SEARCHING FOR COMPUTATIONAL REGIONALISM

A New Middle Eastern Material Culture

FAYSAL TABBARAH
American University of Sharjah, Sharjah, United Arab Emirates
ftabbarah@aus.edu

Abstract. This paper outlines a developing teaching and research pedagogy being developed at an educational institution in the United Arab Emirates that resists contemporary models of Computational Orientalism. The paper describes that exploring the space between Middle-Eastern material culture and contemporary computational design methodologies as generative, systemic, and adaptive conditions allows for the development of a material culture that is both local and driven by the computational zeitgeist. More specifically, the paper outlines the historical relevance of the computational spline and how it is being explored to develop a novel material culture through parametric models and physics-based systems.

Keywords. Generative Design; Middle-East; Regionalism; Parametric Modeling; Splines.

1. Introduction

Driven by the desire to advance of the human condition, Buckminster Fuller vigorously called for a Comprehensive Anticipatory Design Science (CADS) to unsettle human behaviours that are responsible for social, environmental and economic problems. Always the techno-optimist, Fuller developed CADS as a systems-driven worldview that foregrounds the potential of technology and design to anticipate and comprehensively solve problems of living (Fuller and Kuromiya 1992). CADS pushed for a paradigm shift in how we conceive of the built environment towards more productive modes of living. A collective paradigm shift was only possible through the democratization of technology. Today, it is undeniable that democratizing technology in
general, and computation in specific, has brought about this much needed paradigm shift.

Ever since the 1960’s, architecture, as with other fields dealing with information, communication and the public realm, has been undergoing a radical revolution brought about by the universal democratization and pervasiveness of computation. Computational design has become an essential part of both design and production, radically transforming how the discipline generates, develops and communicates ideas of space and living. This is an offspring of the techno-optimism instigated by Fuller. However, as realpolitik continues to prove, democracy, in failing to realize that a one-size-fits-all worldview is its ultimate tragic flaw, exhibits fascist tendencies. In architecture, the democratization of computation has led to an alarming sense of homogeneity that marks infinite architectural endeavours globally, both in academia and practice.

Nowhere is this material homogeneity more evident than in the current architectural trajectories of the Middle East. New urban centres in the region continue to blindly emulate predominantly western models of living and production without sensitivity towards cultural, social and environmental particularities. Educational institutions are not free of blame either, as the Middle-East’s educational programmes continue to operate in one of two failing models; the first operates in defiance towards global models, preferring age-old archaic strategies, and the second imports non-adaptive western educational models.

This paper attempts to outline a model of resistance towards this growing trend of Computational Orientalism through describing the ongoing development of a critical teaching and research pedagogy at the Department of Architecture, American University of Sharjah, U.A.E, in itself a modern-day Constantinople. Computational Orientalism refers to an emerging practice of homogeneity that is being violently imported into the Middle-East through naïve uses of computational design methodologies. At the heart of this pedagogy is the belief that battling the alarming global homogeneity that has made its way into the Middle-Eastern architectural fabric can be achieved by juxtaposing Fuller’s systems-driven approach with a historicocultural attitude towards material culture. Juxtaposing the generative, systemic and adaptive attributes of both computational design methodologies and Middle-Eastern material culture provides a space for a novel computational counterculture. The goal of this pedagogy is to develop a generative architectural language driven by computational spline geometries that is simultaneously local and reflective of the zeitgeist of computation.
2. The Political Context

Exploring the Middle-East’s material culture as a generative condition cannot be done without drawing lines in the sand. The paper breaks away from the mainstream practice of referring to the historical work in question as Islamic. While it is undeniable that the advent of Islam during the 7th century and its rapid expansion into the Middle East and beyond radically affected the development of the region’s art and material culture (Khalili 2005), the paper still prefers the less ideological and more geographic label of the Middle-East. It is less reductive of the complex demographic, cultural and religious condition that has always existed within the region till the present day. As the concept of the Middle-East is in itself controversial and vague, the paper adopts the Arabic speaking world from the edge of the Atlantic in the west to Yemen in the east including modern day Iran.

3. A Brief History of Computation in the Middle East

Islam’s shunning of iconography, the modus-operandi in neighbouring practices such as in Byzantium and Persia, led the emerging culture to seek alternative models of material practices. This search led to a shift away from the representational image-icon towards computational spline-pattern practices. This refers to a technique of creating patterns that are primarily constructed out of interpolated splines due to its logic of construction.

Three computational practices emerged during this aesthetic and material shift. Shown in Figure 1, these practices are respectively (a) calligraphy, the art writing, (b) geometric patterning, the generative practice of symmetrical subdivisions to create high resolution geometric patterns, and finally (c) Arabesques, the practice of creating intricate patterns through computationally organizing foliage-like spline geometries (Sutton 2007). Visually, these practices share all exhibit an overarching use of systemically organized line geometries to generate adaptive multi-scalar patterns.

Figure 1. (a) Folio from a Qur’an, Middle East or North Africa, 8th–9th century. (b) Typical geometric pattern. (c) Relief at the Muttingan Mosque.
Beyond the visual, what connects these material practices with contemporary computational models are three shared values. They are (1) generative, (2) systemic, and (3) adaptive. Generative design is any output that is developed through a rule-based logic. John Frazer notes that "these rules are described in a genetic language which produces a code-script of instructions for form-generation" (1995). In generative design, objective criteria of success must be designed into the rules to allow designers to make decisions. Rules or algorithms need not be executed by a computer as the historical generative practices described above were computed in analogue methods.

All generative output is inherently systemic. Jay Baldwin, long time Fuller friend and scholar, noted that Fuller "didn't say things halfway. The term means exactly what Webster's definitions of words imply" (1992). In this, he means a system is "a regularly interacting or interdependent group of items forming a unified whole" (Webster’s 2013). This is essential to the structure of all computational design methodologies. These systems operate by generating complexity from the ground-up through designing the interaction of simple parts. The attitude of conceiving of the environment through part-to-whole relationships was prevalent in the work of Fuller on all scales and disciplines. This translates to this teaching pedagogy through both, the development of codes and the architectural expression.

Finally, both Middle-Eastern material culture and computational design methodologies are conceived as adaptive systems. The above described practices were adaptive in that they were appropriated for calligraphy, book design, textile design, jewellery, pottery, armoury design, architecture and numerous other material practices. Computer code is adaptive in that it allows for radical differentiation through minimal changes in the code structure.

4. Teaching Pedagogy and Mechanisms

The development of the teaching pedagogy has been ongoing since Fall of 2011 through two interdependent fronts, practice/research and teaching design studios and advanced computational design seminars at the undergraduate level. Understanding the particularities of the institution in which this pedagogy is currently developing highlights how the process can be adaptively deployed within other teaching contexts that are struggling with similar issues in the Middle-East or beyond. The College of Architecture, Art and Design sits within the larger American University of Sharjah, thus, it self-organizes to house a diverse cultural mix amongst faculty members and students.
Understanding this specificity allows for teaching opportunities that might not be available when teaching computation design methodologies elsewhere. While teaching the tools reflects similar modes of teaching globally, students at the higher levels explicitly understand the goals of this teaching pedagogy. Over the last two years, two trajectories have been identified through teaching and research, (a) Spline Assemblies and (b) Spline Formations.

4.1. EARLY COMPUTATIONS

In the first computational design seminar I taught, I asked students to develop rule-based applications targeted at generating existing Middle-Eastern geometric patterns. Historical geometric patterning techniques relied heavily on the interaction between two material elements, a thread of a specific length, with a stake driven into the drawing surface at one end, and a chalk at the other (i.e. compass). This allowed the construction of complex and infinitely line tiled geometries based on fixed-radii circles (Figure 2). The organization of fixed-radii circles was used to extrapolate Cartesian lines that connect at circle centres and intersections. The result is a high resolution, multi-scalar system of Cartesian lines. Here, we are referring to Cartesian geometry because we can extrapolate how they exist in Cartesian space.

![Figure 2. Student Work (Haniyeh Chuhdri): The image on the left follows a procedural logic, while the one on the right is a parametric model.](image)

Early generative exercises of constructing existing and novel geometric patterns occurred through both procedural and parametric models (Figure 2). The procedural model follows a series of rules that govern the organization of circles and resulting pattern formations. A real-time parametric model was introduced to increase efficiency through automation. Students were also asked to find alternative ways to the organization of fixed radii circles to develop the same pattern parametrically. Examples of this are models designed...
through iterative polar arraying operations on lines representing the radius of the base circle. These early exercises laid the groundwork within the student body for a form of generative thinking through minimal means (Cartesian circles and lines) based on familiar patterns due to their ubiquity within the Middle East in general and contemporary practices in the United Arab Emirates in specific. Due to this ubiquity, the focus was not on the visual language of the output but rather on the behaviour embedded within the developed computational models. To fight this ubiquity, we moved into exploring Arabesques as they provided a new computational plateau here-to untouched in contemporary practice: the specificity of spline geometries within Middle-Eastern material culture.

Architecturally, Greg Lynn led the drive towards spline geometries in 1990s through his book, Animate Form. Relevant to the topic at hand is that Lynn viewed spline geometries through the lens of Baroque topologies that are in many ways indifferent to Middle-Eastern topologies.

The generative logics on which spline-driven arabesques are constructed result in formations which cannot be described simply in Cartesian space. While initial set outs responds to Cartesian space through the use of circles, the move away from fixed radii circles allows for a subsequent shift from Cartesian lines towards splines. Computationally, a spline is "any smooth function that is piecewise polynomial but also smooth where the polynomial pieces connect" (Judd 1998). Polynomials are functions that have a number of unknown variables and go through an algebraic procedure such as the singularly indeterminate quadratic equation: $x^2 + 2x + 1 = 0$. When working with splines polynomial interpolation means the ability to compute a function that cannot be described through Cartesian coordinates and has a finite number of unknown variables. In essence, splines are an approximation, and more precisely, an interpolation method that results in a smooth curve. A spline is different from a Cartesian line in that it "is not reducible to exact coordinate" and is "not defined as points but as flows" (Lynn 1999).

Where two-dimensional interpolations are directly related to the variable-radii set outs, three-dimensional examples have an added layer of complexity with the introduction of material behaviour. Objects such as jewellery pieces might seem symmetrical at first, but its scale and material behaviour push artisans towards an interpolation that causes local asymmetry (Figure 3).

4.2. SPLINE ASSEMBLIES

Spline assemblies are conceived as tectonic constructions. Architecturally, they follow in the lineage of projects such as Paxton’s Crystal Palace in that
one can linearly understand and breakdown the hierarchical relationships of its parts to the whole system. Computationally, the troika of the generative, systemic and adaptive is resolved through parametric models.

Assemblies are developed through creating a series of parameters that drive the digital construction of the splines. Parameters include local and global attributes such as the number of the splines within the system, degrees of interpolation, lengths, number of control vertices and their relationships in space. These are fed into an environment that allows for a feedback loop to develop spatial and structural complexity (Figure 4).

Figure 4. Student Work (Hafez Koohestani): Input into this model was a competition between the minimum radii at any single point on the spline, the clear interior height, and length of each spline.

Figure 5 shows variations on a parametric roof structure. Here, the model had an input in the form of number of bays and responds to three criteria: (a) size of ETFE panels, (b) percentage of spline/air distribution system to ETFE panels and (c) amount of covered space. Parametric models are inherently design with CADS in their genetic make-up. They are designed as a
comprehensive system of interdependent elements that exhibit flexibility, allowing designers to anticipate input and output variations and respond accordingly.

![Student Work (Hatem Hatem): Parametric spline assemblies for a roof structure.](image1)

4.3. SPLINE FORMATIONS

While spline assemblies are developed through parametric models, spline formations are developed in two spaces: (a) physics-based software that allows for control and customization through high-level programming language and (b) Object-Oriented Programming (OOP).

In both models, splines organized in a physics-based environment and are programmed to interact with computational particle systems. Coined in 1982, "a particle system is a collection of many many minute particles that together represent a fuzzy object. Over a period of time, particles are generated into a system, move and change from within the system, and die from the system" (Reeves, 1982).

![Student Work (Hafez Koohestani, Nada Abu Shaqra, and Maryam Alzaabi): Example of spline formations.](image2)
Working with particle systems allows for two conditions that parametric spline assembly models did not exhibit. First, particle systems models are inherently dynamic, while parametric models have to be programmed specifically for time-based behaviour. One of the biggest advantages of the dynamism of particle systems is the ability to create variation in particle attributes over time. Second, the use of iso-surfacing techniques to give form to particle systems results in topologically complex models reminiscent of Middle-Eastern space conceptions (Figure 6, Figure 7).

The example in Figure 7, constructed as a teaching example of form-finding through programming in a seminar, portrays the result of connecting a class of particle-chains to spline geometries that are subsequently input into a gravity driven environment. Here, the particles are programmed to increase in size as they move up in space. Particles are also created in the average 3-dimensional distance when two particles from different chains come within a specific distances to allow for maximum topological connectivity.

5. Next Steps

The teaching and research pedagogy described above is a work in continuous progress. The author recognizes a number of areas where work is yet to be done. For example, since the examples described above ultimately reside in the digital, we are exploring feedback loops between digital and material computational models. From an architectural point of view, the focus has been on how form communicates culturally. While this remains at the core
of this exercise, material computation models are currently foregrounding issues of structural and environmental performance.

One of the biggest struggles is the search for novel conditions that communicate culture without reflecting the tendency of computational homogeneity in architecture. Recognizing this will drive the work forward.

## Endnotes


## Acknowledgements

Many thanks must be made to the students and colleagues who have helped in the development of the work discussed above.

## References