-city which resembles the real life city but without being an exact copy of it. We come to following research questions:

- Is it possible to "capture" a city identity into rules?
- Is it possible generating a city which mimics an existing city using above mentioned rules?

We are starting our research to see if it is possible to generate a city based on some rules. First we develop an engine which can generate different city layouts (infrastructure network). The end result must be a city layout which is similar to the original city; it must mimic the target city without becoming an exact duplicate of it. The original city will become a template for the system to generate a lookalike city.

In the next subparagraph we will give an outline of this research.

RESEARCH APPROACH

As mentioned in the previous paragraph we are developing a system, which generates cities. These generated cities must mimic existing cities.



Figure 1
Liberty City.



Figure 2
Pompeii.



Figure 3
Amsterdam.

Figure 4
Network typologies.



a) Population based (Altstadt)

c) Circular (Amsterdam)



b) Grid (Amsterdam)

d) Radial (Paris)



There has been extensive research in the above mentioned research areas. We can recognize two main streams of generating city research, template based en agent based engines. We will discuss these two approaches in some detail.

L-Systems were created by the biologist Lindenmayer as a method to simulate the growth of plants (Prusinkiewicz and Lindenmayer 1990). *“In an L-system, each plant module is represented by a letter, different letters being used for modules of different types or in different states. A sequence of letters forms a word which represents the entire plant. Development is simulated by a process of rewriting; a rewriting or production rule is applied to a letter, resulting in its replacement by a new letter or group of letters”* (Hanan, 1992).

Parish and Muller (Parish 2001) used a template based L-system to generate cityscapes. Parish and Muller recognized in real life cities 4 different types road networks (see figure 4), which they used as a template for their system.

To generate a city with sloped streets they used a gray tone (elevation) map as input parameter to. The topology of the location was depicted in gray tones, each gray tone could be translated into a height.

A different approach was used by Lechner (Lechner et al. 2003, Lechner et al. 2006), they developed a agent based system. The only main input of their system is a terrain description. *“With the aid of agent based simulation we are generating a system*

of agents and behaviors that interact with one and another through their effect upon a simulated environment” (Lechner et al. 2003). “The city generation is implemented by simulating cities using a set of agents that can model specific city entities such as developers, planning authorities and road builders. The system models not only the road network and buildings but also simulates the growth and development of the city over time” (Kelly and McCabe, 2006)

The goal of these approaches isn’t to mimic existing cities; they are constructed from abstract buildings. The goal of both systems lays not so much in reproducing existing cities but in generating convincing and plausible cities.

THE PROPOSED SYSTEM

As we explained in the previous paragraph our system must be capable of at least three different tasks:

1. Generating cities zones.
2. Generating infrastructure network.
3. Generating buildings.

Cities zones

We start by analyzing the map of the original city and make a zoning map. In this zoning -map each zone has its own color. Each color stands for a well defined combination of zone-identity attributes (see first paragraph). The development of this zoning-map is done manually, using existing city maps.

By re-arranging the zones we get an alternative city lay-out which is fundamental the same as the target city. For the aforementioned step we will use the landscape generator developed by Slager (Slager 2011) for generating alternative city-scapes.

This landscape generator uses landscape types as building blocks of plan scenarios. A landscape typology describes a proposed future spatial development and contains spatial and (non)spatial (descriptive) attributes. A 2D reference image indirectly provides objective compositional and configurationally characteristics of the proposed development. These spatial attributes and their target values are retrieved from the compositional and configurational characteristics present in the reference im-

age of the landscape typology. The landscape generator is developed as a multi-objective heuristic optimization modeling approach, and generator contains probabilistic elements (e.g. random starting situation, near-random cell swap), which results in different output, each time it is run with identical input settings. The system is capable of producing a range of landscape configurations for a variety of situations. This generator was developed to produce plausible landscape configurations, so this system lacks the ability to generate adequate infrastructures.

The re-arranged zoning map is input for our system to generate plausible alternatives city zone layouts. This newly generated abstract zoning plan is base for generating alternative infrastructures and buildings. By reading the colors of the re-arranged zoning plan the system “knows” what type of infrastructure, roads and buildings it has to generate within each zone to mimic the identity of the original city.

Infrastructure network typology

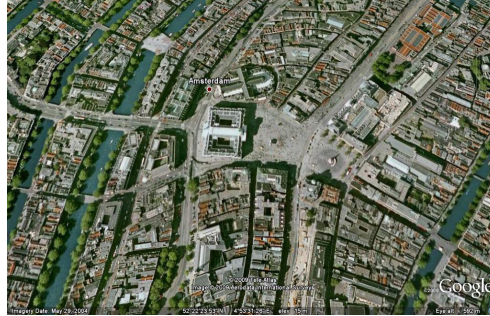
There are lots of different types of infrastructure or patterns (Alexander, 1977) but according to the literature there are four main typologies (see figure 1, the population based, grid, circular and radial networks).

We will use our system to generate Dutch look-alike cities; therefore we have to look if there is a need to localize network typologies. In the Netherlands there is a local network typology, a combination of two distinct infrastructures, the land bound and the water bound. In Amsterdam, Utrecht and more Dutch cities there are canals, on every bank there is a road with bridges connecting those two roads. So to localize the number of existing road types we need to add an extra road type, which is a juxtaposition of two different infrastructure-networks (see figure 4).

Network

We decided to use L-systems (Prusinkiewicz, 1966) as a base to generate the infrastructure. This decision is based on the fact that we want to make a rule-based system, so we can better direct the outcome. The

Figure 4
Juxtaposition (Amsterdam).



traditional L-system has no 'knowledge' of its environment. We will extend the L-system the ability to interact with the 'environment' (Mech, 1996). The environment is in our case the colored zoning map. The L-system can 'read' the colors of the zoning map and act according to it, as each color stands for a zone identity, a combination of zoning attributes. In this way the system can create for instance a road type according to the zone to which it belongs.

Buildings

In order to use an L-system to generate buildings, we have to analyze existing buildings to develop production rules which are stored in a data base. It is our intention to automate this building analyses phase; this will be done by analyzing photos. We think of analyzing the photo's in Google maps. For every zone we have to analyze sufficient buildings of the same type.

We (the system) will perform analyses of the photos in regard of: type of building, type of main entrance, how many floors, what kind of roof, layout of the facade etc., to make up the production rules. These rules will range from number of floor to the layout of window and will be categorized according this level of detail (=LoD). Each rule will have an unique id which is made up of: building type, category it belongs, rule type (number of floor, windows layout etc) and level of detail. This makes it feasible for the system to pick at random for each LoD different rules to generate alternative buildings. Rules which belong to the same type of building, will

generate output which will have some similarity with the existing building type from which the rules where derived.

CityGML

The consecutive results will be put in a model based on the CityGML. This decision is based on the fact that the pipeline of the system resembles that of the LoD used in CityGML. The level of detail of our system will range from LoD 0 (= our zoning plan) to LoD 3 (= our generated building). Our system doesn't generate interior layouts, so the buildings can't be entered.

By using CityGML the outcome of our system can easily imported in other CAD software or viewers, for further visualization or calculations.

DISCUSSIONS

In this paper we discussed the outline of our research to develop a city generator which will generate cities which mimic existing cities.

After finishing the above discussed system we will expand the system with a module which place texture on the surfaces. The materialization of the buildings and roads is also a part of the identity of the city.

It is too soon to conclude if it is possible to write rules which capture the identity of a city. We are still in de phase of the development of the engine. According to the literature it is possible to generate a plausible city.

To validate the system we need to develop a number of different rule sets, each set for a different real-life city. Next step is to generate according to these rule sets 3D 'look a like' cities. We will present these generated 3D cities to a panel of professionals, with the question: "Which city are you looking at/ walking thru?" From their answers we can conclude if our generated cities mimic existing cities or not.

REFERENCES

- Alexander, C 1977, *A Pattern Language*, Oxford university press, New York.
- Haegler, S, Muller P and van Gool L 2009, *Procedural Mod-*

- eling for Digital Cultural Heritage, EURASIP, *Journal on Image and Video Processing*, Volume 2009.
- Hanan, JS 1992, *Parametric L-Systems and their application to the modelling and visualization of plants*, Ph.D. dissertation, University of Regina, Saskatchewan.
- Kelly, G and McCabe, H 2006 A survey of procedural techniques for city generation, ITB Journal issues 14, December 2006, *The academic Journal of the Institute of Technology Blanchardstown*.
- Mech, R and Prusinkiewicz, P 1996, Visual Models of Plants Interacting with Their Environment, in *Computer Graphics Proceedings Annual Conference Series*, ACM SIGGRAPH, pp. 397-410.
- Muller, P, Wonka P, Haegler S, Ulmer, A and van Gool, L 2006, Procedural Modeling of Buildings, *ACM Transactions on Graphics*, vol 25, no3, pp. 614-623.
- Parish Yoav, IH and Muller, P: 2001 Procedural Modeling of Cities, *SIGGRAPH '01 Conference Proceedings*, pp. 301-308.
- Prusinkiewicz, P and Lindenmayer, A 1996, *The Algorithmic Beauty of Plants*, Springer Verlag, New York
- Schirmer, P and Kawagishi, N 2012, Using shape grammars as a rule based approach in urban planning- a report on practice, *Proceedings of the eCAADe Conference, Zurich, Switzerland* pp. 116-124.
- Slager, K 2011, *Landscape Generator, Method to generate plausible landscape configurations for participatory spatial plan-making*, Eindhoven University of Technology, Eindhoven.
- Wonka, P, Wimmer, M, Sillion, F and Ribansky, W 2003, Instant Architecture, *ACM Trans Graph.* 22, pp. 669-677.