

## PROCEDURAL 3D FOUNTAIN MODELING

Meryem KURT\*, Department of Computer Engineering, Ankara Yıldırım Beyazıt University, Turkey, meryemkurt1020@gmail.com

([ID](https://orcid.org/0000-0002-8259-0264) <https://orcid.org/0000-0002-8259-0264>)

Abdullah BULBUL, Department of Computer Engineering, Ankara Yıldırım Beyazıt University, Turkey, abulbul@ybu.edu.tr

([ID](https://orcid.org/0000-0002-2527-2729) <https://orcid.org/0000-0002-2527-2729>)

Received: 01.04.2021, Accepted: 25.07.2021

\*Corresponding author

Research Article

DOI: 10.22531/muglajsci.908164

### Abstract

*Procedural modeling is a method that determines production rules and allows for developing effective structural design in architecture. Fountain (Shadirvan) is an important component of Turkish architecture and having a symmetrical structure and distinctive elements makes fountains suitable for procedural modeling. In this study, we propose a procedural method to design different fountain models similar to ones that exist in Turkish architecture. We specify the parameters by looking at the texture, shape, and quantity of observed fountain images for production rules. Based on the selection of parameters, different fountain models are generated in a procedural manner. We present several different models, determined parameters, and parameter values. We explain how we create shapes such as polygon and pyramid. We show samples of 3d fountain models that our model can produce. Also, we conducted an experiment to evaluate the proposed system in terms of its expressiveness and usability. In this experiment, we asked subjects to produce 3D fountain models similar to the ones in provided images by selecting parameters of the proposed model or by traditional 3D modeling tools.*

**Keywords:** Procedural modeling, Symmetry, Fountains, Mesh

## YORDAMSAL ÜÇ BOYUTLU ŞADIRVAN MODELLEME

### Özet

*Yordamsal modelleme, üretim kuralları belirleyen ve mimaride etkili yapısal tasarım geliştirmeye izin veren bir yöntemdir. Şadirvan, Türk mimarisinin önemli bir bileşenidir ve simetrik bir yapıya ve ayırt edici unsurlara sahip olması, şadirvanları yordamsal modellemeye uygun kılar. Bu çalışmada, Türk mimarisinde var olan şadirvanlara benzer birbirinden farklı şadirvan modellerini tasarlamak için bir yordamsal yöntem öneriyoruz. Üretim kuralları için gözlemlenen şadirvan görüntülerinin dokusuna, şekline ve miktarına bakarak parametreleri belirliyoruz. Parametrelerin seçimine bağlı olarak, yordamsal bir şekilde birbirinden farklı şadirvan modelleri üretilir. Birkaç tane birbirinden farklı şadirvan modellerini, belirlenen parametreleri ve parametre değerlerini sunuyoruz. Çokgen ve piramit gibi şekilleri nasıl oluşturduğumuzu açıklıyoruz. Modelimizin üretebileceği üç boyutlu şadirvan modellerinin örneklerini gösteriyoruz. Ayrıca, önerilen sistemi etkileyiciliği ve kullanılabilirliği açısından değerlendirmek için bir deney yaptık. Bu deneyde, deneklerden, önerilen modelin parametrelerini seçerek veya geleneksel üç boyutlu modelleme araçlarıyla sağlanan görüntüdekilere benzer 3 boyutlu şadirvan modelleri üretmelerini istedik.*

**Anahtar Kelimeler:** Yordamsal Modelleme, Simetri, Şadirvanlar, Mesh

### Cite

Kurt, M., Bülbül, A., (2021). "Procedural 3D Fountain Modeling", *Mugla Journal of Science and Technology*, 7(0), 27-33.

### 1. Introduction

Fountain (Shadirvan) as an architectural term is a place that is usually covered with a dome mostly in mosque courtyards to take ablution. Water flows from the taps around the pool in the middle and the sprinklers in the middle. According to its type, generally, the fountain consists of arches, pools, columns, poles, domes, and pyramidal cones. There are fountains that only have a pool with an open top. Others are covered with a dome or pyramidal cone. The pool of the fountains can be planned as square, rectangle, circle, and polygonal (6, 8, 12 sided). In construction, materials such as stone, lead,

marble, wood, and metal are used. A few samples of fountains can be seen in Figure 1.

Procedural modeling is a computer graphics method which is used to accurately model structured objects following their construction rules. Procedural modeling designs complex models and aims to define them effectively. It is used in various fields such as architecture, urban planning, cultural heritage, gaming, and archeology. In procedural methods, user inputs are significant. For instance, a model created by procedural modeling without user input would be random and away from user intention. On the other hand, as too

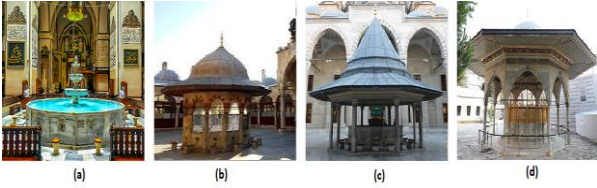


Figure 1. Fountain images from (a) Bursa Ulu Mosque (b) Kadirga Sokullu Mehmed Paşa Mosque (c) Fatih Mosque (d) Ayasofya Mosque.

much user input requires a lot of time and effort to create the model, it reduces the usability of the system [1, 2]. For this reason, the parameters chosen by the users are kept at a certain level.

Having a well defined structure and repeated elements, fountains are suitable to be modeled using procedural modeling. In this paper, we propose a method where fountain models are formed using symmetry information of objects and procedural modeling. In order to achieve this, we have observed fountains that have historical or architectural importance and have different characteristics. According to the observations, we have identified the parameters. The procedural fountain model can be created by determining parameters from a single image similar to Cengiz and Bulbul [3]. Alternative models are offered with parameters that provide different configurations. Models are created with parameters such as shape, texture, and feature, etc. via the provided interface. While currently the parameter selection is done manually, we are planning to automate this process by detecting the parameter values directly from an input image.

The contributions of our work are:

- the design of the system to enable procedural modeling of the complex fountain.
- examining the features of important fountains.
- the evaluation of the system that is created fountain model similar to the image with selected parameters as user input data.

The paper is organized as follows. We present related works in Section 2. Section 3 gives determined fountain parameters along with information regarding how they were created, dependencies of parameters, an overview of our work, and modeling of the fountain. Section 4 shows samples of the fountains that we built in 3D models. Section 4 also gives user results similar to images to analyze our system. We explain the conclusion of generated fountain models in Section 5.

## 2. Related Work

### 2.1. Procedural Modeling

L-systems [4] and procedural modeling can be used to compose 3D models including 3D building models. An L-system is a parallel string rewriting mechanism based on a set of production rules. L-systems are used initially to create plant geometry in computer graphics. Later, Parish et al. [5] generated geometry using L-systems

and created road patterns and large cities with this geometry.

Müller et al. [6] used procedural modeling and proposed shape grammar to generate buildings. Later they developed CityEngine which is a software that procedurally creates 3d city models. 3d models and textures are created from a set of rules with procedural modeling. This tool is widely used, for instance, Singh et al. produced a 3D model of a university campus by using CityEngine and they concluded that it is a good and low-cost tool to create a photorealistic 3-D model of a building. It creates city models in less time. The limitation of CityEngine is that it requires experience and a talented person [7]. Unlike CityEngine, the proposed method doesn't require any knowledge and experience and similarly, it produces a fountain in less time.

Krecklau and Kobbelt [8] create complex interconnected structures such as bridges, catenaries, and roller coasters with a procedural modeling method that uses shape grammar. Connection patterns and geometric queries are used in their production rules. Their grammar supports two types of connection, namely deformable beams and rigid object chains. While rigid object chains are and constructed by inverse kinematics, deformable beams are constructed by spline interpolation. With procedural modeling, Saldana [1] proposes the 'Roman City Ruleset' to generate Roman and Hellenistic architecture. Kelly and Wonka [9] form an interactive and a procedural modeling tool for architectural surfaces with procedural extrusions of building footprints. Their work demonstrates that procedural extrusions can interactively model complex architectural features like curved roofs, dormer windows, interior dormer windows, columns, and alcoves through a user interface.

### 2.2. Modeling of Symmetrical Objects

Symmetry is one of the most common properties of natural or manmade objects and provides an additional constraint for the 3D model. Jiang et al. [10] presented a method to build high-quality architecture models from a single image by exploiting constraints derived from symmetry. They calibrate the camera from a single image by exploiting symmetry. A set of 3D points is recovered according to the calibration and the symmetry. Xue et al. [11] proposed a novel 3-D modeling approach to recover 3-D geometry from a single image of a symmetric object. Given a view of a symmetric object, the user marks symmetric lines. Their algorithm finds a set of planes to approximately fit the object, and then a rough 3-D point cloud is generated. The occluded part of the object is recovered using symmetry information. Their algorithm can successfully recover the 3-D model of a symmetric object from a single image. These studies focus on symmetry for high-quality modeling by ensuring constraints.

### 2.3. Mosque and Fountain Modeling

Cengiz and Bulbul [3] composed 3D models of fountains

Table 1. Fountain Parameters.

| Fountain Parameters | Fatih M. F.                                      | Ayasofya M. F.                                    | Kurşunlu M. F.                                  | Merzifon Kara Mustafa Paşa M. F.         |
|---------------------|--|---|---|--|
| arch                | No   | 8 pointed arch                                    | no  | 8 arch                                   |
| dome                | material: lead<br>shape: pointed cone            | semi-circle dome<br>6 edge roof<br>material: lead | material: lead<br>shape: pyramid                | shape: pyramidal cough<br>material: lead |
| column              | 8 columns<br>material: marble<br>shape: circular | 8 columns<br>material: marble                     | 8 columns<br>material: stone<br>shape: circular | 8 column<br>material: wood               |
| pool                | material: marble<br>shape: 16 edge               | material: marble<br>shape: circle                 | material: stone<br>shape: 16 edge               | material: marble<br>shape: 16 edge       |
| tap                 | 16 taps  | bronze tap  | 16 taps   | 16 taps                                  |
| ground              | shape octagon                                    | shape octagon                                     | shape octagon                                   | shape octagon                            |

from a single image. With the Convolutional Neural Networks method(CNNs), the fountain is detected and labeled on an image. Graph Cut technique segmented out the fountain. The shape of the 3D model is estimated using the contour of the object. Finally, the 3D model is completed by applying the texture of the input image. This method is flexible to model various types of fountains with a single image; however, there may be major geometry flaws. In the proposed study, we offer a method that can generate better defined and more usable 3D models.

Yayla and Bulbul [12] presented a system to generate 3D model of mosques with procedural modeling. The user can determine parameters hierarchically. In addition, the user can manipulate the final 3D model of mosques by changing the low-level parameters. Their aim is to reduce user intervention.

Procedural modeling is used extensively in architecture and can create complex 3D architectural structures therefore we implement procedural modeling method to create realistic fountain 3D models such as effective models that are created by Yayla and Bulbul [12].

#### 2.4. Creating 3D shapes with parametric meshes

Persson and Strang [13] composed meshes like polygons, unit circle, and unit circle with a hole that are high quality by discovering vertex node locations with truss structure and using the Delaunay algorithm. Havemann and Fellner [14] create 3D shapes in place of primitive types. These shapes are gothic windows and an arcade. With Catmull/Clark subdivision surfaces, 3D shape representation is a boundary representation mesh. Similar to the works described above, we created 3D shapes with parametric meshes to obtain polygon, pyramid, and arch like elements.

### 3. Method

#### 3.1. Overview

We propose a procedural method to generate 3D fountain models. While most fountains have a symmetrical shape along their center vertical line, each of them has distinct features which are affected by several factors such as their architectural styles, sizes, used materials, and the era that they are built in. In order to develop a sufficiently expressive procedural method which can cover a big rate of the variety of existing fountain models, we investigated historically and architecturally important fountain models and determined the parameter spaces for the production rules. A user can choose the parameter values from the provided user interface. According to the procedural rules and user choices, then, 3D fountain models are composed of primitive 3D shapes (cubes, cylinders etc.) and meshes generated by parametric functions (arches, pyramids etc.). These shapes are texture mapped and transformed into appropriate sizes and positions to produce the final fountain models. These steps are explained in more detail in the following subsections.

#### 3.2. Fountain Parameter

There is a great variety of fountains, for instance, Bursa Ulu Mosque Fountain, Kadirga Sokullu Mehmed Paşa Mosque Fountain, Fatih Mosque Fountain, and Ayasofya Mosque Fountain (Figure 1). We have observed 15 fountain models. The fountain has architectural elements like an arch, dome, pool, column, tap, seat, and ground. According to our observations of real fountains, these elements are determined as parameters and their values belonging to real fountains are shown in Table 1. The shape of the pool, the shape of the ground, the shape of the dome, the number of columns, and the size

are determined as the fundamental (high-level) parameters of the fountain. Arch texture, column texture, dome texture, sub-dome texture, pool texture, ground texture, seat texture, tap texture, arch number, column shape, position, column thickness, and pool height are determined as the lower-level parameters. Table 2 demonstrates parameter values that can be selected and for each of the parameters.

### 3.3. Parametric Mesh

In order to construct the 3D models of fountains, primitive shapes and parametric meshes are utilized. Cube, square, cylinder, and circle are used as primitive types. Additionally, pyramids, hexagons, arches, and octagons are parametrically created. Each of those mesh models require vertices, faces, and texture coordinates. In our implementation we used Unity's Mesh class to generate them. Vertices, triangles, and texture coordinates are assigned for building a mesh model. As fountain models have symmetrical and circular shapes, we employed cosine and sine functions to determine vertex positions. Each shape is covered with appropriate textures with the parametrically defined texture coordinates.

Table 2. Parameters/Values.

| Fountain Parameters | Values  |
|---------------------|---|
| Arch Texture        | stone, marble, wood   |
| Column Texture      | stone, marble, wood   |
| Dome Texture        | lead, stone   |
| Sub-Dome Texture    | lead, stone, wood   |
| Pool Texture        | stone, marble   |
| Ground Texture      | stone, marble   |
| Seat Texture        | stone, marble, wood   |
| Tap Texture         | bronze  |
| Arch Number         | 0, 4, 6, 8  |
| Column Number       | 4, 6, 8   |
| Column Shape        | cubic, cylinder   |
| Pool Shape          | square, hexagon, octagon  |
| Dome Shape          | circle,<br>four-edged pyramid,<br>six-edged pyramid,<br>eight-edged pyramid |
| Ground Shape        | square, hexagon, octagon  |
| Position            | number  |
| Size                | number  |
| Column Thickness    | number  |
| Pool Height         | Number  |

### 3.4. Dependency

Certain parameters have dependency on other parameters; for instance, gutter where the water is collected, the number of the seats, and the number of the taps have dependency on the number of the edges of the pool. According to the shape of the pool, the gutter is created by subtracting the pool shape from the shape of

the ground using constructive solid geometry operations. The position of the tap has a dependency on the height of the pool. The shape of the sub-dome has a dependency on the number of edges of the pyramid dome. These parameters are automatically generated without any parameter selection. Table 3 demonstrates these dependencies.

### 3.5. Modeling of Fountain

The ground with the gutter is positioned under and the pool is positioned in the middle of this ground. It is positioned so that there are taps and seats around the pool. The columns are positioned symmetrically on the edges of the ground. If there is an arch, columns with the arch are positioned instead of columns. Finally, the dome is positioned to cover the columns. When the shape of the ground and the number of columns change, the position of the columns changes too. Generally, the position of the columns is calculated as shown in Equation 1 and Equation 2. Alpha refers to the angle of the arc between two columns. Size refers to the size of the ground. The arch position is calculated using a similar equation.

$$\alpha = 360/\text{column number} \quad (1)$$

$$\text{position}[i] = \cos(i * \alpha) * \text{size}, 0, \sin(i * \alpha) * \text{size} \quad (2)$$

Table 3. Dependency.

| Dependent parameter                     | Parameter   |
|---|-------------|
| seat number<br>tap number<br>the gutter | pool shape  |
| tap position                            | pool height |
| sub-dome shape                          | dome shape  |

## 4. Result

### 4.1. Samples of Fountain

In this study, according to our observations of existing fountains, we implement procedural modeling using parameters in the production rules in Unity. A user can create 3D fountain models by selecting the parameters. Figure 2 shows samples from the generated 3D fountain models and gutter. On the average, it takes 7 seconds to generate 3D models of fountains by manually determining values of different parameters. Using the proposed method, various fountain models can be quickly generated even by unexperienced users.

### 4.2. Generating fountains based on photographs

In order to evaluate the effectiveness of the proposed approach, we made a user study where we asked user's to generate a 3D fountain model by looking at a real fountain image. We also compared the proposed procedural approach with a commonly used 3D modeling (computer aided design) tool. Three different users including two architects, who are familiar with designing 3D architectural models and one non-

architect try our system to generate fountain models. All users try to produce 3 different fountains in our system by looking at their photos. The first fountain is Kurşunlu Mosque Fountain, the second one is Ayasofya Mosque Fountain, and the third one is Merzifon Kara Mustafa Paşa Mosque Fountain. In order to show that different fountains can be produced, fountains with different features are selected in the experiment. The features of these three fountains are shown in Table 1. For example, the first fountain has only columns, while the others have columns and arches. The dome shape of the second one is circular, while the others are pyramids. In addition, Ayasofya Mosque Fountain is a unique example being the largest and the most magnificent in Ottoman Art. In Figure 3, (a) demonstrates the real fountain image. (b) demonstrates a fountain model that the architect draw using Rhino3D design tool. (c) demonstrates the model that the first user produces in our system. (d) demonstrates the model that the second user produces in our system. (e) demonstrates the model that the third user produces in our system. Figure 4, Figure 5, Figure 6 demonstrate manually selected parameter values to generate the fountain models. Users choose these parameter values by looking at the image. It took 8 minutes to model the first fountain model using Rhino3D. Each of the second and the third fountains required 30 minutes to be modeled in Rhino3D.

The first user creates the model similar to the first fountain image at 3 minutes 20 seconds with manual parameter selection. The user creates the model similar to the second fountain image at 3 minutes 53 seconds with manual parameter selection. The user creates the model similar to the third fountain image at 2 minutes 40 seconds with manual parameter selection.

The second user creates the model similar to the first fountain image at 3 minutes 50 seconds with manual parameter selection. The user creates the model similar to the second fountain image at 3 minutes 1 second with manual parameter selection. The user creates the model similar to the third fountain image at 1 minute 22 seconds with manual parameter selection.

The third user creates the model similar to the first fountain image at 3 minutes and 9 seconds with manual parameter selection. The user creates the model similar to the second fountain image at 2 minutes 53 seconds with manual parameter selection. The user creates the model similar to the third fountain image at 2 minutes with manual parameter selection.

### 4.3. Analysis

We conduct the experiment with three users, comparing the proposed method with the traditional 3D modeling tool. The results suggest that the proposed approach takes notably less time compared to CAD tools and it does not require knowledge or experience of 3D modeling. Moreover, models that are created with the proposed method are quite similar to fountains that are created with traditional tools. Thus, the proposed method is effective and usable.

## 5. Discussion and Conclusion

This paper has introduced that 3D models of the fountains can be built with procedural modeling. Near-realistic fountain models are produced with the specified parameters. We demonstrate the produced fountain models that are formed by different values of parameters. It is possible to generate a great variety of fountains.

In the future, we are planning to improve our current model. More realistic fountain models can be created using parameters such as columns with muqarnas, tap styles, and more defined decorative elements. We aim to provide a procedural method that is capable of modeling most of the real fountains with high fidelity and varying levels of details. Consequently, it will be possible to make a 3D reconstruction of a fountain after automatically detecting the parameters from an image. Similar to Cengiz and Bulbul's study [3], such an approach will enable generating a 3D fountain model from a single image and the resulting model will be more functional and well defined to be used in 3D games and virtual reality applications.

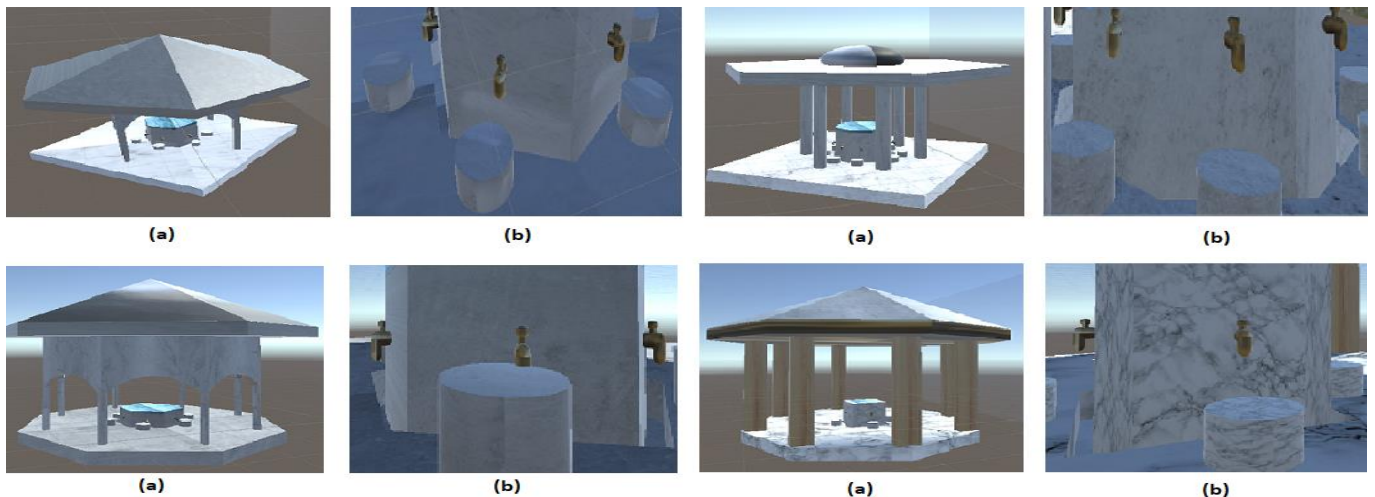




Figure 2. (a): Fountain 3D Models. (b): Images of the gutter.

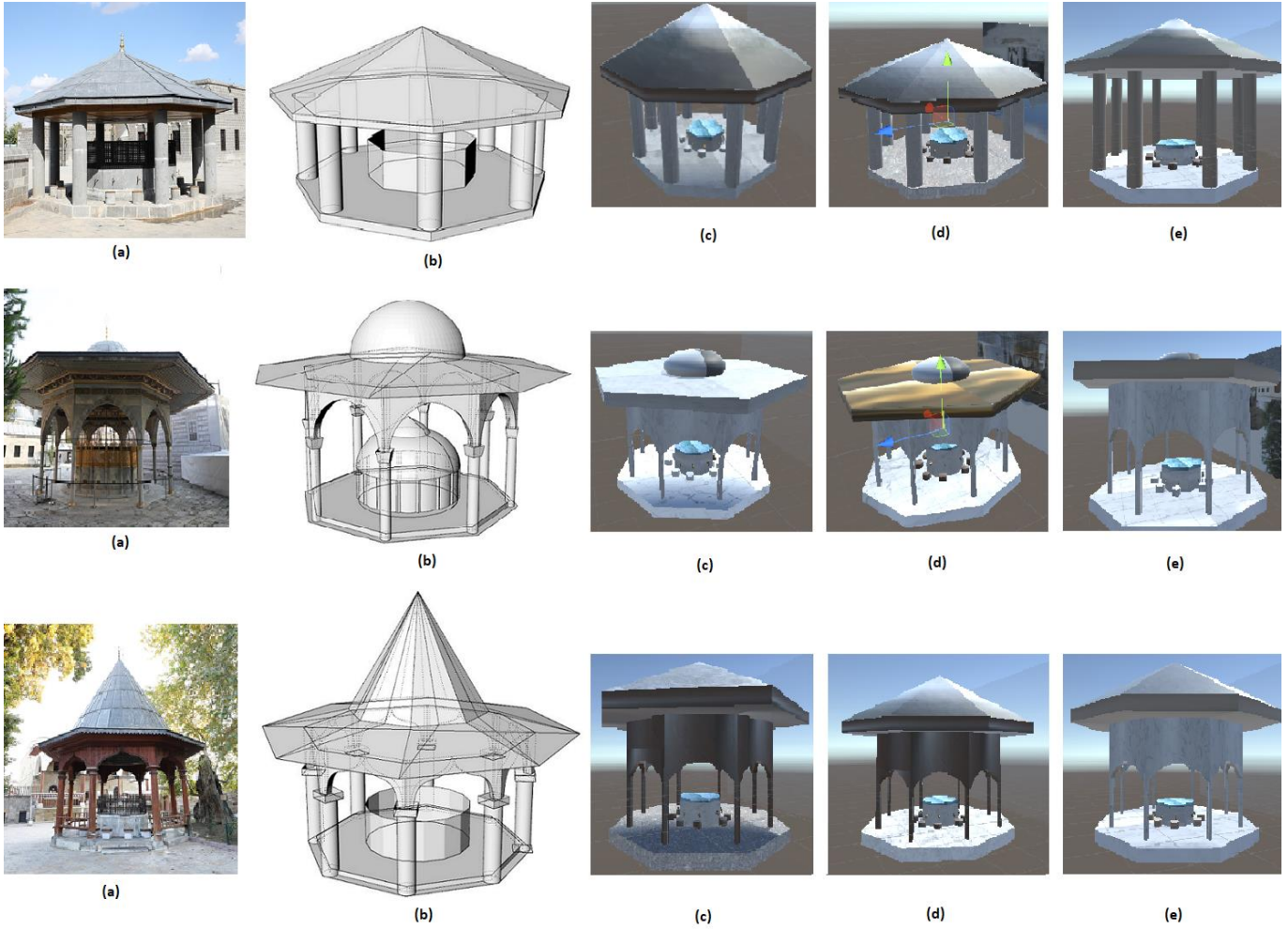


Figure 3.(a): Fountain images. (b): generated models with Rhino3D. (c): 3D fountains created similar to image by first user. (d): 3D fountains created similar to image by second user. (e): 3D fountains created similar to image by third user.

| Parameters       |                    | Parameters       |          | Parameters       |                  |
|------------------|--------------------|------------------|----------|------------------|------------------|
| Position         |                    | Position         |          | Position         |                  |
| x/0              | z/0                | x/0              | z/0      | x/0              | z/0              |
| Size             | 45                 | Size             | 40       | Size             | 50               |
| Column Texture   | stone3             | Column Texture   | marble 1 | Column Texture   | wood2            |
| Dome Texture     | lead4              | Dome Texture     | lead3    | Dome Texture     | lead3            |
| Pool Texture     | stone3             | Pool Texture     | marble2  | Pool Texture     | marble4          |
| Ground Texture   | stone3             | Ground Texture   | marble3  | Ground Texture   | stone            |
| Sub Dome Texture | wood2              | Sub Dome Texture | stone3   | Sub Dome Texture | wood2            |
| Tap Texture      | bronze             | Tap Texture      | bronze   | Tap Texture      | bronze           |
| Seat Texture     | stone3             | Seat Texture     | marble3  | Seat Texture     | wood2            |
| Arc Texture      | None (Texture)     | Arc Texture      | marble2  | Arc Texture      | wood2            |
| Arch             | Yes                | Arch             | Yes      | Arch             | Yes              |
| Arch Number      | Zero               | Arch Number      | Eight    | Arch Number      | Eight            |
| Column Shape     | Cylinder           | Column Shape     | Cylinder | Column Shape     | Cylinder         |
| Column Number    | Eight              | Column Number    | Eight    | Column Number    | Eight            |
| Column Thickness | 8                  | Column Thickness | 8        | Column Thickness | 8                |
| Pool Shape       | Octagon            | Pool Shape       | Octagon  | Pool Shape       | Octagon          |
| Poolheight       | 9                  | Poolheight       | 10       | Poolheight       | 12               |
| Dome Shape       | Eight Edge Pyramid | Dome Shape       | Circle   | Dome Shape       | Six Edge Pyramid |
| Ground Shape     | Octagon            | Ground Shape     | Octagon  | Ground Shape     | Octagon          |

Figure 4. (d): manually selected parameter values by the first user, respectively.

| Parameters       |                    | Parameters       |          | Parameters       |                    |
|------------------|--------------------|------------------|----------|------------------|--------------------|
| Position         |                    | Position         |          | Position         |                    |
| x/0              | z/0                | x/0              | z/0      | x/0              | z/0                |
| Size             | 50                 | Size             | 50       | Size             | 50                 |
| Column Texture   | stone              | Column Texture   | marble4  | Column Texture   | wood2              |
| Dome Texture     | lead3              | Dome Texture     | lead3    | Dome Texture     | lead3              |
| Pool Texture     | stone              | Pool Texture     | marble4  | Pool Texture     | marble4            |
| Ground Texture   | stone              | Ground Texture   | marble2  | Ground Texture   | marble4            |
| Sub Dome Texture | wood2              | Sub Dome Texture | bronze   | Sub Dome Texture | wood2              |
| Tap Texture      | bronze             | Tap Texture      | bronze   | Tap Texture      | bronze             |
| Seat Texture     | wood2              | Seat Texture     | wood2    | Seat Texture     | wood2              |
| Arc Texture      | marble2            | Arc Texture      | marble2  | Arc Texture      | wood2              |
| Arch             | Yes                | Arch             | Yes      | Arch             | Yes                |
| Arch Number      | Zero               | Arch Number      | Eight    | Arch Number      | Eight              |
| Column Shape     | Cylinder           | Column Shape     | Cylinder | Column Shape     | Cylinder           |
| Column Number    | Eight              | Column Number    | Eight    | Column Number    | Eight              |
| Column Thickness | 8                  | Column Thickness | 8        | Column Thickness | 8                  |
| Pool Shape       | Octagon            | Pool Shape       | Octagon  | Pool Shape       | Octagon            |
| Poolheight       | 12                 | Poolheight       | 15       | Poolheight       | 12                 |
| Dome Shape       | Eight Edge Pyramid | Dome Shape       | Circle   | Dome Shape       | Eight Edge Pyramid |
| Ground Shape     | Octagon            | Ground Shape     | Octagon  | Ground Shape     | Octagon            |

Figure 5. (d): manually selected parameter values by the second user, respectively.

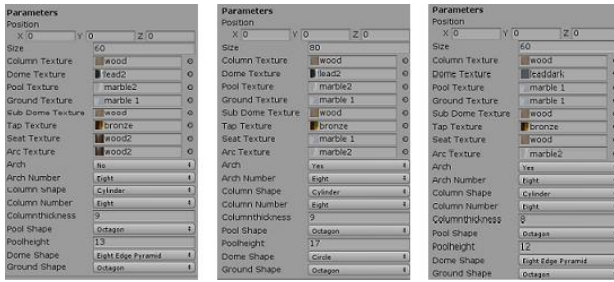


Figure 6. (d): manually selected parameter values by the third user, respectively.

### References

[1] Saldana, M., “An integrated approach to the procedural modeling of ancient cities and buildings, *Digital Scholarship in the Humanities*, 30 suppl1, i148–i163, 2015.

[2] Merrell, P. and Manocha D., “Model synthesis: A general procedural modeling algorithm”, *IEEE transactions on visualization and computer graphics*, 17, 6, 715–728, 2010.

[3] Cengiz, S. and Bulbul A., “3d fountain modeling from single image”, in *Eurasia Graphics 2017*, 1–11, 2017.

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

[5] Parish, Y. I. and Müller P., “Procedural modeling of cities”, in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, 301–308.

[6] Müller, P., Wonka, P., Haegler S., Ulmer, A. and Van Gool L., “Procedural modeling of buildings”, in *ACM SIGGRAPH 2006 Papers*, 2006, 614–623.

[7] Singh, S. P., Jain K., and Mandla, V. R., “Image based virtual 3d campus modeling by using cityengine”, *American Journal of Engineering Science and Technology Research*, 2, 1, 01–10, 2014.

[8] Krecklauer, L. and Kobbelt, L., “Procedural modeling of interconnected structures”, in *Computer Graphics Forum*, 2011, 335–344.

[9] Kelly, T. and Wonka, P., “Interactive architectural modeling with procedural extrusions”, *ACM Transactions on Graphics (TOG)*, 30, 2, 1–15, 2011.

[10] Jiang, N., Tan, P. and Cheong, L.-F., “Symmetric architecture modeling with a single image”, in *ACM SIGGRAPH Asia 2009 papers*, 2009, 1–8.

[11] Xue, T., Liu, J. and Tang, X., “3-d modeling from a single view of a symmetric object” *IEEE transactions on image processing*, 21, 9, 4180–4189, 2012.

[12] Yayla, E. and Bülbül, A., “Procedural modeling of mosques”, in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 2018, 1–4.

[13] Persson, P.-O. and Strang, G., “A simple mesh generator in matlab”, *SIAM review*, 46, 2, 329–345, 2004.

[14] Havemann, S. and Fellner, D. W., “Generative mesh modeling”, 2005.