Automatic Generation of Urban Zones

Mathieu Larive
OktalSE
2, impasse Boudeville
31100 Toulouse
France
mathieu.larive@oktal.fr

Yann Dupuy
OktalSE
2, impasse Boudeville
31100 Toulouse
France
yann.dupuy@oktal.fr

Véronique Gaildrat
IRIT-UPS
118 route de Narbonne
31062 Toulouse
France
gaildrat@irit.fr

ABSTRACT

We present a state of the art of techniques that can be implemented to automatically generate virtual cities (non existing ones). An urban zone constitutes a too vast volume of data to be directly understood, modeled or visualized. The multiresolution approach based on imbricated logical structures allows us to divide this volume of data into successive levels of detail.

Keywords
Urban modeling, architecture, procedural methods, declarative modeling

1. INTRODUCTION

Recent applications in virtual reality, video games and simulation of city expansion as well as the emergence of problems due to the urbanization, such as the influence of electromagnetic radiations and the forecast of the urban transportation network, creates increasing needs in term of digital mock-ups studies and forecasting. The capacity to quickly generate credible digital city models helps the user to fulfill these needs.

A real city satisfies construction rules and depends on multiple influences throughout the time. However the detailed modeling of realistic towns is very challenging for computer graphics. Modeling a virtual city which is detailed enough to be credible for a visiting user is a huge task that requires thousands hours of work. In this context, we think that automatic approaches are well suited for this problem. Hence, they represent a promising research topic which has to be developped because the current results do not satisfy the previously defined needs. For these reasons, we propose a state of the art of the current techniques to distinguish what exists and what remains to be studied.

Conference proceedings ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

2. AUTOMATIC TECHNIQUES

Figure 1 presents a hierarchical division in seven stages of city generation. This division is used as the guideline of this paper to present the existing methods. According to the complexity of every stage, we propose a brief presentation of the formalism and the handled data as well as a discussion about the presented generation methods. A stage can be seen as a level of detail during a multi-scale representation of the city. Techniques described in this paper come from various domains such as artificial intelligence, operational research, town planning or declarative modeling.

2.1 Urban zone

We need to know at least the limits of a city to be able to generate it. We consider the definition of the surface that the city covers as the minimum level of information. This information is generally defined within a GIS (Geographic Information System). It is also possible to constraint the generation in order to respect data map: geographical (altitude, hydrography and vegetation) or socio statistical (population density, street patterns and elevation).

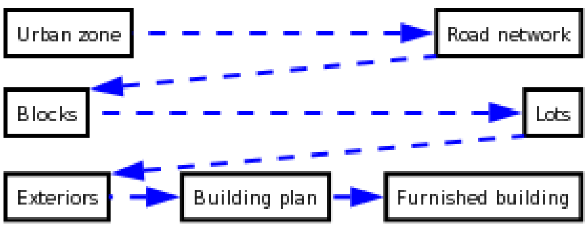

Figure 1: Hierarchical division of city generation

Figure 2: One possible roadmap generated from a set of maps [PM01]

2.2 Road network

Even if it seems illusory to try to characterize all existing cities according to predefined patterns of streets, some classic patterns can be used during the generation. For example, most of the cities in USA are designed according to a checkerboard pattern, whereas European cities tend to follow a radial/concentric pattern. Nevertheless, these observations are to be balanced by the fact that the structure of a city moves during time, according to geographic and socio-economic constraints. In practice, we tend to observe a composition of patterns within the same city.

L-System CityEngine[PM01] uses an extension of the L-System for the creation of the road network. Creating realistic road networks needs a lot of rules with a large number of conditions and parameters. Writing a new rule means rewriting numerous rules, so the extension of a system is actually a difficult task. In order to avoid the growth of the set of rules, Parish and Müller defined an extension of the L-Systems that creates generic successors at every generation stage: they named them *ideal successors*. The parameters of these successors are not instantiated, the call to the global goal function instantiates them in order to satisfy the global goals. Then, the system checks if these parameters satisfy the local goals and modifies them if not.

Declarative modeling of line segments In the field of declarative modeling for urban layout, [LH97] introduces the notions of urban elements that can be combined to obtain more complex urban patterns. The scene is described in a hierarchical and incremental way. The user begins with an approximate sketch, giving only the main features such as the main roads and crossings. This description is used to propose one or several solutions that the designer can refine. The generation process is based on the CSP approach.

Discussion The procedural method based on the L-Systems creates city-scale results, while the one based on declarative modeling suffers from the chosen generation method that does not allow it to generate results from a too vast search space. Nevertheless, the approach stemming from the declarative modeling describes urban patterns not yet managed within procedural approaches. To guarantee a bigger variety of generated road networks, it would be a good idea to integrate the not yet available urban patterns (as the diamond shaped one) into the L-System formalism.

2.3 Blocks

A block is a connected surface delimited by roads or streets. The partition of the city into blocks defines a first stage of hierarchical organization: this decreases the number of constraints to manage, for example in the case of the disjunction constraint. The used model for a block is the polygon, commonly only the convex polygons are used to take advantage of the lesser complex geometric algorithms for this kind of polygon.
The generation process of this stage is not at all complex compared to the previous stage. It can be seen as a simple intersection computation between the segments of the road network as in [PM01].

2.4 Lots

A *lot* is a surface of the same mailing address. Lots are the areas on which the buildings will be placed. It is possible to generate all the lots of a block then to generate again those that do not satisfy the user. There are several similarities with the blocks stage, they use the same model: the convex polygons.
Like for the road networks, the partition of a block into lots is generally done according to a predefined pattern. This resemblance of description does not extend into the generation methods: the small dimensions of a block do not require the use of the L-System. The used methods tend to be based on a geometric partition or on an instantiation of a pattern.

Geometric partition In [PM01], this stage is handled by a recursive process that divides the bigger lot in the middle of its bigger side. This process ends when the surface of the biggest lot is lower than a predefined threshold. Then a post-filtering on the lots

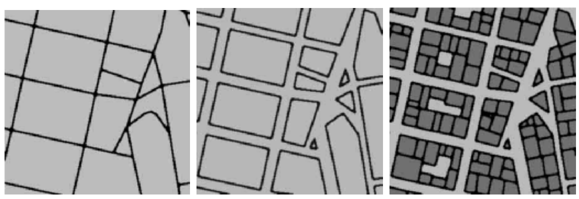

Figure 3: Roads, blocks and lots [PM01]

delete the too small lots as well as those that do not have a direct access to a road.

Pattern guided partition In [LH97] the generation begins from a discretized surface corresponding to the block. A point of this grid is randomly chosen, then the process searches for another point in order to create a new segment that will be compatible with the chosen pattern. This guided search for the extremity of the segment reduces efficently the search space.

Discussion Both methods presented here are extremely different. The first one is effective but creates little varied results. The second offers a big variety of results but uses a generation based on an enumerative process, that leads to restricting the domain of application. According to the degree of realism and efficiency wanted either can be used.

2.5 Buildings

This stage defines the exterior of the buildings. The variables to instantiate during this stage are the shape, height and orientation of the building as well as the distance to the lot's contour.

Shape grammar The LaHave House project studies the industrial architectural design in [RMS96]. This tool is intended to help the designer to conceive aesthetic and functional houses that will be buildable.
From shape grammars, a library of architectural elements is generated then used to visualize working/assembly drawings and 3D scenes. A model is made up of the geometric description and the configuration of the house levels.

Declarative modeling BatiMan ([Cha98]) is a declarative modeller of a small number of buildings. It is based on a learning process stem from the Artificial Intelligence. It classifies the solutions to propose to the user a restricted number of characteristic solutions. Batiman relies on a constraint solver based on the CSP approach on finite domains for the generation phase. The number of parameters defining a building can go to about twenty with a small number of values for each variable.

Split grammars Work described in [WWSR03] derived buildings using split grammars. Theses grammars are parametric set grammars based on the concept of shape. The system uses a second kind of grammar, the *control grammar*, in order to control the propagation of the split grammar's attributes. Using theses two types of grammars, the system can create realistic buildings from high-level or precise descriptions. Nevertheless, if the user wants to modify the rules of the grammar, he must be familiar with the split grammar approach.

Geometric modeling AGETIM [LLGC+05] is an integrated software tool suite that enables 3D environment generation with infrared, electromagnetic and acoustic characterization. The GenVillage module allows the system to create buildings from their base (created in an automatic or manual way) by using predefined templates. Buildings which are created in this way have variable heights and roof forms, and are automatically adapted to their environment (style, orientation and textures).

Discussion The methods described for this stage answer different needs. The construction of the library required for the use of the shape grammar takes a lot of time but allows the user to generate realistic results. The declarative modelling is less realistic but quicker than the first one, whereas the method based on geometrical modelling allows an easy integration according to the building's environment.

2.6 Building plan

This stage was studied as a 2D architectural problem that corresponds to a partition process. Work presented in [Mac91] introduces the building design as a process of constraint satisfaction. This work relies on the framework of AI and Operational Research. The representation of the spatial knowledge in architecture was more particularly studied to be able to define the spatial constraints. More precisely, he studied the link between the symbolical (topological) and the numerical spatial knowledges that had not been properly studied before.
Theses knowledge were used on objects which were modeled using 3D boxes called the *Manhattan boxes*. This representation was selected because of its adequacy with the architectural problems: a building can be seen as a compound of simple shapes. These boxes and the associated operations form the *Manhattan algebra*. A constraint network represents the problem, that is solved by a static propagation mechanism of constraints using heuristics for the instantiation order of the objects.
An extension was developed [Cha93] in order to take into account the geometric specificity of a layout problem, using a new filtering method *semi-geometrical arc-coherency*. Work presented in [MY01] proposes an improvement of previous work by the use of dynamic space ordering heuristics and a topological characterization of the solution space.

2.7 Furnished buildings

It is possible to completely define the instantiation of an object within a 3D model by its geometrical description and its variables of orientation and position.

Figure 4: Scene generated by a tabu search based declarative modeler [LLRG04]

The number of constraints to solve grows exponentially with the number of objects.

CSP approach During the last decade, several constraint solvers based on the CSP approach have been developed to resolve the interior space layout problem. We can distinguish two classes: those limited to an isothetic layout [BP99, LRG03] and those that allow the user to place objects with any orientation [RP02, LR03].

Metaheuristics Stochastic methods (or heuristics) were first studied within the framework of combinatorial optimization in operational research and AI. During the last fifteen years, the stochastic methods have notably progressed with the emergence of a new generation of powerful and general tools called *metaheuristic methods* [Glo86].
Two different classes of metaheuristics were studied for the space layout problem: first the genetic algorithms were studied in [SRGL03, VMC+02], then [LLRG04] studied the interest of a local search metaheuristic, the tabu search.

3. CONCLUSION

The work presented in this state of the art shows the variety of the existing approaches for the city generation. CityEngine, allows a city to be modeled from scratch to exteriors (stages 1 to 5), but no current tool takes into account the seven described stages. Moreover, it is difficult to compare the methods because they result from various research domains.
Some of the seven stages presented in this article have not been extensively studied. The block generation (as well as the lot generation) have been less studied. We have begun to work on this stage in order to evaluate the interest of geometric methods such as Voronoï diagrams and the straight skeleton approach.
It seems possible to create a unified tool that should be able to process all the stages. But this will probably be a difficult task.

4. REFERENCES

[BP99] Bonnefoi P.-F. and Plemenos D. Object oriented constraint satisfaction for hierarchical declarative scene modeling. WSCG'99, 1999.

[Cha93] Charman P. Solving space planning problems using constraint technology. Technical report, Institute of Cybernetics, Estonian academy of sciences, 1993.

[Cha98] Champciaux L. Gestion des contraintes géométriques pour l'aide à l'aménagement urbain. PhD thesis, Ecole des Mines de Nantes, 1998.

[Glo86] Glover F. Future paths for integer programming and links to artificial intelligence. Computers and Operations Research, 13:533-549, 1986.

[LH97] Liège S. and Hégron G. An incremental declarative modelling applied to urban layout and design. WSCG'97, 1997.

[LLGC+05] Latger J., Le Goff A., Champseix N., Cathala T., and Larive M. Automatic 3d virtual scenes modeling for multi sensors simulation. to appear. In Proceedings of SPIE Defense and Security Symposium, 2005.

[LLRG04] Larive M., Le Roux O., and Gaildrat V. Using meta-heuristics for constraint-based 3d objects layout. 3IA'04, 2004.

[LR03] Le Roux O. Modélisation déclarative d'environnements virtuels : contribution à l'étude des techniques de génération par contraintes. PhD thesis, Université Paul Sabatier, 2003.

[LRG03] Le Roux O. and Gaildrat V. Constraint-based 3d isothetic object layout for declarative scene modeling. CISST'03, 2003.

[Mac91] Maculet R. Archipel : intelligence artificielle et conception assistée par ordinateur en architecture. PhD thesis, Université Paris 6, 1991.

[MY01] Medjoub B. and Yannou B. Dynamic space ordering at a topological level in space planning. Artificial Intelligence in engineering, 15:47-60, 2001.

[PM01] Parish Y. and Müller P. Procedural modeling of cities. ACM SIGGRAPH, 2001.

[RMS96] RauChaplin A., MacKayLyons B., and Spierenburg P. The lahave house project: Towards an automated architectural design service. Cadex'96, pp. 24-31, 1996.

[RP02] Ruchaud W. and Plemenos D. Multiformes: a declarative modeller as a 3d scene sketching tool. ICCVG'2002, 2002.

[SRGL03] Sanchez S., Roux O. L., Gaildrat V., and Luga H. Constraint-based 3d-object layout using a genetic algorithm. 3IA'03, 2003.

[VMC+02] Vassilas N., Miaoulis G., Chronopoulos D., Konstantinidis E., Ravani I., and Plemenos D. Multicad-ga : A system for the design of 3d forms based on genetic algorithms and human evaluation. SETN, pp. 203-214, 2002.

[WWSR03] Wonka P., Wimmer M., Sillion F., and Ribarsky W. Instant architecture. ACM Transactions on Graphics, 4(22):669–677, july 2003. Proceedings of ACM SIGGRAPH 2003.