

Pseudo-urban automatic pattern generation

Renato Saleri Lunazzi, architecte DPLG, DEA informatique et productique, master en industrial design.

*Laboratoire MAP aria UMR 694 CNRS – Ministère de la culture et de la Communication.
e-mail renato.saleri@lyon.archi.fr*

Abstract

Notre but dans ce travail est de rechercher et d'expérimenter des méthodes de production automatique de morphologies urbaines ou architecturales. Nous avons jusqu'ici implémenté et fait converger des dispositifs s'appuyant sur une heuristique couplant un moteur de production de séquences pseudo-aléatoires avec un formalisme graphal, de type L-System (Lindenmayer System). L'objectif étant dans un premier temps de produire simplement et "à moindres frais" des environnements géométriques texturés visitables par le biais de technologies de visualisation 3D temps réel. Ce modèle de croissance, qui a déjà donné des résultats intéressants pourra être dans un deuxième temps perfectionné par l'utilisation de systèmes logiciels à comportement émergent, validé par des appréciations perceptives et contraint par des informations d'ordre topographique ou des règles de production urbanistiques ou architecturales. Ceci pourra, par exemple dans le domaine de l'archéologie urbaine et dans un contexte historique donné, permettre de produire rapidement un support d'hypothèses de restitution en présence de données fragmentaires.

In questo nostro lavoro di ricerca ci poniamo come obiettivo di sperimentare metodi di generazione automatica di morfologie urbane e architettoniche. Abbiamo finora sviluppato e fatto convergere taluni dispositivi, come seriazioni pseudo-random e formalismi graftali, o L-sistemi (Lindemayer system) al fine di produrre rapidamente spazi tridimensionali immersivi. Questi modelli di crescita hanno già dato nel nostro campo risultati interessanti. I successivi sviluppi, traendo spunto da inputs topografici e/o ambientali, dovrebbero promuovere la ricerca nel campo del calcolo emergente. Questa ricerca trova una valida applicazione nell'ambito della restituzione archeologica a grande scala, dando il modo ad esempio di produrre "rapidamente" ipotetici ma plausibili tessuti urbani storicamente contestualizzati.

This research task aims to experiment automatic generative methods able to produce architectural and urban 3D-models. At this time, some interesting applicative results, rising from pseudo-random and l-system formalisms, came to generate complex and rather realistic immersive environments. Next step could be achieved by mixing those techniques to emerging calculus, dealing with topographic or environmental constraints. As a matter of fact, future developments will aim to contribute to archeological or historical restitution, quickly providing credible 3D environments in a given historical context.

1. Introduction

Since the end of the 70's, the "fractality" of our environment raised as an evidence, pointing some peculiar aspects of everyday phenomena. Some micro and macro-scopic internal-arrangement principles appear to be similar or even auto-similar, leading the reasoning through general explanatory theories. Physicians and biologists regularly discover fractal processes through natural morphogeneses such as cristalline structures or stellar distribution. Human creations also seem to be ruled by fractal fundamentals and since 15 years, the "fractality measure" of some human artefacts can be somehow achieved.

Fractal investigation through urban patterns mainly focused on two subsequential aspects : the direct analysis of spatial organisation, and thus the formalization of self-generating geometrical structures. The growth of urban models is at this time fulfilled either by time-based spatial simulators or by simple static generators. Spatial simulators are usually based on simple "life-game" (cellular automata) devices or even by "diffusion limited aggregation" formalisms (DLA). In this paper, we will mainly focus on some generative techniques involved in 2D and 3D automatic builders.

2. Research task context

The mainframe of this research task consists in real-time rendering of huge 3D databases. Different aspects of this goal have already been explored, considering from the top that rendering techniques should be optimal for a given applicative context. Therefore, the main aspect of MAP-aria participation in this project consists in building plausible urban structures related to some given historical or archeological context.

Early stages of our investigation pointed the discontinuous properties of growth phenomena. In other words we barely believed in the existence of a possible continuous morphological development model, according to the evidence of micro and macro-scopic observable morphological differences on one hand and through bidimensional and threedimensional topological discontinuities on the other. In other words, we focused some "scale-based formalisms", related to specific urban scale-types, as listed in the following section:

3. Applications

The description of the following formalisms is broadly summarized. Further refinement on geometrical models, architectural primitives and morphological break-down are under development...

3.1 Multi-scale pattern generator

A "top of the heap" wide range concentric propagator, whose aim is to distribute, filter and drop geometric locators above a given terrain mesh.

The deal is here to develop a "general land-scaled model", mostly a variant of the L-system model depicted below. The initial distribution of locators basically follows a concentric distribution. Their final positioning can be meanwhile modified by some disruptive factor, mostly depending on simple angular non-overlapping constraints. The graph below shows

three different steps of the computation : locators displacement, neighbourhood tracking and plot drawing.

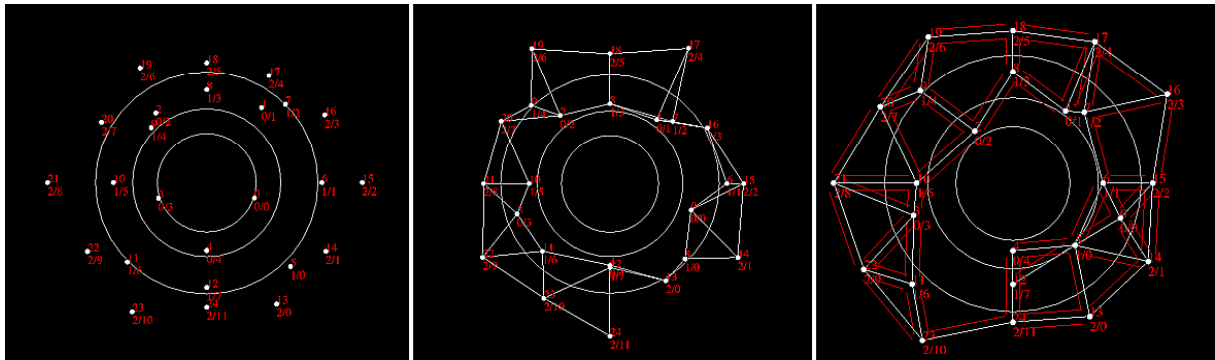


Figure 1. Deployment of a 2D geometric model.

A local geometric transformation transforms the initial structure to a position-related “constructible zone“, starting from two initial input variables, named here d' and d'' . At the moment, inevitable angular occlusions occur with sharp and wide angles. This drawback should meanwhile be solved in a very next release of the applet.

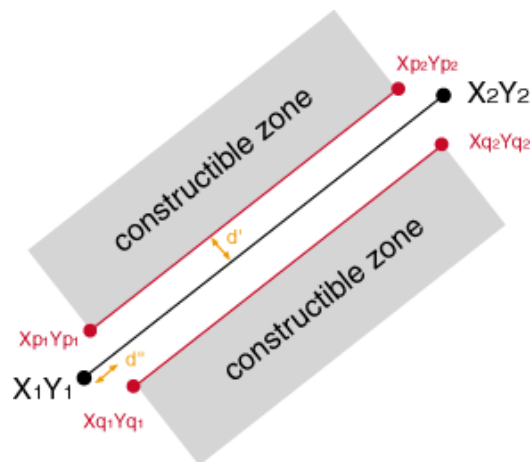


Figure 2. Geometric deduction of “constructible zones“.

Extracting the n closest neighbours and drawing the respective bijective connexions leads the entire process, and we can finally hybrid this bidimensional mesh to allocation rules and topographic constraints, to produce the models shown on the figures below : the skeleton and the final rendering.

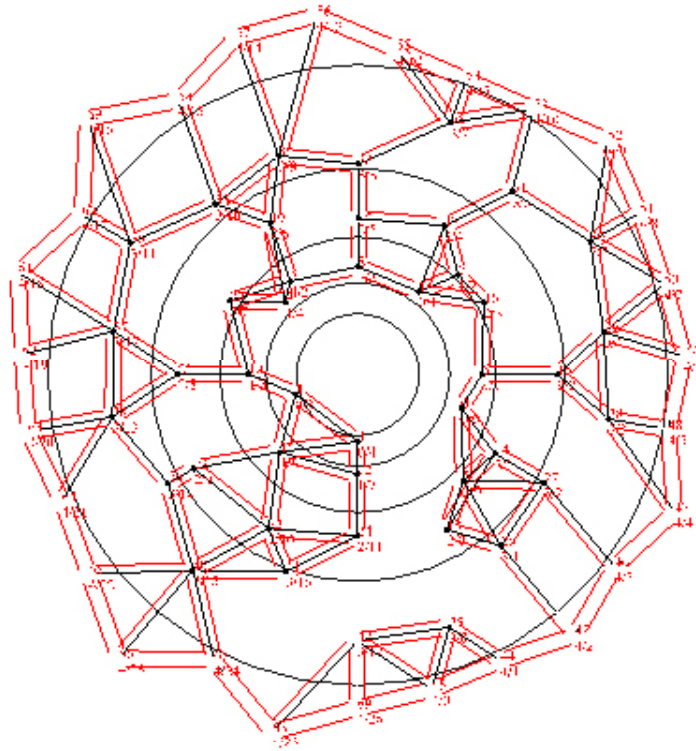


Figure 3. The geometric skeleton...



Figure 4. ...and it's 3D expression.

In this example, only one architectural primitive is distributed over the map ; a “hull – filling “ generator (shortly described below) could be implemented to create a more realistic perceptive variety.

3.2 Graphtal or L-system generator

Graphtal or L-system, applied to local building and block propagation.

The L-System, or Graphtal, starts from a simple recursive substitution mechanism. This rules-based generator, described in the late 60’s by A. Lindenmayer [1], can quickly provide complex geometric developments. It’s characteristic deal with simple substitution rules, recursively applied to a sprout, as shown below :

All we need to start is an alphabet, listed hereby : 0,1,[,]

In this example, 0 and 1 occurrences will “produce geometry“ while [and] will provide a simple affine transformation (rotation and/or translation). We can now describe simple substitution rules, applied to alphabetic elements :

0 : 1[0]1[0]0 **1** : 11 **[** : [**]** :

If we recursively apply those substitution rules to an initial sprout (applied from the top to the rule of letter “0“) we obtain :

11 [1[0]1[0]0] 11 [1[0]1[0]0]1[0]1[0]0

Two “generations“ or recursive steps later we obtain :

```
11 11 11 11 [ 11 11 [ 11 1[0]1[0]0 ] 11 [ 1[0]1[0]0 ] 1[0]1[0]0 ] 1111 [ 11 [ 1[0]1[0]0 ] 11 [
1[0]1[0]0 ]1[0]1[0]0 ] 11[ 1[0]1[0]0 ] 11 [ 1[0]1[0]0 ] 1[0]1[0]0 ] 11 11 11 11 [ 11 11 [11 1[0]1[0]0 ]
11 [ 1[0]1[0]0 ] 1[0]1[0]0 ] 1111 [ 11 [1[0]1[0]0 ] 11 [ 1[0]1[0]0 ]1[0]1[0]0 ] 11 [ 1[0]1[0]0 ] 11 [
1[0]1[0]0 ] 1[0]1[0]0 ] 11 11 [ 11 [ 1[0]1[0]0 ] 11 [1[0]1[0]0 ] 1[0]1[0]0 ] 11 11 [ 11 [1[0]1[0]0 ] 11 [
1[0]1[0]0 ] 1[0]1[0]0 ] 11[ 1[0]1[0]0 ] 11 [ 1[0]1[0]0 ] 1[0]1[0]0 ]
```

The “trick“ consists here in replacing the brackets by specific 3D operations – typically affine transformations, such as rotations or translations - and the “0“ and “1“ occurrences by 3D pre-defined objects. We notice how the transformations and object creations are invoked in the following source code (obviously part of the main program, implemented within a “switch“ JAVA object) The resulting output sourcecode is VRML 97, mimed with a CosmoPlayer© plug-in.

```
case '1'      :

    devX=devX+0.5/propag;

    writeln(' Transform \{ rotation 0 1 0 '+-k+' translation '+devX+' 0 0
children \[ Transform \{ rotation 0 1 0 '+k+' children Shape \{ appearance
Appearance \{ material Material \{ diffuseColor 1 1 1 emissiveColor 0.3
0.3 0.3 \}\} geometry Extrusion \{convex FALSE crossSection \[-0.5 -0.5 , -
0.5 0.5 , 0.5 0.5 , 0.5 -0.5 , -0.5 -0.5 \] spine \[ 0 '+alt+' 0 , 0 0.1 0
, 0 0.1 0 , 0 0.15 0 , 0 0.15 0 , 0 0.2 0 , 0 0.2 0\] scale \[ 1.5 1.5 ,
1.4 1.4 , 1.3 1.3 , 1.3 1.3 , 1.25 1.25 , 1.25 1.25 , 1.2 1.2 \]
\}\}\}\}\}\}');

    nbobj=nbobj+1;
```

Generative Art 2002

```
break

case '0'      :

    devY=devY+0.5/propag;

writeln(' Transform \{ rotation 0 1 0 '+-k+' translation 0 0 '+devY+'
children \[ Transform \{ rotation 0 1 0 '+k+' children Shape \{ appearance
Appearance \{ material Material \{ diffuseColor 1 1 1 emissiveColor 0.3
0.3 0.3 \}\} geometry Extrusion \{convex FALSE crossSection \[-0.5 -0.5 , -
0.5 0.5 , 0.5 0.5 , 0.5 -0.5 , -0.5 -0.5 \] spine \[ 0 '+alt+' 0 , 0 0.1 0
, 0 0.1 0 , 0 0.7 0 , 0 0.8 0 , 0 0.8 0 , 0 0.9 0\] scale \[ 1.5 1.5 , 1.4
1.4 , 1 1 , 1 1 , 1 1 , 1.05 1.05 , 1.05 1.05 \]\}\}\}\}\}\}');

    nbobj=nbobj+1;

break

case '['      :

    ouvcrochet=ouvcrochet+1;

    alt=alt-0.2;

writeln(' Transform \{ rotation 0 1 0 '+-k+' translation '+devX+' 0.2
'+devY+' children \[ Transform \{ rotation 0 1 0 0 children \[');

writeln(' Transform \{ rotation 0 1 0 '+k+' translation '+devX+' 0 '+devY+'
children \[ Transform \{ rotation 0 1 0 '+k+' children Shape \{ appearance
Appearance \{ material Material \{ diffuseColor 1 1 1 emissiveColor 0.3
0.3 0.3 \}\} geometry Extrusion \{convex FALSE crossSection \[-0.5 -0.5 , -
0.5 0.5 , 0.5 0.5 , 0.5 -0.5 , -0.5 -0.5 \] spine \[ 0 '+alt+' 0 , 0 0.45 0
, 0 0.55 0 \] scale \[ 1 1 , 1 1 , 1.05 1.05 \]\}\}\}\}\}\}');

    nbobj=nbobj+1;

break

case ']'      :

writeln(' Transform \{ rotation 0 1 0 '+-k+' translation '+devX+' 0
'+devY+' children \[ Transform \{ rotation 0 1 0 '+-k+' children Shape \{
appearance Appearance \{ material Material \{ diffuseColor 1 1 1
emissiveColor 0.3 0.3 0.3 \}\} geometry Extrusion \{convex FALSE
crossSection \[-0.5 -0.5 , -0.5 0.5 , 0.5 0.5 , 0.5 -0.5 , -0.5 -0.5 \]
spine \[ 0 '+alt+' 0 , 0 0.05 0 , 0 0.05 0 , 0 0.4 0 , 0 0.4 0 , 0 0.5 0 \]
scale \[ 1.2 1.2 , 1 1 , .95 .95 , .95 .95 , 1 1 , 1 1\] \}\}\}\}\}\}');

writeln(' \]\}\}\}\}\}');

    ouvcrochet=ouvcrochet-1;
```

```

alt=alt+0.2;

nbobj=nbobj+1;

break

```

Depending on initial rules, such a model can quickly “run out of control“ and generate huge 3D databases. It’s specific initial generative inputs are the only condition for the whole evolution process – which is meanwhile eminently determinist; nevertheless, geometry partial overlaps are frequent and due to concatenated affine transformations previously described. Hereby we show a four-steps generated VRML model, made of solely 2 architectural primitives. Some extra visual artefact is provided by the height change of the objects, depending on their distance to the first geometric settlement.



Figure 5. A L-System-based growth engine.

Most of these generative models are developed within a web browser interface : a javascript code which dynamically generates a VRML source displayed by a CosmoPlayer plugin. We are studying by now other geometrical algorithms, in order to constrain these L-system, such as Voronoï diagrams or Delaunay triangulations.

3.3 Random or pseudo-random “hull-filler“

Random or pseudo-random “hull-filling“ generators for single-building construction.

The “hull-filling“ model offers by itself rather interesting investigative perspectives : in this model the specific positioning of architectural types or sub-types could be guided by a prior analysis that tends to break down or disassemble some historically-contexted architectural types by a morphological factorization. The process is obviously reversible and could be achieved by a rules-based grammar. The amazing Palladio 1.0 Macintosh© Hypercard Stack [2] is a noteworthy example of such a morphological synthesis. We also must here quote the scientific goal of the research team “Laboratoire d’Analyse des Formes“ from the architecture

school of Lyon that leads somehow this specific aspect of this research task.[3] Their aim is to identify major stylistics guidelines from distinct architectural families, dispatching them through pre-identified morphologic, functional, architectonic and compositional occurrences [4]. A similar search will soon commence, leaning on Claude-Nicolas Ledoux architectural production, whose factorizable characteristics appear as an evidence.

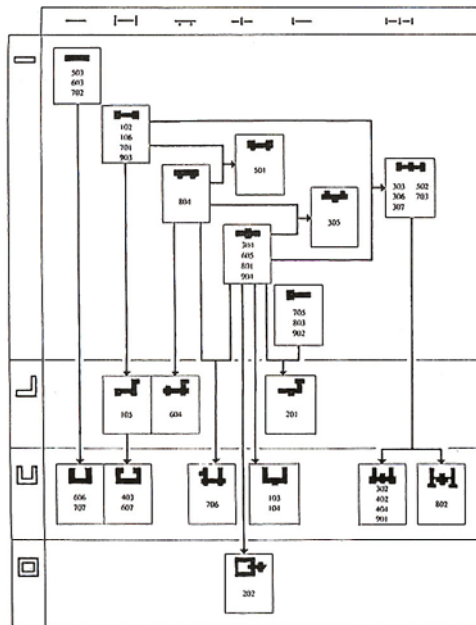


Figure 6. A graph-based morphological parser.

Courtesy of “Laboratoire d’Analyse des Formes”

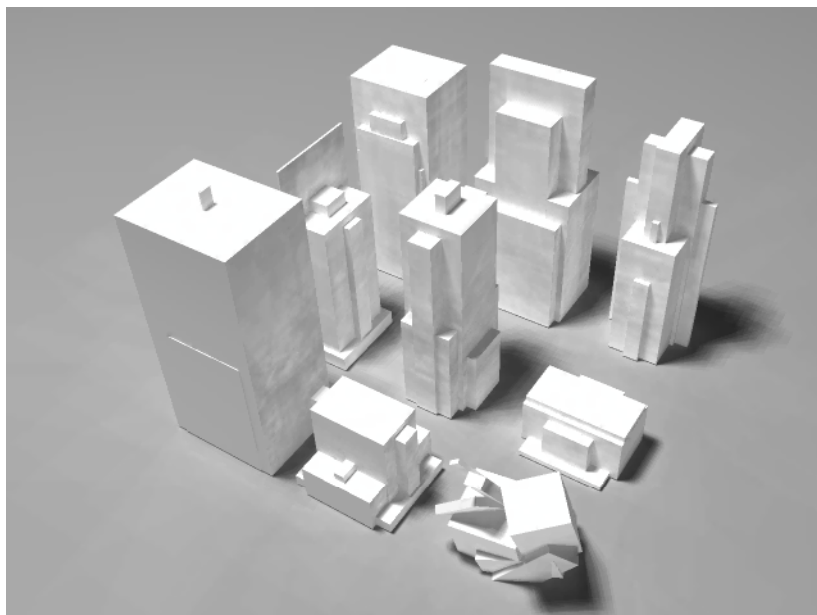


Figure 7. Some “hull-filled“ objects.

At the moment, this complex formalism is barely drafted; it is therefore interesting to point out the relevant difference of the “ugly duckling“ bottom right object, that descends from the same construction formalism but differs from 1 single input attribute.

3.4 Random 2D – 3D generators

Random or pseudo-random simple pattern generators applied to facades, according to buildings height or local floors indentations.

This very first applicative experiment was only acquired to test some early combinational conjectures. Some 3D “hull-filled“ objects are textured with simple combinational patterns ensuring somehow an intrinsic global coherence in order to avoid 2D and 3D possible mismatch. This could be achieved by establishing for instance a common spatial framework, arbitrarily bounded here by 2,5 meters-sided cubes. As shown in the picture below, the intrinsic coherence of the texture itself depends on the pertinence of single texture patches positioning, known as inner, top, left, right and bottom occurrences : on the illustration, the gray-filled board zone invoke specific ledge-type instances as the inner white zones use generic tiles. Right underneath, some texture patches that come with the 2D library and below, two facade variants.

These examples are here intended as “ironic standalone designs“ : the (im)pertinence of these random objects is obvious. Meanwhile, if coupled with accurately-sized 3D objects, the visual impression could be effective, as shown on Figure 6 .

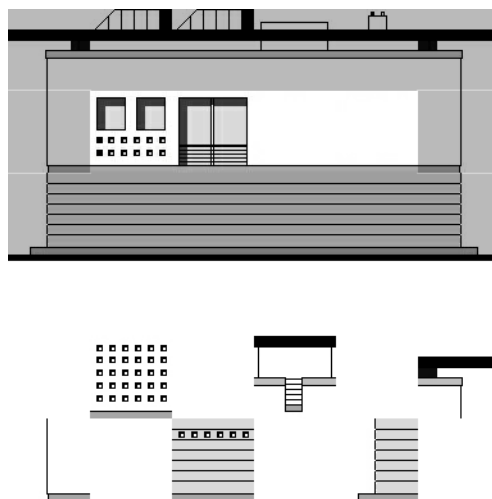


Figure 8. The automatic facade builder :o)

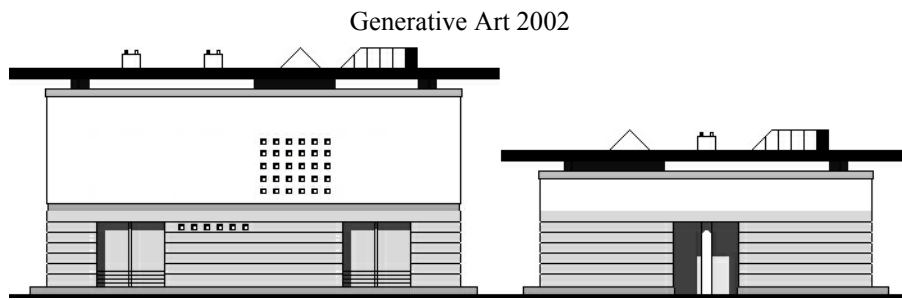


Figure 9. Some “automatic“ facades.

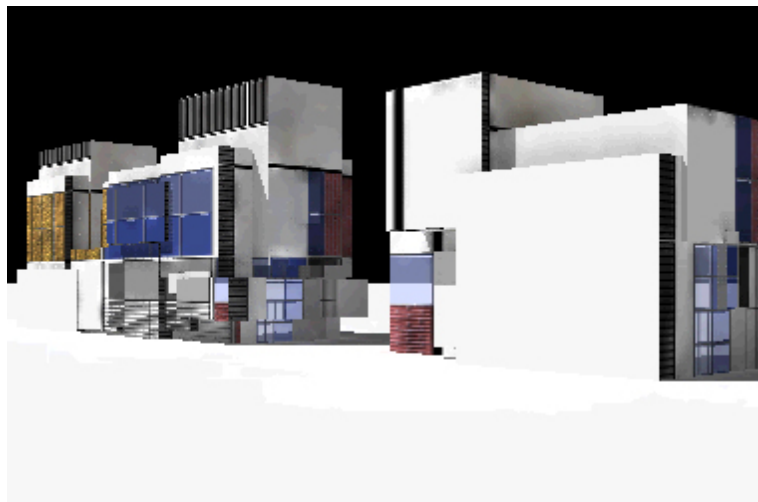


Figure 10. Applying and rendering colored tiles on specific 3D objects.

4. Conclusion

Virtual reality hardware and software costs and means are still relevant today. Trying to partially solve this peculiar aspect of leading 3D rendering techniques is part of the regional DEREVE project, whose aim is to build a convergent know-how, trying to extend hardware and software intrinsic performances through methodological and algorithmic applications, in terms of modeling and rendering. As a matter of fact, the specific involvement of the “MAP-aria“ lab in this research task deals with 3D scenes building, leaning on his specific architectonic culture and virtual reality previous experimentations.

5. References

5.1 Related bibliography

[1] **Lindenmayer, A.** “Mathematical models for cellular interactions in development“, parts I-II. *Journal of Theoretical Biology* 18: 280-315, 1968.

[2] **Freedman, R.** “Palladio 1.0“, Apple Macintosh© Hypercard Stack, 1990.

[3] **Paulin, M – Duprat, B.** “De la maison à l’école, élaboration d’une architecture scolaire à Lyon de 1875 à 1914“, Ministère de la Culture, Direction du Patrimoine, CRML, 1991.

[4] **Ben Saci, A.** “Théorie et modèles de la morphose“, Thèse de la faculté de philosophie sous la direction de B. Deloche, université Jean Moulin, 2000.

Pierre Frankhauser “L’approche fractale : un nouvel outil de réflexion dans l’analyse spatiale des agglomérations urbaines “, Université de Franche-Comté, Besançon, 1997.

Pierre Frankhauser *La Fractalité des Structures Urbaines*, Collection Villes, Anthropos, Paris, France, 1994.

Catherine Khamphang Bounsaythip “Algorithmes évolutionnistes“ in “*Heuristic and Evolutionary Algorithms: Application to Irregular Shape Placement Problem*“ Thèse - Public defense: October 9, 1998 (NO: 2336)

Jean-Claude Heudin “L’évolution au bord du chaos“ Hermès Editions 1998

Bryan Horling “Implementation of a context-sensitive Lindenmayer-System modeler“ Department of Engineering and Computer Science and Department of Biology, Trinity College, Hartford, CT 06106-3100, USA 1996

Sikora S., Steinberg D., Lattaud C., Fournier C., Andrieu B. – 1999- “Plant growth simulation in virtual worlds : towards online artificial ecosystems. Workshop on Artificial life integration in virtual environments“. European Conference on Artificial Life (ECAL’99), Lausanne (Switzerland), 13-17 september 1999.

Barber, C.B., Dobkin, D.P., and Huhdanpaa, H.T., "The Quickhull algorithm for convex hulls," *ACM Trans. on Mathematical Software*, Dec 1996.

Batty M., Longley P.A., “*Fractal Cities: A Geometry of Form and Function*“, Academic Press, London and San Diego, CA, 1994

Paul Torrens, “How cellular models of urban systems work” , CASA, 2000.

5.1 Related webgraphy

<http://www.geom.umn.edu/locate/qhull>

<http://www.ctpm.uq.edu.au/virtualplants/ipivp.html>

<http://www.cpsc.ucalgary.ca/projects/bmv/software.html>

<http://www.cs.hope.edu/~algnim/ccaa/algo.html>

www.casa.ucl.ac.uk