3D-GIS Parametric Modelling for Virtual Urban Simulation Using CityEngine

Ibrahim M. Badwi, Hisham M. Ellaithy and Hidi E. Youssef

Department of Architectural Engineering, Faculty of Engineering, Beni-Suef University, Beni-Suef, Egypt

ABSTRACT

Modelling and visualization of three-dimensional (3D) models for cities is a great challenge for computer software and graphics. Recently, 3D city modelling has grown due to advances in applications accompanying the information technology revolution. 3D Geographic Information Systems (3D-GIS) have evolved enormously due to the availability of large-scale 3D modelling techniques. These technologies have become very important in representing large cities and conducting various analyses in the city’s virtual environment to support urban decision-making. CityEngine is one of the most recent 3D-GIS modelling applications. CityEngine can be described as parametric modelling using Procedural Modelling (PM) to create 3D urban elements through macros and routines. This paper highlights the importance of 3D Procedural Modelling (PM) of cities in the GIS environment using ESRI CityEngine and presents a parametric concept for designing urban spaces. This issue has been addressed in three respects. First, discuss the concept and strength of parametric design. Second, the concept of procedural modelling and its power to generate complex 3D models using a set of rules is discussed. Finally, CityEngine was evaluated through a real-world case study of a neighbourhood in the new city of Beni-Suef, Egypt. The results confirm the effectiveness of CityEngine as a 3D-GIS modelling software that generates dynamic 3D models from 2D spatial data. While the results are promising, it is important to investigate more complex cases. The CityEngine modelling approach enables comprehensive urban analyses such as sequence vision, façade studies, urban fabric and character, and statistical operations based on attribute database.

1. Introduction

In the field of the fourth generation of Information and Communication Technologies (ICTs), emerging technologies have an impact on almost all fields worldwide (Talvitie 2004). As a result, several changes in urban planning and urban design methodologies are taking place. These changes will assess our understanding of cities and the development of future-oriented urban growth (Badwi, El-Barmelgy, and Ouf 2015). Engineers and city planners are looking for the best way to organize their cities in order to manage the complexity of urbanization. Three-dimensional modelling has become a key knowledge in many applications (Uddin et al. 2021).

Today, 3D models are widely used in exciting applications such as archaeology, animation, architecture, urban planning, reverse engineering, industrial design, museums, movies, games, toys, and web-design. Modelling 3D-GIS is the best way to visualize and simulate real cities (Singh, Jain, and Manda 2013). 3D modelling for cities has become feasible with the availability of recent applications and computers that have large memory, processing and graphics capabilities of today’s hardware. Using of the 3D modelling applications are not only in the fields of education and engineering, but also in the development of virtual models, computer games and filmmaking (Parish and Müller 2001), in addition to smart transport systems and modelling of tourist and archaeological sites.

The Geographic Information System (GIS) is one of the advanced technologies for collecting, compiling, storing, updating, retrieving, analysing and processing spatial data. GIS organizes geo-referenced spatial data stored in a database (Goodchild 2009). The strength of GIS lies in the link between the spatial features and their attributes (Database). Therefore, it has become very important in representing cities and performing a variety of analyses in the city’s virtual environment (Lin and Chen 2015), such as searching, queries and various statistical analyses, unlike other 2D applications that do not have these capabilities.

There are several urban modelling applications that use 2D spatial data in a GIS environment and can provide a variety of spatial and statistical analyses, as well as simulation and prediction functions such as the Future
Land Use Simulation (FLUS) model (Liu et al. 2017), the Conversion of Land Use (CLUE) model (Verburg et al. 2002), the Informal Hosing Growth Model (IHGM) (Badwi, El_Barmelgy, and El_Din Ouf 2021), the GIS-based land use change (GEOMOD) model (Pontius and Chen 2006), the what if? Planning support system (Klosterman 1999), the FUTURE Urban-Regional Environment Simulation (FUTURES) model (Meentemeyer et al. 2013), and the Urban Simulation (UrbanSim) (Waddell 2002).

In the past, 3D urban analyses that could not be performed in a 2D-GIS environment were described in descriptive text (Tsou 2014). Currently, GIS is no longer 2D, but 3D. GIS is experiencing a major shift towards 3D-GIS, where 2D geographic data can be modelled into a three-dimensional environment using advanced computers that provide excellent visualization with minimal effort and file size (Elwannas 2011) (Deren et al. 2009). 3D-GIS models will allow performing a variety of urban studies, such as urban fabric, urban pattern, building density, land registry, urban design, and traffic control, etc.

3D-GIS models enable urban designers to control building shape, height, and colour at the micro-level, furthermore environmental assessments of plazas and streets based on visual impact analysis at the macro level. On the basis of 3D-GIS models, it is possible to effectively control and improve the quality of urban space. There would be great progress when we design the urban space with an operational 3D-GIS system, like CityEngine (Xia and Qing 2004).

CityEngine is one of the most recent 3D-GIS modelling applications; it is a stand-alone desktop application for generating large-scale 3D models for the design, planning and modelling of urban environments. It was originally developed by Procedural Inc. and is currently being further developed by the Environmental System Research Institute (ESRI). The CityEngine relies on Procedural Modelling (PM) (Parish and Müller 2001), which is based on L-Systems Algorithms (Lindenmayer and Prusinkiewicz 1990). CityEngine uses an integrated programming language called ‘Computer Generated Architectures’ (CGA) based on the use of rules and algorithms to create urban models in a 3D environment (Watson et al. 2008). The CGA rule can control the number of floors, the height of each floor, the style and colour of the openings, the texture of the façades, the type of finishing and the shape of the ceiling and its height, as well as other parameters that control the external form of the building.

CityEngine provides a variety of complementary scripts that are used in the furniture of external spaces such as seats, lighting poles, shades, and various site layout elements, therefore these codes can be applied to any number of buildings or furniture types to perform a 3D model in a very short time. Additionally, the latest version of CityEngine shows that it is becoming a more specialized tool for urban design. CityEngine now offers integration with ArcGIS Urban. ArcGIS Urban can be used for visualizing master plans, monitoring development projects across the city, managing zoning codes and engaging with citizens (ESRI 2021).

2. The Concept of Parametric Modelling

Parametric modelling is a modelling process that allows the designer to change the geometry of the model by modifying its dimension values and evaluating design scenarios based on the integration of geometry variables and rules in an iterative design process (Burry 2003). Generative, parametric, and algorithmic models are all synonyms of computational design, intended to use ‘Artificial Intelligence’ and advanced mathematics to generate complex forms of virtual reality. Parametric modelling is implemented using computer programming code such as a script to define the dimensions and geometry of the model. The special nature of parametric models arises from the results obtained on the basis of the mathematical probabilities based on statistics, since it enables the user to change the values of one variable and to deduce the effects on other variables, and thus creating a new design scenario based on the values of all variables, therefore there is no need to redraw the model whenever it needs a change as shown in Figure 1. In other words, parametric modelling allows the designer to modify the entire shapes of the model, not just individual members.

Currently, the concept of modelling using parameters has been used in a wide range of ready-made programs such as: CATIA (Computer Aided three-dimensional Interactive Application), APOLLON, SolidWorks, Grasshopper 3D, BIM: Autodesk Dynamo, and GIS: CityEngine (Badwi, El-Barmelgy, and Ouf 2015), (Fink and Koenig 2019), (Ribeiro, de Almeida, and Ellul 2014), and (Dobraja 2015).
Modelling using these techniques can contribute to the design stages of large projects, as well as solving operational problems. Many advantages are defined, including centralized data management, descriptive information (spatial data), improve cost control, construction schedule maintenance, and dealing with stakeholders from various backgrounds (Shaqour 2021).

The Urban Parametric Model (UPM) can be defined as a computer-based urban model that describes the components of the urban system and transforms each component into a series of variables that control the urban element composition in third dimension such as building, road, sidewalk, lighting poles, trees, shades, seats … etc. The parametric design approach provides a methodology for expressing the selected design elements as variables (ElBatran and Ismaeel 2021). The user can control and modify the initial values of these variables, as well as integrate various pre-defined rules and criteria, producing a large number of alternatives for a single building footprint. (ElBatran and Ismaeel 2021). Thus, the urban context’s vision can be restored, and an infinite number of scenarios can be generated.

For this reason, the parametric design tools allow the design to remain open for modification with a high level of detail in order to understand the essential qualities and elements of the proposed design. Figure 2 shows the importance of parametric design. For example, in the case of a residential building, the parameters can be described as the total height of the building, the number of floors, the height of the typical floors, the height of the ground floor, the width of the windows, the height of the windows, the number of windows, the main façade, the side façades … etc, as shown in Figures 3 and 4.

3. Background of CityEngine

The Swiss company Procedural Inc. is the main developer of the CityEngine program and one of the leading companies in the design and development of 3D programs and games. The program was initially developed by the Swiss Institute of Technology (ETH Zurich) through the main program developer Dr. Pascal Müller, who designed and programmed a set of techniques to generate 3D architectural elements using a technique called Procedural Modelling (PM).

In 2001, Dr. Müller and others published many papers to clarify how procedural modelling works (Müller et al. 2006). In 2007, he published a paper on how procedural modelling works on building façades, using photographs of building façades (Müller et al. 2007). Weber et al. published a paper on interactive urban modelling in 2008 by developing a 3D urban simulation system over time (Weber et al. 2009). At the end of 2008, the first version of CityEngine was released, which specializes in developing 3D models using procedural modelling in a fast and easy way.

Finally in 2011 the Environmental Systems Research Institute (ESRI) acquired CityEngine from Procedural Inc and included the program in the family of ESRI-GIS applications (ArcGIS – ArcCatalog – ArcScene … etc.).

3.1. The Modelling Framework of CityEngine

CityEngine software provides a framework for generating large 3D Models from 2D-GIS data. CityEngine integrates the city’s geographic data with other infrastructure data such as water, sewage, and electricity network, as well as integrating zoning legislation and building height limitations. The main information layer in the city model is the 2D-GIS spatial layers. 3D-GIS allows architects and city
Figure 3. (a-d) Different layout compositions on the same parcel footprint: a. Linear blocks, b. Closed blocks, c. Patio houses, d. Tower blocks (Steinø and Obeling 2014).

Figure 4. (a-d) Examples of different building shapes with the same parcel footprint: a. Simple slab, b. horizontal setbacks, c. Vertical setbacks, d. vertical setbacks and height variations (Steinø and Obeling 2014).
planners in simulating virtual 3D urban environments, shaping better novel structures, and predicting potential problems (Ghorbanian and Shariatpour 2019).

When generating a 3D city model or residential neighbourhood from 2D-GIS geographic data, a set of operations is carried out to create the 3D model. Initially, the road network is constructed, and then the resulting blocks are subdivided into parcels. Finally, 3D building models are created based on a horizontal projection of a parcel or building footprint using one of the CGA rules (ESRI 2021), as shown in Figure 5. During this stage, landscape, water features, vehicles ways, street furniture, etc., can be generated based on one of the CGA rules and 3D assets. Infrastructures above and below the Earth's surface are also represented in this stage.

The final 3D-GIS urban model (Figures 6 and 7) can be exported to other software specialized in 3D visualization, presentation, or analysis (e.g. ArcGIS, Maya, 3ds Max, Autodesk Revit, Google Earth, and RealityServer). The workflow is very flexible within the CityEngine environment, and the context of each stage can be adjusted based on the user’s vision. The user can store all of the CityEngine scenes as different data layers to represent various stages (Esri 2014), as shown in Figure 8.

3.2. Concept of Procedural Modelling (PM) in CityEngine

City Engine was developed to facilitate the generation of 3D cities from existing 2D-GIS data. The main concept of the CityEngine is the ‘procedural’ approach towards efficiently modelling. The user develops a code-based ‘procedure’ that represents a series of commands executed instead of a ‘classic’ manual user interaction; this task is described as ‘abstract’ in the rule file. (Ribeiro, de Almeida, and Ellul 2014).

Procedural Modelling (PM) has the ability to generate complex 3D models using a set of parameters and variables to control the formation of buildings. Procedural Modelling (PM) is a type of computer programming paradigm that allows the user to simulate the characteristics of the building 3D model (Ghorbanian and Shariatpour 2019), and to automatically generate an infinite number of 3D architectural components.

Procedural Modelling uses the Computer Generated Architecture Rules (CGA-Rules) that specify a series of well-structured coding statements, subroutines and functions in the program to achieve the desired output. The Commands provided in CityEngine’s CGA shape grammar, such as ‘extrude’, ‘split’ or ‘texture’ are widely

![Figure 5](image5.png) The basic concept of generating 3D model from 2D-GIS road network map.

![Figure 6](image6.png) Modelling a street using the road rule and the surrounding landscape (Van Maren, Shephard, and Schubiger 2012).

![Figure 7](image7.png) Generating Large-Scale urban environment with CityEngine (ESRI CityEngine 2021).
known commands in 3D modelling applications (Ghorbanian and Shariatpour 2019). In addition, the models produced are interactive and have the capability to be adapted to the new rules at any time. A single CGA rule can be used to generate a number of 3D models. For example, the rule may apply the attributes of the urban element stored in 2D-GIS data, such as floor number, floor height, windows and doors width, roof type, wall material type, etc. The more urban element attributes the more accurate the model generated.

Researchers Parish & Müller are the first to apply this technique to a 2D footprint of building and transform it into a 3D building (Parish and Müller 2001). In addition, this building is covered with a special pattern of openings on the façade of the 3D building. The researchers explained the possibility of constructing and modelling an entire city in a very short time without the need to model each building separately (Parish and Müller 2001). Figures 9 and 10 summarize the steps for creating a 3D model using Procedural Modelling (PM).

The 3D modelling process can be divided into four stages: In the first stage, the 2D geographic data is imported into CityEngine as spatial layers. In the second stage, the parameters that determine the building characteristics are inserted, such as the height of the building, the number of floors, the width and height of the openings, the width and height of the doors, the external columns, the shape of the upper floors, etc. In the third stage, the urban developer develops a CGA script that includes all functions and subroutines. The CGA script translates the building information into 3D elements, which are then combined to form the final 3D model.
In the fourth stage, the code is applied to the entire 2D city plan. At this step, the predefined assets in CityEngine can be used, which contain textures, opening shapes, or images describing the building’s exterior composition. So the final output is a 3D model for all the city buildings. Figure 11 (a-c) shows the process of applying the CGA rules and the insertion of building features at each stage.

The properties of the CGA rule are used in all the city buildings and can be modified through a separate interface for transactions entered into the CityEngine program. The following example shows how to model a simple building using the CGA rule, consisting of 3 floors and each role has 4 slots using the Split Grammar function as in Figure 12.

First, the characteristics of the building are determined, such as the height of the building, the height of the ground floor and of each floor, the width of the façade divided module, the colour of the walls and other characteristics as follows:

- Attr height = 12(12M)
- Attr groundfloor_height = 5(5M)
- Attr floor_height = 3 (3M)
- Attr tile_width = 4.5 (4.5M)
- Attr wall_color = "# FFFAA6 " (yellow)

Second, the 3D model can be created using the Extrusion function by using the building height which defined in the building characteristics definitions

\[ \text{Lot} \rightarrow \text{Extrude (height)} \]
\[ \text{Building} \]

Third, the front, back and side façades of the building are defined, where each façade is divided horizontally and vertically to form each side of the building.

\[ \text{Building} \rightarrow \text{Comp(f)} \text{ (front: FrontFacade | side: SideFacade | back: BackFacade | top: Roof)} \]

Fourth, using the Split function to define the façade and perform the splitting operation, each façade of the building can be formulated separately in order to

\[ \text{Figure 11. (a–c) The 3D model evolved by adding different details at each stage of the CGA rule. (ESRI 2021) (Radies 2013) (Kim and Wilson 2015).} \]

\[ \text{Figure 12. Modelling of simple elevation using the Split Grammar function.} \]
simulate the real façade of the building. For example, the main façade can be divided into a group of floors, by setting the height of the ground floor and the typical floors using the Y-direction split operator. The following example shows how to vertically divide the main façade of the building.

FrontFacade —> Split (y) { groundfloor_height: Groundfloor | (~floor_height: Floor)*}

Fifth, each floor is divided into several parts, each of which contains a window, a door, or a wall . . . etc. depending on the type of architectural element on the façade. For example, the main façade can be divided into a set of vertical divisions/ tiles using the X-direction split operator. The following example shows how to horizontally divide the ground floor of the main façade.

Groundfloor —> Split (x) { 1: Wall | (~tile_width: Tile)* | ~tilewidth: EntranceTile | 1: Wall }

The following code shows how to divide the part that contains the entrance door on the main façade. Figure 13 shows an example of the graphical representation of a rule file in the CityEngine software.

EntranceTile —> Split (x) {~1: SolidWall | 2: Split(y){2.5: Door | ~2: SolidWall} | ~1 ~* SolidWall }

4. Case Study

The study area was chosen for one of the housing projects located in the new city of Beni-Suef governorate. This site has been assigned to one of the housing associations for the construction of a social housing project.

This project was chosen to measure the capacity of CityEngine to construct 3D models of large areas using the Procedural Modelling (PM) programming language with the application of codes on residential and service buildings. The study area is approximately 70 acres. The project consists of 156 residential buildings, a commercial market, a primary school, two kindergartens, a mosque and an administrative building dedicated to postal services and firefighting, as well as parking areas and green spaces, as shown in Figure 14.

The 2D-GIS data was prepared using the AutoCAD and ArcGIS 10.8 programs; the geodatabase consists of layers representing residential buildings, roads, walkways, schools, commercial markets, mosques, service buildings, parking areas, green spaces, and playgrounds. The geodatabase was projected based on the Egyptian Transverse Macerator Coordinate System (ETM), according to the Geodetic Datum, Helmert Ellipsoid 1906. Figure 15 shows the flowchart of the procedural modelling using CityEngine.

5. Building Models and Results

After creating the 2D-GIS geodatabase and entering the features attribute data, CityEngine R-2019 was used to build 3D models using Procedural Modelling (PM) and creating the rule files. The rules file contains a set of rules and properties responsible for forming the 3D urban element. In previous versions of CityEngine, a rule file can be created by editing or visual programming interface, but newer versions further than 2017 can only be created by editing the code, as shown in Figure 16.

Attr building_height = 20
@startRule
Lot —> Color (1,1,0)
Extrude(world.y, building_height)
BuildingMass

In the beginning, all parameters, constants, and methods for forming façades and external cladding are defined in order to develop each façade code. Then,
these parameters and properties are used as inputs in the rule file, which can be set or changed by the user in order to produce the required architectural formation. The three-dimensional model is created through a series of operations such as Extrude, Rotate, Copy, Move, ... etc. All of these functions can be used in the
rule code to create 3D architectural models which can be applied to any 2D polygon with the same characteristics in order to convert it into 3D elements. The following sections will clarify the rules used to generate the 3D model of this project.

5.1. Residential Buildings’ Modelling

Real images of the building façades were used to represent the building 3D model. Since the façade of the building has a protruding block and a countersunk block, the façade image is divided into several parts; each image represents a complete part with a single surface that does not contain any level.

The modelling process for the residential building begins with the front façade function, which requires the overall height of the building, the number of floors, the height of the ground floor and the typical floors, as well as the height of the roof parapet.

The next step is to divide the façade vertically to create the building floors by defining the locations of starting and threshold locations for both windows and balconies using the front façade splitting tool. As well as dividing the façade horizontally so as to define the widths of the windows and balconies. Figure 17 shows the horizontal and vertical division of the main façade.

The following programming codes show a portion of the written code used to create a 3D residential model within the project, the final output shown in Figure 18.

```
@StartRule
Generate ~> alignScopeToGeometry(zUp,any,world.lowest)
alignScopeToAxes(y)
FacadeElevationCheck
FacadeElevationCheck ~> case scope.elevation-initialShape.origin.py
| GroundfloorHeight*0.5 || Pattern == "vertical": # is face above ground?
```

Figure 16. Creation of a rule file by editing or visual programming interface in CityEngine (ESRI 2021).

Figure 17. The division of the main façade into vertical sections.

Figure 18. The 3D residential model.
split(y){ ~1: UpperFloors | UpperfloorHeight*0.5: Wall} else:
split(y){GroundfloorHeight: GroundFloor | ~1: UpperFloors | UpperfloorHeight*0.5: Wall} GroundFloor —> case Pattern == "grid":
  split(y){ ~1: UpperfloorHeight*0.5: Wall} ~1: VerticalPattern | ~1: Wall} case Pattern == "horizontal-closed":
split(x){~TileWidth: Wall | TileWidth*0.5: Window | ~1: Wall} else:
Window UpperFloors —> case Pattern == "vertical":
VerticalPattern case Pattern == "grid":
split(y){ ~1: UpperfloorHeight*0.3: Wall} ~1: VerticalPattern
  {UpperfloorHeight*0.5: VerticalPattern
  | ~1: Wall}* | UpperfloorHeight*0.5: VerticalPattern} case Pattern == "horizontal-open":
split(y){ ~1: UpperfloorHeight*0.2: Wall} ~1: VerticalPattern
  {UpperfloorHeight*0.8: Window
  | ~1: Wall}* | UpperfloorHeight*0.8: Window} case Pattern == "horizontal-closed":
split(y){ ~1: UpperfloorHeight*0.3: Wall} ~1: VerticalPattern
  {UpperfloorHeight*0.4: Window
  | ~1: Wall}* | UpperfloorHeight*0.4: Window} else:
Window VerticalPattern —> split(x){ ~1: TileWidth*0.6: Wall | {TileWidth*0.7: Window | ~1: TileWidth*0.3: Wall}* | TileWidth*0.7: Window | ~1: TileWidth*0.6: Wall} Wall —> color(SecondaryColor)
Window —> set(material.specular.r,0.5) set(material.specular.g,0.5)
set(material.specular.b,0.5) set(material.reflectivity,0.3)

5.2. Roads Modelling

The roads were represented using a built-in rule inside CityEngine program including road width, pavements, lighting columns. This rule was modified to be suitable with the study area.

Roads were represented using one of the built-in rules of CityEngine program, including road width, pavements, lighting poles. This rule has been modified to be appropriate for the study area. CityEngine presents the polyline features as roads or corridors and gives the possibility to choose the width of the road, the width of the lanes and the width of the left and right pavements. For example, when applying this rule to roads, the user defines road material, a sidewalk, traffic signals, a pedestrian crossing, a bus lane, a car lane, and a bike lane.

In addition, the rule allows the user to represent parking areas, cars, people, buses and bicycles with the density needed in the form of a 3D obj format. Figures 19 and 20 show the 3D elements used from the program library, as well as an example of road modelling in CityEngine. The following programming codes explain a portion of the variables used to generate a 3D road model in the project.

# Roads CGA Rule
Level_of_detail: Low (optional)
Street Layout
Nbr_of_left_lanes: 2
Nbr_of_right_lanes: 2
Lane_width: 3
Median_width: 2
Lawns: true

Figure 18. An example of a 3D model generated from a real two-dimensional image.
Arrow_marking: true
Crosswalk
Crosswalk_width: 0
Crosswalk_style: European (not used)
Lights
Traffic_lights: true
Lamps: true
Lamp_distance: 30
Trees
Tree_percentage: 100
Tree_distance: 1
Tree_max_height: 1
Traffic Density
Vehicles_per_km: 10
Bus_Percentage: 0
Sidewalk
People_percentage: 10
Props_percentage: 10
Sidewalk_height: 0.2

5.3. Green Areas Modelling

The rule was used to represent gardens, including trees, grass, and footpaths. This rule allows the user to define the width of the paths, the height of the sidewalks and the direction of the paths within the gardens. The following is an explanation of the most important features of the garden rule.

# ——— Garden CGA Rule
@Group("MODEL OPTIONS",0) @Order(1)
@Range("Formal","Natural")
attr Green_Space_Type = “Formal”

@Group("MODEL OPTIONS") @Order(2) @Range(0, 0.2)
attr Sidewalk_Height_Match = 0.2
# ———— Pathways CGA Rule
@Group("PATHWAYS",1) @Order(1) @Range(10,50)
attr Unit_Width = 15
@Group("PATHWAYS") @Order(2) @Range(0,180)
attr Rotation = 0

Figure 19. 3D models stored in the program (.Obj) used for furnishing of the external spaces.

Figure 20. Example of intersections between the main and secondary roads, including sidewalks, trees and lighting Poles.
5.4. Football Playground Modelling

The playground is represented by a rule written for it; this rule uses an actual image of the playground to give it a realistic appearance; as shown in Figure 21. The following programming code explains the part of the CGA rule that is used to represent the playground by defining the goalpost characteristics and specifying its width and height. In addition, the definition of the images used as textures for the playground and the 3D objects representing the goalpost and the football.

```plaintext
attr Orientation_Change = false
tag Goalpost_Width = 6
tag Goalpost_Height = 1.85
tag Goalpost_Depth = 1.2
tag Number_Balls = 7
soccerFieldTex = "assets/soccerField/soccer.jpg"
goalpostAsset = "assets/soccerField/soccer_Goal.obj"
Ball = "assets/soccerField/Ball.obj"
```

The final result is a three-dimensional (3D) model for the housing project, including residential and service buildings, roads, sidewalks, green areas, playgrounds, as well as landscape and architectural features such as trees, people cars, and bicycles that make the model realistic, as shown in Figures 22, 23, 24 and 25.

6. Discussion & Conclusions

CityEngine is a powerful parametric urban design tool. CityEngine uses the Procedural Modelling (PM) technique to construct CGA rules and algorithms capable of generating a large 3D model of a city from a 2D building footprint based on the buildings descriptive characteristics. In a few minutes, a city 3D model can be generated at a low cost compared to other applications.

CityEngine is based on a simple scripting language, making it relatively simple to learn for architects and planners. (Steinø and Obeling 2014). However, using a scripting language to express the urban designer’s ideas is a challenge when it comes to using CityEngine in the urban design process because the programming interface is not intuitive to lay-users. The procedural modelling with CGA requires minimal programming knowledge. The ability to use Esri-ArcGIS programming languages is a good starting point for GIS developers to build CGA rules (Dobraja 2015).

This paper used CityEngine to model a residential project in the new city of Beni-Suef. The modelling process involves collecting data, creating 2D-GIS data,
importing GIS layers to CityEngine, writing CGA rules and applying them to each corresponding layer, and finally creating a 3D model.

The obtained results confirm the effectiveness of CityEngine as a 3D modelling program in urban planning processes; performs well in the generation of dynamic 3D models from high-quality 2D spatial data. Furthermore, you can change the properties of any 3D object by manually modifying its parameters or through the CGA rule file. The 3D urban analyses can be performed such as skyline, landscape, sequence vision, visibility, and enclosure analysis. Moreover, the analyses based on attribute data can be performed well, such as searching, inquiry, query, and statistical analyses related to areas or heights.

CityEngine includes a library of shapes used to create shades, fences, sidewalks, trees, shrubs, light columns, cars, furniture elements in open spaces, and a variety of landscape features (Steinø and Obeling 2014). CityEngine has the capabilities to simulate

Figure 23. 3D model of the housing project using CityEngine.

Figure 24. The commercial centre, the school and the mosque within the service area of the housing project.
shadows by months of the year, as well as displaying aerial images, satellite images and Digital Elevation Models (DEM).

The limitation of CityEngine is the additional functions required to create 3D models capable of simulating urban design. The main issue is how to interpret arcs and circles because the program does not support these objects. CityEngine divides the façade using the split function, where the separation is done on the X, Y-axis, while it is not possible to draw curved shapes using the freehand drawing tool. (Saldana 2015). Therefore, CityEngine users find it extremely difficult to model historical buildings due to their unique characteristics and the fact that, in most cases, such buildings do not share repetitive patterns, and these facades feature numerous organic and geometric details.

Figure 25. Landscape features, spaces furniture, roads, sidewalks, traffic signals, and pedestrian crossings.
Another issue is that CityEngine does not accept the addition of polygons containing internal spaces such as courtyards. So once the polygons containing holes are imported, the software converts them into a single polygon and fills these holes (Saldana 2015).

Furthermore, CityEngine exports the data as layers to a new GIS database in the form of non-editable layers in traditional GIS software. These data can only be exchanged through the ArcScene and ArcCatalog program included in the ArcGIS package. This means that the data model is incomplete and it will be interesting to see where this development leads (Dobraja 2015).

Future studies for this research focus on adding new Levels Of Detail (LOD) to represent objects as close to reality as possible and adding infrastructure paths to the housing project by specifying maintenance sites so that they are represented in an integrated manner with urban elements (Saldana 2015), (Dobraja 2015). In addition, exporting 3D model output to a web interface (WebScene) will create a platform to support decision-making and regulate the city management process (Schaller et al. 2015). This methodology can support urban design processes and provide the partners with a realistic representation of urban elements to achieve compatibility between different projects and intelligently via a web-based interface.

**Author bio**

Ibrahim M. Badwi received his Bachelor’s degree in Architectural Engineering from Cairo University; he successfully received GIS/DBMS Scholarship Diploma from Information Technology Institute (ITI), then a M.Sc. and Ph.D. degree from Cairo University. Dr. Badwi always preserved a dual career between practices of urban planning and GIS besides academic work. He taught at Beni-Suef University, Faculty of Art Ain Shams University, Al-Sa’wa High engineering institute, and Giza engineering institute (GEI) in Egypt. He has served as a consultant for the General Authority for Physical Planning (GOPP) in Egypt.

Hisham M. Taher Ellailthy received his Bachelor’s degree in Architectural Engineering from October High Institute for Engineering and Technology, Egypt. He successfully received Diploma in Cultural Anthropology, from Institute of African Research and Studies, Cairo University, Egypt. Then a M.Sc. and Ph.D. degree from Cairo University. Dr. Hisham always preserved a dual career between practices of architecture and heritage conservation besides academic work. He taught at Beni-Suef University, Faculty of Archaeology Cairo University, and Giza engineering institute (GEI) in Egypt. He has served as a consultant for the Architectural Design, Urban Anthropology in Egypt.

Hidi E. Youssef is a lecturer of urban Planning at Beni-Suef University. She received her M.Sc. and Ph.D. from Cairo University in regional planning. She authored several researches in the fields of architectural and urban planning. She is also interested in the field of heritage conservation. She taught at and Giza Engineering institute in Egypt, and served as a consultant for the General Authority for Physical Planning (GOPP) in Egypt.

**Disclosure statement**

No potential conflict of interest was reported by the author(s).

**Funding**

The authors have no funding to report.

**ORCID**

Ibrahim M. Badwi [http://orcid.org/0000-0002-8640-5513](http://orcid.org/0000-0002-8640-5513)

Hidi E. Youssef [http://orcid.org/0000-0001-6818-2371](http://orcid.org/0000-0001-6818-2371)

**References**

ArcGIS CityEngine (2021). Available at: [www.esri.com/software/cityengine](http://www.esri.com/software/cityengine)

Badwi, I. M., M. M. El_Barmelgy, and A. S. El_Din Ouf. 2021. “Modeling and Prediction of Expected Informal Growth in the Greater Cairo Region, Egypt.” *Environment and Planning B: Urban Analytics and City Science*, 49 (2): 23998083211 002207.

Badwi, I. M., M. M. El_Barmelgy, and A. S. E. D. Ouf. 2015. “Modeling and Simulation of Greater Cairo Region Urban Dynamics Using SLEUTH.” *Journal of Urban Planning and Development* 141 (4): 04014032. doi:10.1061/(ASCE) UP.1943-5444.0000193.

Burry, M. 2003. “Between Intuition and Process: Parametric Design and Rapid Prototyping.” In *Architecture in the Digital Age: Design and Manufacturing*, 148–162. RMIT University.

Çağışkan, O. 2012. “Design Thinking in Urbanism: Learning from the Designers.” *Urban Design International* 17 (4): 272–296. doi:10.1057/udi.2012.21.

Deren, L., W. Mi, H. Qingwu, and H. Fen. 2009. “On Three-dimensional Visualization of Geospatial Information: Graphics Based or Imagery Based?” *Annals of GIS* 15 (2): 75–84. doi:10.1080/19475680903250749.

Dobraja, I. (2015). “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*.

ElBatran, R. M., and W. S. Ismael. 2021. “Applying a Parametric Design Approach for Optimizing Daylighting and Visual Comfort in Office Buildings.” *Ain Shams Engineering Journal* 12 (3): 3275–3284. [https://doi.org/10.1016/j.asej.2021.02.014](https://doi.org/10.1016/j.asej.2021.02.014)

Elwannes, R. (2011). “3D GIS: It’sa Brave New World”. FIG Working Week, Marrakech, 9.

Esri. 2014. “ArcGIS Resources. 3D Urban Content Creation and Design Using CityEngine”. Available at: [http://resources.arcgis.com/en/communities/cityengine/01w900000008000000.html](http://resources.arcgis.com/en/communities/cityengine/01w900000008000000.html)
Ribeiro, Rades, Müller, Müller, Lindenmayer, Kim, Goodchild, Ghorbanian, Fink, ESRI (2021). “Integrated Parametric Urban Design in Grasshopper/Rhinoceros 3D-Demonstrated on a Master Plan in Vienna”.

Ghorbanian, M., and F. Shariatpour. 2019. “Procedural Modeling as a Practical Technique for 3D Assessment in Urban Design via CityEngine.” Irving University of Science & Technology 29 (2): 255–267.

Goodchild, M. F. 2009. “Geographic Information Systems and Science: Today and Tomorrow.” Annals of GIS 15 (1): 3–9. doi:10.1080/19475680903250715.

Kim, K., and J. P. Wilson. 2015. “Planning and Visualising 3D Routes for Indoor and Outdoor Spaces Using CityEngine.” Journal of Spatial Science 60 (1): 179–193. doi:10.1080/14498596.2014.911126.

Klosterman, R. E. 1999. “The What if? Collaborative Planning Support System.” Environment and Planning B: Planning and Design 26 (3): 393–408. doi:10.1068/b260393.

Lin, H., and M. Chen. 2015. “Managing and Sharing Geographical Knowledge in Virtual Geographic Environments (Vges).” Annals of GIS 21 (4): 261–263. doi:10.1080/19475683.2015.1099568.

Lindenmayer, A., and P. Prusinkiewicz. 1990. The Algorithmic Beauty of Plants. Vol. 1. New York: Springer-Verlag.

Liu, X., X. Liang, X. Li, X. Xu, J. Ou, Y. Chen, … F. Pei. 2017. “A Future Land Use Simulation Model (FLUS) for Simulating Multiple Land Use Scenarios by Coupling Human and Natural Effects.” Landscape and Urban Planning 168: 94–116. doi:10.1016/j.landurbplan.2017.09.019.

Meentemeyer, R. K., W. Tang, M. A. Dorning, J. B. Vogler, N. J. Cunniffe, and D. A. Shoemaker. 2013. “FUTURE: Multilevel Simulations of Emerging Urban–rural Landscape Structure Using a Stochastic Patch-growing Algorithm.” Annals of the Association of American Geographers 103 (4): 785–807. doi:10.1111/0004-5608.2012.707591.

Müller, P., P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool (2006). “Procedural Modeling of Buildings”. In ACM SIGGRAPH 2006 Papers (pp. 614–623).

Müller, P., G. Zeng, P. Wonka, and L. Van Gool. 2007. “Image-based Procedural Modeling of Façades.” ACM Transactions on Graphics (TOG) 26 (3, August): 85. doi:10.1145/1276377.1276484.

Parish, Y. I., and P. Müller (2001, August). “Procedural Modeling of Cities”. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (pp. 301–308).

Pontius, R. G., Jr, and H. Chen (2006). “GEOMOD Modeling”. Clark University, 44. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.4466&rep=rep1&type=pdf (Accessed February 2021).

Radies, C. (2013). “Procedural Random Generation of Building Models Based Geobasis Data and of the Urban Development with the Software CityEngine”. Bernburg, Germany, 175–184.

Ribeiro, A., J. P. de Almeida, and C. Ellul (2014). “Exploring CityEngine as a Visualisation Tool for 3D Cadastre”. In 3D Cadastres 2014-4th International FIG 3D Cadastre Workshop (pp. 197–218).

Saldana, M. 2015. “An Integrated Approach to the Procedural Modeling of Ancient Cities and Buildings.” Digital Scholarship in the Humanities 30 (suppl_1): i148–i163. doi:10.1093/lilc/fqv013.

Schaller, J., O. Ertac, S. Frelter, C. Mattos, and Z. Rajcevic. 2015. “Geodesign Apps and 3D Modelling with CityEngine for the City of Tomorrow.” In Digital Landscape Architecture, 59–70.

Shaqour, E. N. 2021. “The Role of Implementing BIM Applications in Enhancing Project Management Knowledge Areas in Egypt.” Ain Shams Engineering Journal 13 (1): 101509. https://doi.org/10.1016/j.asej.2021.05.023.

Singh, S. P., K. Jain, and V. R. Manndla. 2013. “Virtual 3D City Modeling: Techniques and Applications.” ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-2/W2: 73–91. doi:10.5194/isprsarchives-XL-2-W2-73-2013.

Steine, N., and E. Obeleng. 2014. “Developing a Parametric Urban Design Tool. Some Structural Challenges and Possible Ways to Overcome Them.” Architecturae et Artibus 6 (1): 51–57.

Talvitie, J. 2004. “Incorporating the Impact of ICT into Urban and Regional Planning.” European Journal of Spatial Development 10: 1–32.

Tsou, M. H. 2014. “Big Data: Techniques and Technologies in Geoinformatics.” Annals of GIS 20 (4): 295–296. doi:10.1080/19475683.2014.944934.

Uddin, M. N., Q. Wang, H. H. Wei, H. L. Chi, and M. Ni. 2021. “Building Information Modeling (BIM), System Dynamics (SD), and Agent-based Modeling (ABM): Towards an Integrated Approach.” Ain Shams Engineering Journal 12 (4): 4261–4274. doi:10.1016/j.asej.2021.04.015.

Van Maren, G., N. Shephard, and S. Schubiger (2012). “Developing with Esri CityEngine”. Proceedings of the ESRI Developer Summit, San Diego, CA, USA, 26–29. (available at: https://proceedings.esri.com/library/userconf/devsummit12/papers/developing_with_esri_cityengine.pdf) (Accessed Feb. 2021).

Verburg, P. H., W. Soepboer, A. Veldkamp, R. Limpiada, V. Espaldon, and S. S. Mastura. 2002. “Modeling the Spatial Dynamics of Regional Land Use: The CLUE-S Model.” Environmental Management 30 (3): 391–405. doi:10.1007/s00267-002-2630-x.

Waddell, P. 2002. “UrbanSim: Modeling Urban Development for Land Use, Transportation, and Environmental Planning.” Journal of the American Planning Association 68 (3): 297–314. doi:10.1080/01944360208976274.

Watson, B., P. Müller, O. Veryovka, A. Fuller, P. Wonka, and C. Sexton. 2008. “Procedural Urban Modeling in Practice.” IEEE Computer Graphics and Applications 28 (3): 18–26. doi:10.1109/MCG.2008.58.

Weber, B., P. Müller, P. Wonka, and M. Gross. 2009. April. “Interactive Geometric Simulation of 4d Cities”. In Computer Graphics Forum. Vol. 28, 481–492. Oxford, UK: Blackwell Publishing.

Xia, Z., and Z. Qing. 2004. “3D City Models Based Spatial Analysis to Urban Design.” Annals of GIS 10 (1): 82–86. doi:10.1080/1082400409480658.