ESKİŞEHİR TECHNICAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
A- APPLIED SCIENCES AND ENGINEERING

16th Digital Design In Architecture Symposium
16th DDAS (MSTAS) - Special Issue 2022

2022, Vol.23, pp. 50-59, DOI:10.18038/estubtda.1168718

ARCHITECTURAL DESIGN WITH GENERATIVE ALGORITHM AND VIDEO PROJECTION MAPPING

Buse BÖLEK [1], Hatice Günseli DEMİRKOL [2], Mehmet İNCEOĞLU [3]

[1] Department of Architecture, Architecture and Design Faculty, Eskişehir Technical University, Eskişehir, Türkiye
[2] Department of Architecture, Architecture and Design Faculty, Eskişehir Technical University, Eskişehir, Türkiye
[3] Department of Architecture, Architecture and Design Faculty, Eskişehir Technical University, Eskişehir, Türkiye

ABSTRACT

The innovative and creative approach that is produced for architectural design problems involves quite complex stages. In architectural design, the designer's approach and numerical data make the design process a multi-layered structure. The integration of parametric modeling programs and generative algorithms makes it easy to carry out multi-layered design processes. Generative systems defined through generative algorithms enable conceptual approaches, various geometrical constructs, and different forms to be created in digital environments in solving design problems. The creation of the generative algorithm of the design eliminates the restrictive role of modeling software in the architectural design. The designer gets the opportunity to create every parameter and design criteria specific to his approach.

In this study, our aim is to develop a generative algorithm in a computer environment by using the principles of fractal geometry function. The generative system creates by overlaying the developed generative algorithm with the environmental factor parameters. In this system proposal, the goal is to produce various facade design proposals by coding the algorithm fractal functions and to make use of the visualization of various potentials of the generative algorithm in parametric facade proposals. Video projection mapping method is chosen to discuss the functionality of the parameters in the proposed generative system of the developed generative algorithm. Computation of the transformation relating consecutive image frames is an essential operation in video mapping. In this context, the design created with the generative system is reflected on a real scale medium , specifically a column in this study, and various image frames letting new design proposals produced by the algorithm were evaluated. In concluding rmarks, the advantages of architectural designs modeled by coding with the generative algorithm compared to the algorithms obtained directly with the software are discussed.

Keywords: Generative Systems, Algorithm Design, Generative Fractal Algorithm, Video Projection, 3D Mapping

1. INTRODUCTION

Architectural design tools have changed and transformed over time with the developing technology. Thus, architects integrated computer-aided approaches into architectural designs. Modeling programs and software interfaces facilitate design processes, but also bring limitations in mental production processes. Sometimes design ideas cannot be visualized directly through these programs, so we have to use conventional design tools. However, this situation complicates the calculations in the digital design processes. If the designer codes the generative algorithm of his design initially, software becomes a developing tool. In the studies conducted with architecture students, they observed the limitations of the software interfaces in the individual works of the students. However, they also evaluated that the students who knew how to code their algorithm could directly transfer their designs to the digital environment [1].

The thinking process and logical operation in a hand-drawn line based on the design coincide with the algorithm-based line. Architectural designs with algorithms allow parameters to self-organize [2]. The situations that are unpredicted in the design processes can be formulated by manually changing

*Corresponding Author: busebolek@eskisehir.edu.tr
Received: 31.08.2022 Published:23.12.2022

parameters or running algorithms iteratively. In addition to that, relationships that the designer cannot predict can also appear by manually changing parameters or by running algorithms iteratively [3]. Algorithms are essentially mere form of expression of the thinking process [4]. Through this form of expression, seminal parametric designs occur. Designing with algorithms is a new design skill as well as a new way of discovering the potential of design [5]. With this new design skill, the designer can formulate his own geometric algorithm. Thus, geometric algorithms remove the boundaries between modeling software and the designer [6].

Parametric designs depends on algorithms formed by the combination of different parameters and consist of a combination of algorithms that allow variations [7]. On the other hand, algorithms involve a certain number of steps in reaching possible solutions to the defined problem [8]. Modeling of parametric designs depends on the creation of algorithms. The parametric design not only allows for the diversified geometric organizations but also for the different variations [9] and complex forms (Schumacher, 2009). It provides flexibility in the design process as geometric relationships can be clearly defined in the parametric design process [11]. It is also necessary to determine the constraints in the design processes for constructed parametric geometries and to create the parameters in a way that can diversify the design [12]. Parameters provide a variety of solutions as they allow the development and numerical control of geometries in the design [13]. It enables the identification and control of the relationships among the elements [14] and creates meaningful tectonic potentials for the elements [15]. Thus, design schemes that are not able with traditional methods, appear in the second and third dimension through algorithms. Creating algorithms is no longer the domain of a particular discipline. This method is open to architectural design, and expressing architectural designs in the form of algorithms offers the opportunity to collaborate with other study disciplines [16].

Fractal geometries provide various design potentials especially when working with algorithms. Fractals consist of compression, rotation, and nonlinear transformations of a given geometry. Fractal geometries provide the relationship between nature's complexity and architecture. This aspect of fractal geometry facilitates prototyping in surface designs in architecture [17]. Forms produced with fractal geometry turn into complex structures [18]. Functions of complex structures can be reduced to fractal functions and thus they are converted into generative algorithms [19]. Fractals exposes a form of grammar that facilitates the assimilation of syntactic information in architectural designs diversified with a generative algorithm [20]. Fractal geometries are qualified to design aids with fewer rules, more repetitions, and more formal similarity than other shape grammars [21]. Complex system designs in architecture, which cannot be created with traditional methods, are made through generative algorithms [22]. By this means, it is possible to design complex systems providing numerical consistency.

Visualization of algorithm-based complex designs at real scale provides the opportunity to experience the design. Architectural design can be transmitted in the second and third dimension in one-to-one scale with the video projection mapping method. This method proposes projection of a created video on an architectural surface or communication of the video with an architectural design [23]. The use of video projection mapping in the architectural field is a method that includes the subject as the designer or user as a viewer in the visual projected element. This method is also usable in the design and concept stages of the projects [24].

The integration of designs with video projection mapping method, it is possible to experience digital models in the second and third dimension. This methodology ensures that the images integrated in the desired geometric form are transferred on to the real scaled design surfaces (Figure 1). In this study, the actual dimensions of the design become one of the parameters of the generative algorithm. In order to discuss the effect of the parameters in the generative algorithm, the design coded with the generative algorithm prompt is projected in real scale with video projection mapping.

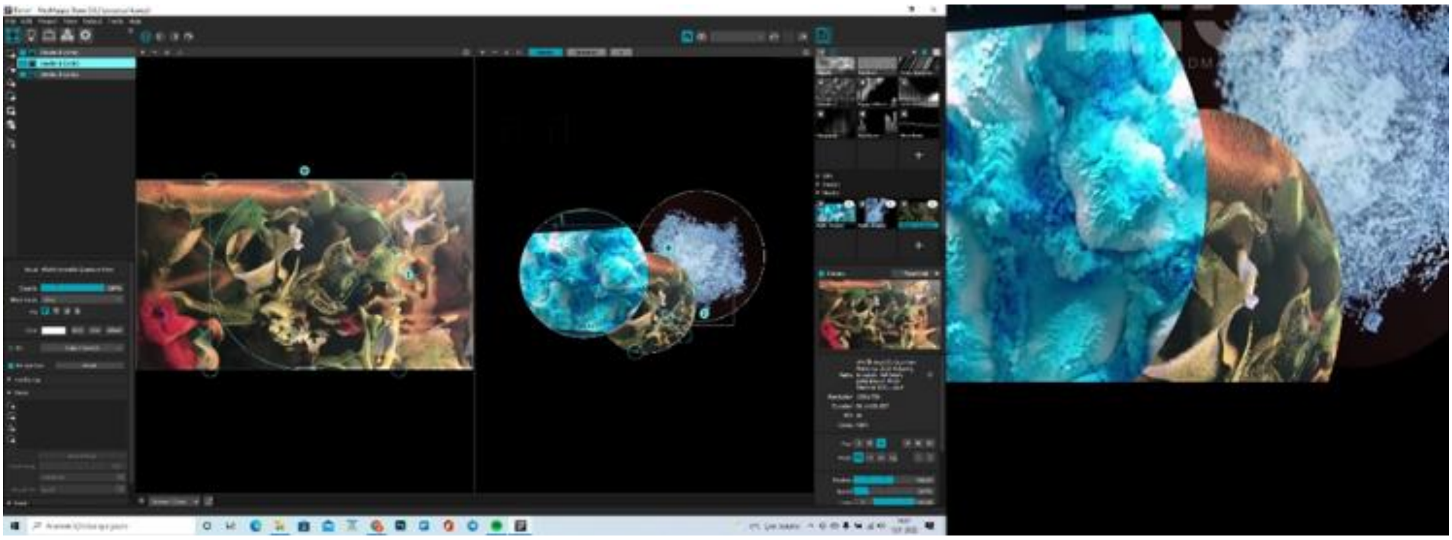


Figure 1: Madmapper video projection mapping working with curvilinear geometries [25].

In this study, we used fractal function in the generative algorithm system developed within the scope of the study. The selection of fractal function enables to provide the control of the point and vector positions of the geometry after the third iteration, which is not possible without computer. After the third iteration, vectors and point coordinates in space geometry become incalculable. However, the generative system, which consists of the combination of the generative algorithm with other parameters, can produce this information regularly. Thus, a generative algorithm based on the features of fractal geometry on projected surface design as in architectural parametric facade designs is created, benefiting from its continuity. The syntactic scheme of the generative algorithm is coded in the Python scripting language accessible from Grasshopper interface. The visualization process of the syntactic language is generated by the Rhino interface. The design-specific coding of the generative algorithm provides guidance and varied features in the design. This approach, which shows the topological and geometric aspects of the architectural pattern, has been integrated and presented as a generative system in which various parameters managed. The proposed generative system algorithm is supposed to discuss the various potentials of parametric facade designs in the early design phases.

2. GENERATIVE SYSTEM DESIGN AND METOD

In order to design the generative fractal algorithm system through the object-oriented algorithm, a writing language is developed that consisted of three main steps as expressed in the flowchart (Figure 2). These steps are respectively;

• Designing the generative fractal algorithm
• Determination of the start and targets of the parameters
• Visualization of the design

Once the concept design is created the system automates the other stages of the flowchart. For this reason, only the initial function belongs to the designer. In this direction, the coded algorithm in the Rhino-Grasshopper environment through the Python software language works as a generative system in other stages. With the automatic iterative operation of the algorithm, the visualization is provided in Rhino and relatively the design is diversified iteratively in the process by producing various output values.

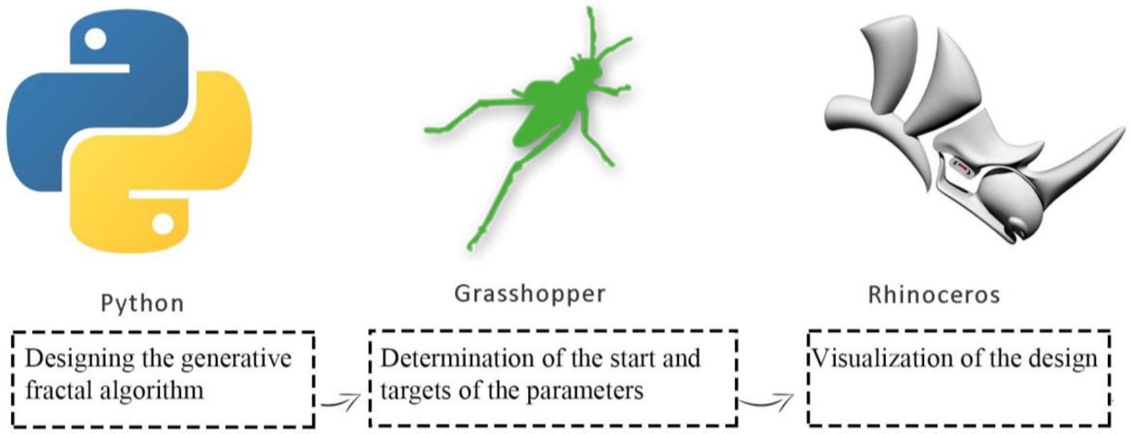


Figure 2. Flow chart summarizing the sequence of use of the software and coding language used in the study process. (Prepared by the authors.)

The flowchart of the algorithm is defined to design a generative algorithm system (Figure 3). During the coding process of the algorithm, the actual dimensions of the column selected as the sample projection surface are measured at the first stage. The dimensions of two consecutive surfaces of the column are also measured, respectively. According to the measurement results, the algorithm defines the three-dimensional geometry of the column. Thus, the changes in the designs suggested by the algorithm are observed in the real size changes with video projection mapping. The algorithm first calculates the coordinates of each vertex to divide the object surfaces. These coordinates are grouped into lists. The values in the list are associated with parameters named P_up, P_down, Level. This parametric change creates the generative fractal loop associated with the diversification of coordinates. In each production, the coordinates are calculated and listed by the algorithm. In order to generate the fractal loop iteratively, the data in the new list is redefined as the variables of the fractal function, and new surfaces are created by the generative operation of the algorithm. The defined surfaces are automatically moved axially by the algorithm. Various suggestions are obtained at each stage during this automatically moving process. During the iterative generation of the algorithm, new points, position vectors, surfaces are defined on the design geometry. The position of each new geometry in the spatial geometry is coded by the algorithm as a data list. In addition, the number of fractal units belonging to each generative design is determined in the data list. In this way, the spatial coordinates inform about the geometry obtained by the generative algorithm. In order to generate the fractal loop iteratively, the data in the new list is redefined as the variables of the fractal function and new surfaces are created by the generative operation of the algorithm.

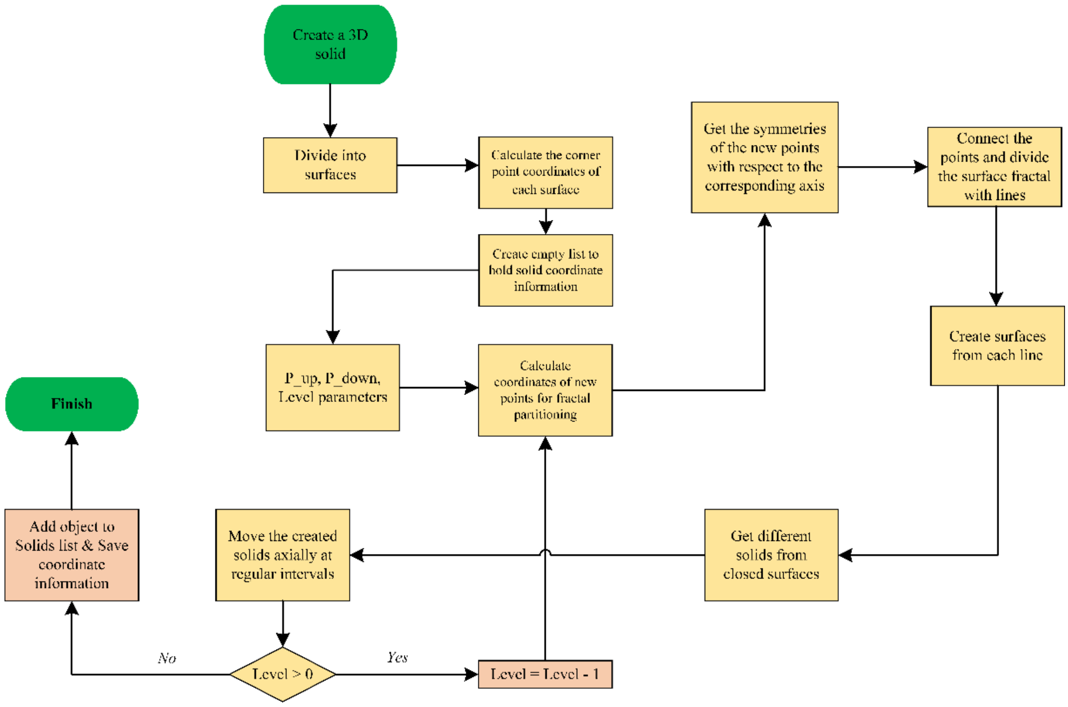


Figure 3. Python algorithm flowchart (Prepared by the authors.)

3. FINDINGS

In this study, a generative fractal algorithm is designed, which is coded through an object-oriented software language. With this algorithm, a generative system has been defined with the aim of diversified facade designs. Within the scope of the study, iterative production of the concept design by the algorithm on the surface of the column, which was determined as the projected surface, gives us as a sample scenario. The fractal algorithm coded in the object-oriented Python programming language in the Grasshopper interface provides various parameters added to the algorithm in which the numerical ranges can be changed from within Grasshopper. It has been observed that defining the parameters both in the algorithm and in the Grasshopper interface is effective in diversifying the design (Figure 4).

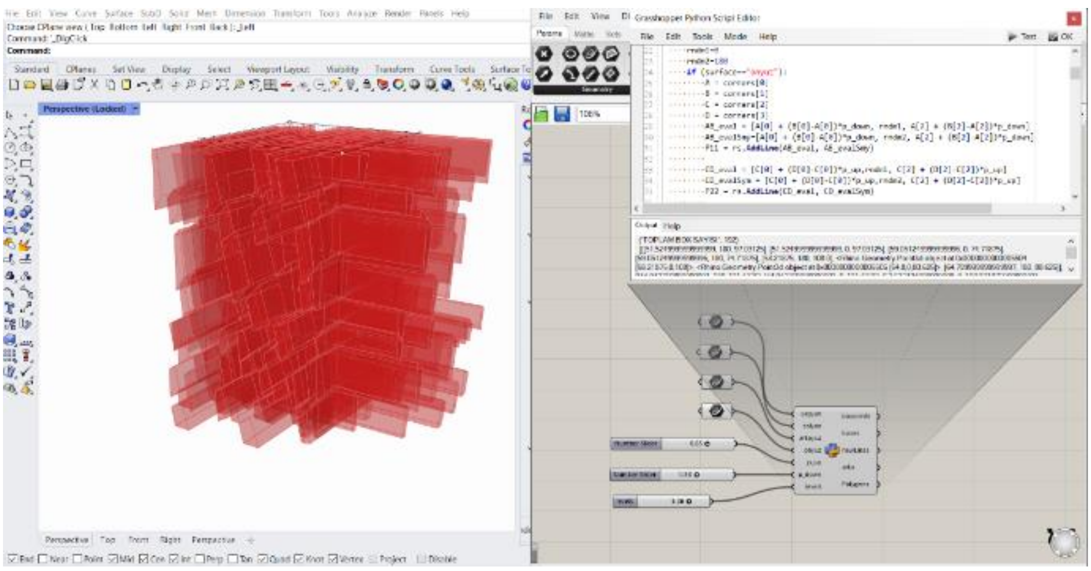


Figure 4. Python-based generative Grasshopper algorithm (Prepared by the Authors)

In the process of designing the fractal function algorithm firstly, the algorithm was coded to give outputs in the second dimension. Various outputs are obtained with the automatic generation of the fractal. In the process of obtaining the outputs of the fractal geometry algorithm, first it makes a random selection from the two parallel sides of the rectangle. Then it defines a line by connecting two randomly chosen points over the selected parallel edges. Respectively, two more lines are defined by selecting two more points on the line created in the previous step by defining points from the mutually parallel sides that were not selected. From this stage onwards, the fractal geometric loop becomes visually perceptible (Figure 5).

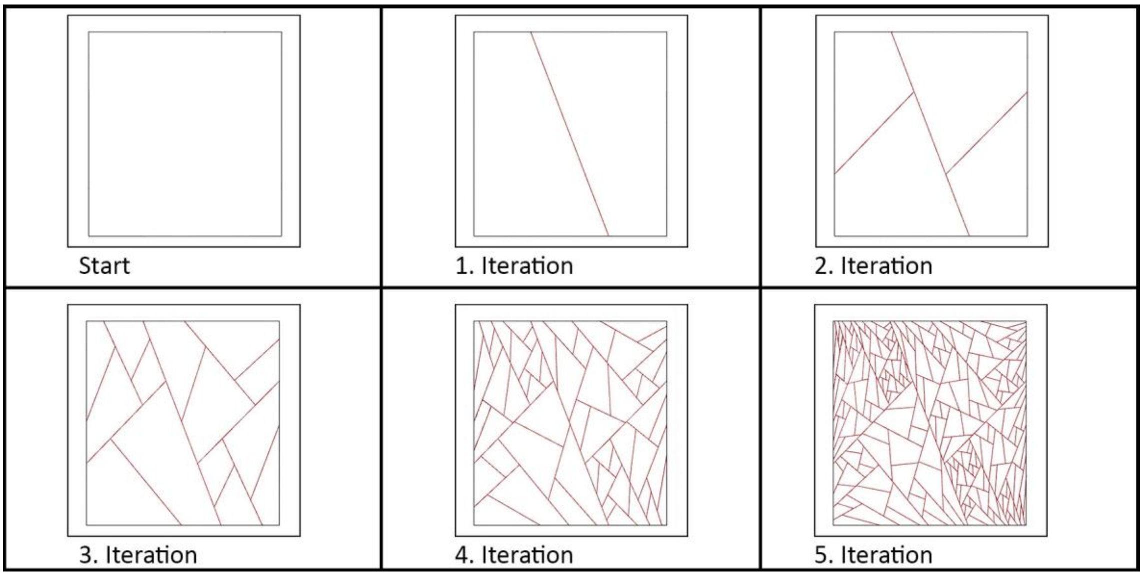


Figure 5. Second dimension outputs of the generative fractal algorithm designed via Python (Prepared by the Authors)

While the algorithmic loop continues, it also recognizes and divides different quadrilaterals with the same mathematical solution. A complex geometry undrawn by conventional methods has been obtained by advancing the cycle one-step further. The algorithm is limited with five iterations. If different iteration numbers and parameters are used, different geometries can be obtained. The design outputs obtained depends on the mathematical expression of the fractal function in the coding lines within the scope of the study.

The second-dimensional outputs provided by the generative fractal algorithm were useful in deciding the angle parameter defined in the fractal function. After this stage, the generative algorithm system iteratively provided generative proposal outputs in the third dimension. In the first step of generating the outputs, the generative fractal algorithm system enabled the coordinates to be defined from the two parallel sides of the column. Consecutively, a line is defined on the edges by connecting two points randomly chosen by the generative algorithm, and by this means, the defined lines together form surfaces. Surfaces take place in various coordinates in each iteration through the function that defines random axial motion in the algorithm. In addition to axial variation, each iteration produces fractal geometry in three dimensions. In the process of generating new geometries of the generative system, for each stage of the algorithm, a new axial variation occurs. With the increase in the number of iterations, the concept design has turned into a complex structure (Figure 6).

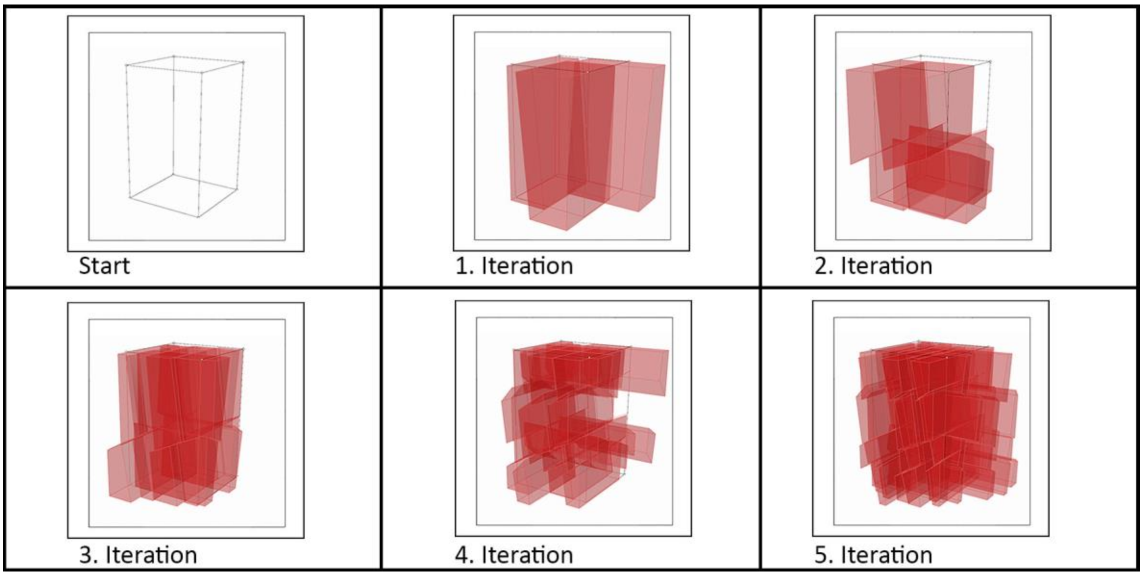


Figure 6. Third dimension outputs of the generative fractal algorithm designed with Python (Prepared by the Authors)

Throughout the process, it has been observed that the axial mobility of the fractal surfaces formed as a result of new productions at each stage by the algorithm allows for generative system design diversification, although the algorithm is produced based on a single fractal function. The production of various proposals of "design by the algorithm" has led to the generation of a large number of fractal geometry, such as a unit in the algorithm during the design process. For this reason, in order to control the geometric organization in the algorithm, the fractal unit number, as a result of each iteration provided to the algorithm, and each point in the space geometry system defined and listed. In this way, complex geometry became definitive and controllable (Figure 7).

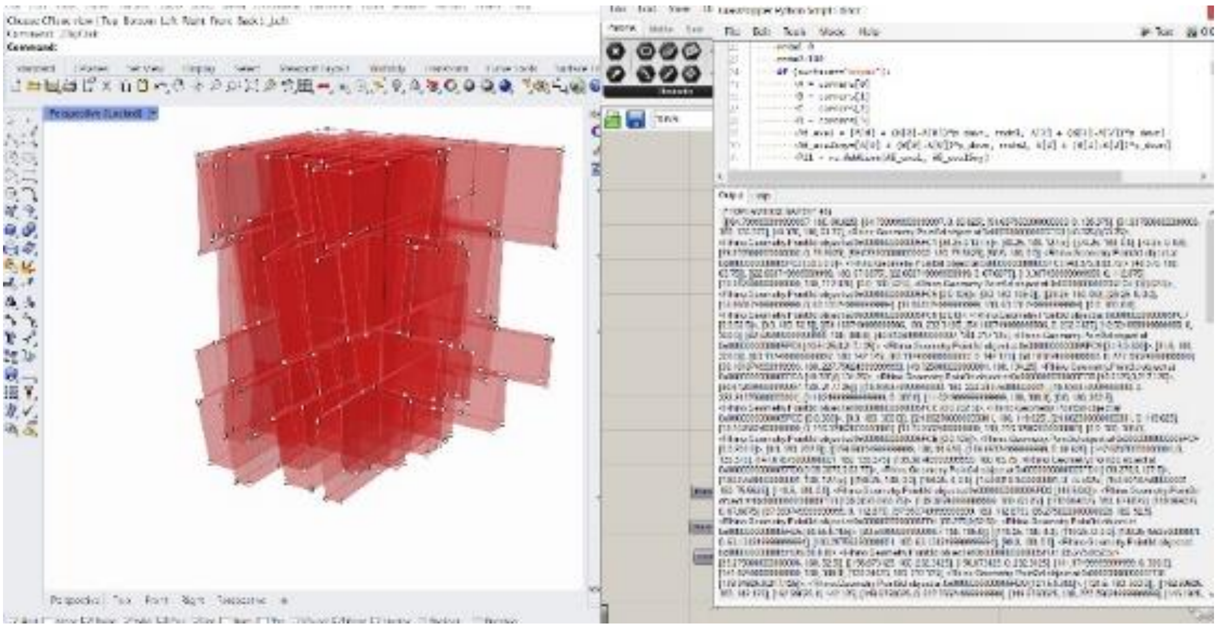
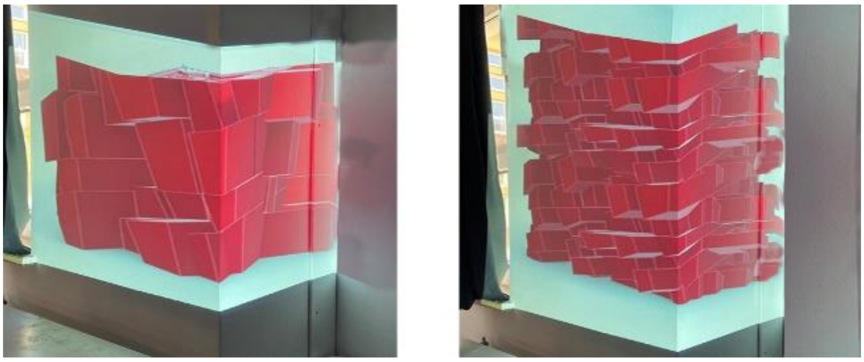


Figure 7. The location coordinates of the geometry produced by the algorithm and the generation of the fractal unit number by the algorithm (Prepared by the authors.)

As a further step in the study, the design outputs of the design geometry produced by the generative fractal algorithm system were integrated with the video projection mapping method to the region defined on the column (Figure 8). In order to do that, video projection mapping was performed by defining the column dimensions as parameters in the algorithm (Figure 8.a). As it is observed, the algorithm gives a new output with the change of the vertical dimension parameter on the geometric output produced by the generative system (Figure 8.b). It is clear that, the size parameters of the column surface and the algorithm determined at the beginning were important factors in the production process. As a conclusion, there are differences in the output production potentials provided by the algorithm with the diversification of the size constraints.



(a) **(b)**

Figure 8. Transfer of design outputs produced by the generative fractal algorithm system with video projection mapping (Prepared by the authors).

4. RESULTS

If the algorithm is defined as a certain number of parameters running in a finite time period of defined work, it can also be solved by iterative generation algorithms which are the basis of fractal geometry. This feature of fractal geometry facilitates prototyping in architectural designs. Thus, the different outputs provided by the generative algorithmic systems at the beginning of the design phases allow various solutions to the architectural design problems in the second and third dimension.

In this study, the generative algorithm is visualized in Rhino, which is encoded in Python in the Grasshopper interface. A variety of potentials in the design process is proposed by obtaining the second and third dimensional outputs of the generative fractal algorithm system. The planning decisions to be made in the design phases are diverse with the algorithm producing outputs in the second dimension. The outputs in the third dimension enabled the production of suggestions for the diversification of the multiple fractal units that come together on the architectural picture plane as forms of image frames. The generative algorithm system provides outputs in the second and third dimensions simultaneously. Planametric and third dimension variations, which are the main elements of architectural design, produced "a coordinated generative algorithm system". Thus, parametrically controllable and definable fractal geometry outputs were obtained with the generative algorithm system. During the production of geometrical outputs, the function of fractal geometry defined in the Python code was the most effective parameter in both the second and third dimension. Although the axial mobility in the third dimension within the generative algorithm system is produced depending on the fractal algorithm function that is defined at the beginning, it has been observed that the generative system continues to diversify on the geometric outputs. The number of fractal units in the diversified geometric outputs and the position vector of each point in space geometry can be defined; calculated and observed as output. It is envisaged that these calculated outputs can be useful in interdisciplinary studies during the realization process of the design (architectural surface, column, façade, ground floor etc…) and contribute to the working process in future.

The integration of generative algorithm-based architectural designs with the video projection mapping method has enabled the visualization of the change in the actual measurement parameters of the design. It has been observed that the actual dimensions of the design are included as parameters in the algorithm. Furthermore, it was predicted that the various outputs of the generative algorithm may be efficient in the production process. In addition, it ensures that the design achieved in the design incorporates into the actual dimensions without the need for materials and construction processes by defining the actual dimensions as parameters. Therefore, it is expected that the study will contribute to the research and development of diverse potentials in architectural facade or plane designs with generative systems.

In this study, the importance of the designer's development of the original algorithm in digital design was emphasized because the outputs generated by the generative algorithm cannot be directly produced computationally through softwares and modeling programs. As the generative algorithm of geometries is established by the designer, the constraints of the interfaces of modeling programs are removed. In conclusion, the interfaces of modeling programs are no longer a restrictive factor in the design process, so these various potentials of the design are observable in computational design processes. In addition, it has been evaluated syntactic coding languages can take place as "a new solution skill" in the design discipline with the development of designer-specific algorithms.

CONFLICT OF INTEREST

The authors stated that there are no conflicts of interest regarding the publication of this article.

REFERENCES

[1] Leitão A, Santos L. Programming languages for generative design: Visual or textual? 29th. In eCADe Conference, Slovenia. 2011.

[2] Rafael Iván, Pazos Pérez. Blurring the Boundaries Between Real and Artificial in Architecture and Urban Design through the Use Artificial Intelligence. 2017.

[3] Sutherland IE. Sketchpad a Man-Machine Graphical Communication System. SIMULATION. 1964; 2(5): 3- 20.

[4] Rian IM, Asayama S. Computational design of a nature-inspired architectural structure using the concepts of self-similar and random fractals. Automation in Construction. 2016; 66: 43–58.

[5] Oxman R. Thinking difference: Theories and models of parametric design thinking. Design Studies. 2017; 52: 4–39.

[6] Pauly, M., Mitra, N. J., Wallner, J., Pottmann, H., & Guibas, L. J. (2008). Discovering structural regularity in 3D geometry. In *ACM SIGGRAPH 2008 papers* (pp. 1-11).

[7] Caetano I, Santos L & Leitão A. Computational design in architecture: Defining parametric, generative, and algorithmic design. In Frontiers of Architectural Research Higher Edcation Press Limited Company. 2020; 9(2): 287–300.

[8] Terzidis K. Algorithmic architecture. Routledge. 2006.

[9] Jabi W. Parametric design for architecture. Hachette UK. 2013.

[10] Schumacher P. Parametricism: A New Global Style for Architecture and Urban Design. Architectural Design. 2009; 79(4): 14–23.

[11] Oxman R, Gu N. Theories and Models of Parametric Design Thinking PDT Parametric Design Thinking View project The Impacts of Urban Density on Social Sustainability: A Computational Approach View project. 2015.

[12] Burry M, Murray Z. Architectural Design Based on Parametric Variation and Associative Geometry in Challenges of the Future. 15th. In eCAADe Conference, Vienna. 1997.

[13] Hudson R. Knowledge Based Strategies for Parametric Design in Architecture. 2014; 77–81.

[14] Vyzantiadou MA, Avdelas AV, Zafiropoulos S. The application of fractal geometry to the design of grid or reticulated shell structures. CAD Computer Aided Design. 2007; 39(1): 51–59.

[15] Alvarado RG & Muñoz JJ. The control of shape: Origins of parametric design in architecture in xenakis, gehry and grimshaw. Metu Journal of the Faculty of Architecture. 2012; 29(1): 107-118

[16] Lee J, Gu N, & Williams AP. (2014). Parametric Design Strategies for the Generation of Creative Designs. International Journal of Architectural Computing. 2014; 12(3): 263–282.

[17] Osama A, Sherif L, Ezzeldin S. Fractal Geometry in Architecture: From Formative Idea to Surfical Skin DESIGN. RCHDESIGN. 2014; 14: 39–49.

[18] Çağdaş G. Fraktal geometri ve bilgisayar destekli mimari tasarımdaki rolü. CAD+ Bilgisayar Destekli Tasarım ve Ötesi. 1994; 23: 28-31.

[19] Amenta N, Choi S, & Kolluri RK. The power crust. Proceedings of the Symposium on Solid Modeling and Applications. 2001; 249–260.

[20] Ediz Ö, Çağdaş G. Mimari tasarımda fraktal kurguya dayalı üretken bir yaklaşım. İTÜ Journal. 2010; 4(1).

[21] Schmitt GN, Chen CC. Classes of design classes of methods classes of tools. Design Studies. 1991; 12(4): 246-251.

[22] Mayatskaya I, Kashina I, Gerlein N, Yazyev B. Fractal geometry and design of modern strcures. E3S Web of Conferences. 2021.

[23] Catanese R. 3D Architectural Videomapping. International Archives of the Photogrammetry, Remote, Sensing and Spatial Information Sciences, Digital Fabrication in Architecture. 2012; 165–169.

[24] Aksu M. Mimarlıkta Video Projeksiyon Haritalama Kullanımı. Tasarım Enformatiği. 2019; 01(02):107–117.

[25] Bölek B, Yavuz B, Demirkol G, İnceoğlu, M. Video projeksiyon haritalama ve mimarlık. 2022, (sumitted for publication).