Int. J. Architect. Eng. Urban Plan, 29(2):255-267, December 2019
DOI: 10.22068/ijaup.29.2.255

RESEARCH PAPER

General Architecture

Procedural modeling as a practical technique for 3D assessment in urban design via CityEngine

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

[1]*Assistant Professor, School of Architecture & Environmental Design, Iran University of Science & Technology (IUST), Tehran, Iran.*

[2]*MA. Student of Urban Design, School of Architecture & Environmental Design, Iran University of Science & Technology (IUST), Tehran, Iran.*

Received: May 2019, Revised: November 2019, Accepted: November 2019

Abstract

3D modeling and visualization in large-scale cities is a major difficulty for computer graphics. Studying photos of a big city reveals a variety of street patterns, buildings, shapes, and textures. This paper presents "procedural modeling" as a new approach to create three-dimensional modeling, especially for large-scale modeling. Procedural modeling deals with the production of semi-automatic (semi) content using a program or routine. This kind of modeling was historically utilized for 3D visualizing the natural properties, however, by releasing the CityEngine in 2008, the technology can easily be applied to urban environments as well. In this paper, this approach using a descriptive-analytical method will be examined, with particular emphasis on the extent to which procedural modeling is used in the design of cities, streets, buildings, etc., by CityEngine software. Therefore, while reviewing how architectural form grammars are used to generate 3D modeling procedures, two types of modeling including traditional and procedural modeling will be studied and compared and the benefits of procedural modeling will be revealed. After a description of the features of CityEngine, it will be introduced as a technique that follows procedural modeling. Finally, due to the advantages of procedural modeling, this type of modeling (algorithmic) is suggested as an alternative to manual modeling (traditional methods in urban design) to increase the accuracy, speed, and efficiency of the design and in addition, this will increase the flexibility of the design via this approach. In this regard, the workflow of the procedural modeling process will be provided using the CityEngine technique and a number of outputs of this technique that have been provided in the CGA will be presented.

Keywords: *3D modeling, Visualization, Procedural modeling, CityEngine software.*

1. INTRODUCTION

Urban designers need to create 3D models for all buildings within the design area; models that are dynamic and flexible and capable of rapid change that can create different scenarios at once. Various software applications have come into the scope of 3D modeling of building models, but none have been capable of fast and interactive modeling. In this way, procedural modeling techniques can be very efficient and useful for urban designers.

Recently, Procedural Modeling (PM) has attracted increasing attention in the research field. Numerous novel studies are published each year while missing a comprehensive overview seemingly. Hence, in the present work, we review PM approaches for creating both man-made and natural structures existing in the cities [4].

Creating the compelling models is a vital task in developing successful computer games and movies. Nevertheless, it is very expensive to model large 3D environments like cities and require numerous man-years' worth of work. In this work, we will use procedural modeling utilizing shape grammars able to proficiently create huge cities with great geometric detail and up to a billion polygons. Hence, replicating these outcomes with existing modeling software It would be extremely time-consuming to [24].

* *Corresponding author: Ghorbanian@iust.ac.ir*

This paper is extracted from the master thesis of the second author in Iran University of Science and Technology, entitled "Smart Community Design Technology Based on Procedural Modeling Algorithm Using CityEngine; Case Study: Narmak, Tehran" under supervision of the first author.

2. LITERATURE REVIEW

2.1. Procedural

Procedural techniques are code segments or algorithms that specify some characteristics of a computer-generated model or effect. For example, a procedural texture for a marble surface does not use a scanned-in image to define the color values. Instead, it uses algorithms and mathematical functions to determine the color [3].

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 artwork (1971), and fractals as presented by Benoît Mandelbrot in his book (1977): “Fractals: Form, Chance, and Dimension” [4- 5].

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 subroutines in a program’s architecture by specifying all the steps that the computer must take to reach a desired state or output [6].

2.2. 3D Modeling

For three-dimensional graphics, modeling is the first and the most serious foundation. It can be determined as “the procedure of generating a three dimensional model in the computer” [27] including three key phases. The first phase is three-dimensional data attainment, the next phase includes modeling and the last stage involves rendering. The data attainment step deals with collecting data regarding the size and depth of the object. Two main data acquisition approaches including contact (for instance, measurements via tape) and non-contact (for instance, laser scanning and measurements taken with overall stations). The modeling stage involves refining the primary shape, for instance, surface construction [8]. Increasing the supply of remotely sensed data regarding the three-dimensional environment also contributes to making more popular and feasible three-dimensional visualizations of the cities [9].

Today, three-dimensional models are extensively utilized in exciting applications areas such as Archaeology, Animation, Dentistry, Architecture, Fashion, Textiles, and Education, Foot Wear, Games, Forensics, Manufacturing, Industrial Design, Medical, Multimedia, Movies, As-built Plants Rapid Prototyping, Museums, Sculpture, Reverse Engineering, Mold Making, Toys, and Web Design. Three-dimensional modeling has become a key knowledge in numerous applications [10].

2.3. History of procedural modeling

Procedural content creation has been studied for over 30 years. This pattern was originally used to visualize natural content like landscapes and plants. The use of procedural methods in urban modeling is relatively new both practically and academically. Technical modeling of reproductive systems L-systems, split grammars and shape grammars have been developed, allowing for the construction of complex structures from small inputs sets [2].

L-systems are created as the basis for geometric plant modeling [11].

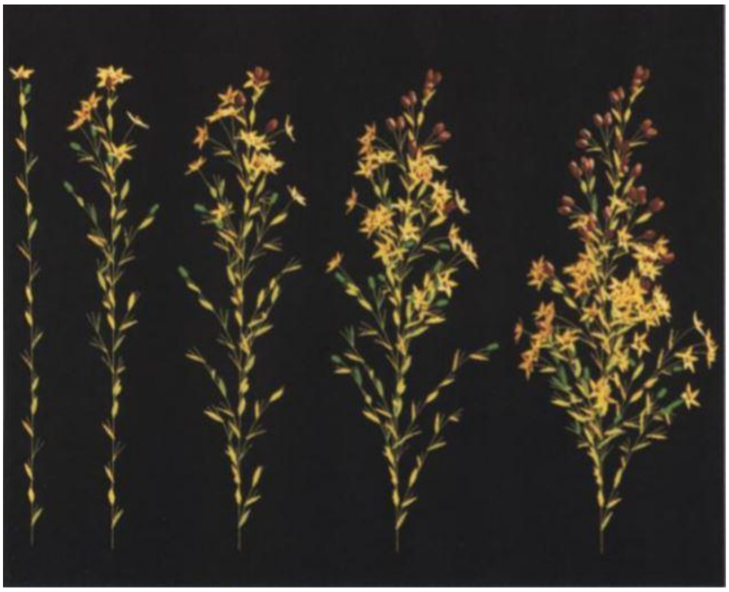

Fig. 1 An example of Algorithmic Generation of Plant Models Utilizing L- Systems [11].

A shape grammar determines a group of shapes known as a language containing all the shapes s created by the shape grammar with no related signs, that is, labeled shapes of the form <s, 0> each resultant from the primary shape by employing the shape rules; each is fabricated of shapes or sub-shapes of shapes in the set S [12].

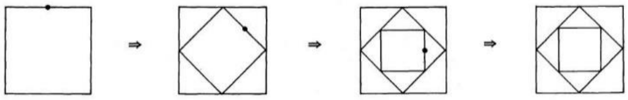

Fig. 2 Simple way of generating shape using shape grammar [13].

A shape grammar is a group of shape-conversion rules and a generation engine selecting and processing those rules step by step, initiating from a primary shape, to create a design language. A way of substituting the particular shapes is specified by the rules and the manner is described in which they are substituted. To date, shape grammar has been utilized in different areas of design, ranging from engineering designs to painting. It can be utilized as an analytical and design instrument. Shape grammar was mostly established and utilized as an instrument for analysis, making a knowledge base for comprehending the features of a particular design. This property has made it a suitable instrument in the learning procedure. The development of novel and original designs through shape grammars was derived from the analytical method [14].

The release of the CityEngine software in the year 2008 has attracted interest in using procedural production in some fields, such as urban planning and archeology [15].

2.4. Procedural Modeling

Procedural modeling is the designing technique extensively utilized in computer graphics technology. The term "procedural modeling" indicates a programming procedure of modeling that allows us to model the rules via the computer, attain automatic modeling purposes, get rid of heavy manual modeling. Presently, the procedural modeling of the city is a hot subject. By the quick expansion of computer graphics hardware, complex urban scenes are easily dealt with [16]. The term ''procedural modeling'' is a concept of computer graphics that produces information objects (entities) that have let obtain access to their creation as a sequence of instructions [17].

Procedural modeling (PM) was an active research area for over 30 years and has extensive applications in various areas like modeling the textures, terrain, plants, buildings, road networks, urban areas, art creation or rivers. No single definition exists for procedural modeling; it includes various generative methods able to (semi-)automatically create a particular kind of content in terms of a set of input parameters. An extensively approved generic description is that PM offers content through a program or a process [15]. Procedural modeling is associated with numerous different areas among which the computational simulation is the closest one. Numerous procedural models are principally generative illustrations of nature-inspired processes like plant development or man-based processes, like urban simulations. PM also connects to physics-oriented simulations [1].

Procedural modeling is a general method able to simplify the explanation of complex geometric structures by encrypting their structures into some procedural rules. Hence, executing the procedural rules represents the encrypted structure geometrically. Though it is possible to use the procedural models in describing any shape, they are particularly beneficial when existing some recursive or repeated components in the encoded structure that can be characterized by a few simple rules [18].

Possibly Lindenmayer (1968) initially introduced the first formal definition of a procedural model and explained a parallel rewriting grammar for generating branching cellular structures [19]. Lindenmayer and Prusinkiewicz inserted the grammar, known as the Lindenmayer system (L-system) into computer graphics (1990) and provided a geometric explanation of symbols from the L-system alphabet. It was confirmed by the authors that L-systems are particularly beneficial for encrypting linear branching structures like plants. They suggested various procedural models in terms of L-systems to represent the algorithmic nature of numerous 8 plant species. In the meantime, procedural modeling became prevalent in other fields of computer graphics like fractals [10], noise generation [15], and texture synthesis [3, 15, 18].

Presently, urban modeling turns to be another are in which procedural modeling is the outstanding content-authoring technique. Novel procedural models were established for automatic modeling of individual buildings [2], street layouts, building facades [14] and city blocks [20]. Various new formal definitions were introduced to represent the procedural models because of the changing nature of most of these novel procedural problems. Several works from urban modeling are on the basis of an idea of recursive shape replacement presented by Stiny and Gips (1971) in their work regarding shape grammars. Later, shape grammars were prolonged to CG architecture (CGA) grammars [2] and split grammars [9] that are more appropriate to model the buildings and facades, respectively [18]. We also increase the parametric control power that makes it possible to allocate a meaningful concept to an element (for example a number making mountains smoother or rougher). Amplifying the modeler/animator's efforts is also provided by the parametric control: some elements result in further details; It was denoted as database amplification by Smith (1984) [21]. The user is unburdened in this parametric control from the specification of detail and low-level control. We also obtain the serendipity characteristic in procedural methods: the unanticipated behaviors of processes, particularly stochastic procedures make us pleasurably amazed [3]. However, it is difficult to control procedural modeling, hence, it is hard to use it as an exploratory design instrument making it available only to specialists [1].

2.5. Types of Modeling

In general, two types of modeling (Manual 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.

Too much manual work is regularly required by the traditional 3D modeling instruments, therefore, their application is overly high-cost for archaeological projects. However, it was proved that our method is efficient and fairly simple. Moreover, our procedural modeling method makes possible to test various hypotheses by setting some elements [2].

Using procedural methods is a powerful solution for large-scale urban modeling [2]. Parish and Müller presently explored the idea of modeling urban environments by means of shape grammars [22, 9]. Müller and Parish provided a way for generating large urban environments, in which each building includes simple mass models and shader for facade detail. They also [9], demonstrated the generation of geometric details on individual buildings' facades. Preferably, we intend to incorporate these two ideas to make detailed and large urban environments [2].

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 formation of detailed large-scale three-dimensional city models [25]. ▪ Data Amplification Capability: Using a simple set of input parameters or a few generations to produce various models [21]. ▪ Data compression: a compact procedural model and a set of elements can represent a slightly complex geometric model, however, the actual geometry is created only when required [21]. ▪ Easy creation of multiple 3D models [29]. ▪ Potential to severely decrease the quantity of modeling effort needed for creating digital content: since its procedures are regularly stochastic, PM is able to create various outcomes from a group of input parameters [21]. ▪ The modeling procedure can be controlled faster and randomness. Updating the rule files is easy and it can be reused them for other projects since the creation procedure has been recorded [8]. ▪ The PM's benefits make it mainly attractive for making virtual environments. Virtual worlds are imperative in numerous applications, such as various (serious) simulations and games [21].	▪ In classic manual modeling method for large-scale modeling is duplicate and hard. ▪ 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 SOFTWARE

CityEngine has been developed well in the gaming industry since its commercial arrival in 2008. This software was widely utilized for creating a greatly detailed three dimensional model of urban landscapes and fictional cities. CityEngine contains the possibility of importing different types of architectural and geographical datasets like CAD and GIS, and quite auspicious options for modeling real landscape [23].

The "CityEngine", a procedural modeling solution (Procedural, Inc. in Zurich, Switzerland) uses a shape-grammar-based geometry generation system known as "CGA shape grammar" to competently make a large scale three dimensional environments within parameter ranges and definite rules. The scripting language utilized by CityEngine was resultant by enhancing the shape and set grammar syntax established in the last decades, moreover, it is enhanced for architectural content. It allows to vary or control volumes, proportions, architectural assets, materials, and rhythms [15].

A procedural modeling method is used by CityEngine to automatically create models via a predetermined rule set. A CGA shape grammar system defines the rules by allowing to create complex models. It makes possible for the users to add or change the shape grammar as much as required offering room for new designs. Via CGA Shape Grammar system, it is possible to read Esri-Oracle format datasets straightly, and it acts as a top-bottom generation tree: it creates complex components from simple Shapefiles polygons/poly-lines/points while each leaf and branch of the generation tree is not able to interact with others. There are differences between this method and main-stream shape grammars like Grasshopper in Rhinoceros three dimensional and Dynamo in Autodesk Revit [25].

Fig. 3 This scheme indicates using CGA shape, a new shape grammar to model procedurally the computer graphics architecture. First, procedural variations of the building mass model are created by the grammar via volumetric shapes and then facade detail is generated according to the mass model. Within the context-sensitive rules, it is ensured that entities such as doors or windows do not cross with other walls since the doors give out on terraces or the street level and the terraces are bounded by railings, etc [24].

4. THE FEATURES OF CITYENGINE

4.1. Procedural Modeling Core

The concept of CityEngine is mainly based on the "procedural" method for efficiently modeling. A code-based "procedure" is given to the computer to provide some commands in this context geometric modeling commands, which then will be implemented. The task is explained "abstractly" in a rule file, despite the "classical" interference of the user interacting manually with the model and modeling 3d geometries. The commands provided in CityEngine's CGA shape grammar including "extrude", "texture" or "split" are extensively known commands in most three-dimensional applications, therefore, they are easy to use by any user to create complex architectural forms within a short time [26].

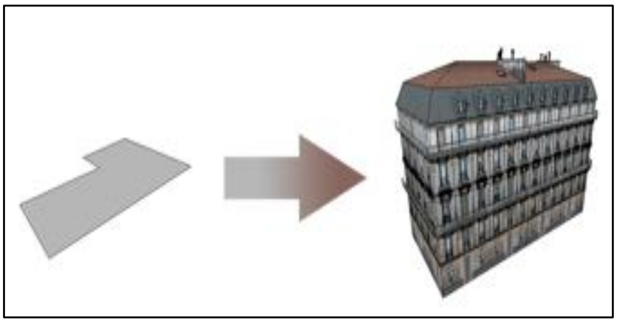

Fig. 4 Procedural Modeling Core [26].

CityEngine involves 3 separate PM systems:

- Automatic generating street networks through an instrument named "Street Growth Wizard"
- Algorithms to subdivide blocks into separate building lots
- CGA shape grammar for creating 3D-content in terms of the made street and lot shapes

According to Paris, the automatic generation of street networks is on the basis of an extended L-system [22]. The system has some differences with the deliberated L-systems to represent the various topological nature of street networks in comparison with the branching structures for which L-systems initially were established. A basic three-step procedure for creating the resulting network is the drawback of the street generation algorithm. Originally, a primary output of branching streets is produced by the L-system [30].

4.2. Dynamic City Layouts

The Dynamic City Layouts provide an influential instrument for the user to make interactive street networks automatically updating in real-time. Streets, whole Blocks and Sidewalks creating the particular urban context adapt competently to the input of the user and provides an instinctive way for the user to design the complete cities' design. Indeed, all the geometries relying on the design of the underlying Dynamic City Layout are updated as well on the fly, you can watch your buildings being reconstructed while you edit the adjacent streets' width [25].

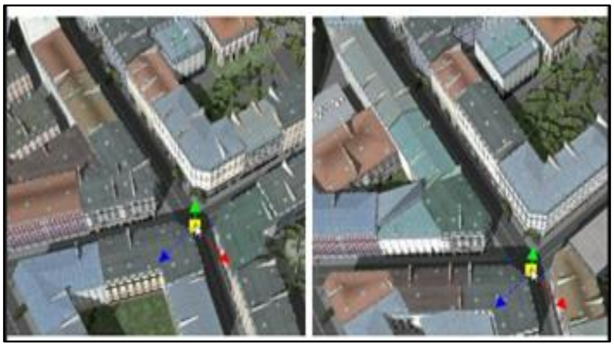

Fig. 5 Dynamic City Layouts [25].

5. THE CITYENGINE MODELING PIPELINE

CityEngine, as a software providing a design framework for geographic data, makes full three-dimensional cities models all over the world. CityEngine integrates the geographical data of a city with other critical data points such as zoning law limitations and the water pipes location. By a completely replicated 3-D city model, architects and urban planners are able to shape better novel structures and evade potential problems prior to occurring.

The geographic information system maps are the main information layer in these city models. The GIS allows the individuals in any industry to visualize well what actually occurring. GIS is potentially able to aid some disciplines in terms of PCs, from urban planning to international development [31].

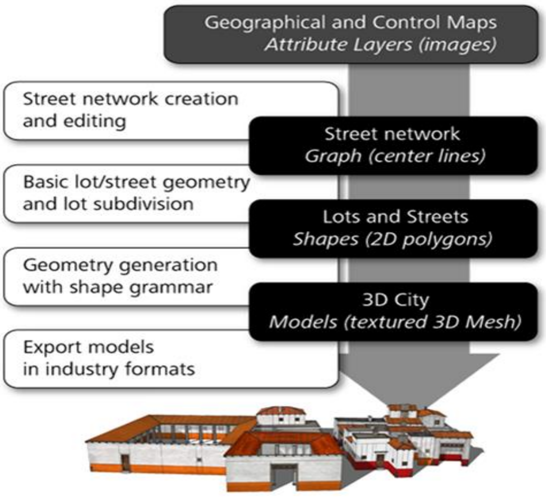

Fig. 6 The CityEngine Workflow [25].

Characteristically, the street network is made in the initial phase, then, the resultant blocks are subdivided into lots. Ultimately, utilizing the CGA rules, the three-dimensional buildings models are created. Polygonal building models are the yield of CityEngine. ACityEngine scene is stored as different data layers to represent various phases. The pipeline is flexibly able to enter at various stages. For instance, building masses or street blocks can be inserted and further processed [25].

6. CREATING LARGE-SCALE URBAN LAYOUTS

To model a city, there are some problems with computer graphics. A transportation network is included in every urban area following environmental influences and population, and regularly a superimposed pattern plan. The appearances of the buildings trace statutory, historical, and aesthetic rules. A roadmap should be designed to make a virtual city and numerous buildings should be created. CityEngine is a system that uses a procedural method in terms of L-systems to model cities. Based on different image maps provided as input like population density and land-water boundaries, in our methods a system of streets and highways is created, the land is divided into lots, and the proper geometry is created to build on the individual allotments [18].

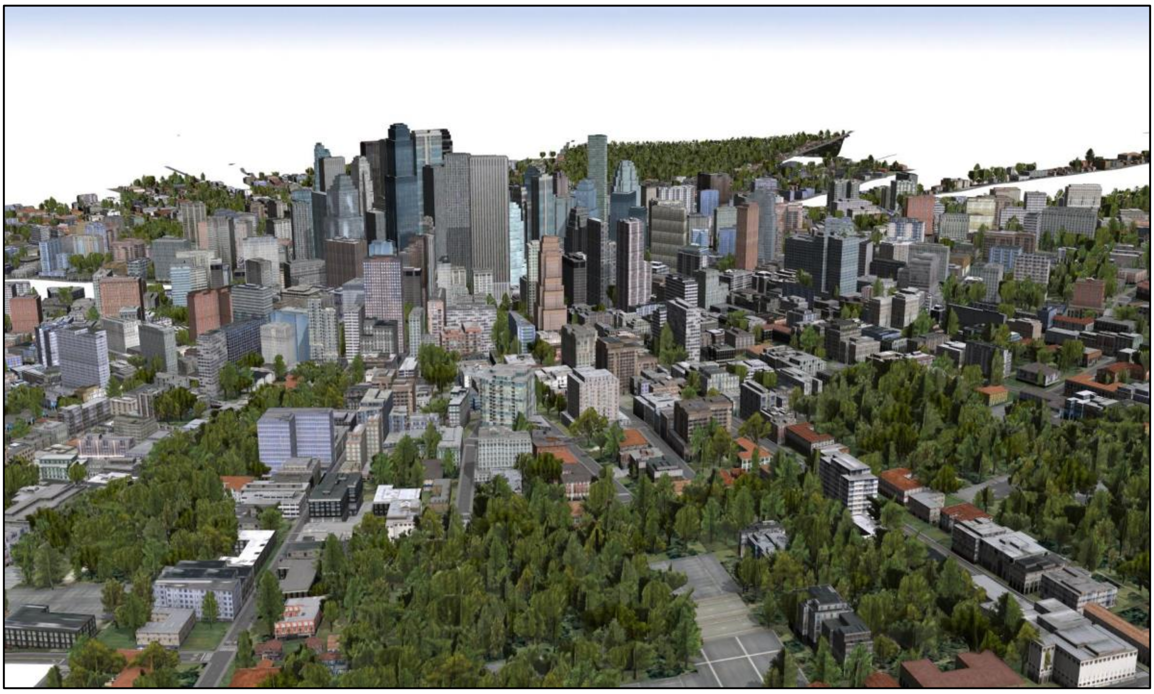

Fig. 7 Generating Large-Scale Urban Layouts with CityEngine.

7. GRAMMAR-BASED MODELING

"Grammar-based" or "procedural" modeling is extensively utilized, however, it is mostly applied, when numerous iterations of a design or many objects should be made following certain standardized rules [25].

Procedural modeling is a three-dimensional model creating process through algorithms and rules. It includes a base geometry, for instance, building footprints with data regarding roof types and building heights as well as procedural rules. As stated in [2], numerous financial resources and various man worth of labor for years are required for modeling huge three-dimensional environments. Hence, saving money and time is vital in creating a large-sized city model. The procedural modeling is a solution to meet these necessities. According to Figure 5, there is a definite time and cost-saving benefit for using procedural modeling when requiring numerous three dimensional design iterations or modeling [8].

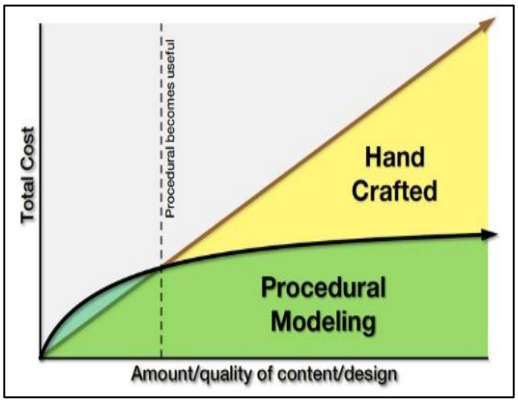

Fig. 8 The efficiency of procedural modeling compared to the manual modeling [29].

7.1. CGA-language

CGA is a language roughly on the basis of an L-system data structure which acts on shapes rather the symbols to explain their transformations through actions written based on the CGA language requirement. The CGA scripts are stored as plain text files with .cga extension. The primary shape from which the generative production starts is external to the language requiring production by means of modeling the abilities of CityEngine or any other modeling software then inserted to CE [30].

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 [32].

Fig. 9 Principle of procedural modeling [20].

CityEngine works based on the formalism of the L-Systems. The initial geometry and the axiom ω are attributed to geometric rule sets that change the initial state via parameter or contents. 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 [20].

The building geometries are generated with the CGA shape grammar as:

1. The building lots are either made by CityEngine with the pre-mentioned instruments or imported. Mass models (building envelopes) can be the starting point as well rather than polygons.
2. The user chooses the rule file (.cga) based on own wishes for applying onto these shapes. It is possible for the user to assign one rule to all buildings or allocate rule sets on a per-building base.
3. Then, the user is able to start using the rules on the chosen shapes. Hence, it is imperative that the shape's *Start Rule* exists in the rule file. Then, it is not possible to invoke any rule. Then, the created models can be discovered in the three-dimensional viewer of CityEngine. Regarding the huge models, generating all buildings in CityEngine is not suggested owing due to memory constraints.
4. There are various possibilities to edit the resultant three-dimensional models,: (1) editing the rules, (2) overwriting a rule set's parameters on a per-building base, and (3) it is possible to alter the random seed of all or single buildings in case using stochastic rules (rules with random parameters).
5. By finalizing the design, the user is able to export the chosen streets or buildings to the hard disk (such as textures). It should be noted that no memory limitations exist in the exporting mode [25].

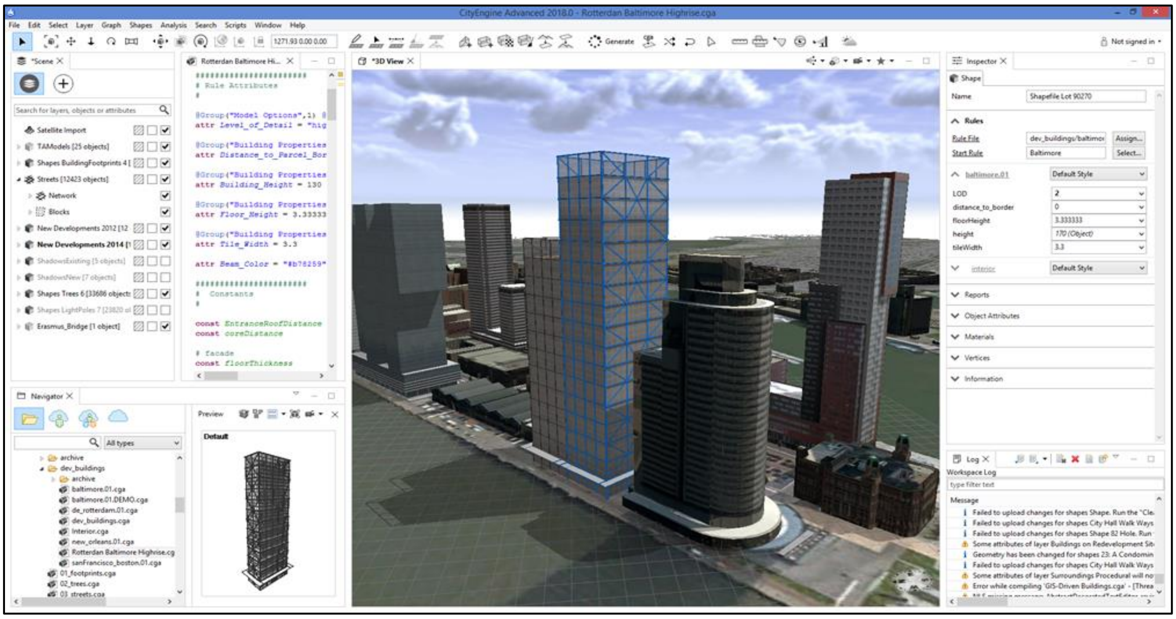

Fig. 10 The creation of building geometries with the CGA shape grammar [25].

8. CONCLUSION

To create large size three-dimensional city models with low costs and in a short time, the procedural modeling is one of the most proper solutions since 3D textures and geometries are made utilizing algorithms and rules compared to manual modeling.

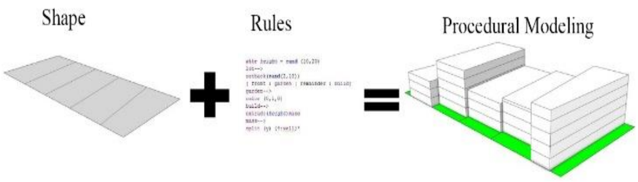

Fig. 11 The basic concept of procedural modeling in model production.

It makes possible to alter in the 3D content more easily for the modeler, as well as to reuse the already made rules for other projects. The modeling procedure includes preparation, data collection, and importing into the software, model creation and export for the created model. The CityEngine software performs well in the generation of the model itself this software is powerful for creating realistic three-dimensional models from two-dimensional data.

Fig. 12 Procedural Modeling by CityEngine in one parcel with different results.

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. This produced model is interactive and has the capability of being up-to-date and can adapt to the new rules at any time. This modeling approach is much more efficient than manual modeling by assigning parameters to the model to make the model controllable.

Fig 13. Neighborhood design with the procedural algorithm.

Above is a complex of the model designed by the procedural language in the CityEngine software.

Some of the analytical outputs of this software are:

Visibility Analysis: Visibility analysis is utilized in analyzing and quantifying the buildings' visual landscape and the sunshine period of buildings within the daytime [17, 28]**:**

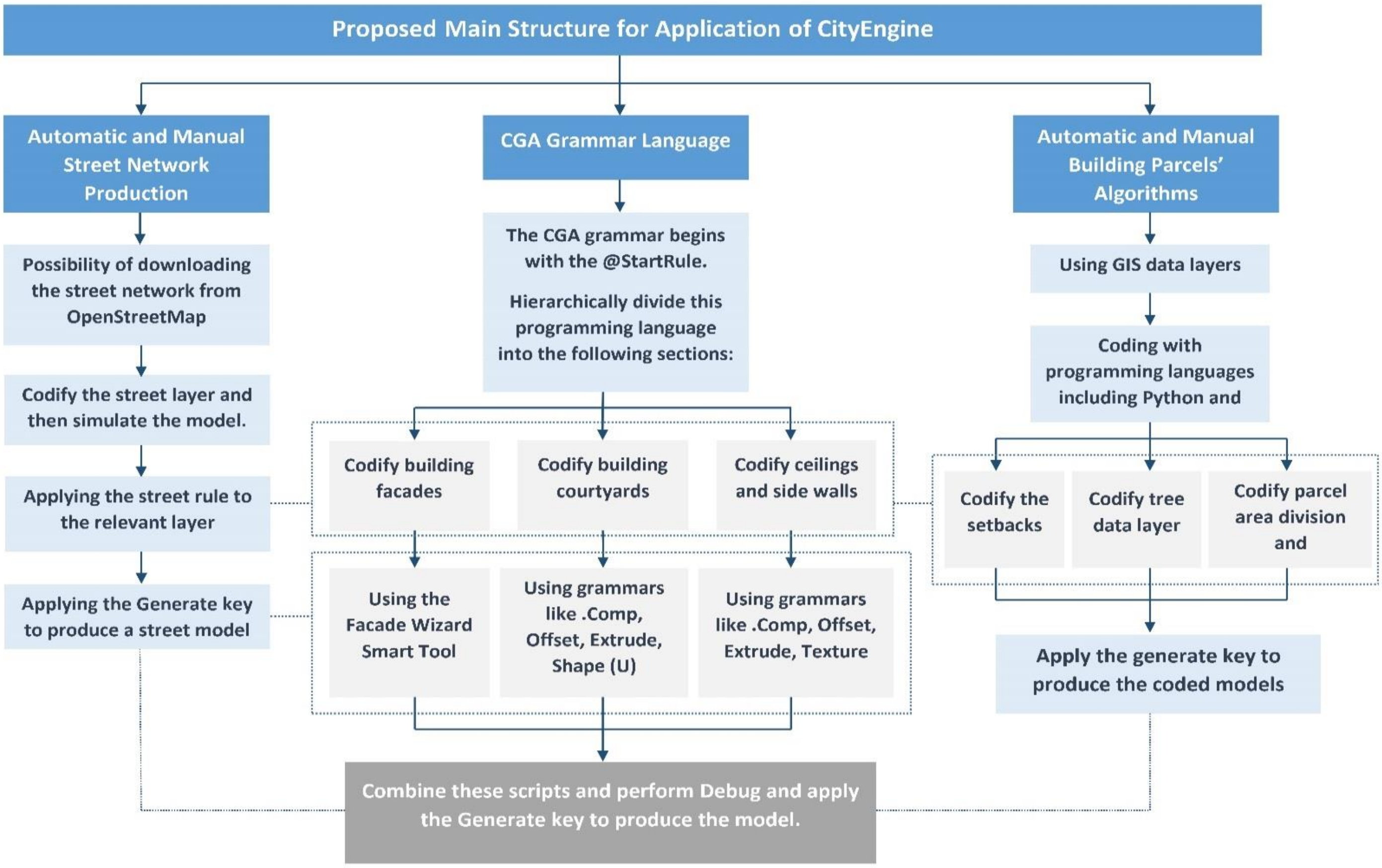

Fig. 14 Workflow of the Procedural Modeling using CityEngine

1. Landscape Analysis: In the landscape analysis, the building is considered as an observer, and the landscape wall is taken to represent Hill as a target [1].

2. Sunlight Analysis: it takes sun points as the target. A value is allocated for each observation point on the building to show the visibility between the sun and the observation point [1].

3. Building 3D Modeling: The two stages are mostly included in the three-dimensional modeling procedure, the first step is to generate three-dimensional external models of buildings in a rapid and batch mode, the second step is to make the target building's three-dimensional internal model [17].

The workflow of the procedural modeling using CityEngine is illustrated as below:

The steps of procedural modeling using cityEngine:

The method suggested is to create scripts in CGA language and then modify, combine, and apply the codes in any study area. In the picture below a part of the rule written for an example study area is shown. The procedural modeling procedure in CityEngine initiates with creating a rule file which is a set of attributes, functions, and rules and is written in scripts.

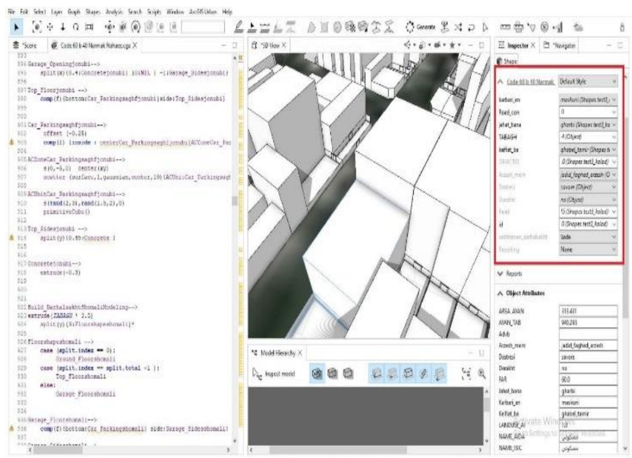

Fig. 15 The rule file with the written attributes (red box)

While writing the rule file, the CGA language creates a copy of the primary form and completes the entire modeling process on the secondary form. Indeed only the basic form of the production process must be defined.

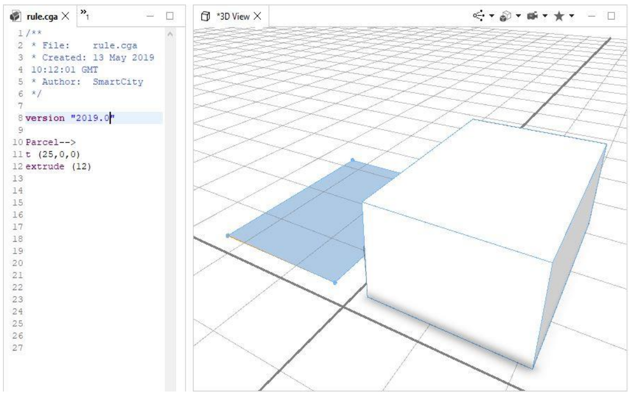

Fig. 16 The CGA language makes the initial parcel and performs the next coding operation over it.

The 3D-end is the result of a sequence of conversion operations (e.g. rotation, size, texture, displacement, etc.) used to the primary shape. By creating the rule file, it will apply to the primary shapes and create 3D content.

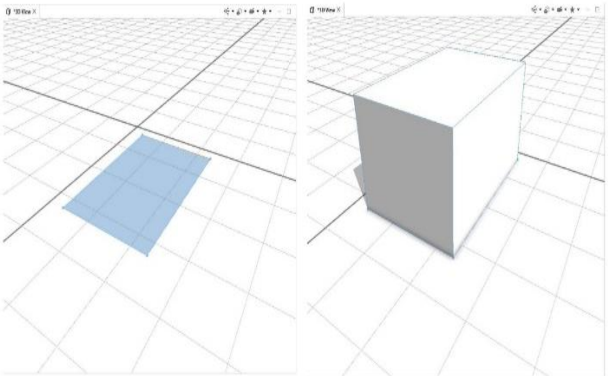

Fig. 17 Application of the code to the parcel (left) and the result is displayed in 3D (right).

What is important for the coding process is an accurate understanding of the model building process as well as a combination of parameters and code layouts that can make the coding more accurate and complete. For example, to calculate the height of buildings, the number of floors in each building can be multiplied by a height of 3.5 meters as the height of each floor.

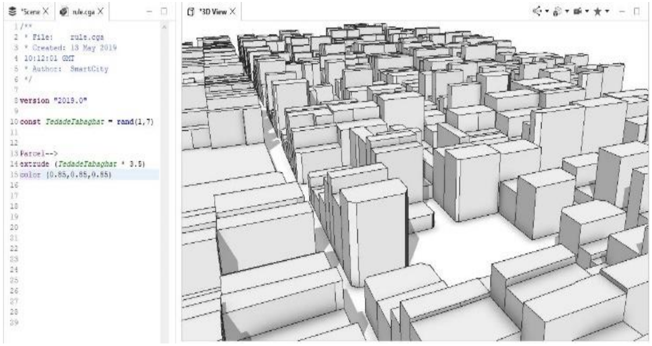

Fig. 18 Calculates the number of floors of the building automatically

Even to further clarify the simulated final model, the facade photos can be taken via area survey and then coded into rules as textures. Finally, by modeling in the rule file, one can get instant reports of parcel areas, building densities and percentages of occupancy, which are flexible, with any changes such as aggregation or disaggregation, changes in the retreat of buildings, etc. All reports are in real-time and they will be displayed instantly so that you can access up-to-date information quickly and intelligently.

Fig. 19 An example of code-generated views and the surveyed two-dimensional images are generated in 3D.

REFERENCES

[1] Nishida, G., Garcia-Dorado, I., Aliaga, D. G., Benes, B., & Bousseau, A., *Interactive sketching of urban procedural models.* ACM Transactions on Graphics, 2016. Vol. 35, No. 4, pp. 1–5.

[2] Marvie, J.-E., Perret, J., & Bouatouch, K., *The FL-system: a functional L-system for procedural geometric modeling.* Visual Computer, 2005. Vol. 21, No. 5, pp. 329–330.

[3] Ebert, D.S., et al., *Texturing & modeling: a procedural approach*. 2003: Morgan Kaufmann.

[4] Smelik, R.M., et al. *A survey on procedural modeling for virtual worlds*. in *Computer Graphics Forum*. 2014. Wiley Online Library.

[5] Mandelbrot, B.B., *The fractal geometry of nature*. Vol. 173. 1983: WH freeman New York.

[6] Techopedia. *Procedural Language*. 2019 [cited 2018; Available from: https://www.techopedia.com/definition/8982/procedural-language.

[7] ArcGIS Resources. *Developing with Esri CityEngine*. 2012 August 4,2014]; Available from: http://video.arcgis.com/watch/1147/developing-with-esri- cityengine.

[8] Dobraja, I., *Procedural 3D modeling and visualization of geotypical Bavarian rural buildings in Esri CityEngine software.* Unpublished MA dissertation, Technische Universität München, Faculty of Civil, Geo and Environmental Engineering, Department of Cartography, 2015.

[9] Shiode, N., *3D urban models: Recent developments in the digital modeling of urban environments in three-dimensions.* GeoJournal, 2000. Vol. 52, No. 3, pp. 263-269.

[10] Luan, X., et al., *Research and development of 3D modeling.* IJCSNS International Journal of Computer Science and Network Security, 2008. Vol. 8, No. 1,

pp. 49-53.

[11] Prusinkiewicz, P. and A. Lindenmayer, *The algorithmic beauty of plants*. 2012: Springer Science & Business Media.

[12] Stiny, G., *Introduction to shape and shape grammars.* Environment and Planning B: planning and design, 1980. Vol. 7, No. 3, pp. 343-351.

[13] Stiny, G. and J. Gips. *Shape grammars and the generative specification of painting and sculpture*. in *IFIP Congress (2)*. 1971.

[14] Mandić, M. and B. Tepavčević, *Analysis of shape grammar application as a tool for urban design.* Environment and Planning B: Planning and Design, 2015. Vol. 42, No. 4, pp. 675-687.

[15] Dylla, K., et al., *Rome reborn 2.0: A case study of virtual city reconstruction using procedural modeling techniques.* Computer Graphics World, 2008. Vol. 16, No. 6, pp. 62-66.

[16] Su, P. and L. Xiong. *Procedural modeling technology in Urban design*. in *Advanced Materials Research*. 2012. Trans Tech Publ.

[17] Müller, P., et al. *Image-based procedural modeling of facades*. in *ACM Transactions on Graphics (TOG)*. 2007. ACM.

[18] Parish, Y.I. and P. Müller. *Procedural modeling of cities*. in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 2001. ACM.

[19] Lindenmayer, A., *Mathematical models for cellular interactions in development I. Filaments with one-sided inputs.* Journal of theoretical biology, 1968. Vol. 18, No. 3, pp. 280-299.

[20] Radies, C., *Procedural random generation of building models based Geobasis data and of the urban development with the software CityEngine.* Bernburg, Germany, 2013: pp. 175-184.

[21] Smith, A.R., *Plants, fractals, and formal languages.* ACM SIGGRAPH Computer Graphics, 1984. Vol. 18, No. 3, pp. 1-10.

[22] Müller, P., et al. *Procedural 3D Reconstruction of Puuc Buildings in Xkipché*. in *VAST*. 2006. Citeseer.

[23] Edvardsson, K.N., *3D GIS modeling using ESRI's CityEngine: a case study from the University Jaume I in Castellon de la Plana Spain*. 2013.

[24] Müller, P., et al. *Procedural modeling of buildings*. in *Acm Transactions On Graphics (Tog)*. 2006. ACM.

[25] ArcGIS Resources. *3D urban content creation and design using CityEngi*. [cited 2014; Available from: http://resources.arcgis.com/en/communities/cityengine/01w900000008000000.html.

[26] ArcGIS Resources. *What is CityEngine?* [cited 2014 October 23, 2014]; Available from: http://resources.arcgis.com/en/communities/city engine/01w90000000m000000.htm.

[27] Govil-Pai, S., *Principles of Computer Graphics: Theory and Practice Using OpenGL and Maya®*. Vol. 190. 2006: Springer Science & Business Media.

[28] Zhang, H., et al., *The Application of GIS 3D Modeling and Analysis Technology in Real Estate Mass Appraisal-Taking landscape and sunlight factors as the example.* The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2014. Vol. 40, No. 4, pp. 363.

[29] Piccoli C. *CityEngine for Archaeology. InProceedings of the Mini Conference 3D GIS for Mapping the via Appia, Amsterdam, The Netherlands* 2013 Apr (Vol. 19).

[30] Viinikka J. *Adopting procedural information modeling in urban planning*, 2014.

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

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

AUTHOR (S) BIOSKETCHES

M. Ghorbanian., *Assistant Professor, School of Architecture & Environmental Design, Iran University of Science & Technology (IUST), Tehran, Iran.*
Email: *Ghorbanian@iust.ac.ir*

F. Shariatpour., *MA. Student of Urban Design, School of Architecture & Environmental Design, Iran University of Science & Technology (IUST), Tehran, Iran.*
Email: *Farshad_shariatpour@arch.iust.ac.ir*

HOW TO CITE THIS ARTICLE

M. Ghorbanian., F. Shariatpour., (2019). Procedural modeling as a practical technique for 3D assessment in urban design via city engine. Int. J. Architect. Eng. Urban Plan, 29(2): 255-267, December 2019.

URL: http://ijaup.iust.ac.ir/article-1-207-en.html