ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20180713143638

Original scientific paper

AUTOMATIC PROCEDURAL 3D MODELLING OF BUILDINGS

Robert ŽUPAN, Katarina ŽUPAN, Stanislav FRANGEŠ, Adam VINKOVIĆ

Abstract: The problem of modelling, especially of large-scale virtual urban environments such as city areas is today very challenging for cartographers. Cities are difficult to model in detail because of their often complex geometries. This paper describes the applied and tested new procedures for the development of a test three-dimensional urban area model using the Esri CityEngine software, which is based on procedural modelling. This process involves three steps. The first step is to collect the available data, as well as the Digital Model Relief data. The second step is to create a Computer Generated Architecture (CGA) file that contains a set of rules used by the software for an automatic generation of the model. The third step is to create and visualize 3D models in the CityEngine software because it can be programmed, for example, according to the rules of cartographic visualization. The Level of Detail (LOD) 2 was reconstructed. Several potential uses of such 3D visualization are also described.

Keywords: CGA; CityEngine; procedural modelling; 3D modelling; 3D visualization

1 INTRODUCTION

3D models facilitate planning in the areas such as urban planning, energy, noise control, telecommunications, risk management, etc. 3D models of the existing state of an area also make it easier to visualize a potential future state which is good for the investment business. In tourism and in the real-estate area, 3D models make it easier to imagine space and orientation. A 3D model designed for professional users (planners) must emphasize the accuracy (geometric and attributive) and the required level of detail, while a 3D model intended for non-professional users (tourists, investors, real estate buyers, etc.) must have an accent on appearance. An exception to this rule is when a 3D model is created as a background for visualizing a future project, for example, when architects design a future building and want to visualize it together with the environment. It is then important that the existing condition around the plot on which it is built is accurate and looks nice, as the architect will "build" his building on that 3D model of an existing state that must be graphically and in detail aligned with the visualization of the building [1].

2 PREVIOUS WORK

3D cadastre refers to a cadastral system that represents the legal status of a property, not just the 2D cadastral parcels, but the 3D legal units with the rights, responsibilities and unique and homogenous constraints [2, 3]. The development of an efficient 3D cadastral system includes the consideration of several legal, institutional and technical aspects [3, 4]. The basic factors of the 3D cadastre are properties, buildings and other constructions, special parts of properties, ores and other underground natural resources, as well as the public utility infrastructure. The amount and complexity of data that are kept on the basis of regulations in public authorities are constantly increasing. This process is conditioned by the development of technology and takes place alongside it. The list of opportunities offered by new technologies is steadily increasing. The adoption of technological achievements, particularly in the field of information-communication technologies, and their implementation in the enrolment systems facilitated the maintenance, access and exchange of data. This opens the possibility of collecting some new, additional data that had not been previously collected because their maintenance was technically overwhelming [5].

The pressure on the construction of buildings, especially in city centres and on the seashore, has led to a dense construction of complex buildings. These structures are intertwined. As a rule, the enrolment of rights to parts of buildings is enabled, and the spatial display of the extension does not usually exist [6]. An increasing number of tunnels, infrastructure facilities (plumbing, gas pipeline, electric and other grids) located below or above the land, and not owned by the landlord, require a better description and enrolment in the public registers of such facilities.

As far as the technical aspects are concerned, the 3D data collection, 3D data modelling, and 3D database management systems are factors in the development of 3D cadastral applications. However, 3D visualization plays a significant role and is recognized as an issue of extreme importance in several international workshops (1st, 2nd and 3rd International Workshop on 3D Cadastre) by several authors [7, 8].

Visualization systems are necessary not only to display the physical objects but also to visualize their legal parts. Legal parts may be of limited or unlimited volume [9]. For the cadastre, several specific prototypes have been proposed, such as the spatial databases with CAD and GIS access-parts [3, 8, 10-14]. On the other hand, the technological advancement and the ease of use of web application visualization have made them very popular among users. This popularity has led to the construction of several 3D visualization prototypes using web technologies [15-21], although they still require the maturation and confirmation by users, so that they can be used in real life situations. Thus, 3D visualization for the cadastre requires further research [22, 7].

2.1 Procedural Modelling

The best method for creating a large 3D city model in a short time and at a low cost is procedural modelling, which means creating 3D models using rules and algorithms. It is an economical solution for generating 3D urban content [23]. The procedural modelling software from Esri called CityEngine is mainly used to create 3D urban area models. Modelling is based on CGA (Computer Generated Architecture) rules. CGA is a scripting language suitable for creating architectural 3D content [24]. Previous research in this area have shown that CityEngine is suitable for modelling cities. CityEngine is a three-dimensional (3D) modelling software developed by Esri R & D Center Zurich (formerly Procedural Inc.) and specializes in creating 3D urban environments. With procedural approaches to modelling, CityEngine enables you to efficiently create a detailed 3D city model displayed in a large scale in just a few mouse clicks instead of a long-lasting and exhausting way of creating and manually setting objects [25].

2.2 3D Visualization

The focus of this paper is primarily on 3D visualization, not on data management or data delivery. Van Oosterom [22] stated that "visualization and/or interaction with 3D cadastral particles requires more attention and can be quite different from the more familiar visualization, e.g. 3D city models", and added: "Some specific key points are as follows:

(1) how to visualize dense 3D volumes such as in complex buildings because the first visible outer layer of 3D space unit blocks the look of others;

(2) how to display open or unlimited particles,

(3) how to include the Earth's surface and / or other reference objects (e.g. as CityGML) into a 3D cadastre,

(4) how to display underground infrastructure."

Shojaei [15] proposed a set of conditions that should meet 3D cadastral visualization systems and tested them on several visualization systems to determine to what extent they implemented the necessary preconditions. They also reviewed several common 3D visualization solutions.

3D models created in CityEngine have different applications, for example, in cultural heritage, urban planning and film industry. Procedural modelling can be applied to the creation of 3D video content [23]. Research by Radies describes the principle of 3D modelling of buildings using gradual modelling procedures in the CityEngine software. The basis of procedural modelling is based on form grammar with production rules to generate detailed 3D objects from simple initial geometric features. It begins with the extrusion of the initial shape on a model that is later divided into shapes, for example, facades and roofs. To create a detailed high-level model, the created shapes are divided into elements, for example, windows and doors, and textures or colors, and added to these elements.

In order to create better and more realistic 3D urban content, several suggestions have to be made, as Watson [23] said: "Urban content is a little more than building a shape." Creating a real-time impact on urban content and wear elements can be applied to materials and models, for example, the accumulation of rust and dirt. The model of the city looks more realistic if there are cars and pedestrians in it.

3 CITYENGINE

CityEngine is a standalone computing application for designing, planning and modelling urban environments in 3D. It was designed to help professional users in GIS, CAD and 3D systems to:

- quickly generate 3D cities from the existing 2D GIS data;
- realize conceptual design in 3D based on the GIS data and procedural rules;
- model virtual 3D urban environments for simulation and entertainment.

Consequently, experts that are using CityEngine come from the following industries: urban planning, design and development (architectural design and local government); entertainment (movies, commercials, video games); simulation of the real world (Fig. 1), emergency services and defence; and, of course, education.

Figure 1 Scene from software supporting procedural modelling [26]

The main concept of CityEngine is the "procedural" approach to modelling. The computer is a code that represents a number of commands - in this context, geometric modelling commands - which will then be executed. Instead of "classical" user intervention, i.e. manual interaction with the model and 3D geometry modelling, the task is described as "abstract" in the "rule" file. Commands available in the CGA shape grammar format of CityEngine such as "extrude", "split" or "texture" are widely-known commands in most 3D applications so that each user can easily adopt and create complex architectural shapes in a relatively short period of time [26].

One "rule" file can be used to generate many 3D models. For example, the rule may use information about attributes stored in the GIS data such as the number of floors, floor height, roof type, wall material type, etc. - to generate a range of alternative 3D models that accurately represent the properties of each feature. The more attributes you have, the more accurate the model can be. The 3D model is nothing more than a 3D object that is the result of an extruded 2D form according to the rules defined in the CGA "rule" file. The origin of these 2D forms is variable:

- it can be imported from Esri Shapefiles or File Geodatabase;
- it can be manually build in CityEngine;
- it can be generated through CGA rules.

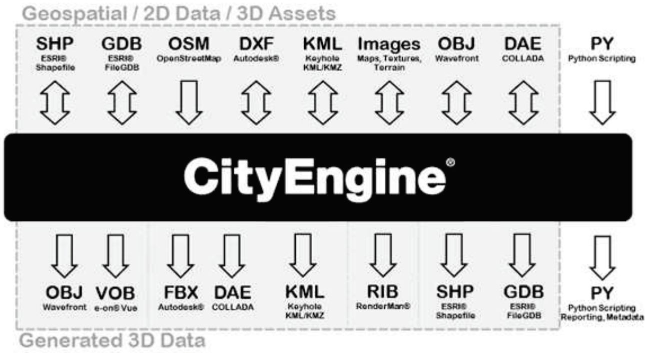


Figure 2 CityEngine Software Input and Output File Formats [26]

Three-dimensional objects represented in CityEngine do not have to be all generated within CityEngine. They can be downloaded through the formats shown above in Fig. 2. However, only the geometry of objects from the multiple Esri Shapefiles or File Geodatabase can be edited and later upgraded to the original files. 3D models are 3D objects generated in CityEngine through procedural modelling. The remaining objects (3D or 2D) are called Shapes. By changing the urban design (i.e. by regenerating the "regenerate") commands, they are automatically applied and updated [26].

The next advantage of CityEngine is the variety of data formats supported for import and export. CityEngine supports geographic data, 2D data, 3D data (polygonal networks) and Python scripting as input data types and offers a wide variety of formats for exporting 3D data. The further advantage of procedural modelling is its ability to dynamically edit. For example, when importing a street network, the software generates plots inside the street and 3D buildings are generated inside the parcel, and if the street is moved, then the buildings are also moved because they are procedurally associated [26].

The lack of procedural modelling compared to manual modelling techniques is less "user-friendliness" [27]. The knowledge of CGA scripting for writing proper files makes the process of procedural modelling more complex at the beginning if the user has no experience with it.

3.1 A tool for Procedural 3D Modelling

The main idea behind CityEngine is to create an urban environment from scratch. At the very beginning of the project, central street lines are usually generated based on pre-programmed chart algorithms whose value can be changed by the user. These street lines close the 2D polygons representing the cadastral particle (2D particle). In both cases, the user can modify the values of the parameters used. In the third stage, the buildings are built in 3D, based on the CGA "rule" file, which contains data on the number of floors, heights, type of roof, color and texture. 3D buildings can be generated directly from the particle, citing, for example, the offsets or pre-loaded 2D floor planes. At this stage, vegetation, water, vehicles, furniture, etc., can also be generated based on the CGA rules. Infrastructures above and below the Earth's surface are also represented at this stage. If needed, the created urban environment can be loaded into another software specializing in 3D visualization, presentation or analysis (e.g. ArcGIS, Maya, 3ds Max, Google Earth, Unity, Unreal, RenderMan and RealityServer) [28].

For some imported 3D objects, you may only need to change the display mode (color, thickness, transparency, texture, etc.). In the case of 2D objects, it is necessary to further define the CGA rules that are based, for example, on the attributes from the GIS attribute table. Another important feature of CityEngine is the fact that, in the case of features from the Geodatabase file, the existing relationships between the tables are imported. They can be viewed and searched through a code (CGA rules or Python). Let us assume that apartment A, located at level L of building B located on plot P, is owned by O1 and O2. If these relationships are modelled in the File Geodatabase, then the Inspector Window will show information about the physical 3D object "Apartment A". The attribute "Owner" of the object "Apartment A" will appear in the form of a list, as there are two owners - O1 and O2 [28].

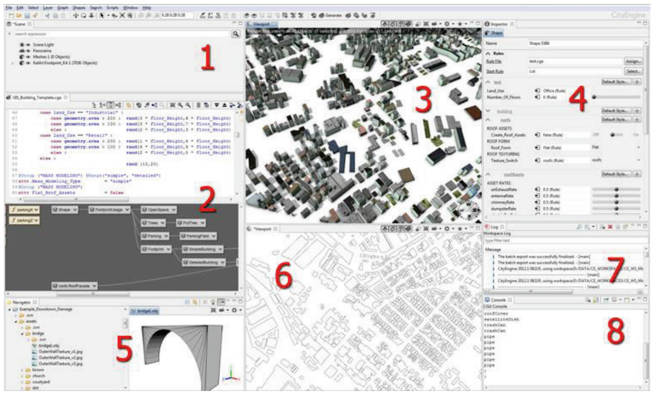
Programming in Python can also present great flexibility in the workflow. By programming in Python, it is possible to define which objects must belong to each layer. For example, if there is interest in separate legal objects, by placing them in different layers, they can later be separately controlled for their visibility. Especially in the case of separation of flats per floor, laying a layer for each floor. Moreover, attribute queries can only be made by programming in Python. For example, to select all multi-owner objects, and then create a layer with that information or export them to a text file in the form of a report [28].

According to Watson [23], Esri CityEngine is one of the most prominent procedural modelling tools available. It is a standalone software that transforms 2D GIS data into smart 3D cities models. This improves urban planning, architecture and design. CityEngine combines 3D modelling and grammatical modelling methods to generate 3D content. Shape grammar recognizes certain shapes and its possible substitutes. The forms that are to be replaced and the way in which this will be executed are listed in the rules. Split grammar divides the 3D object into its components, for example, the front facade, edges or peaks.

For a successful use of the form and split grammar in the modelling process, the following two steps should be taken into account and implemented. The first step is gaining a good understanding of basic building elements such as windows, doors, pillars, walls and others. The next step is to create grammatical rules for combining these basic elements [23].

Typical modelling sessions include steps shown in Fig. 3.

Some of the elements needed for city modelling in CityEngine are the terrain, controlled map layers, street networks, building blocks and CGA "rule" files. Street network and floor plans can be automatically or manually created in CityEngine or imported from DXF and SHP files [27].


File Edit Select Layer Graph Shapes Search Scripts Window Help
Scene
Scene Light
Panorama
Meshes 1 (0 Objects)
GIS_Building_Template.cga
1
2
3
4
5
6
7
8

```
        case Land_Use == "Industrial" :
            case geometry.area > 200 :  rand(3 * Floor_Height,6 * Floor_Height
            case geometry.area > 100 :  rand(3 * Floor_Height,7 * Floor_Height
            else :                      rand(2 * Floor_Height,3 * Floor_Height
        case Land_Use == "Retail" :
            case geometry.area > 200 :  rand(1 * Floor_Height,4 * Floor_Height
            case geometry.area > 100 :  rand(4 * Floor_Height,9 * Floor_Height
            else :                      rand(2 * Floor_Height,5 * Floor_Height
        else :
                                        rand (10,20)

@Group ("MASS MODELING") @Range("simple", "detailed")
attr Mass_Modeling_Type          = "simple"
@Group ("MASS MODELING")
attr Flat_Roof_Assets            = false
```

Figure 3 Windows in the CityEngine software

The CGA (Computer Generated Architecture) script that defines the rules or procedures for creating 3D content is the core of this software. CityEngine can be used for the entire developmental cycle of procedural modelling, as it includes scenes with the already existing data, modelling street geometry, writing and analyzing CGA script files, applying CGA files to create 3D models, and exporting created models for further processing.

The two primary properties that link the initial form to the generation process are the "Rule file" that defines that the correct CGA file assigned to the initial form, and the "Start rule" that defines the production rule from which the process of creation begins. Other attributes that describe the initial form (e.g. height of building, number of floors, roof shapes, etc.) can be used as parameters in the CGA rule file [29].

The advantage of CityEngine is the easy option of creating 3D urban scenes based on the existing GIS data. High-quality 3D content can be created with only a combination of 2D data, attributes, and procedurally defined rules. The components of CityEngine are geometry, attributes, and procedurally defined rules.

The complexity and actual accuracy of the generated model depends on the amount of detail envisaged for each component. When 3D city content is generated using the parameter rules, it is possible to create a 3D city design and further analysis (for example, visibility and shadow analysis) of the model. In short, creating a 3D city model in CityEngine can be done in 5 steps:

- First of all, there should be a 2D Geodatabase or 2D GIS data, CAD data or data of other formats that will be imported into CityEngine.
- After import, GIS data can be edited and manipulated. This step involves 3D streets, blocks and parcel creation.
- The next step involves 3D extrusion of buildings, production of roofs, street furniture and other elements in the model.
- After the model is made, the next step includes the in-depth detailing of the model details using procedural rules, the selection of detail levels (LoD) and the addition of vegetation.
- The last step of the process is to export the created city model to another 3D application. It can be displayed in another application, exported to a video game, and shared online using the Esri WebScene. It is also possible to export the created models back to the Geodatabase file for a more accurate 3D analysis in ArcScene [30].

4 APPLICATION OF PROCEDURAL MODELLING IN THE TEST AREA

The test area includes 95 buildings in the Ilica street (Zagreb) from the Ban Jelačić Square to the British Square. For building a 3D model of a building, the Shapefile of the floor plan (Fig. 4) was used with attribute information about the building height. The files have been provided by the GDi GISDATA in Zagreb. Shapes were initially in the old Croatian coordinate system of HDKS. They were transformed using an open source software of QGIS into the official Croatian coordinate system of HTRS96.

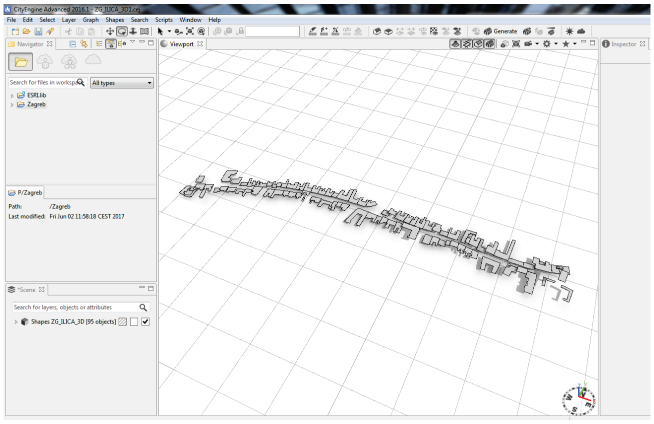

CityEngine Advanced 2016.1 - ZG_ILICA_3D.cej
File Edit Select Layer Graph Shapes Search Scripts Window Help
Navigator
Viewport
Inspector
Generate
Search for files in workspace
All types
ESRI.lib
Zagreb
P/Zagreb
Path: /Zagreb
Last modified: Fri Jun 02 11:58:18 CEST 2017
*Scene
Search for layers, objects or attributes
Shapes ZG_ILICA_3D [95 objects]

Figure 4 Source data uploaded and visualized to Esri CityEngine

For a more realistic scene, a DRM (Digital Relief Model) was used. It was downloaded with a spatial resolution of 1 arc-second from the free web service: https://asterweb.jpl.nasa.gov/gdem.asp. Since CityEngine has no ability to load DMRs in the form of grid points, by using the open source software SAGA GIS, a shaded relief view was created as a geotiff image and loaded into CityEngine.

4.1 Set of CGA Rules

The definition of the CGA set of rules follows once all the required data has been collected. Attributes that are different for each building are defined. These are the height of the building, ground floor height, height of the floor, width of walls on the buildings, facade texture, type of entrance door, window, floor and the type and texture of the roof. In this case, the height of the building and the texture of the facade are given for each shape according to their ID. Entrance doors, sliding doors and windows are the same in all buildings due to the shrinking workload. The roof type is also the same for all buildings, but the texture of the roofs is assigned by the "random" function that randomly assigned the texture of each building between 6 different textures of red and gray bricks to mimic the appearance of the DOF (digital orthophoto) building (Fig. 5).

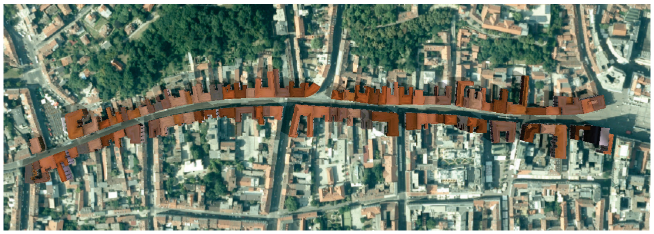


Figure 5 Roof textures assigned by the "random" method

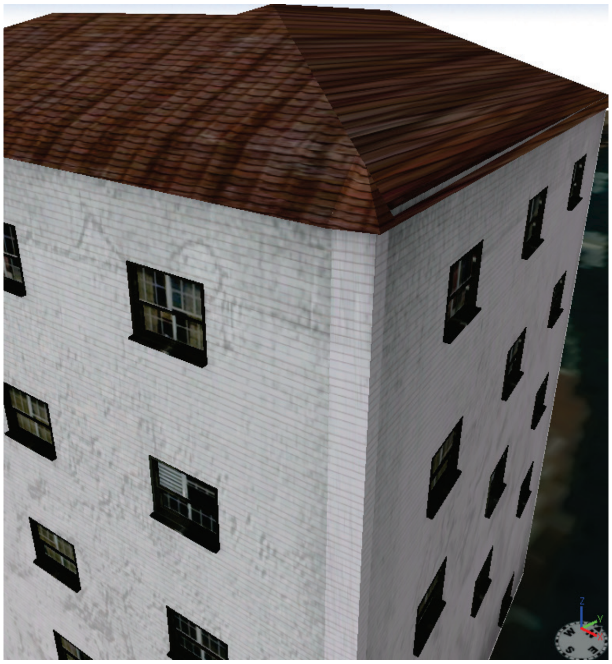


Figure 6 Problem of roof modelling

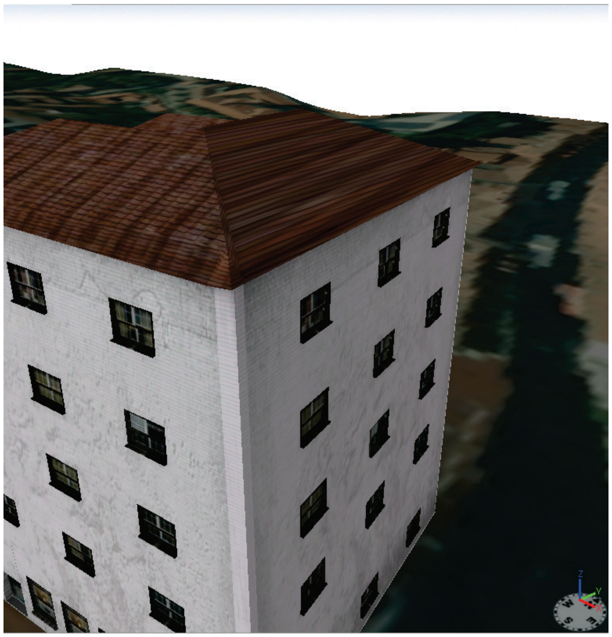


Figure 7 Problem of roof modelling

The height of the floors and the width of the windows defining the number of windows on the facade were changed in the Inspector Window of CityEngine according to the information obtained from the photo of the building. The "extrude" function generated the height of the building, which is then divided into the front facade and all the other facades and the roof. The width of the floors on the ground floor and the ground floor are defined where the ground floor is marked as a fixed floor, and where the floors are "floating", meaning that they are repeated to infinity on the same pattern. The type of the roof (RoofHip), as well as the roof gradient (20°) were also defined. Since a roof with a gradient of 0.5 m above the edge of the facade was defined, the problem with roof modelling was that the facade wall "cut" the roof on the edges (Fig. 6). This problem has been solved by the code line *t* (0, 0, 0, 15), which projected the roof by 15 cm in the *y* axis direction, thus avoiding the overlapping with the walls (Fig. 7).

Generating the window and door on the edges with other objects is disabled by using the code:
Window → case touches: Wall
Door → case touches: Wall

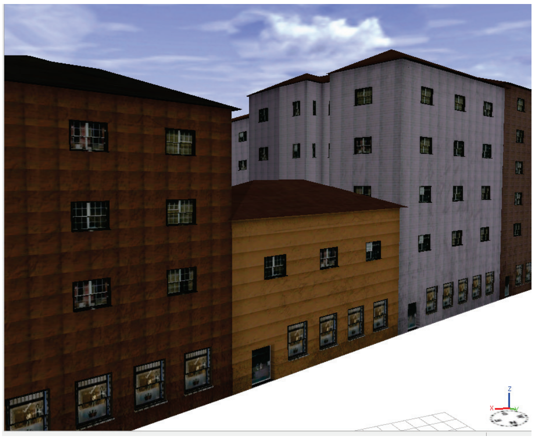


Figure 8 Viewing a 3D model of a part of the city of Zagreb

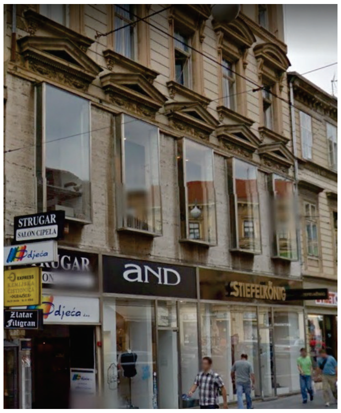


Figure 9 Facade bends

The balcony, different facade tiles (Fig. 9), and the decorative elements on the facade have not been modelled, since for this level of modelling in CityEngine, a polygonal rather than a procedural type of modelling is foreseen.

Finally, the 3D model of buildings was developed to the LOD 2 level of detail (Fig. 8).

The entire set of rules used for modelling buildings can be seen on the web at [31].

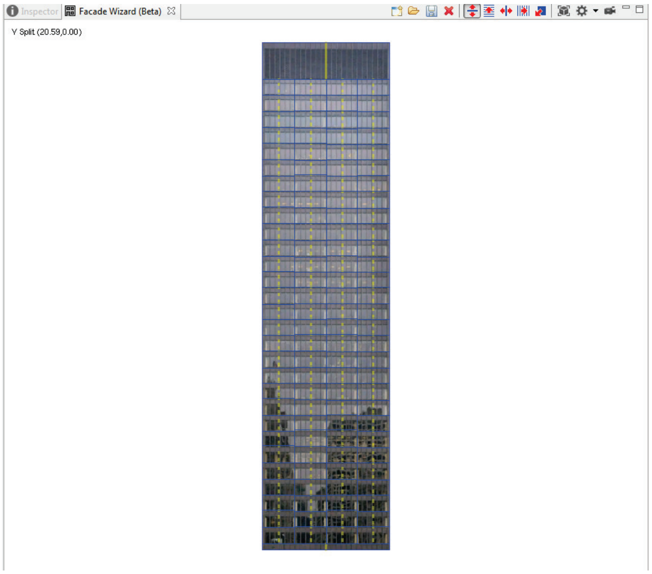


Figure 10 Distribution of the facade on the floors and surfaces in Facade Wizard

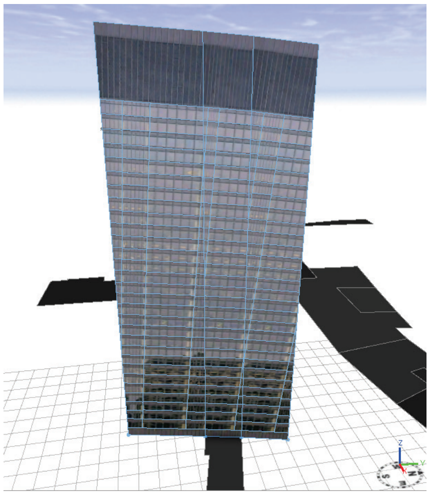


Figure 11 Result obtained with the Facade Wizard

After all the LOD 2 level buildings were built, we decided to manually create the Ilica skyscraper, since it is the only building with its shape and modern architecture to stand out from the rest of the buildings built in the late 19th and early 20th centuries. For creating the skyscrapers, the Facade Wizard was used. This tool creates a 3D model based on a facade photograph. Unfortunately, the Facade Wizard was not able to take a photograph of the skyscraper that covered the whole building which is why the texture, looking as a hyphen skyscraper, was downloaded from the Esri library. The Facade Wizard functions in a way that the facade is divided into functional and aesthetic elements, which are then given a "fixed" or "floating" value, as well as a z-component (i.e. the descent or ascent in the facade) (Fig. 10). As the obtained results were not satisfying (Fig. 11), we had to use a tool for automatic error correction - Shape Cleanup.

Fig. 12 and Fig. 13 show the results of skyscraper modelling in the Ilica street, as well as a comparison of a scene from CityEngine.

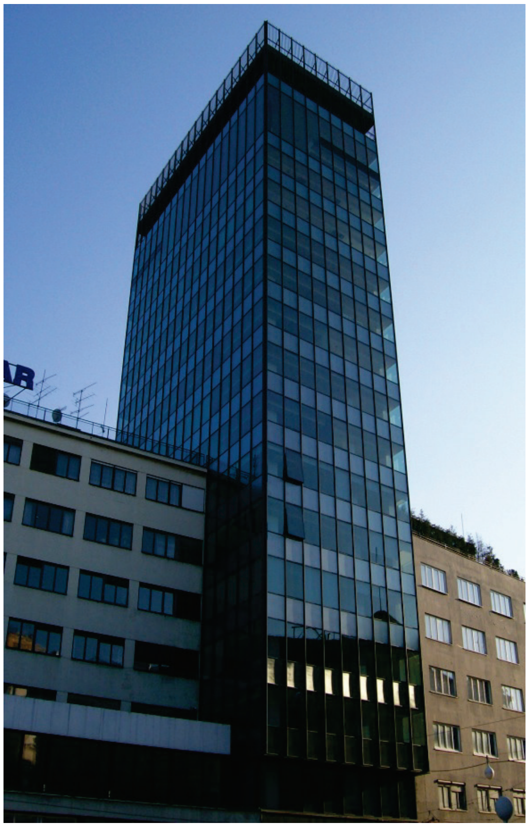


Figure 12 Skyscraper in the Ilica street - photo

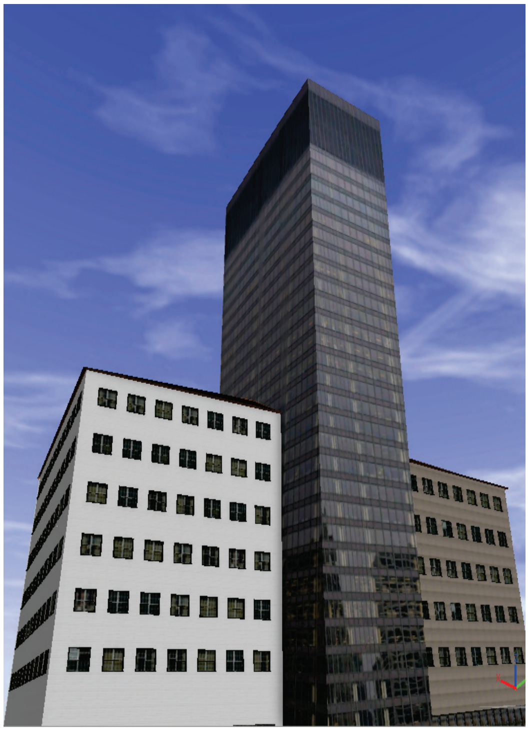


Figure 13 Skyscraper in the Ilica street - 3D model obtained as result of automatic procedural modelling

When modelling the facade based on an image, the main problem in the process is to generate a credible 3D model with semantic information from one input image. The process consists of four phases:

1) Detection of facade structure - an agitation that automatically divides the facade image onto floors and surfaces.
2) Definition of the surface - this is the phase in which the planes are divided into smaller regions. The method recursively selects the best splitting line of a given surface. The division of the structure is a common concept in procedural modelling [32, 33] and gives the hierarchy elements.
3) Element Identification - the procedure for matching the elements obtained by the previous step with a huge library of 3D objects.
4) Editing and extraction of the grammatical set of rules - at this stage of interpretation, the facade is encoded as a tree shape. Due to the lack of depth information, it is necessary to edit the depth of facade elements to improve the visual perception.

This method can be used for urban reconstruction or reconstruction of facades if original images have an appropriate quality. The full resulting programmed code can be found at [31]. The finished 3D model of a single street in the city of Zagreb is shown in Fig. 14. The 3D model consists of 95 buildings in the Ilica street (Zagreb) from the Ban Jelačić Square to the British Square.

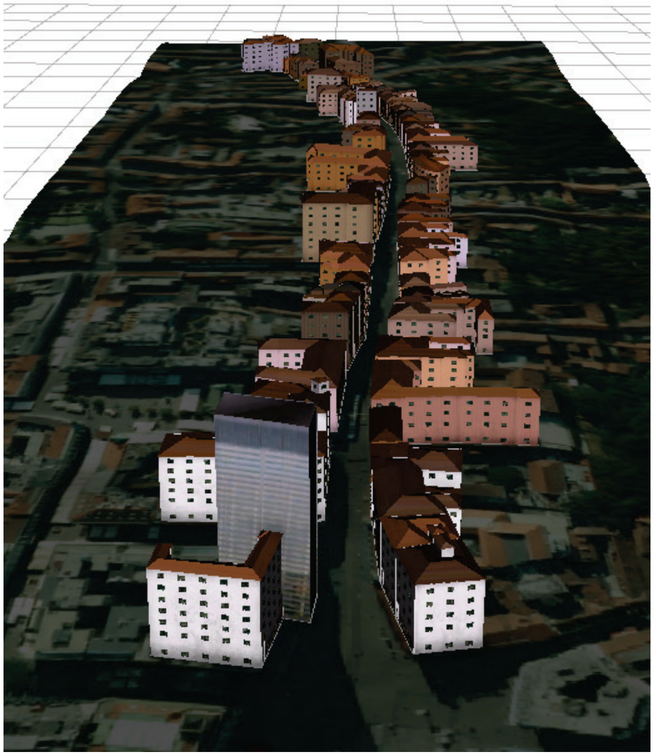


Figure 14 Finished 3D model of a part of the city of Zagreb

5 CONCLUSIONS

In this paper, it is shown that there is no need for expensive solutions and a new software for procedural 3D modelling (because there is no software for automatic 3D modelling so far). The software used in this research is a tool for generating 3D models from 2D GIS data and it was programmed to do that automatically. However, following the fundamental policy of procedural modelling, namely that one rule applies to a large number of buildings or entire cities, one cannot obtain a satisfactory and completely faithful representation of buildings. In case we want a more detailed view of buildings with all ornamental details on the facade, or at least a model that can be used to calculate the volume or the grossly developed surface which contains all balconies, terraces, etc., we need to write a special code for each building and in most cases, further elaborate details on the object using a polygonal modelling tool. For areas and cities where buildings differ in the architectural aspect, procedural modelling only further complicates and extends the modelling process. As shown in this work, procedural modelling is the best choice for creating a LOD1 building model of large areas or LOD2 building model for areas where buildings resemble one another. Furthermore, what CityEngine is missing are spatial analysis tools, including the simpler ones such as length measurements. However, this can be solved by having the 3D model exported and loaded into another software after making it in postprocessing.

Procedural modelling, once adopted, becomes a quick and effective way to generate 3D low to mid-level detail models. These models can be very useful in the field of urban planning, in the design of bases for smart cities, as a subject of visualization for tourist purposes and in many other areas. By building a 3D model of objects that surround us, we are making a step forward in the development of virtual reality.

6 REFERENCES

[1] Gorički, M. (2015). Modeliranje 3D kartografskog prikaza za potrebe planiranja pametnih gradova, *Diploma thesis,* Sveučilište u Zagrebu, Geodetski fakultet.

[2] van Oosterom, P. (2012). Summary of the Third International FIG Workshop on 3D Cadastres–Developments and Practices. In *Third International FIG Workshop on 3D Cadastres–Developments and Practices*, Shenzhen, China. http://www.gdmc.nl/3dcadastres/literature/3Dcad_2012_30.pdf (23.06.2017)

[3] Stoter, J. E. (2004). 3D cadastre. *PhD Thesis*, TU Delft, Delft, Nizozemska.

[4] Pouliot, J., Vasseur, M., & Boubehrezh, A. (2011, November). Spatial representation of condominium/co-ownership: comparison of Quebec and French cadastral system based on LADM specifications. In *2^nd^ International Workshop on 3D Cadastres*.

[5] Mađer, M. (2012). Model of linking cadastre with related registers, *Doctoral dissertation*, Geodetski fakultet, Sveučilište u Zagrebu).

[6] Roić, M. (2012). *Upravljanje zemljišnim informacijama-katastar*, Sveučilište u Zagrebu, Geodetski fakultet. ISBN 978-953-6082-16-2, Zagreb.

[7] Pouliot, J., Vasseur, M., & Boubehrezh, A. (2011, November). Spatial representation of condominium/co-ownership: comparison of Quebec and French cadastral system based on LADM specifications. In *2nd International Workshop on 3D Cadastres*, 274-289.

[8] Aditya, T., Iswanto, F., Wirawan, A., & Laksono, D. P. (2011, November). 3D cadastre web map: Prospects and developments. In *2nd International workshop on 3D cadastres*, 16-18.

[9] Lemmen, C., Van Oosterom, P., Thompson, R., Hespanha, J. P., & Uitermark, H. (2010). The modelling of spatial units (parcels) in the Land Administration Domain Model (LADM).

[10] Rahman, A. A., Hua, T. C., & Van Oosterom, P. (2011, May). Embedding 3D into multipurpose cadastre. In *FIG working week*.

[11] Billen, R., & Zlatanova, S. (2003). 3D spatial relationships model: a useful concept for 3D cadastre? *Computers, Environment and Urban Systems, 27*(4), 411-425. https://doi.org/10.1016/S0198-9715(02)00040-6

[12] Hassan, M. I., Ahmad-Nasruddin, M. H., Yaakop, I. A., & Abdul-Rahman, A. (2008). An integrated 3D cadastre–Malaysia as an example. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 37*(B4), 121-126.

[13] Stoter, J. E. & Ploeger, H. D. (2003). Property in 3D—registration of multiple use of space: current practice in Holland and the need for a 3D cadastre. *Computers, Environment and Urban Systems, 27*(6), 553-570. https://doi.org/10.1016/S0198-9715(03)00014-0

[14] Shen, Y. I. N. G., Renzhong, G. U. O., & Lin, L. I. (2011, November). Design and Development of a 3D Cadastral System Prototype based on the LADM and 3D Topology. In *The 2nd International Workshop on 3D Cadastres*, Delft, The Netherlands.

[15] Shojaei, D., Kalantari, M., Bishop, I. D., Rajabifard, A., & Aien, A. (2013). Visualization requirements for 3D cadastral systems. *Computers, Environment and Urban Systems, 41*, 39-54. https://doi.org/10.1016/j.compenvurbsys.2013.04.003

[16] Dimovski, V., Bundaleska-Pecalevska, M., Cubrinoski, A., & Lazoroska, T. (2011, November). WEB portal for dissemination of spatial data and services for the needs of the agency for real estate cadastre of the Republic of Macedonia (AREC). In *2011 International Workshop on 3D Cadastres*, 6-18.

[17] Vandysheva, N., Tikhonov, V., Van Oosterom, P. J. M., Stoter, J. E., Ploeger, H. D., Wouters, R., & Penkov, V. (2011). 3D Cadastre modelling in Russia. In *Proceedings of the FIG Working Week 2011" Bridging the Gap between Cultures" & 6th National Congress of ONIGT*, Marrakech, Morocco, 18-22 May 2011. International Federation of Surveyors (FIG); Ordre National des Ingénieurs Géomètres Topographes (ONIGT).

[18] Ying, S., Guo, R., Li, L., & He, B. (2012). Application of 3D GIS to 3D Cadastre in Urban Environment. *3rd International Workshop on 3D Cadastres: Developments and Practices*, 25-26.

[19] Guo, R., Li, L., He, B., Luo, P., Ying, S., Zhao, Z., & Jiang, R. (2011, November). 3D cadastre in China-a case study in Shenzhen City. In *Proceedings of 2nd international workshop on 3D cadastres*, Delft, The Netherlands, 16-18.

[20] Coors, V. (2003). 3D-GIS in networking environments. *Computers, Environment and Urban Systems, 27*(4), 345-357. https://doi.org/10.1016/S0198-9715(02)00035-2

[21] Stoter, J., & Salzmann, M. (2003). Towards a 3D cadastre: where do cadastral needs and technical possibilities meet? *Computers, environment and urban systems, 27*(4), 395-410. https://doi.org/10.1016/S0198-9715(02)00039-X

[22] van Oosterom, P. (2013). Research and development in 3D cadastres. *Computers, Environment and Urban Systems, 40*, 1-6. https://doi.org/10.1016/j.compenvurbsys.2013.01.002

[23] Watson, B., Müller, P., Veryovka, O., Fuller, A., Wonka, P., & Sexton, C. (2008). Procedural urban modelling in practice. *IEEE Computer Graphics and Applications, 28*(3), 18-26. https://doi.org/10.1109/MCG.2008.58

[24] Parish, Y. I. & Müller, P. (2001, August). Procedural modelling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 301-308.

[25] See http://en.wikipedia.org/wiki/CityEngine (23.06.2017)

[26] See http://www.esri.com/software/cityengine (23.06.2017)

[27] Piccoli, C. (2013). CityEngine for Archaeology. In Amsterdam: 3D GIS for mapping the Via Appia.

[28] Ribeiro, A., Duarte De Almeida, J., Ellul, C. (2014): Exploring CityEngine as a Visualisation Tool for 3D Cadastre, Dubai, United Arab Emirates.

[29] Viinikka, J. (2014). Adopting Procedural Information Modelling in Urban Planning. Aalto University.

[30] http://www.youtube.com/watch?v=zScyrwQLf0k (23.06.2017)

[31] http://mojoblak.srce.hr/public.php?service=files&t= 9601bd0acec842acddba1479b996ac15 (15.06.2018)

[32] Müller, P. et al. (2006). Procedural Modelling of Buildings. ACM Transactions on Graphics - *Proceedings of ACM SIGGRAPH, 25*(3), 614-623. https://doi.org/10.1145/1141911.1141931

[33] Wonka, P., Wimmer, M., Sillion, F., & Ribarsky, W. (2003). Instant architecture. In ACM SIGGRAPH, New York, NY, USA.

Authors' contacts:

Robert ŽUPAN, Assoc. Prof.
Faculty of Geodesy
Kačićeva 26, 10000 Zagreb, Croatia
01/4639-184, rzupan@geof.hr

Katarina ŽUPAN, mag. Ing. geod & geoinf.
Faculty of Geodesy
Kačićeva 26, 10000 Zagreb, Croatia
kzupan@geof.hr

Stanislav FRANGEŠ, Full Time Prof.
Faculty of Geodesy
Kačićeva 26, 10000 Zagreb, Croatia
01/4639-231, sfranges@geof.hr

Adam VINKOVIĆ, mag. ing. geod. et geoinf.
Faculty of Geodesy
Kačićeva 26, 10000 Zagreb, Croatia
01/4639-381, avinkovic@geof.hr