The Role of Algorithmic Applications in the Development of Architectural Forms (Case Study: Nine High-Rise Buildings)

Ahmadreza Khalili [a], Hossein Soltanzadeh [b,*], Seyed Hadi Ghoddusifar [c]

[a] Department of Architecture, Faculty of Art and Architecture, South Tehran Branch, Islamic Azad University, Tehran, Iran
[b] Department of Architecture, Faculty of Architecture and Urban Planning, Central Tehran Branch, Islamic Azad University, Tehran, Iran
[C] Department of Architecture, Faculty of Art and Architecture, South Tehran Branch, Islamic Azad University, Tehran, Iran

Received: 02 November 2018 - Accepted: 16 January 2019

Abstract

The process of developing architectural forms has greatly been changed by advances in digital technology, especially in design tools and applications. In recent years, the advent of graphical scripting languages in the design process has profoundly affected 3D modeling. Scripting languages help develop algorithms and geometrical grammar of shapes based on their constituent parameters. This study aimed to discover new concepts through the use of computer tools, especially algorithmic tools, in the form design process. The statistical population included high-rise buildings constructed worldwide in the past two decades. The main research questions include: 1) What are the main features and properties of algorithmic applications in comparison with conventional drawing applications? 2) How do algorithmic applications affect the process of devising an architectural form? In this case study, the quantitative research method was employed along with computer simulation. In addition, desk studies and the Internet references were reviewed to collect data. According to the research results, the use of graphical scripting languages as form design tools can develop a smart and generative geometrical framework. Such a smart geometrical framework can relate geometrical grammar to the mathematical relations of shapes. As a result, the changed geometrical parameters of a shape can be reflected on the other variables of dependent forms to change the primary features and specifications of the form. Moreover, changing the existing parameters can result in morphogenic forms, the development of which complies with specific principles.

Keywords: Algorithmic application, Design tool, High-rise buildings, Architectural form.

1. Introduction

The form of a building can be regarded as the goal of a designer or architecture in the design process. The design tool is of great importance among the factors affecting the development of a form such as geometry, performance, beauty, stability, and safety problems. In fact, the use of different tools is controlled by a designer's competence and knowledge in addition to selecting the primary geometry to achieve an ideal form in a design process. Digital technologies have changed the information and software penetration into the field of architectural design tools. As a result, there has been controversial conditions in new environments with various capacities in comparison with manual tools. Users can benefit from digital technologies to test their knowledge, ideas, and requirements in a new environment with a wide variety of capacities. In fact, technologies can change the human experience (Rosenberger & Verbeek, 2015, 10). Different digital technologies have not been used in the process of architectural education and design for a long time. However, the use of graphical drawing applications (2D, 3D, etc.) as design tools is a controversial subject in the architectural design process because it is characterized by specific features compared with manual design tools (freehand design, manual models, etc.). In architecture, a software application acts as a tool helping a designer draw the outcomes more accurately and quickly. The early applications were able to draw and display 2D lines. With the advent of computerized design, computers have taken a role beyond that of design tools. In the past, applications like AutoCAD helped accelerate the design process. However, if a geometrical component of a shape is changed or deleted in such applications, the modeling process may be gone. However, the geometrical grammar of objects can be transferred to a computer through an algorithmic language command. Unfortunately, software applications are regarded only as tools in the architecture society, especially in Iran. Digital tools are not employed merely for display methods, design purposes, and ease of use in specific areas. However, the method of using them will change design model processes (Khabazi, 2013, 13). In fact, each of such applications is characterized by specific features, which should be analyzed and evaluated separately. Hence, this study aimed to discover new ideas based on the use of computer tools, especially algorithm applications, in the process of designing forms. The following research questions were to be addressed: 1) What are the main features and properties of algorithm applications in comparison with conventional drawing

* Corresponding author Email address: h72soltanzadeh@gmail.com

applications? 2) How do algorithm applications affect the process of devising an architectural form?

In architecture, a modern challenge is to design high-rise buildings, which are very important due to the formalistic approach adopted by designers to devise their forms. High-rise buildings, also known as skyscrapers, are characterized by specific features due to certain structural, urban, and climatic constraints. Therefore, it can definitely be stated that every design company needs to employ computer tools, especially software applications, to draw and analyze the specific features of such buildings. For this purpose, the authors utilized software simulation to analyze the formation of skyscrapers after classifying their topologies based on the geometry of their plots.

2. Research Method

In this case study, the quantitative research method was employed along with computer simulation. First, data were collected on 9 high-rise buildings, selected as the statistical population, through desk studies. Then the authors classified the buildings as typological categories based on the geometry of their plots. In the next step, the authors employed computer simulation to analyze the form development of the buildings. For this purpose, AutoCAD was used as a non-algorithmic application first. Then Rhino, Plugin, and Grasshopper were used as graphical scripting languages to run the tests.

3. Research Literature

Every architectural style has a specific form of grammar. There are two types of form grammar: standard design and parametric design. In standard designs, most of the attributes are constant. However, forms are more likely to be flexible and variable in parametric designs (Stiny, 1985: 39). According to Roudavski, digital morphogenesis is an architectural term, by which a digital medium is utilized to create a specific form, match a context, or respond to a design problem rather than being used as a display tool. In fact, this concept has a metaphorical association with morphogenic processes of the nature such as gradual evolution or adaptation and certain concepts such as generation, self-organization, or formation (Roudavski, 2009). Neil Leach pointed out computer generative design, in which modeling stems from the design logic rather than an introduced subject. Leach also analyzed the use of computer as a tool to develop forms along with architectural and urban spaces (Leach, 2009). Rivka Oxman proposed five paradigmatic classes of digital design models based on different relationships between designers, applicable design processes, and the design tool itself: CAD models, formation models, generative models, performance models, and integrated compound models (Oxman, 2012). In *Digital Design Processes*, Khabbazi dealt with the contexts for the formation and introduction of areas for digital architecture presentation by introducing the field of process and methodology. The digital design methodology was also introduced in five different layers as separate discourses. The analysis and methodology of digital design processes represent a significant difference in the contemporary design approach (Khabbazi, 2014, 4). In Expressive Form, Terzidis analyzed the idea of expressiveness in architecture through computerized and computational methods to also evaluate some of the formation concepts such as morphology in digital architecture (Terzidis, 2014, 6).

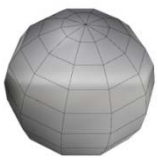
4. Software Applications as Form Design Tools

According to previous theories and studies on the relationship between formation analysis, geometry is a method of achieving design goals in the development of a consistent, appropriate, and stable form. In fact, geometry represents the architectural design language of a building. An architectural concept is based on a geometrical space concept. Architecture is defined in relation to the geometrical space, especially in the design process (Soltanzadeh, 2009). The form of a building is the first item taken into account to find the geometrical shapes in architecture (Evans, 1995). Although geometry may seemsoulless in the design and plans of a building, it is present actively in the abovementioned spaces. Geometry is used as a logical basis and proper tool for reasoning and creating proper design forms. At the same time, a form is a component of a building architecture and the final product of the design process. What helps a designer achieve the ideal form is always a specific design tool. Knowing the architectural design tool and its position in the architectural design process requires the identification of constituent features and components. Once the designing process becomes a conscious action, it will be affected by knowledge, competence, perseverance, tenacity, and design tools. Although design tools are used for explaining a specific action more often, it includes certain features in close relation to the architectural design because they can be employed to view thoughts, analyze ideas and structures, and present plans. Hence, an architect needs certain design tools to display mental images of the design problem in addition to record and keep visions and thoughts for the long term. Such tools also enable architectures to share and evaluate their mental images to make possible changes (Ching & Juroszek, 2010, 2-120). Thus, design tools should be able to express design ideas and concepts very well to put them into reality, something which has a significant role in the realization of thoughts and ideas developed by designers. Design tools help designers by the power to objectify thoughts in different design stages. These tools should be able to develop the design process and present the quality of design accurately in addition to enabling designers to analyze and evaluate plans and designs (Lockard, 1982, 35). Every architectural design tool has specific functions and capabilities in the abovementioned process. Regarding manual design tools, designer record information and data through freehand drawing, manual sketches, and manual modeling and rendering. Therefore, perceptions of design problems and effective factors can be expressed manually. Digital technologies have been acting as tools to boost human efforts. In this area, architecture-related applications have enabled designers to develop primary ideas, test plans, 2D and 3D drawings, material layers, and light based on architectural mentalities in a virtual environment. In fact, such applications receive information inputs and produce different outputs based on a designer's demands or needs through digital data. Considering the information explosion, data processing is very important in the architectural design process and information visualization as conceptual and applied diagrams in today's digital world. These applications act as certain tools in a graphical design editing environment for image processing (3D modeling, rendering, animation) and structural data modeling and designing based on generators and parameters (Ganji Khabiri & Diba, 2014). In both cases, designers complete the design process to analyze and evaluate data to develop a final design. However, the final design is achieved in different durations because different design tools can be used for analysis.

5. Algorithm and Coding in the Design Process

The use of computers in architectural design started with computer aided design (CAD). It has recently ended up with computer design. Therefore, there has been a great development in the mindsets of architectures. CAD is based on entering accurate information into the computer to process them in different applications and finally obtain outputs as data required by users. In this applications, the main functions include editing lines and dots. Computers and software applications were used in architecture offices by Frank Gehry first in 1989. Gehry's office first employed CATIA in Walt Disney Concert Hall and then in Guggenheim Bilbao Museum (Bani, Masoud, 2009, 335). After that, AutoCAD was launched for accurate 2D drawings and sometimes 3D images for both professional and academic purposes. It was then become popular very fast. Other advantages of such applications included the distribution, correction, and transfer of information. Then more advanced environments were developed for the formation of surfaces. In such environments, volumes were defined through covering surfaces. Finally, the SOLID model was proposed. It is based on the idea of creating volumes, analyzing them based on numbers, and converting them into feasible machine work (Golabchi, 2011, 3). Since design applications are based on mathematical and computational relationships, the formation of spatial volumes should also be based on mathematical definitions. There are two common 3D workspaces in computer graphic design systems: polygon mesh and NURBS. In the polygon mesh modeling system, volumes are designed based on the drawing logic in the Cartesian 3D space, in which all of the surfaces and volumes are based on triangles or rectangles. In NURBS, the spatial definitions of surfaces are even and smooth. Table 1 indicates certain features of both modeling methods. The most common 3D modeling applications are 3D MAX, SketchUp, and Rhino. Apart from volume modeling, another classification of image processing can include applications like V-Ray for designing textures and materials and rendering final models. There are also other well-known applications like Revit, used for building information modeling (BIM).

Table 1
Comparing Two Conventional Modeling Methods for Computer Graphic 3D Designing and Modeling

Surface Differences in Drawing a Sphere	Features	Graphical Modeling
	Drawing lines, surfaces, and volumes based on Cartesian points; Shapes of surfaces and volumes based on triangles and rectangles (fragmental surfaces); Providing outputs for digital construction hardware	Polygon (Mesh)
	Drawing lines, surfaces, and volumes based on relative points; Shapes of surfaces and volumes based on free forms (even and smooth surfaces); No outputs of digital construction hardware	NURBS

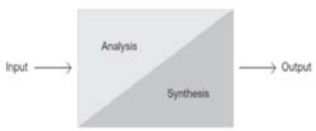
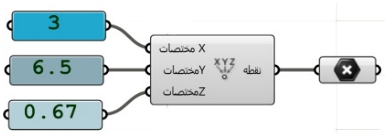
In recent years, algorithmic scripting languages have greatly affected 3D modeling. An algorithm is a computational process which includes multiple parameters or a group of parameters as inputs (Cormen, 2010: 10). After finalizing processes, algorithms produce certain variables in the output. Therefore, an algorithm shows a series of computational steps transforming inputs into outputs (Khabbazi, 2012: 58). In algorithmic methods, scripting languages can be employed to achieve a geometrical instruction by applyinga series of commands or regulations on the initial shape (Woodbury, 2010, 46). Another important factor affecting the algorithmic architecture is the parametric mindset, defined for the architecture design process. In recent years, the direct entrance of scripting languages into architecture has resulted in an approach known as parametric design to control parameters affecting the design process. Parametric design, also known as geometrical grammar, is a new discipline of computer sciences. It is aimed at discovering and utilizing the logical and mathematical relationships between numbers and figures in addition to shapes (Oxman, 2017: 45). Scripting languages were first used as texts in applications (C#, Python, etc.). However, they were not very popular due to difficulty in learning and use of scripting languages by designers. As a result, software developers tried to devise graphical scripting languages as a more user-friendly environment. A graphical scripting language is provided as an extension for users so that it can be installed on the main application. In fact, these parametric extensions enable users to edit every stage of a process based on selective commands due the nature of commands pertaining to the programing language and the algorithm. However, designers and users had not been able to perform such tasks. Table 2 shows the most commonly-used extensions based on graphical programming languages. Regarding parametric applications, the first and most widely-sued graphical programming language is Grasshopper, an extension of Rhino.

Table 2
A List of Most Well-Known Extensions of Graphical Scripting Languages

	First Release	Developer	Parametric Extension	Application
1	2007	McNeel	Grasshopper	Rhino 3D
2	2011	Autodesk	Dynamo	Revit
3	2016	Trimble	Viz Pro	Sketchup
4	2016	Autodesk	Max Creation Graph	3D Max

In graphical scripting applications, the execution of commands is based on defining parameters by their data types. In this method, the primary data are given as a defined input to a command packet acting as a processing operation. Then the final data will be formed through the type of operations performed on the input. This process looks very much like the initial diagram of the design process in the relevant references (Table 3). Therefore, every command issued by a designer is a single-loop process. As a result, the algorithm design process is a series of interconnected commands through a linear procedure. Commands are connected through the types of input data, operations, and outputs. They provide expansion and connection to the next parts. Given the nature of input and output data formed in architecture graphic applications, there are four data groups: 1) visual data: images, drawings, footages, and diagrams; 2) geometrical data: dots, lines, plates, and volumes; 3) numerical data: distance, angle, length, and interval; 4) textual data: tables, diagrams, and mathematical relations. In the parametric design, relevant parameters are defined in practice. Then an algorithm explains the relationships between the variables.Such a design process results in a form, which is easily changeable and analyzable. Such a form is described as parametric.

Table 3
Different Data Types in Graphical Scripting Languages for Architectures and the Process of Creating Data Based on Inputs and Outputs

Basic Diagram in the Design Process Concept	**Relationships between Commands**	**Data Types in Scripting Languages**
		Audiovisual Data (image, footage, etc.) Geometrical Data (dot, line, plate, volume, etc.) Textual Data (tables, diagrams, mathematical relations, etc.) Numerical Data (angles, distance, length, etc.)

6. Findings

The phenomenon of high-rise building has been dealt with more or less in the history of architecture. The first high-rise buildings were built in Chicago in the late 19^{th} Century (Bani Masoud, 2011, 211). Following different advances in technology, the high-rise building approach became very popular to solve certain problems such as the availability of sufficient land, the impossible horizontal expansion of buildings in certain areas, and the optimization of energy consumption. In the last two decades, different design and architecture companies have tried to design buildings in various forms worldwide with respect to new software features and capacities. The appropriate form of a high-rise building along with a reasonable geometrical design can bring about very good advantages in certain contexts such as the proper reactions of the building to orthogonal forces and increased stability against side forces such as winds or earthquakes. Furthermore, the general forms of high-rise buildings are greatly affected by their plots (Hyeong-ill, 2004). The common geometrical forms of towers have been triangles, circles, ellipses, squares, and rectangles in recent years. These shapes are selected due to the use of simple and regular forms in addition to helping the stability and statics of buildings to increase safety (Golabchi, 2010, 61). Now the authors would like to analyze different types of high-rise buildings based on their plots to classify them in their topological plans. Table 4 shows 9 skyscrapers selected as the research samples. They were selected because of their plot forms, certain challenges and differences in formation, and their 3D modeling methods. High-rise buildings, also known as skyscrapers, follow certain features due to constructional, urban, and climatic constraints. As a result, it can be stated that all of the designer companies need to benefit from computer tools and design software applications to design and analyze the specific features of such buildings.

Table 4
Classification of High-Rise Buildings in Their Geometrical Plans, Form Features, and Architectural Styles (http://www.ctbuh.org)

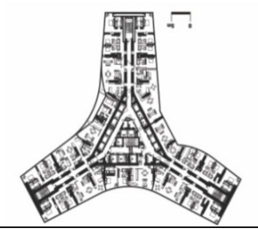
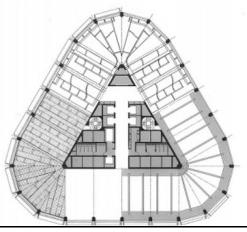
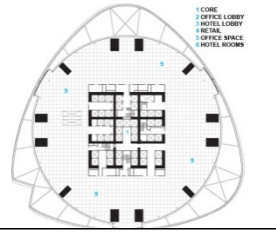
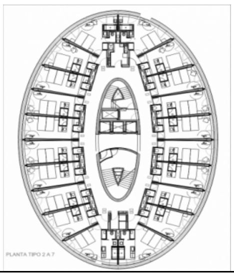
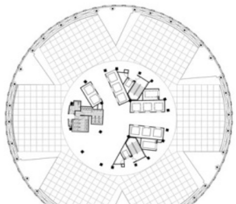
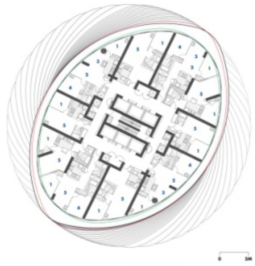
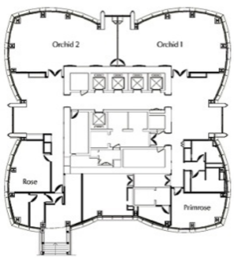
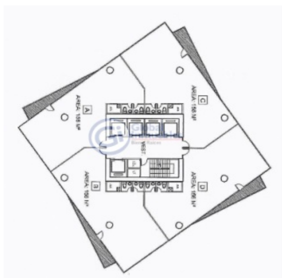
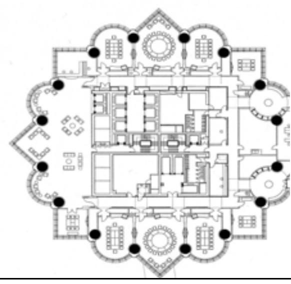
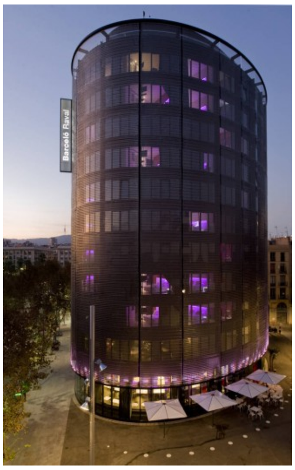
Triangle Plans	Tehran Tower	Riverside-Centre Plan	Shanghai Tower plan
			
	Designer: - Construction Site Tehran, Iran Construction Duration: 1998-2005 The Number of Floors: 54 The Height: 162 meters	Designer: Harry Seidler Construction Site Brisbane, Australia Construction Duration: 1986 The Number of Floors: 40 The Height: 142 meters	Designer: Gensler Construction Site Shanghai, China Construction Duration: 2009-2015 The Number of Floors: 128 The Height: 632 meters
Circular or Elliptic Plans	Raval Hotel plan	Swiss Re Plan	Absolute Tower plan
			
	Designer: CMV Architects Construction Site Barcelona, Barcelona Construction Duration: 2005-2008 The Number of Floors: 12 The Height: 53 meters	Designer: Norman Fooster Construction Site London, England Construction Duration: 2000-2004 The Number of Floors: 40 The Height: 1798 meters	Designer: MAD Architects: Burka Architects Construction Site Michigan, Canada Construction Duration: 2007-2012 The Number of Floors: 50 The Height: 1579 meters
Rectangular Plans	Rose Rotana Tower plan	F&F Tower plan	Petronas twin
			
	Designer: Khatib & Alami Construction Site Dubai, the UAE Construction Duration: 2004-2007 The Number of Floors: 71 The Height: 331 meters	Designer: Pinzon Lozano Construction Site Panama City, Panama Construction Duration: 2003-2008 The Number of Floors: 53 The Height: 2364 meters	Designer: Cesar Pelli Construction Site Kuala Lumpur, Malaysia Construction Duration: 1998 The Number of Floors: 128 The Height: 5621 meters

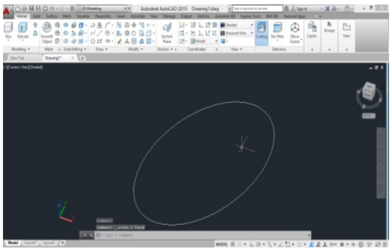
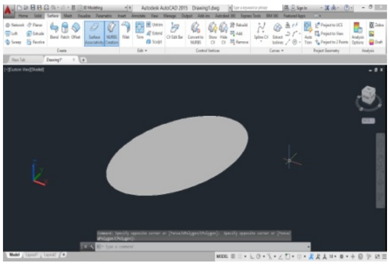
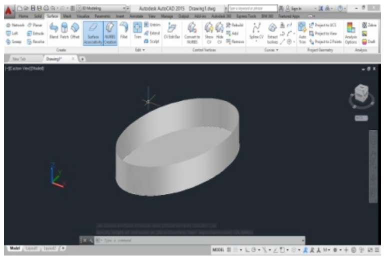
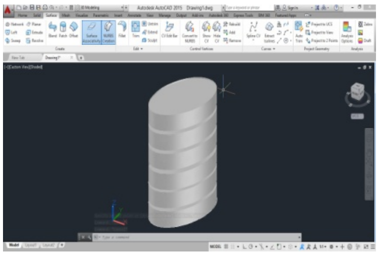
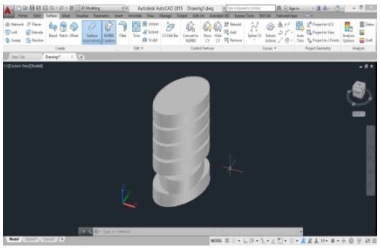
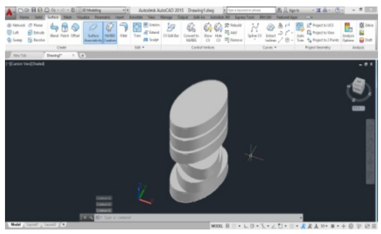
Table 5
Images of High-Rise Buildings Analyzed and Simulated in this Study (http://www.ctbuh.org)

Tehran Tower	**Riverside-Centre Tower**	**Shanghai Tower**
		
Raval Hotel Tower	**Swiss Re Tower**	**Absolute Tower**
		
Rose Rotana Tower	**F & F tower**	**Petronas twin**
		

7. Analysis

After classifying the high-rise buildings in their plans, their modelling processes were performed in Rhino. The commands are executed step by step in Rhino. Therefore, it is not possible to modify or correct commands in the design process. If designers or users make any mistakes, the whole process should be performed from the beginning (Table 6).

Table 6
Selecting Different Commands to Develop and Model a Form in AutoCAD

		
Step 1: using the Rectangle command to draw the plan	Step 2: using the surface command to transform the plan into a shell	Step 3: using the Extrude command to transform a plate into a solid volume (the first floor)
		
Step 4: using the array command to multiply floors at Z	Step 5: using the Rotate command to rotate every floor	Step 6: repeating Step 5 and reusing the Rotate command

Then the modeling method were analyzed along with the way of selecting commands in Grasshopper. In this step, the form design process algorithm was redefined on the basis of the plan geometry by using graphical scripting languages in Grasshopper (Figure 1). In this algorithm, the effective parameters include the plan geometry (data type: geometrical), the height of each floor (data type: integer), the distance between floors (data type: integer), the number of floors (data type: integer), and the rotational angle of each floor based on the previous floor. With the help of constituent parameters of towers, the geometric drawing algorithm was written in Grasshopper. In fact, the input was a plan, whereas the output was solid.

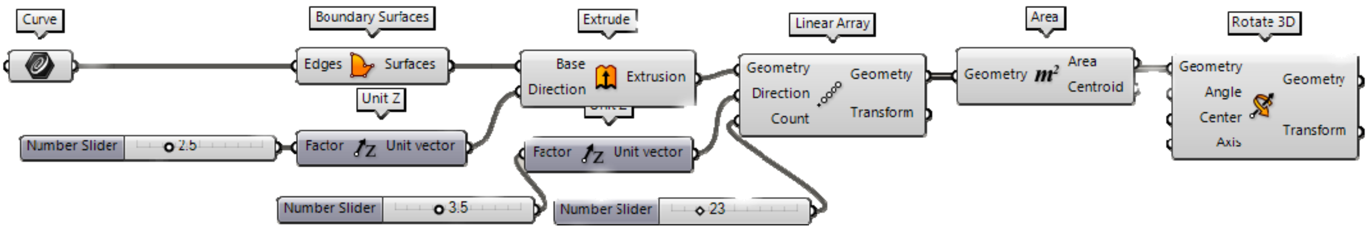


Fig. 1. The Parametric Modeling Algorithm Based on the Types of Commands, Data, and Effective Parameters in Grasshopper

As a result, commands were executed in a continuously integrated way in algorithm applications like Grasshopper, in which designers can modify and change commands and parameters of every step.
Therefore, the most important difference between the algorithm and non-algorithm applications is the use and execution of commands by designers in the form modeling process.
The features affected by this kind of command execution were then discovered. Thus, the drawing process of triangle-based buildings was analyzed and evaluated. Accordingly, control points (on edges and middle of sides) were employed to draw the plan for Tehran Tower based on an equilateral triangle. Then the drawn geometry was defined as the tower plan input in the algorithm. It was then inserted into Grasshopper to perform the necessary processes and obtain the tower volume as the output. The process enabled the researchers to obtain morphogenic triangle plans without drawing and modeling each of towers and only through plan modification (Figure 2). As a result, the developed geometrical algorithm was generalizable. In other words, the selected buildings were characterized by the same algorithm, despite having different forms. In fact, the forms of other towers could be obtained by changing the plan geometry (mophogenic forms) of a building selected as the algorithm input without redrawing the volumes. This generalizability feature is specific to algorithm applications because they are based on writing algorithms in scripting languages. However, nonparametric applications lack this ability, and the form and volume of each building should be drawn separately.

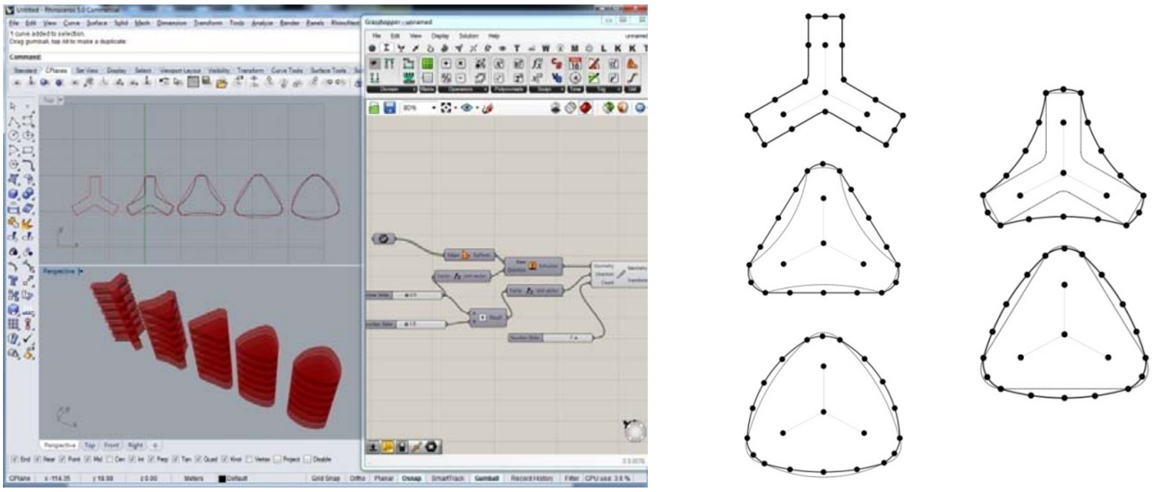


Fig. 2. The Generalizability of Triangular Shapes Based on Control Points and the Formation of Towers Based on a Constant Algorithm

Other capabilities of the algorithmic design methods include writing mathematical relationships of geometrical shapes based on geometrical parameters. For instance, elliptic relations are used in the algorithmic programming language to design curved towers based on two parameters: opening and focal distance. Not only does this process enable designers to draw morphogenic shapes like circles, but it also let them form various shapes by changing numerical parameters such as the number of floors and rotational angle (Table 5). This method can lead to a smart geometry plan so that designers can achieve different forms by changing only a number of numerical parameters in mathematical relationships of geometry plans. The smart and generative geometry can result in unpredictable forms generated by computers. Therefore, designers feel free to select their favorite forms.

Table 7
Formation of Smart Geometry Based on Mathematical Relationships and Changed Numerical Parameters

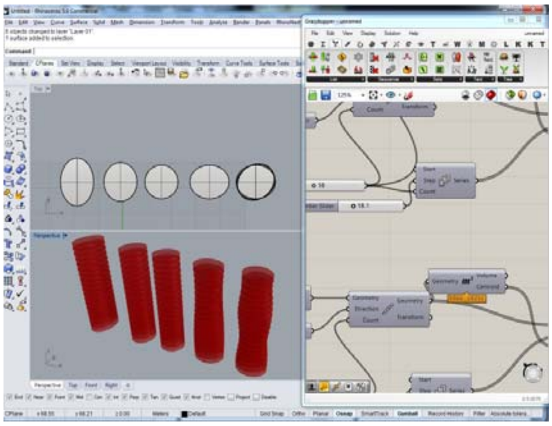
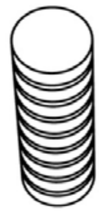
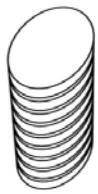
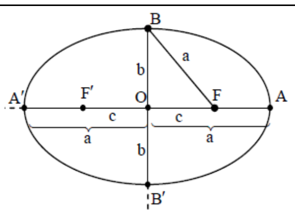
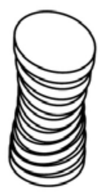
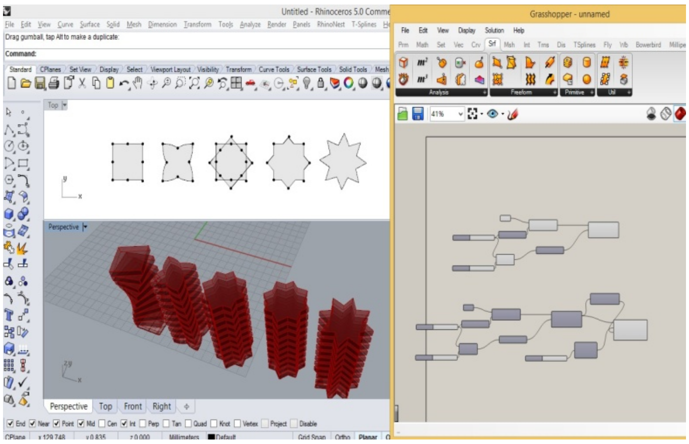
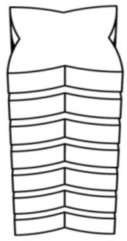
			Circular Plan Tower (Swiss Re Tower); Drawing an Ellipsis with the Ratio of Focal Distance to Opening: 1(circle)
			Elliptic Plan Tower (Raval Tower); Drawing an Ellipsis with the Ratio of Focal Distance to Opening: greater than 1
FF'= 2c =Focal Distance AA'= 2a= opening(a=first radius) B Second Radius $=\sqrt{a^2 - c^2}$			Rotated Epileptic Plan Tower (Absolute Tower); Rotating Each Floor on the Previous Floor around the Vertical Axis
Epileptic Mathematical Relation Based on Opening and Focal Distance			

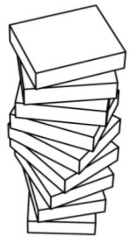
Table 8
Tower Modeling Based on Square Plans in Rhino and Grasshopper



A

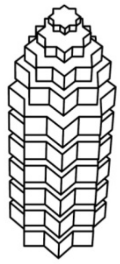


B

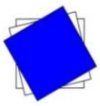


C

D



Plan Formation Based on Control Points



Plan Formation Based on the Rotational Angle of Floors

Formation of Plan Geometry Based on the Number of Sides, Side Angles, and Floor Scales

Table 9
The Useful Features of Algorithmic Applications Compared with Non-Algorithmic Applications in Form Development

New Features	Usefulness
Making Data Parametric	Changing parameters based on data types Editing and modifying commands in every step of the modeling process
Smart Geometry Formation	Form Compliance with the input data types Relating the geometrical grammar development through mathematical relations
Creating the Generative Geometry	Developing unpredictable and novel forms by computers Feasibility, generalizability of shapes, and command expansion

8. Conclusion

Architecture tools and applications are directly related to the design process. Software developing companies design different features matching data types required by users. The most important features of architectural software applications include 2D and 3D drawing, obtaining different analytical outputs in the form of textual and visual data, and creating animations. However, the advent of graphical scripting tools in architecture applications has greatly affected the architectural design process. In this study, 9 high-rise buildings were modelled in terms of their geometrical plans. First, Rhino was used as a non-algorithmic application. Then Grasshopper was used as an algorithmic graphical scripting language to analyze the effects of such applications on the architectural form development. According to the results, scripting languages enable designers to create algorithms and geometrical grammar of shapes based on their constituent parameters. The most important difference between algorithm and non-algorithmic applications is the execution of commands by designers in the design modeling process. In non-algorithmic applications, commands are formed and executed separately step by step. However, the execution of commands is a continuously integrated process in algorithmic applications. The distinct feature of algorithmic applications is that designers can make forms parametric based on the input data. The parametric feature enables designers to use a system in which a group of relationships interact with each other between several variables contributing to a larger unit. In this method, it is possible to change or modify every command packet. In other words, if an error occurs in the selection of the input data by the designer, it will not be necessary to restart all of the commands and redo the modeling process. In fact, it is possible to modify and change every command whenever necessary. Another useful feature of this methods is the use of algorithmic design to achieve the smart geometry, in which a mechanism is defined to redefine geometrical grammar and regulations of shapes in a series of variables and mathematical relations by suing the scripting language. Therefore, constant and variable algorithms can be used in the algorithm to form a particular style of geometry which can operate smartly toward changes in variables. This system relates the geometrical grammar to the mathematical relations of shapes. If the parameters of a shape are changed, the modification can be reflected on the variables of other dependent shapes to change the primary features and specifications of the form. The third and final feature of the research process was the generative quality of the resultant geometry. The quality of being generative can bring about novel forms by changing input data. Such novel forms are beyond a designer's prediction and imagination. However, the non-algorithmic design of the initial formsshould be developed mentally by designers to some extent. They can then employ applications to draw and edit the initial forms. Another useful feature of the generative geometry is the generalizability or expandability of the written algorithms. Generalizability provides an environment in which direct and curved lines, dots, surfaces, and volumes can be transformed into each other. Moreover, changing the existing parameters can bring about morphogenic shapes, which comply with specific principles.

References

1) Asefi, M., Imani, E. (2017) Analyzing the Effect of Digital Application on Creativity Improvement in Architectural Design Training, Urban Identity, 11 (32): 79-92.
2) Bani Masoud, A. (2010) Post Modernization and Architecture (Overview of Intellectual Paradigms in Western Contemporary Architecture (1960-2000). Esfahan: Khak Publications. (In persian)
3) Bani masoud, A. (2011) Western Architecture, Roots and Concepts. Tehran: Honar-e Memarie gharn. (In persian)
4) Ching, Francis D.K. & Juroszek, Steven P. (2010) Design Drawing, John Wiley & Sons, Inc., 2nd Edition, New Jersey.
5) Colakoglu, M. (2001) Design by Grammar: Algorithmic Design in an Architectural Context. Massachussetes: MIT.
6) Day, M. (2005) Generative Components. AEC Magazine.
7) Dibaj, S. M. and Soltanzadeh, H. (1997) Architecture and the Philosophy. Tehran: Daftare. Pazhoheshhaye Farhangi. (In Persian)
8) Evans, R. (1995) The Projective Cast. Architecture and Its Three Geometries. Cambridge, Massachusetts: The MIT Press.
9) Faruque, O. (1984) Graphic Communication as a Design Tool. New York: Van Nostrand Reinhold Co.
10) Ganji Khabiri, A, Diba, D. (2014) Form Compliance with Data: Context-Oriented Architecture in the Digital Era, Iran's Scientific Council of Architecture and Civil Engineering, 1 (10) 63-82. (In persian)
11) Gharib Pour, A. (2014) Analytical Comparison between Freehand and Computer Architecture Design Processes, Fine Arts, 19 (1), 5-14. (In persian)
12) Golabchi, M., Andeji Garmaroudi, A., Bastani, H. (2012) Digital Architecture. Tehran: Tehran University Press. (In persian)
13) Golabchi, Mahmud. (2001) Criteria for Designing High-Rise Buildings, Fine Arts, No. 9, 52-62. (In persian)
14) Hemberg E., Gilligan C., O'Neill M., Brabazon A. (2007) A Grammatical Genetic Programming Approach to Modularity in Genetic Algorithms. In: Ebner M., O'Neill M., Ekárt A., Vanneschi L., Esparcia-Alcázar A.I. (eds) Genetic Programming. EuroGP 2007. Lecture Notes in Computer Science, vol 4445. Springer, Berlin, Heidelberg.
15) Hernandez, C. (2006) Design Procedures: A Computational Framework for Parametric Design and Complex Shapes in Architecture. Massachusette : MIT.
16) Hyeong-ill K. (2004) Space efficiency in mixed-use high-rise building. Illinois Institute of Technology.
17) Jabi, W. (2016) Linking design and simulation using non-manifold topology. Architectural Science Review, 59(4), 323-334.
18) Khabbazi, Z. (2013) Algorithmic Architecture Paradigm. Mashhad: Kasra Publications. (In persian)
19) Khabbazi, Z. (2013) Digital Design Process, Vol. 1, First Edition. Mashhad: Kasra Publications. (In persian)
20) Langsdorf, L. (2016) From Interrelational Ontology to Instrumental Ethics: Expanding Pragmatic Postphenomenology, Techné: Research in Philosophy and Technology, 20 (2),112- 128.
21) Levelle, E. (2017) The Morpheus Hotel: From design to production. Retrieved from. www.rhino3d.com/go/morpheus.
22) Lockard, William Kirby. (1982) Design Drawing. New York: Van Nostrand Reinhold Co.
23) Meredith, M. (2008) From control to design: Parametric/algorithmic architecture. Barcelona, Spain: Actar.
24) Nianquan Z. (2001) A computer-based environment for preliminary structural design, design collaboration and design automation of tall buildings. Hong Kong University.
25) Picon, A. (2010) Digital culture in architecture: An introduction for the design profession. Bern, Switzerland: Birkhäuser.
26) Picon, A. (2013) Digital Culture in Architecture, translated by Morteza Khayat Pour Najib, Tehran: Parham Nashr Publications.
27) Porter, T. (1990) Architectural Drawing. New York: Van Nostrand Reinhold Co.
28) Rivka, O. (2017) Thinking difference: Theories and models of parametric design thinking, Design Study, 52 .pp 4-39.
29) Rosenberger, R., & Verbeek, P. P. C. C. (2015) A field guide to postphenomenology. In R. Rosenberger, & P-P. Verbeek (Eds.), Postphenomenological Investigations: Essays on

Human-Technology Relations (pp. 9-41). (Postphenomenology and the Philosophy of Technology). Lexington Books.

30) Roudavski, S. (2009) Towards morphogenesis in Architecture, International Journal of Architecture Computing, 345-374.

31) Sang Min P., Mahjoub E., Sharpe, D.C., Krawczyk, R.J. (2004) Tall Building Form Generation by Parametric Design Process,Tall Buildings in Historical Cities - Culture & Technology for Sustainable Cities confrance, seoul: Architectural Institute of Korea.

32) Soltanzadeh H. (2009) The Role of Geometric & Numerical Symbols in Iranian Utopia, Journal of Population Association of Iran, 3(6),67-88. (In persian)

33) Soltanzadeh H., Ghaseminia M. (2016) Intertextual Relationships in the Contemporary Aarchitecture of the Iran, During 1961-1977, Space Ontology International Journal, 5(4),39-50.

34) Stiny, G. (1985). Computing with Form and Meaning in Architecture: Blackwell Publishing on behalf of the Association of Collegiate Schools of architecture , Automn 1985, Journal of Architectural Education, 39(1) pp. 7-19.

35) Terzidis, K. (2005) Expressive Form A Conceptual Approach to Computational Design. New York: Spon Press.

36) Terzidis, K. (2014) Expressive Form, translated by Kaveh Bazrafkan et al., First Edition. Mashhad: Kasra Publications. (In persian)

37) Woodbury, R. (2010) Elements of Parametric Design. New York: Routledge.

38) Wortmann, Th. & Tuncer, B. (2017) Differentiating parametric design: Digital workflows in contemporary architecture and construction, Singapore University of Technology and Design,Design study, 52 .pp 173-197.

39) Zaero-Polo, A., & Moussavi, F. (2013) Morphogenesis. FOA’s ark. Barcelona, Spain: Actar.