PatGen: Islamic Star Pattern Generator

Birgül Çolakoğlu<sup>1</sup>, Tuğrul Yazar<sup>2</sup>, Serkan Uysal<sup>3</sup>. *Yildiz Technical University, Computational Design Unit, Turkey* 

http://www.bot.yildiz.edu.tr

<sup>1</sup>bigi@alum.mit.edu, <sup>2</sup>turuly@yahoo.com, <sup>3</sup>serkanbot@gmail.com

**Abstract.** This paper describes an introductory computational design course for graduate students of design in which Islamic star patterns are utilized in teaching computational logic and design rules. Formal descriptions of eight pointed Islamic star pattern that depict a variety of geometrical structures and constraints of the Euclidean shapes are extracted with the shape grammar method. Then star pattern generator that runs on a specific CAD system is developed by encoding these formal descriptions in systems scripting language. The examples of new designs are generated, and then fabricated using various CAM technologies.

**Keywords.** Computational design, generative algorithms, Islamic star patterns, architectural education.

## Introduction

Advanced computing is a feature of every field of science and is widely used in technical work for solving various problems. Architectural design which incorporates science and art is facing a challenge of integrating computing into design process.

Computational design, (CD) a fairly new interdisciplinary field, integrates design, computer science, mathematics, and the ability to graphically visualize and provide answers to design questions or to explore new directions in design that are difficult or impossible to build. CD denotes the multidisciplinary combination of techniques and knowledge developed to solve design problems through computer tools. It is a new sub-area in design which merges computational thinking - a collection of diverse skills to do with problem solving which results from the studying of the nature of computation- and design thinking. Computational thinking also consists of some very basic problem solving skills such as the ability to think logically, algorithmically, and innovatively. The act of designing is usually defined as a black box and intuitive process. Therefore, designers are rarely interested to learn computing and to involve in computation of design. They see computer as a tool that provides support for rationalization and visualization of a design project.

However, recent developments in CAD/CAM systems necessitate designers to operate on explicit knowledge. Architects are exploring new computational design methodologies that allow them to go beyond the mouse-screen interaction, into the logic of formal language. (Goulthorpe, 2006; Cache, 2003). They are discovering non-visual or numerically driven methods of computation which assist and enhance their designs.

### An exploratory model for teaching computational design

Computational design is a novel and developing research field without a widely accepted curriculum. This approach in education was first introduced by Mitchell, Ligget and Kvan in *The Art of Computer Graphics Programming* (1987). This book created a source of inspiration for a new research area that explores computational design teaching models (Yakeley, 2000; Celani, 2002) in architecture. New design courses that aim to investigate and explore the process and theories of computational

design have been developed (Nagakura, 1998; Terzidis, 2002; Celani, 2004; Duarte, 2007).

The work presented in this paper, is an example of student work developed in Introduction to Computational Design graduate course titled "Designing the Design" at Yildiz Technical University. The course focuses on the use of algorithms as design tools, rather than coding experiments. The course runs parallel to shape grammar course, in which basic concepts of computation is thought through visual design exercises. The *shape grammar course* teaches rule based logic, algorithmic thinking and formal approach to design. In this course, Islamic star patterns that depict a variety of geometrical structures and constraints of the Euclidean shapes are used to teach above mentioned concepts.

The presented work named **PatGen: Islamic Star Pattern Generator** is developed as final project for above briefly explained courses. The students followed the instructional framework given below for the development of their study.

- 1. Identify a pattern construction problem,
- 2. *Develop* a background knowledge,
- 3. Perform research on Islamic star patterns,
- 4. Develop content understanding of patterns,
- 5. Gather pattern examples,
- 6. Learn to use new generative modeling tools -scripting-
- 7. *Simplify* pattern design problem. Determine accurately the design of pattern via shape grammar method.
- 8. *Identify* variables and relationships for modeling patterns on shape rule schemata,
- 9. *Explore* research findings on small exercises to determine viable relationships between patterns,
- 10. Represent mathematical relationships of parametric patterns,
- 11. Translate variables and relationships into computational model,
- 12. Develop a pattern generator using research findings,
- 13. Generate new pattern designs that derive from existing pattern.
- 14. *Modify* patterns for CAM production,

# PatGen: Islamic star pattern generator

Islamic star patterns tend to be rigidly geometric in design, and feature star-shaped polygonal regions. The largest sets of Islamic geometric patterns are rosettes and starliked shapes. These sets are rotationally symmetric around their centers. There is little information how Islamic star patterns are originally devised. The techniques of constructing these patterns were kept secret and passed from master to apprentice and ultimately lost in history. Although constructing these patterns is an intriguing puzzle; once their mathematical nature is extracted, vast variations of patterns can be constructed. There have been studies conducted on mathematics and software implementation of Islamic patterns (Ozsariyildiz, 1991; Kaplan, 2000).

The analysis of the pattern may vary depending on the designer's cognition and also the way the designer's choice of parametric structuring. Different parametric structuring of the same pattern would yield different variations of the original pattern. The pattern analysed in this study (shown in Figure 1a) is evaluated as a square tiled composition of 8 pointed star rosettes and rhombi located on four corners of the modules (Figure 1b).



Figure 1 (a)The original motif, (b) Decomposition of the pattern.

The original motif is divided into square modules.  $P_1$  represents the size of a module. Each module includes 8 arms and 4 rhombi (Figure 1b). In order to create the shapes (arms and rhombi), various points (such as A,B,B' etc..) are placed using a polar coordinate technique. In this technique; first, poles (O and C) are placed on the module, and then, points (A and B) are defined by distance and angle parameters relative to the poles.

Primary pole (O) is placed at the center of a module. Secondary pole (C) is placed at the middle of the top edge.

After that, two points (A and B) are added. A is placed using relative polar coordinates with primary pole (O). P<sub>2</sub> represents the distance between A and O, and P<sub>0</sub> represents the angle of the line AO with X axis. B is placed using relative polar coordinates with secondary pole (C). P<sub>3</sub> represents the distance between B and C, and P<sub>4</sub> represents the angle of the line BC with the top edge of the module.



Figure 2 (a) Definition of an arm, (b) Parameters of the arm.

The lines between three points A, B and C, define the half arm of a rosette. To create the arm, this shape is mirrored along the OC axis (Figure 3a). In order to define a rhombus, the arm created above is rotated  $45^{\circ}$  (Figure 3b). Then, the line BC is extended until the top edge of the module, defining a new point, D (Figure 4a). The line DC is mirrored to create point C' (Figure 4b). The half-rhombus (points C,D and C') are mirrored along the OC axis to create the rhombus (Figure 4c).



Figure 3. The reflectional symmetry and repositioning of an arm.



Figure 4. The construction of an interconnecting rhombus

In order to reach the final shape of the rosette (Figure 5c), the rhombus is rotated from the center 3 times at  $90^{\circ}$  (Figure 5a); and the arms, 7 times at  $45^{\circ}$  angular intervals (Figure 5b) within a module.



Figure 5. The rotational symmetry of the arms and the rhombi.

### The script interface

The pattern generation system was developed in *3ds Max scripting environment*. Due to its flexible software architecture, the user can develop plug-ins requiring no extensive knowledge on procedural programming languages. Maxscript along with the *macro scripting* facility of 3ds Max provides easy access to most of the software components, thereby expanding their capabilities. Also, it allows developers to create GUI for the plug-ins. The pattern generation script in this study is taking advantage of this feature (Figure 6). PatGen generates new Islamic star patterns using the language of the analysed original pattern.

- Par	ameters		
-Pattern Parameters-			
No. of Modules (X):	2	:	
No. of Modules (Y):	2	:	
Module Size:	200,0	:	
Inner Star Radius:	30,0	-:	Limits V
Inflection Radius:	30,0	:	~
Inflection Angle:	22,5	:	
3D Parameters			
☑ 3D Pattern 0n/	Dff		
🔽 Beveled Profile	On/Off		
Line Width: 5.	.0	=	
Line Depth: 1	0,0	•	
Ger	nerate		
	bout		_

Figure 6 The script interface in 3ds Max

The parameters in the GUI of PatGen are grouped under two sections: *pattern parameters*, and the *3D parameters*. Under the pattern parameters section, the parameters needed to construct the pattern in 2D are listed. These parameters are: the inner star radius ( $P_2$ ), the inflection point radius ( $P_3$ ) and angle ( $P_4$ ), the module length ( $P_1$ ) and the number of repetitions on X and Y axes.

The value ranges of three parameters ( $P_2$ ,  $P_3$  and  $P_4$ ) are limited to prevent heavy distortions (Figure 6). These constraints and associations are derived from the geometrical analysis of the original rosette. But in order to support further design experimentation, parameter limiting is made optional.

The 3D parameters section contains the options for the visual representation of a generated pattern, such as 2D/3D switch, profile selection and the relief depth.

It is possible to export the digital data of the generated designs into other CAD software for further improvements. The functionality of the script was tested under different combinations of parameters (Figure 7). Later, one of the generated patterns is selected for manufacturing. The digital data created by the plug-in is processed by a plasma-cutter.



*Figure 7.Star patterns generated by applying different parameters to the script.* (*a*):P2: 15 | P3: 15 | P4:22.5 (*b*):P2: 60 | P3: 41.4 | P4:20 (*c*):P2: 30 | P3: 41.4 | P4:8 (*d*):P2: 115 | P3: 80 | P4:45

The script is further advanced by adding a parameter variation feature. This feature makes it possible to create patterns with varied rosette combinations (Figure 8). In order to develop this feature, new constraints and parameters are defined in the PatGen algorithm.

Instead of planar grid systems, three-dimensional geometric potentials are experienced but left as a further research.



Figure 8. Parameter variations.

#### Conclusion

Computational methods have become a key technology for a wide spectrum of applications in design and fabrication. Computational design is a research driven field that connects faculty interested in computation to those interested in application.

In addition to computational skills, the next generation of architects specialized in computational design need to have sufficient exposure to cross-disciplinary research between computation, sciences, engineering and other domains of design.

Computational design education encourages design students to improve their "competitive edge" by cross training in computational science and arts. The course presented in this paper emphasizes that computational thinking is not only programming but rather conceptualizing that operates in the multiple layers of abstraction simultaneously. It points out that computational thinking is abstract, involves problem-solving and is a contemporary design skill. It explores new ways of teaching computational design to students who do not have background in computation but are educated as designers. It introduces the students with systematic thinking and formal languages and teaches them basic concepts of programming with <sup>3rd</sup> generation CAD software that utilize the graphic capabilities of symbolic computation through scripting.

This approach challenges students to take responsibility for their own learning. It is a learner-centered approach in which students learn better by actively interacting with computer and their colleagues.

#### References

Cache B.: 2003, "Towards a Fully Associative Architecture", in *Architecture in Digital Age – Design and Manufacturing* edited by Kolarevic, B.

Celani G.: 2002, "Beyond Analysis and Representation in CAD: a New Computational Approach to Design Education," Ph.D. submitted to Department of Architecture at Massachusetts Institute of Technology.

Celani G.: 2004, The Symmetry Exercise: Using an Old Tool in a New Way, SIGraDi 2004.

Duarte. J.: 2007, "Inserting New Technologies in Undergraduate Architectural Curricula," Predicting the Future 25th eCAADe Conference Proceedings.

- Goulthorpe M.: 2006, "Scott Points: Exploring Principles of Digital Creativity," in Architecture in Digital Age Design and Manufacturing edited by Kolarevic,. B.
- Kaplan C.S.: 2000, "Computer Generated Islamic Star Patterns", in R. Sarhangi (ed.), Bridges 2000 Proceedings.
- Mitchell WJ., Ligget R.S & Kvan T. (1987) The Art of Computer Graphics Programming, New York: Van Nostrand Reinhold.
- Nagakura, T. 1998, Formal Design Knowledge and Programmed Construct, course taught at MIT, http://cat2.mit.edu/arc/4.207/
- Ozsariyildiz. I.S.: 1991, "Conceptual Design by means of Islamic-geometric-patterns within a CAAD environment," Ph.D. submitted to Department of Architecture at TU Delft.

Terzidis K.: 2002, Algorithmic Architecture, course taught at GDS Harvard

Yakeley M.: 2000, "Digitally Mediated Design: Using Computer Programming to Develop a Personal Design Process," Ph.D. submitted to Department of Architecture at Massachusetts Institute of Technology.