Procedural Modeling as an Analytical Tool for 3D Survey in Urban Design Assessment

Ghorbanian, M. [1], *Shariatpour, F. [2]

[1]*Assistant professor, School of Architecture & Environmental Design, Department of Urban Planning and Design, Iran University of Science & Technology (IUST), Tehran, Iran.*

[2]*M.A. Student of Urban Design, School of Architecture & Environmental Design, Department of Urban Planning and Design, Iran University of Science & Technology (IUST), Tehran, Iran.*

Abstract:

This paper examine procedural modeling as a tool for 3D modeling creation. Procedural modeling historically has been used for 3D visualization of natural features, but with the release of the software CityEngine in 2008 the technology can easily be adopted also in problem domains dealing with urban environments.

Then, we will examine and compare two types of modelling, traditional and procedural modeling and consider the advantages of procedural modeling. furthermore, reviewed the features of CityEngine software and introduced CityEngine as a software application that uses the procedural modeling system.

Procedural modeling deals with (semi-)automatic content generation by means of a program or procedure. We survey procedural methods that are useful to designe roads, buildings, and entire cities. In this survey, we focus particularly on the degree of the use of procedural modeling in the design of cities, streets, buildings, etc, in by CityEngine software. This paper examines how architectural shape grammars can be used to procedurally generate 3D modelling.

In the end, considering the advantages of the procedural modeling, we propose a procedural modeling (Algorithmically) to replace the manual modeling (Traditional methods) to increase the accuracy, speed, and design efficiency, furthermore, flexibility of this approach helps to quickly change and regenerate the model as new model.

Keywords*:* Procedural, 3D Modeling, Procedural Modeling,CityEngine**.**

E-mail addresses:
ghorbanian@iust.ac.ir (Ghorbanian, M)
farshad_shariatpour@arch.iust.ac.ir (Shariatpour, F)
*Corresponding Author

1. INTRODUCTION

In recent years, the field of Procedural Modeling (PM) has been receiving increased attention in the research community. Many new papers are being published yearly, and we felt that a comprehensive overview was missing. We therefore survey PM methods for generating both natural and man-made structures that can be found in city [26].

The creation of compelling models is a crucial task in the development of successful movies and computer games. However, modeling large three-dimensional environments, such as cities, is a very expensive process and can require several man years worth of labor. In this paper we will employ procedural modeling using shape grammars capable of efficiently creating large cities with high geometric detail and up to a billion polygons. It would be extremely time consuming to replicate these results with existing modeling software [21].

2. LITERATURE REVIEW

2.1. Procedural

Procedural generation techniques in digital graphics have been applied since at least the early 1980s. Due to the all-encompassing nature of the term, the history of the field is difficult if not impossible to summarize to a single narrative. Some generalizations about the development of the subject through the decades however can be made. The theoretical foundations for many of the techniques applied later were laid out in the 1960s and 1970s. This took place for example in the work of the Hungarian biologist Aristid Lindenmayer who developed a formal language called L-system to model plant growth (1968), the shape grammars developed by Stiny and Gips to generate art work (1971), and fractals as presented by Benoît Mandelbrot in his book (1977): "Fractals: Form, Chance and Dimension" [31].

A procedural language is a type of computer programming language that specifies a series of well-structured steps and procedures within its programming context to compose a program. It contains a systematic order of statements, functions and commands to complete a computational task or program. A procedural language, as the name implies, relies on predefined and well-organized procedures, functions or sub-routines in a program's architecture by specifying all the steps that the computer must take to reach a desired state or output (https://www.techopedia.com).

2.2. 3D Modelling

Modeling is the first and the most critical pillar of 3D graphics. It can be defined as "the process of creating a 3D model in the computer" [12].The process includes three main steps. The first step is a 3D data acquisition, the next step is modeling and the last step is rendering. In the data acquisition step, data about object's size and depth are collected. There are two main methods of data acquisition – contact (for example, measurements using tape) and non-contact (for example, laser scanning and measurements taken with total stations). The modeling step includes refinement of the initial shape, for example, surface construction [4].

Nowadays, 3D Models are used in a large range of exciting applications areas: Animation, Archaeology, Architecture, Dentistry, Education, Fashion and Textiles, Foot Wear, Forensics, Games, Industrial Design, Manufacturing, Medical, Movies, Multimedia, Museums, As-built Plants Rapid Prototyping, Reverse Engineering, Sculpture, Toys, Mold Making, and Web Design. 3D modeling has become a key technology in many applications [52].

2.3. Procedural Modeling

The term ''procedural modeling'' is a concept of computer graphics which produces information objects (entities) which have let obtain access to their creation as a sequence of instructions [25].

Procedural modeling (PM) has been an active research topic for over thirty years and it is applied to a wide variety of areas such as modeling of textures, plants, terrain, buildings, urban areas, road networks, rivers, or art creation. There is not a single definition of procedural modeling; it encompasses a wide variety of generative techniques that can (semi-)automatically produce a specific type of content based on a set of input parameters. A widely accepted generic definition is that PM provides a content by means of a procedure or a program [5]. Procedural modeling relates to many different areas, the closest one being computational simulation. Many procedural models are essentially generative representations either of processes inspired by nature, such as plant development, or of man-centered processes, such as urban simulations. PM also relates to physics-based simulations [26].

Procedural modeling is a popular technique that can simplify the description of complex geometric structures by encoding their structures into a set of procedural rules. Execution of the procedural rules

then creates the geometric representation of the encoded structure. Although procedural models can be used to describe any shape, they are especially useful when the encoded structure contains some repeated or recursive components that can be represented by a few simple rules [27].

Probably the first formal definition of a procedural model was introduced by Lindenmayer (1968), who described a parallel rewriting grammar that was used to generate branching cellular structures. This grammar, called the Lindenmayer system (L-system), was brought into computer graphics by Prusinkiewicz and Lindenmayer (1990) who presented a geometric interpretation of symbols from the Lsystem alphabet. The authors demonstrated that L-systems are especially useful for encoding linear branching structures, such as plants, and they proposed several procedural models based on L-systems to show the algorithmic nature of many 8 plant species. Meanwhile, procedural modeling became popular in other areas of computer graphics, such as noise generation [5], fractals [19], and texture synthesis [6] [5] [27].

Recently, urban modeling became another field where procedural modeling is the prominent content-authoring technique. New procedural models were developed for automatic modeling of street layouts [22], city blocks [30], individual buildings [21], and building facades [22]. Because of the varying nature of many of these new procedural problems, several new formal definitions for the representation of procedural models were introduced. Many works from urban modeling are based on a concept of recursive shape replacement that was introduced by Stiny and Gips (1971) in their work about shape grammars. Shape grammars were later extended to split grammars [32] and CG architecture (CGA) grammars [21] that are more suitable for modeling of facades and buildings, respectively [27].

2.4. Types of Modeling

In general, two types of modeling (Manuall modeling & Procedural modeling) are used. The following is a brief explanation of each and then their comparison and the study of the advantages of procedural modeling compared to traditional modeling.

Traditional 3D modeling tools often require too much manual work and their application is therefore overly expensive for archaeological projects. In contrast, our approach proves to be efficient and fairly simple. Furthermore, our procedural modeling approach allows for the testing of several hypotheses by adjusting some of the parameters [21].

A powerful solution to large-scale urban modeling is the use of procedural techniques [21]. The idea of modeling urban environments using shape grammars was recently explored by Parish and Müller [23] and [32]: On the one hand, Parish and Müller showed how to generate large urban environments where each building consists of simple mass models and shader for facade detail. On the other hand, [32] demonstrated how to generate geometric details on facades of individual buildings. Ideally, we would like to combine these two ideas to generate large and detailed urban environments [21].

The following table indicates procedural modeling compared to traditional modeling.

Table 1: Comparison of procedural and traditional modeling.

Procedural modeling (Algorithmically)	**Traditional methods/ Classic manual modeling**
▪ Supports the creation of detailed large-scale 3D city models [8]. ▪ Data Amplification Capability: Using a simple set of input parameters or a few generation to yield a wide variety of models [29].	▪ In classic manual modeling method for large-scale modeling is duplicate and hard.

▪ Data compression: a rather complex geometric model can be represented by a compact procedural model and a set of parameters, while the actual geometry is generated only when needed [26]. ▪ Easy creation of multiple 3D models [24]. ▪ Potential to drastically reduce the amount of modeling effort required to create digital content: because its methods are often stochastic, PM can create a variety of results from one set of input parameters [26]. ▪ The modeling process faster and randomness can be controlled. It is easy to update the rule files and they are re-usable for other projects because the process of creation has been recorded [4]. ▪ The benefits of PM make it particularly attractive for creating virtual environments. Virtual worlds are important for many applications, including a wide variety of (serious) games and simulations [26].	▪ The traditional modeling does not have the ability to upgrade the data and parameters due to the lack of a programming language. ▪ The traditional modeling process is tedious and repetitive, and the ability to use the model parameters in another model does not exist, which makes the modeling a lot of time and costly. ▪ The traditional modeling process does not have the ability to create multiple and simultaneous models. ▪ In the traditional modeling of data and random parameters is meaningless because this type of modeling does not have the ability to use written language. In practice, this kind of method is non-flexible and does not have the ability to control and change immediately. This makes the process of modeling take a lot of time.

3. CityEngine as a Procedural Modeling

CityEngine has established itself well within the gaming industry since its commercial release in 2008. The software has been used extensively to create highly detailed 3d model of fictional cities and urban landscapes. CityEngine has the option of importing various forms of geographical and architectural datasets such as GIS and CAD, its possibilities of modeling real landscape are quite promising [7].

The "CityEngine" software, a procedural modeling solution from Procedural, Inc. in Zurich, Switzerland, employs a shape-grammar-based geometry generation system called "CGA shape grammar"[1] to efficiently create large scale 3D environments within defined rules and parameter ranges. The scripting language used by CityEngine is an enhancement of the set and shape grammar syntax developed in the last decades and is optimized for architectural content. It makes it possible to control or vary volumes, architectural assets, proportions, rhythms, and materials [4].

CityEngine uses a procedural modeling approach to automatically generate models through a predefined rule set. The rules are defined through a CGA shape grammar system enabling the creation of complex models. Users can change or add the shape grammar as much as needed providing room for new designs. CGA Shape Grammar system can read Esri-Oracle format datasets directly, and it operates as a top-bottom generation tree: it generates complex components from simple Shapefiles polygons/poly-lines/points whereas each branch and leaf of the generation tree cannot interact with others. It is different than main-stream shape grammars like Grasshopper in Rhinoceros 3D and Dynamo in Autodesk Revit [9].

[1] *CGA Shape* is a grammar suitable for architectural design. In this subsection we will give a short introduction to CGA Shape necessary to encode the designs in the previous section. A more comprehensive description of CGA Shape is given in.

Fig.1*:* This figure shows the application of CGA shape, a novel shape grammar for the procedural modeling of computer graphics architecture. First, the grammar generates procedural variations of the building mass model using volumetric shapes and then proceeds to create facade detail consistent with the mass model. Context sensitive rules ensure that entities like windows or doors do not intersect with other walls, that doors give out on terraces or the street level, that terraces are bounded by railings, etc [21].

4. The CityEngine Features

4.1. Procedural Modeling Core

The main concept of CityEngine is the "procedural" approach towards modeling efficiently. The computer is given a codebased "procedure" which represents a series of commands - in this context geometric modeling commands - which then will be executed. Instead of the "classical" intervention of the user, who manually interacts with the model and models 3d geometries, the task is described "abstractly", in a rule file. The commands which are provided in CityEngine's CGA shape grammar, such as "extrude", "split" or "texture" are widely known commands in most 3d applications and thus any user can adapt them easily and create complex architectural forms in short time [11].

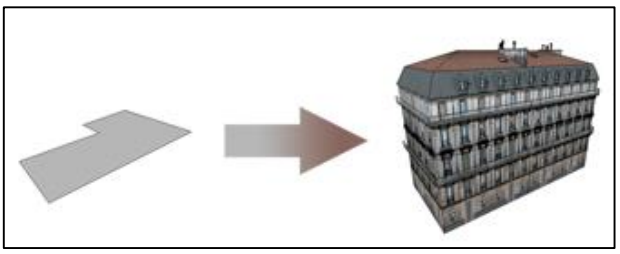

Fig.2: *Procedural Modeling Core [8].*

CityEngine features three distinct PM systems:

- Automatic generation of street networks using a tool called “Street Growth Wizard”
- Algorithms to subdivide blocks into individual building lots
- CGA shape grammar to create 3D-content based on the created street and lot shapes

The automatic generation of street networks is based on an extended L-system as described by Paris & [23]. The system differs from the discussed L-systems in a number of ways to account for the different topological nature of street networks as compared to the branching structures for which L-systems originally were developed. The street generation algorithm iterates a basic three step process to create the resultant network. At first the L-system produces an initial output of branching streets [31].

4.2. Dynamic City Layouts

The Dynamic City Layouts give the user a powerful tool to create interactive street networks which automatically update in realtime. Streets, Sidewalks and whole Blocks, which form the specific urban context adapt efficiently to the user's input and give the user an intuitive way to design the layout of complete cities. And of course, all the Geometries which depend on the layout of the underlying Dynamic City Layout are also updated on the fly! Watch your buildings getting rebuilt while you edit the width of the surrounding streets [9].

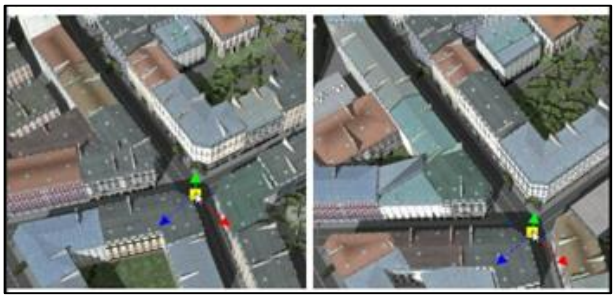

Fig.3: *Dynamic City Layouts [9].*

5. The CityEngine Modeling Pipeline

CityEngine, a software that provides a design framework for geographic data, creates full 3-D models of cities around the world. CityEngine combines the geographical information of a city with other crucial data points, like zoning law restrictions and the location of water pipes. With a full replicated 3-D city model, urban planners and architects can build better new structures and avoid potential problems before they occur. The main layer of information in these city models is the geographic information system maps. GIS lets people in any industry better visualize what's really happening. From urban planning to international development, GIS has the potential to help a number of disciplines, and it's all based on PCs [16].

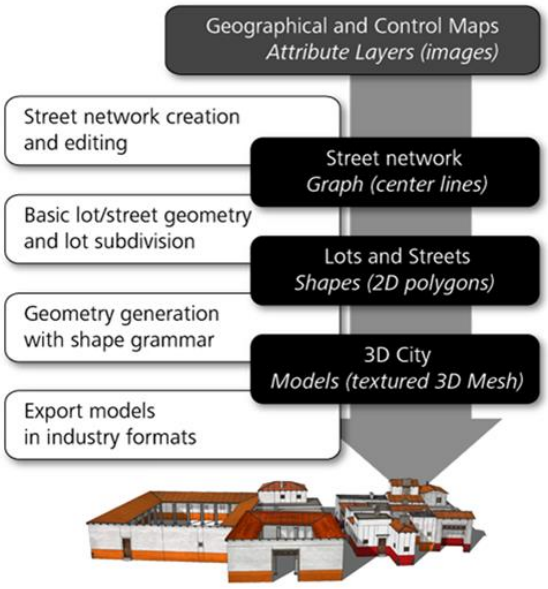

Fig.4:*The CityEngine Workflow [9].*

Typically, in the first step, the street network is created, afterwards the resulting blocks are subdivided into lots. Finally, the 3D models of the buildings are generated using the CGA rules. The output of CityEngine is polygonal building models. a CityEngine scene is stored as layers of different data types representing the different stages. The pipeline is flexible and can be entered at different stages. For example, street blocks or building masses can be imported and further processed [9].

6. Generating Large-Scale Urban Layouts

Modeling a city poses a number of problems to computer graphics. Every urban area has a transportation network that follows population and environmental influences, and often a superimposed pattern plan. The buildings appearances follow historical, aesthetic and statutory rules. To create a virtual city, a roadmap has to be designed and a large number of buildings need to be generated. CityEngine is a system using a procedural approach based on L-systems to model cities. From various image maps given as input, such as land-water boundaries and population density, our system generates a system of

highways and streets, divides the land into lots, and creates the appropriate geometry for the buildings on the respective allotments [23].

Fig.5: *Generating Large-Scale Urban Layouts with CityEngine.*

7. Grammar-based Modeling

"Grammar-based" or "procedural" modeling has a widespread range of applications, but mostly it is applied, when large numbers of iterations of a design or large numbers of objects have to be created which obey certain standardized rules [9].

Procedural modeling is a 3D model creation process using rules and algorithms. It consists of a base geometry, for example, building footprints with the information about building heights and roof types, and procedural rules. According to [21], modeling of large size 3D environments requires a lot of financial resources and several man years' worth of labor. For this reason in creation of a large sized city model saving time and money is important. To fulfill these requirements, procedural modeling is one of the solutions. As the figure 5 shows, there is a definite cost and time saving benefit for using the procedural modeling when a lot of 3D modeling or design iterations are needed [4].

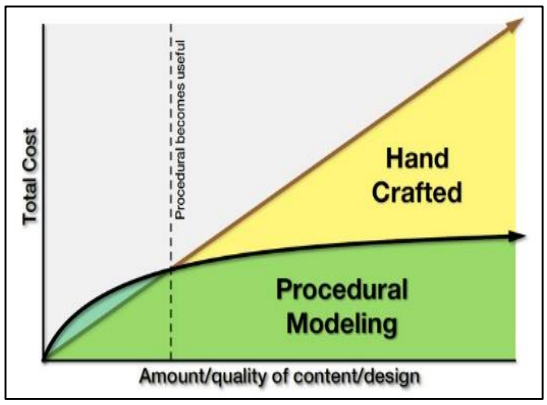

Fig.6: Efficiency of a procedural modeling in comparison with a manual modeling [24].

7.1. CGA-language

CGA is a language loosely based on an L-system data structure which instead of symbols operates on shapes to describe their transformations using operations written according to the CGA language specification. The CGA scripts are saved as plain text files with .cga extension. The initial shape from which the generative production begins is external to the

language and needs to be produced using the modeling capabilities of CityEngine or any other modeling software and later imported to CE [31].

The rule files in CityEngine are user-written descriptions of how two-dimensional shapes should be converted into three dimensional models. Rules are stored in CGA, or computer generated architecture, files and every shape in the scene (that will be used) should have an associated CGA file. CGA is not a programming language in the usual sense it can be thought of as describing how a single shape is replaced by one or more shapes and/or models (the mapping of a single shape is a rule). For procedural construction of buildings/roads, this approach is very effective since branching from one rule to the next can be randomized and/or driven by the scene context [13].

Fig. 7: *Principle of procedural modeling [25].*

CityEngine works based on the formalism of the L-Systems. The initial geometry and the axiom ω are attributed geometric rule sets which change the intial state via parameter or contants. Axioms containing polygons or points are designated as initial shapes and are the start elements for all production rules. The production rules are described as CGA rule files and are stored as ASCII-format. CGA stands for Computer Generated Architecture and generates as a shape grammar with different production rules detailed 3D objects out of simple geometric outlines [25].

The generation of building geometries with the CGA shape grammar works as follows:

1. The building lots are either created by CityEngine with the above mentioned tools or imported. Instead of polygons, mass models (building envelopes) can also be the starting point.

2. The user selects which rule file (.cga) she or he wishes to apply onto these shapes. The user can either assign one rule to all buildings, or assign rule sets on a per-building basis.

3. Then, the user can trigger the application of the rules on the selected shapes. Therefore it is important that the *Start Rule* of the shape is present in the rule file. Otherwise no rule can be invoked. The generated models can then be explored in the 3D viewer of CityEngine. In the case of very large models, it is not recommended to generate all buildings in CityEngine due to memory constraints.

4. In order to edit the resulting 3D models, different possibilities exist: (1) edit the rules, (2) overwrite the rule parameters of a rule set on a per-building basis, and (3) if stochastic rules are used (rules with random parameters), the random seed of all or single buildings can be altered.

5. After the design is finalized, the user can export the selected buildings or streets to the hard disk (including textures). Note that there are no memory constraints in the exporting mode [9].

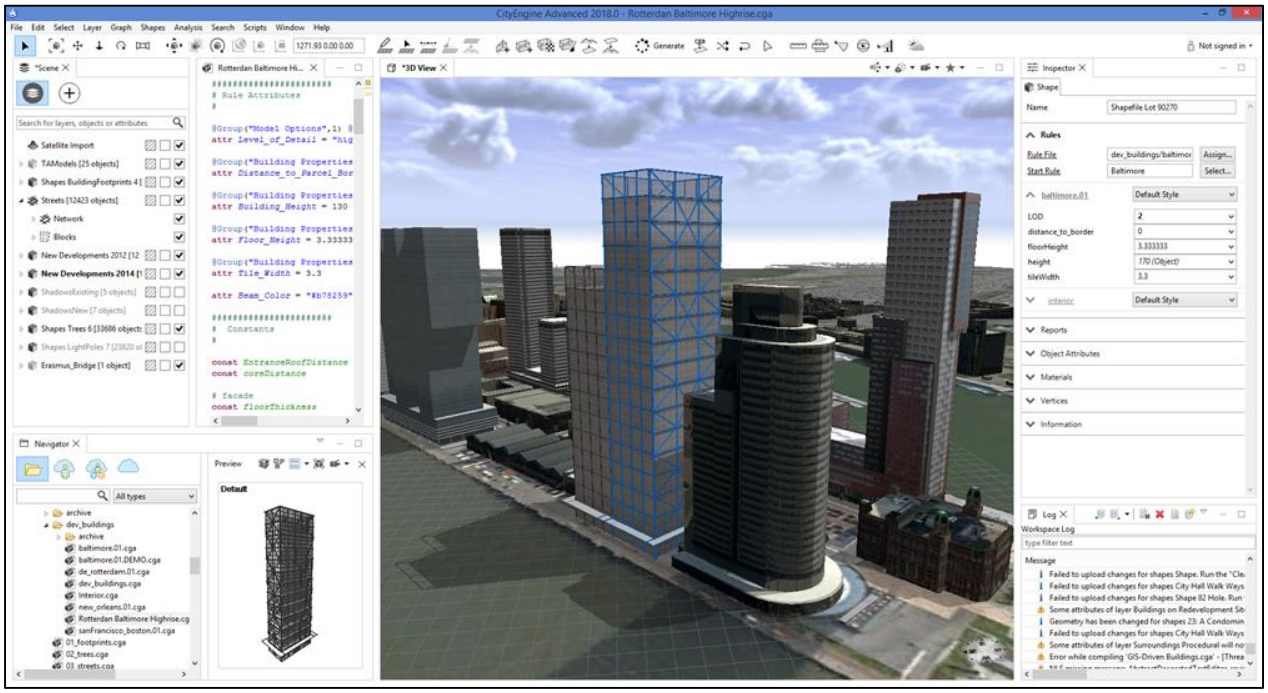

Fig. 8: *generation of building geometries with the CGA shape grammar [9].*

8. Conclusion

Procedural modeling is one of the most appropriate solutions for creating large size 3D city models in a short time and with low expenses because, compared to manual modeling, 3D geometries and textures are constructed using rules and algorithms. It allows the modeler to make changes in the 3D content more easily, and already created rules can be re-used for other projects. The modeling process consisted of data collection, preparation, import into the software, model creation and export for the model created.

The CityEngine software perform well in the generation of the model itself this software is a powerful for the creation of realistic 3d models from 2d data.

As mentioned above, CityEngine uses procedural language in modeling, which is called CGA-language in CityEngine software. CGA a hierarchy of programming language, which, due to the speed of change and high flexibility, generates multiple models at a time. This programming language is designed to increase design speed, flexibility in the model. Below is a sample of the model designed by the procedural language in the CityEngine software. This model is written with the CGA language that, as it is seen, the model can change at the same time.

Fig. 9: *Procedural Modeling with CityEngine.*

REFERENCES

[1] Bolognesi, M., Furini, A., Russo, V., Pellegrinelli, A., and Russo, P. “Accuracy of Cultural Heritage 3D Models by RPAS and Terrestrial Photogrammetry.” International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences — ISPRS Archives 40, no 5(2014):19-113.

[2] Chen, G., Esch, G., Wonka, P., Pascal, & Zhang, M. E. (2007). Interactive procedural street modeling. ACM Trans. Graph., 27(3), 35.

[3] De Reu, J., G. Plets, P. Verhoeven, M. De Smedt, B. Bats, W. Cherretté, D. De Maeyer, Deconynck, Herremans, Laloo, Van

Meirvenne, and De Clercq“ .Towards a Three-dimensional Cost-effective Registration of the Archaeological Heritage.” Journal of Archaeological Science , 2012: 1108–1121.

[4] Dobraja, I. (2015). Procedural 3D modeling and visualization of geotypical Bavarian rural buildings in Esri CityEngine software. Munich Technical University. 8-9.

[5] Dylla, K., Frischer, B., Mueller, P., Ulmer, A. & Haegler, S., 2008. Rome Reborn 2.0: A Case Study of Virtual City Reconstruction Using Procedural Modeling Techniques. Computer Graphics World, 16 (25), 63.

[6]Ebert,D.S.,Musgrave,F.K.,Peachey,D.,Perlin,K.,&Worle y,S.(2002). Texturing and modeling: A procedural approach (3rd ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.15-23

[7] Ebert, D. S, Worley. S, Musgrave. F. K, Peachey. D, ,Perlin K.: Texturing & Modeling, a Procedural Approach,3rd ed. Elsevier, 2003.5-8

[8] Edvardsson, K.N, 2013. 3d GIS modeling using ESRI´s CityEngine. The University Jaume I. 8.

[9] Esri,2014. ArcGIS Resources. 3D urban content creation and design using CityEngine. Available at: http://resources.arcgis.com/en/communities/cityengine/01w900000000008000000.html.

[10] Esri,2014.CityEngine Help. Available at: http://cehelp.esri.com/help/index.jsp [Accessed August 26, 2014].

[11] Esri, n.d.(b). ArcGIS Resources. What is CityEngine? Available at:http://resources.arcgis.com/en/communities/cityengine/01w90000000m000000.htm[Accessed October 23, 2014].

[12] Esri, 2012. ArcGIS Resources. Developing with Esri CityEngine. Available at: http://video.arcgis.com/watch/1147/developing-with-esri-cityengine [Accessed August 4,2014].

[13] Govil-Pai, S., 2004. Principles of Computer Graphics. In New York: Springer. 83.

[14] Goodenough ,A. Brown,S. CityEngine/SUMO Scene Construction Manual, 2008, Rochester Institute of Technology, p.11.

[15]https://cehelp.esri.com/help/topic/com.procedural.cityengine.help/html/toc.html.

[16]https://www.techopedia.com/definition/8982/procedural-language.p.1.

[17] Kaufman, E. This 3-D Software Could Change the Way We Plan Our Cities,12 November 2015. 1-23.

[18] Lindenmayer, A. (1968). Mathematical models for cellular interaction in development: Parts i and ii. Journal of Theoretical Biology, 18.

[19] Luan, X.-D. et al., 2008. Research and Development of 3D Modeling. IJCSNS International Journal of Computer Science and Network Security. 52

[20] Mandelbrot, B. (1983). The fractal geometry of nature. New York: W. H. Freeman and Comp.

[21] Müller, P . Vereenooghe, T . Wonka, P . Paap, I. Van Gool ,L, Procedural 3D Reconstruction of Puuc Buildings in Xkipché,2006.

[22] M¨uller, P., Wonka, P., Haegler, S., Ulmer, A., & Gool, L. V. (2006). Procedural modeling of buildings. In Siggraph ’06: Acm siggraph 2006 papers (pp. 614–623). New York, NY, USA: ACM Press.

[23] M¨uller, P., Zeng, G., Wonka, P., & Gool, L. V. (2007). Image-based procedural modeling of facades. In Siggraph ’07: Acm siggraph 2007 papers (p. 85). New York, NY, USA: ACM

[24] Parish, Y. I. H., & Müller, P. (2001). Procedural modeling of cities. In Siggraph ’01: Proceedings of the 28th annual conference on computer graphics and interactive techniques (pp. 301–308). ACM Press.

[25] Piccoli, CH. CityEngine for Archaeology, Mini conference: 3D GIS for mapping the Via Appia VU University Amsterdam , 19 April 2013. 14-15.

[26] Radies, C., 2013a. Procedural Random Generation of Building Models Based Geobasis Data and of the Urban Development with the Software CityEngine. In Bernburg, Germany, pp. 175– 184.

[27] Ruben ,M. Smelik, Tim Tutenel, Rafael Bidarra and Bedrich Benes, 2014, A Survey on Procedural Modeling for Virtual Worlds. 1-3.

[28] Stava, Ondrej, 2012. INVERSE PROCEDURAL MODELING OF TREES, Purdue University ,USA.7- 8.

[29] Stiny, G., & Gips, J. (1971). Shape grammars and the generative specification of paintingandsculpture. InSegmentationofbuildingsfor3dgeneralisation.in: Proceedings of the workshop on generalisation and multiple representation , leicester.

[30] Smith, A. R.: Plants, fractals, and formal languages. In SIGGRAPH ’84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques (New York, NY, USA, 1984), ACM Press, pp. 1–10.

[31] Vanegas, C. A., Aliaga, D. G., Beneˇs, B., & Waddell, P. A. (2009). Interactive design of urban spaces using geometrical and behavioral modeling. In Acm siggraph asia 2009 papers (pp. 111:1–111:10). New York, NY,

[32] Viinikka, J. (2014). Adopting Procedural Information Modeling in Urban Planning. Department of Real Estate, Planning and Geoinformatics, School of Engineering, Aalto University.pp.10-20.

[33] Wonka, P., Wimmer, M., Sillion, F., & Ribarsky, W. (2003). Instant architecture. ACM Trans. Graph., 22(3), 669–677.