Procedural 3D Urban Content Generation in Simulation and Games

Noura El Haje
Jean-Pierre Jessel
Institut de Recherche en Informatique
de Toulouse
Paul Sabatier University
Toulouse, France 31400
Email: Noura.El-Haje@irit.fr
Email: Jean-Pierre.Jessel@irit.fr

***Abstract*—As virtual urban environment generation becomes a widespread research topic, the need to create enriched urban worlds, which group both the geometric and semantic properties has become a necessity, the purpose being to obtain interactive, adaptive and consistent world entities. This paper suggests a representational model of urban environment generation based on both geometric and semantic knowledge, the future work being to create a modeling tool of a 3D urban environment, which stores the semantic information and allows the final non expert user to acquire information about the environment before and after its creation and to explore this information in an interactive way.**

***Keywords*–3D Content; Semantic-Based; Enriched Environment; Web Content.**

I. Introduction

Virtual environment generation techniques and modelers have become a powerful and efficient way for creating plausible 3D environments destined for visualization, simulation or serious games. The automatic generation of these virtual environments has also reduced the time and cost charges of the manual generation, which can be very laborious and repetitive for games developers. Although works were established on visualization and generation tools concepts, there is not so far an intuitive generation solution that makes use of the semantics in a way that allows the final user not only to acquire basic knowledge about the world, but to deal with the semantics at a higher level.

In Section 2 of our paper, we will briefly elaborate the most relevant urban generation techniques with their advantages and limitations. We will then describe in Section 3 our representational model called UDG (Urban Data Generation) along with the standards and techniques used to create this model. Section 4 develops the mecanism of the representational model and discusses what needs to be done and the challenges behind information retrieval from the city generated before concluding our paper in Section 5.

II. Related Works

This section develops the most relevant works to our research focus, mainly with the preliminary works on procedural urban models generation to the most recent approaches in this domain. The semantic-based generation topic, along with its advantages and challenges is also elaborated.

A. Urban Generation

A common approach for procedurally generating cities is to start from a dense road network and identify the polygonal regions enclosed by streets. Then, the subdivision of these regions results in lots, which are populated with buildings in two ways: either the lot shape is directly used as a building footprint or the building footprint is fitted on the lot. Finally, by simply extruding the footprint to a random height, it becomes possible to generate quite complex models [1]. Some approaches like split grammar [2] and shape grammar [3] have been used for generating several kinds of pattern structures by executing constructive solid geometry operations on a set of components selected by query mechanisms. An approach that also retains the advantages of grammar-based solutions is the procedural content graphs [4]. The content generation procedures are specified by a generic approach, which gives a richer and a more expressive design specification than that offered by other grammar and graph-based approaches.

A more recent concept in the domain of procedural modeling, the inverse procedural modeling, was discussed by Musialski et al. [5]. It aims at discovering both the parametrized grammar rules and the parameter values that yield to a prespecified output when applied in a particular sequence. The most important advantage of the inverse approach is that it leaves the designer entirely out of the process: the Procedural modeling is opaque so the designer does not need to know about it at all [6]. Another work on the topic of inverse modeling was established by Vanegas et al. [7] who propose a framework that enables intuitive high-level control to an existing urban procedural modeling by interactively edit urban model. Their system is capable of discovering how to alter the parameters of the urban procedural model so as to produce the desired 3D output. The limitation of this method is the lack of accuracy of their engine with parameters increase, especially when a large number of dependant parameters are included.

Most of the approaches presented above are based on traditional procedural modeling, which was and is still considered a significant contribution for the computer graphics community. However, the major drawback of this approach is that the number of rules to be defined is quickly increasing with the size of the environment and the level of detail (LOD), which becomes difficult to control and maintain.

B. Semantic-Based Generation

The majority of procedural methods are specialized in generating one specific type of content or feature. To be really useful in the design of complete virtual worlds, heterogeneous data needs to be assembled, explored and integrated in a modeling tool. The advantage of making use of heterogeneous data for urban environment generation is that it allows designers to focus on high level concepts in the world design rather than on the model level [8]. In this direction, the authors described their framework (SketchaWorld) based on a declarative modeling approach and which combines the strength of manual and procedural modeling by allowing designers to state their intent using simple, high-level constructs. Their declarative approach builds upon established research results on parameterized procedural generation, constraint solving and semantic modeling in order to automatically tarnslate statements into a matching 3D virtual world. The consistency of this world is automatically maintained using a semantically rich model of all its features and their relations, analogous to the automatic maintenance of interior scenes based on object semantics.

The lack of rich information suitable for consumption by the game artificial intelligence (AI) was exposed in [9] where the solution suggested the improvement of the embedded information contained in immersive game worlds by focusing on the symbolic annotations of environmental elements. Some authors [10] focused on applying the semantics for managing digital representations of buildings for architectural applications and analyse their transformations over time. Their description model defines three levels of the building morphology: the semantic, the structure and the representation one and the temporal dimension is joined to these three levels. Also in the context of the city change over time, an interactive agent-based behavioral system was elaborated in [11]. This system is linked to the geometrical modeling in order to create plausible urban models. Their platform (NUBES) focuses on the definition of an informative system on an architectural scale, which exploits the relations between the 3D representation of the building and heterogeneous information coming from various fields (technical, documentary or historical). It aims at organizing multiple representations (and associated information) around a model of semantic description with the aim of defining a system for the multi-field observation of historic buildings.

III. Our Proposed Representational Model

This work in progress suggests a representational model of urban data that associates semantic information to 3D urban data. This model will be implemented in a modeling tool so that the final non expert user could visualize the data and explore the semantics in order to create his own modified urban environment. Another usability of the semantics will be to make the environment architecturally more complex by adding texture and geometric details. The previous works explained above showed how buildings and complete cities could be generated based on heightmaps and procedural rules.

A. Standards and Tools

Our proposed model called urban data generation (UDG) is built around some standards and tools for a complete environment representation. First, we focused on extracting geospatial data layers of cities from the openly licensed map, OpenStreetMap which is a viable alternative to commercially offered data sources. The exported format, the OpenStreetMap (OSM) format is coded in extensible markup language (XML) and contains geographic data in a structured, ordered manner.

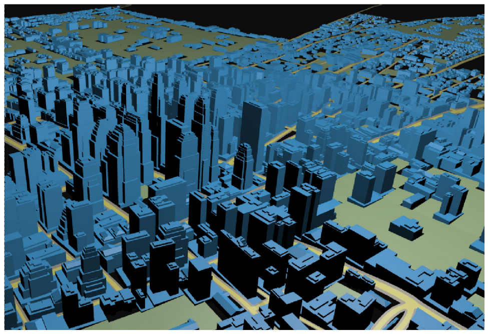

Figure 1. X3D file display for an intermediate city size in html page

In order to represent the OSM file in 3D, we focused on the Open-Source web graphics library (WebGL) framework and in particular the document object model (X3DOM) for two main reasons: First it's an ISO standard file format which represents 3D graphics using XML syntax, which makes it easy to undestand and modify. Second, its runtime environment manages interoperation between the extensible 3d (X3D) browser and host application for purposes like file delivery, hyperlinking, page integration and external programmatic access. Finally, this format is supported by many 3D modeling softwares such as Blender, Meshlab and 3Ds max(via a plugin) which enables the environment to be externally edited by the designers. Figure 1 shows an example of 3D urban data displayed as a webpage that takes an urban file in the open street map (OSM) format and displays it as an extensible 3d (X3D) file.

The X3D file is then imported into the game engine Unity3D using the X3D plugin for Unity. The use of Unity3D in the development allows a better control of the city layers and an easier fusion of the semantic information with the geometric one. The constraints of only using the X3D format to visualize the city is the lack of the semantic information and thus a lack of ”character” to the buildings. To solve this problem we needed to enrich the extensible 3d (X3D) file by adopting two approaches: 1- adding metadata tags inside the extensible 3D (X3D) file and 2- defining relations between layers models.

The geographic markup language (CityGML) model was our inspiration for enriching the world with metadata. CityGML is a common information model and an XML-based encoding that describes 3D objects with respect to their geometry, topology, semantics and appearance. It also defines five levels of details (LODs). The LOD concept in CityGML is different from the one in computer graphics since it denotes the model spatio-semantic adherence to its real-world counterpart [12].

Buildings at LOD0 in CityGML can be represented in two ways: a footprint and a roof edge (in general). Both the LODs representation of a building footprint and a roof edge have to be a horizontal surface pursuant with CityGML specifications. If a footprint is in reality situated on a slope then the lowest

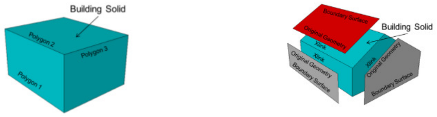

Figure 2. Different LODs of a building taken from CityGML specifications

value has to be used (as specified in CityGML). Figure 2 shows the difference between LODs. Left: and LOD1 solid without surfaces and right: an LOD2 solid with accompanying modelled surfaces.

B. Our Representational Model Description

Our interface of city generation is given some parameters as input. This is the default scenario which is parametrized by the user. Some of these parameters are : the city type (large city, village, small town...), by the sea or by a river and the society type (isolated, mixed...). Our environment consists of three generators: the terrain generator, the street generator and the buildings generator.

1- Terrain Generator: The terrain is generated using the ”CoherentNoise” library, a library for Unity3D that allows the use of noise of all kinds. The library has the ability to modify and combine different noises together, all with a simple and clean object-oriented API. CoherentNoise library allows for unlimited possibilities.In most basic form, noise function can be used to distort some pattern - that may be itself generated procedurally or made by hand. Noise could be directly added to the pattern, or noise values are used to perturb the pattern. Perturbation of function is achieved by adding noise value to the function input. Figure 3 shows the effect of addition and perturbation on a 2D pattern and the terrain sample using the coherent noise function.

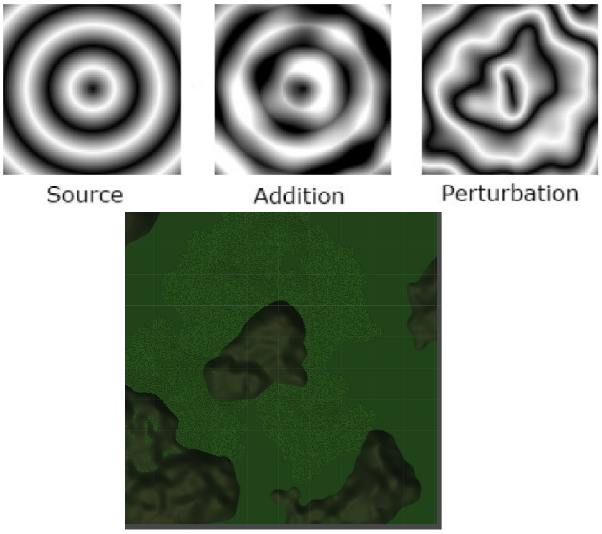

Figure 3. Effect of adding noise to a pattern and the sample terrain we generated using Coherent Noise

2- Street Generator: To create street patterns, we used the famous Voronoi Diagram technique. This technique creates cell-like or network-like patterns. A diagram is based on an (infinite) number of control points, that are randomly distributed in space. Such distribution is not completely random, but it is enough for most practical applications. Base classes for Voronoi diagrams determine distances to three closest control points and this allows to make own diagrams by combining these distances in creative ways. All Voronoi diagrams have a 2D variant. These variants are exactly the same as their 3D counterparts, except they ignore the Z- coordinate. This makes them considerably faster (as less control points must be considered). Figure 4 explains the concept of Voronoi and a street pattern we generated by this method.

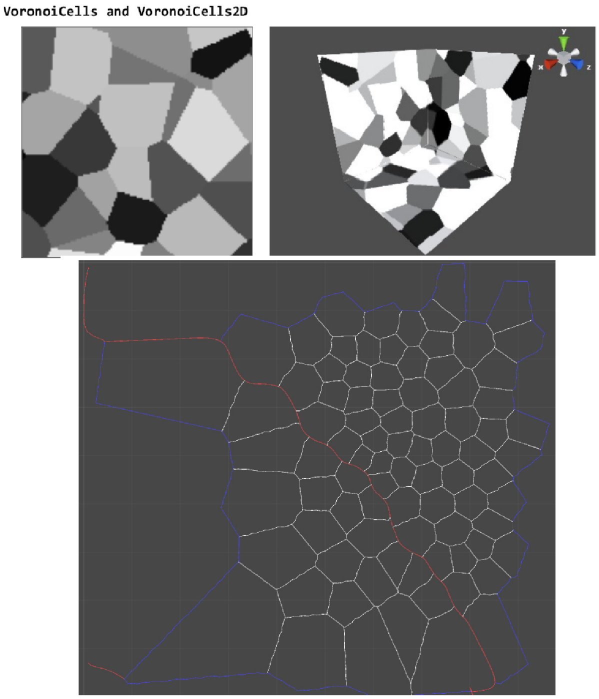

Figure 4. Voronoi graph and streets generated using this technique

3- Building Generator: As we explained before, we are using X3D format to generate the buildings in our environment. The drawback of the X3D is its lack of high level semantic information. The information regarding the buildings is only limited to basic internal properties such as placement, size and material color. Therefore, we resorted to CityGML, a standard that provides five standard LODs: LOD0 is a 2D footprint, LOD1 is a block model obtained with extrusion, LOD2 is an upgrade of the former with simple roof structures and semantically enriched boundary surfaces, LOD3 are architecturally detailed models with fenestration, and LOD4 contains interior [13].

IV. Framework Mecanism and Discussion

The mecanism of our representational model UDG with the different blocks is shown in Figure 5 and will be integrated in a modeling interface largely dependant on the semantics. The modeling interface takes as input the default scenario that gives specification about the city from the user and a default X3D

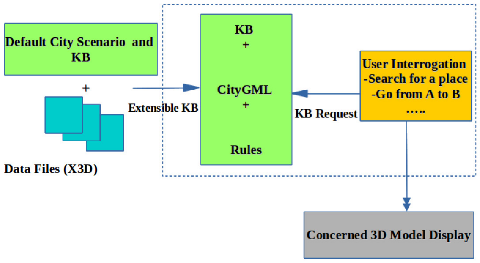

Figure 5. Representational model of the city generation

file. The user using parameters from the terrain generator, the street generator and the buildings generator explained above respectively allows adding more visual details to the city and thus enriching the X3D file. The knowledge base makes use of the CityGML nodes and should be automatically updated. Up to now, the visualization is limited to a general level of detail and the visualization and manipulation of buildings interior is not to be considered at this stage of our work.

After the city generation, the user shoud interact with the city by launching some sort of a query (for example looking for a building that have more than 3 stories or how to go from point A to point B). The knowledge interpretation and its visibility in front of the user is an aspect we are currently investigating. It could be satisfied graphically (by displaying tags on the buildings according to their types or characteristics), or visualizing the acceptance area depending on the user request.

To summarize what we explained before, this is what needs to be done: First, accomplish the development of the modeling interface which receives as input the extensible 3D (X3D) file and displays it. Second, we need to define the interaction types and ways that the user could have with the city generated, which is important for our system functionality validation. And third, rules and constraints should be specified to maintain the world consistency while paying attention to not over-constrain the system, which could lead to unrealistic urban representation.

V. Conclusion and Future Work

We have introduced an approach to combine the geometric urban data with semantic information, both high and low level. The implementation task is in progress and aims at reaching a complete modeling interface for city creation based on many standards, tools and models that make use of the world knowledge. At this moment, the process to provide semantic contents is manually established. We plan at a later stage to automate the generation process of the semantic database.

As a future perspective, semantics could be also interesting in improving the geometric models by adding more architectural details and textures. Our models are basic geometric shapes extruded or refined with no accessories such as windows, roofs or balconies. The semantics could therefore make the environment visually more appealing to the user by allowing various LODs. Finally, as we have seen from the related works on the different levels of semantics, there are promising uses for semantic information in virtual games worlds and in the near future, we expect increasing effort to be made in the merging process of different data type for virtual worlds creation destined for simulation and games.

References

[1] S. Greuter, J. Parker, N. Stewart, and G. Leach, " Real-time procedural generation of 'pseudo infinite' cities," Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia 2003, NY, USA., pp.: 87-ff http://doi.acm.org/10.1145/604471.604490

[2] L. LeBlanc, J. Houle, and P. Pulin, " Component-Based Modeling of Complete Buildings," Proceedings of the Graphics Interface 2011, May 25-27 Canada, pp.: 87-94 http://doi.acm.org/10.1145/237170.23719

[3] M. Larive and V. Gaildrat, "Wall Grammar for Building Generation," Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and South-East Asia 2006, Canada, pp.: 429-437 http://doi.acm.org/10.1145/237170.23719

[4] S. Pedro Brandao and E Eisemann, "Procedural Content Graphs for Urban Modeling," International Journal of Computer Games Technology ,volume 2015, Article ID 808904, 15 pages http://dx.doi.org/10.1155/2015/808904

[5] P. Musialski and M. Wimmer, "Inverse Procedural Methods for Urban Models," Eurographics Workshop on Urban Data Modeling and Visualization, May 2013, pp.: 31-32, Editors V.Tourre and G.Besuievsky

[6] R.M. Smelik, T.Tutenel, R. Bidarra, and B. Bedrich, "A Survey on Procedural Modeling for Virtual Worlds," Computer Graphics Forum 2014, Volume 33, issue 6, pp.: 31-50 http://dx.doi.org/10.1111/cgf.12276

[7] C. Vanegas, I. Garcia-Dorado, D. Aliaga, B. Benes, and P. Waddell, "Inverse Design of Urban Procedural Models," ACM Trans. Graph.31, November 2012, Article 168, 11 pages http://dx.doi.org/10.1145/2366145.2366187

[8] R.M. Smelik, B.Tutenel, K.J. De Kraker, and R. Bidarra, "A Declarative Approach to Procedural Modeling of Virtual Worlds ," Computers and Graphics 35, 2011, pp.: 352-363 http://dx.doi.org/10.1016/j.cag.2010.11.011

[9] G.M. Youngblood, W.P. Frederic, D. Heckel, H. Hale, and P.N. Dixit, "Embedding Information into Game Worlds to Improve Interactive Intelligence," Artificial Intelligence for Computer Games, 02 February 2011, pp.: 31-53

[10] L. De Luca, C.Busayarat, C. Stephani, P. Veron, and M. Florenzano, "A Semantic-Based Platform for the Digital Analysis of Cultural Heritage," Computers and Graphics 35, Issue 2, April 2011, pp.: 227-241 http://doi.acm.org/10.1016/j.cag.2010.11.009

[11] C. Vanegas, D. Aliaga, B. Benes, and P. Waddell, "Interactive Design of Urban Spaces Using Geometrical and Behavioral Modeling," ACM Siggraph Asia 2009, ACM NewYork, USA, Article 111, 2009, 10 pages http://doi.acm.org /10.1145/1661412.1618457

[12] F. Biljecki, H. Ledoux, J. Stoter, and J. Zhao, " Formalisation of the level of detail in 3D city modelling," Computers, Environment and Urban Systems 48, November 2014, pp.: 1-15

[13] T.H. Kolbe, " Representing and exchanging 3D city models with CityGML," In 3D Geo-Information Sciences, Zlatanova S., Lee J., (Eds.). Springer Berlin Heidelberg, 2009, pp. 15-31. 2

Copyright (c) IARIA, 2016. ISBN: 978-1-61208-464-0