Hand Crafting Computational Design Thinking in Basic Design Studios

John H. Abell

Washington State University

Introduction

This paper presents examples of the author's approach to introductory architectural design instruction. The approach balances computational design thinking concepts and skill sets with traditional design principles and hand-craft skill sets. The paper is a qualitative-explanatory case study. The case examples are drawn from assignments that entail computation design thinking and specifically parametric and algorithmic concepts to explore tectonic pattern making and form finding goals with linear and planar material elements. Examples (Fig. 1 - Fig. 4) provisionally answer questions concerning how to balance traditional and non-traditional skill sets early in the design curriculum to better prepare students for design thinking and architecture design practice today.

Generally, the main problem introductory architectural design courses tackle is how to help students learn basic skills, concepts, and creative practices that are significant in architectural design practice. An assumption reflected in introductory architectural design textbooks is that it is best to focus on traditional design principles and techniques. However, design thinking research suggests that acquisition of concepts and skills molds problem solving practices and informs the generation of design ideas and design results.[i] Thus, if the goal is to encourage design thinking and design practices that balance traditional and non-traditional knowledge and skill sets, both should be introduced and balanced early in the curriculum. Unfortunately, instructional material is scarce on balancing traditional concepts and skills with those that are non-traditional like computational design thinking and associated algorithmic and parametric concepts.

This paper includes background discussion on the issues described above and discussion of assignments balancing traditional concepts and skills and computational design thinking and documentation of how the balance is reflected in student results.

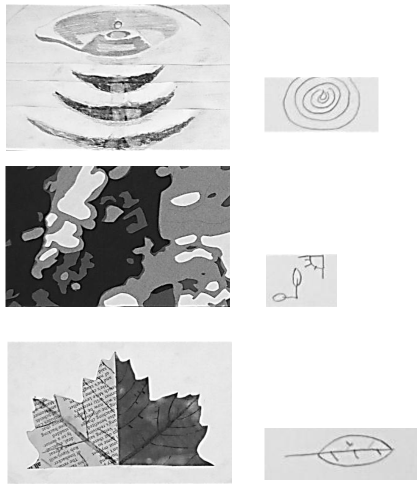

Fig. 1. Pattern identification exercise, representation of repetition & variation. Mixed media paper collage, pencil drawing. Top. Water drop "releases its energy across the entire surface…all connected." Middle. "shadow…of light bending around or penetrating leaves." Bottom. "the structure of the veins in a leaf defy…symmetry [man] can still learn from nature's most basic building blocks." WSU Arch 101 '12, Instructor John Abell, Student, Jordan Beck.

Balancing traditional and non-traditional skill sets

Recent design thinking and problem solving research call attention to the benefits of balancing traditional and non-traditional design knowledge and practices. The research suggests that vertical-traditional problem solving methods, once established in the mind of the designer, can become an obstacle to problem solving. Early on, Wertheimer (1959), a founder of Gestalt theory, cautioned that traditional patterns in ill-defined design problem solving can prevent one from seeing the problem situation in terms other than those that fit fixed traditional problem solving patterns. Vertical-traditional problem solving is associated with breaking a problem into parts and focusing on parts. Gestalt theory suggests that ideally problem solving entails having principles with which to identify, define and work with problems holistically and dynamically. Holistic problem solving entails iterative reformulations of problems and part to whole relationships. In other words, in an ideal world, this would entail seeing holistic relationships among the parts of design problems and the dynamic nature of the relationships.[ii]

Also recently, Alhusban observed significant differences in horizontal and vertical problem solving abilities based on whether or not architectural design students were exposed to alternative ways of thinking about and defining design problems and practices. In the study, vertical design thinking was defined as “traditional” thinking and as having a “depth” and “detail” focus. Horizontal design thinking (a.k.a. lateral design thinking) entails “breadth” and “critical” thought processes connecting different parts of the design problem in new ways. Vertical thinking lends itself to design developments after conceptual or schematic design. Horizontal design thinking lends itself to bridging and synthesizing alternatives during early stages of problem solving.[iii]

Recent studies by Carnegie (2012) and Alhusban (2012) reinforce the idea that design innovation for ill-defined problems entails the ability to restructure the problem space iteratively to co-evolve problem and solution relationships. Further, these studies reinforce the theory that the designerly way of thinking is a generative trial and error process and that this process is abductive.[iv] However, if a design process is contingent on the application of traditional problem solving strategies to bridge the problem-solution space, the resulting design is less likely to holistic connect and synthesize the parts of the design problem.

The discussion above supports the supposition that the acquisition of concepts and skills prior to design problem solving, as well as during problem solving, molds problem solving practices and informs the generation of design ideas and design results. The discussion also supports the idea that introducing architecture students to alternative problem solving concepts and skills early in their training will benefit their design thinking. Differences in traditional and non-traditional design thinking are not difficult to define, even if only provisionally. What is less well understood is how traditional and non-traditional problem solving strategies can be balanced in the introductory architectural design studio to best prepare students for architectural practice. One difficulty is that the literature on introductory design focuses on traditional design knowledge. On the other hand, examples of non-traditional design focus on digital design results rather than computational design thinking.

Architecture, Form, Space and Order and *Design Drawing* by Francis Ching are standard reference manuals for introductory architectural design studio teaching and learning. *Form, Space and Order* illustrates a broad range of traditional architecture design principles: axis, symmetry and asymmetry, hierarchy, datum, center/edge, rhythm static and dynamic order, repletion, transformation, solid/void and figure/ground relationships, proportioning systems and regulating lines, as well as balancing order and variety in design. *Design Drawing* directly addresses design thinking and practice, specifically “Speculative Drawing,” “Creative Process” and “Digital Concepts.” One of the overall suppositions in both manuals is that relationships between the diagram, material form and design practice are close and mutually influential. Further, these relationships have explicitly informed architectural discourse since the Acropolis, 5C BC. Ching’s diagrams illustrate how elementary geometric shapes, patterns of shapes, as well as repetition and variation and transformation in patterns form a universal basis for understanding architectural design principles and practices.

However, Ching’s brief discussion on digital design concepts in *Design Drawing* focuses on the use of computer software rather than computation design thinking. Ching mentions layering and undoing operations afforded by software for managing graphic information and how with “Boolean Operations” one can merge shapes additively and subtractively[v]. It is not difficult to see how one might expand Ching’s brief discussion, even if only provisionally, to balance traditional concepts and skills with non-traditional computational design thinking and associated algorithmic and parametric concepts.

Mid-century modernism & proto-computational design thinking

Ching’s section drawing of Aalto’s Concert and Convention Hall, Helsinki, 1967-71, offers an example of proto-computational design thinking in mid-century modern architecture. In Aalto’s design, one sees an intricate configuration of elements, a configuration that clearly exhibits algorithmic and parametric relationships associated with computational design thinking today. The system of curved acoustical baffles lofted above the seating establish an asymmetrical yet ordered pattern of repeated basic elements, and algorithmic variation in the size, location and orientation of the elements. Clearly, this pattern expresses sound wave behavior parameters. It is not surprising that one of Aalto's projects can be accurately described with concepts that define computational design thinking, or that the material elements of the design reflect algorithmic and parametric design concepts. This is not surprising given the functional imperatives of modern architecture, and given the systemization of the design process in modern design practice. Also, it is not surprising that Aalto's proto-computational design thinking informed his creative processes. Aalto's systematic step-by-step design procedure for the Baker House, MIT, Mass, 1946-49 are well-known. The steps entailed the repetition and incremental variation of basic linear, planar and volumetric elements based on form finding and site configuration parameters to generate alternatives. This formed the basis for his comparative evaluations of design alternatives and design decision making. In this way, building elements and site configuration were correlated with goals concerning adjacencies, paths-space relations, public-private zoning of activities and relationships, massing, views, and day lighting.[vi]

Computational design thinking and parametric design

Aalto's emphasis on the dynamic relationships among interrelated parts of the overall design offers a bridge between traditional and non-tradition design entailing computational design thinking. However, much of the architectural literature on computational or parametric design in academic design studio settings emphasizes digital design at advanced studio levels. There is however several helpful examples emphasizing hand crafted computational design thinking in advanced design studios. Two cases are explicit about the computational basis for design activity. Both emphasize parametric and algorithmic modeling of patterns to design associative systems comprised of basic shapes, lines and planes. In both cases creative modeling and problem solving entail tectonic pattern making, form finding and system configuration activities to create non-linear patterns exhibiting repetition and variation.[vii] Parametric and algorithmic modeling of non-linear patterns (often with linear and planar elements) is fundamental to computational design thinking and parametric design in the architecture, engineering and construction (AEC) industry.

Computational design thinking in Architecture today is defined in many ways, often with reference to natural material systems behavior, genetics, morphogenesis, emergence and self-similarity (e.g. in nature, in swarm and in crowd behavior). Practice oriented definitions of computational design emphasize performance based associative design, systems thinking for design control and optimization.[viii] The concepts of patterns and parametric relationships among pattern elements are central to the many different ways of defining computational thinking.

In *Elements of Parametric Design,* Robert Woodbury defines parametric design as "relating" and "modeling patterns" of interdependent "parts." Diagrams play a central role in parametric design in so far as parametric design is "graph" based with "nodes" that define constraint variables with input–output instructions, rule sets, and procedural instructions defining what the system can and can't do at the node. Algorithms define the instructions.[ix] The graph and the nodes diagrammatically define the performance of the building system. This kind of thinking and modeling in architectural design is particularly useful for performance based form-finding. In practice, successful design decision-making hinges on leveraging material constraints to explore opportunities, particularly incremental variations in form finding and site configuration parameters to generate, evaluate, and revise alternatives. Certainly today there are many important and very public practice-based examples of parametric form-finding and site configuration in the AEC industry. Woodbury highlights two of them: the courtyard roof for the Smithsonian Institution Patent Office Building, (Nigel Young and Norman Foster + Partners), and the International Terminal Waterloo Station by Nicholas Grimshaw & Partners.

While computational design often entails the use of computers to perform various steps and operations, computation itself is procedural, entailing repeatable routines with information-inputs. Computational thinking, whether computer based or hand-craft based (analog) enables a systematic exploration of input-output relationships. While routines are repeatable, input-output results can vary because input can vary. Essentially, computational routines are sets of rules and procedures, algorithms for exploring input-output relationships. Algorithms are often used to model and explain complex and dynamic systems in nature. Nature's patterns are characterized by repetition and variation: patterns like weather systems, cloud systems, flocks of birds, and crowd behavior.

Natural phenomena like flocks and cloud systems do not depend on or require a computer or digital software, but they can be explained, modeled and 'designed' with algorithms and computational thinking. Computers can help with design information modeling particularly when a lot of information needs to be processed procedurally to explore and to better understand input-output relationships in systems that are characterized by component repetition and variation. However, computational design thinking is a way of thinking about design and explaining design, and a way of designing. Computational design entails rule sets and procedural steps to explore input-output relationships, but it does not necessarily entail digital design or the use of computer design software.

This is not to suggest that computational thinking whether digital or analog is equivalent to natural and social systems or processes or phenomena. Rather, the focus here is on how the introductory design studio can balance the traditional analog modeling techniques in architecture with non-traditional techniques in architecture practice.

One intended benefit of this balance is to better prepare students to think critically about digital methods. This could help avoid the uncritical use of digital technologies whether based on traditional or non-traditional assumptions. Another intended benefit is to introduce students to important distinctions regarding material phenomena in the life-world and the abstract modeling of the life-world computationally or digitally or traditionally.

Pattern identification in nature, linear & planar elements, 2D representation and modeling.

The introductory design studio case examples discussed in the following are drawn from assignments that seek to balance traditional design thinking and computational design thinking and particularly parametric and algorithmic design concepts. The results of a studio exercise shown in Figure 1 focus on identifying patterns with basic linear and planar elements and representing patterns with lines, outlines, and sheet materials. The emphasis is on patterns found in the exterior life-world, particularly patterns found in nature, cultivated nature, and the built environment. The goals of the exercise were to introduce the concept of pattern identification and pattern repetition and variation with linear and planar elements. As discussed above, these are core concepts in computational design thinking.

Also, the emphasis in the exercise is on traditional concepts and skills sets; direct observation, insight, interpretation, abstraction and process. Direct observation emphasizes going out into the external life-world to identify and document examples of patterns through sketching and photography. Insight focuses on perceiving the underlying structural pattern of relationships in the observed setting. Interpretation entails explaining the particular meaning of the pattern or subjective experience of the pattern. Students write brief descriptions to explain this (quotations in Fig. 1 caption). Interpretation also entails editing and inflecting patterns, lines and sheet materials to represent meaning or experience. Abstraction emphasizes the distinction between the actual patterns observed and the modeling of the pattern with linear and planar elements. Patterns often lay beneath the surface of what is observed but none the less configure what is observed on the surface. Process entails working materials to study and represent patterns and their component elements, to model them two dimensionally. Process work emphasizes feedback loops including testing and refining visual design communication ideas with materials and techniques, instructor and peer consultations.

Continuous patterned surface design, folding paper, 2D & 3D representation and design modeling

This exercise (Fig. 2 & 3) focuses on mutually influential relationships between 2D pattern diagrams and the 3D material articulation of the pattern as a continuous folding surface. The emphasis is on working back and forth iteratively, 2D-3D-2D-3D. The exercise entails algorithmic procedures and parametric constraints that govern the process and the outcome (input–output relationships). The goal is for the models to materially articulate the diagram in 3D, and for the diagrams to articulate the material form in 2D.

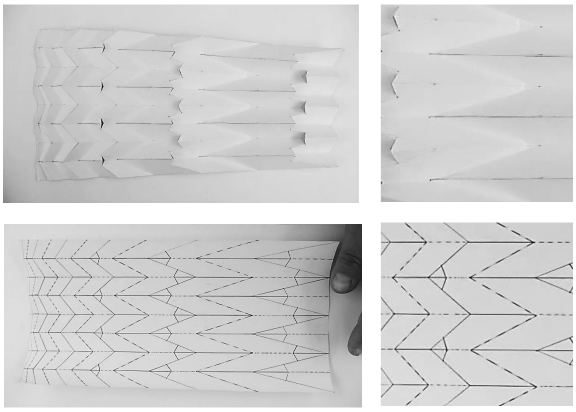

Fig. 2. Continuous pattern surface design: folding paper, 2D & 3D representation and design modeling, Bristol paper, pencil. Top left, herringbone pattern folding paper model. Top right, detail of model. Bottom left, diagram of fold pattern. Bottom right, detail of diagram. WSU Arch 101 ’12, Instructor John Abell. Student, Matthew Staley.

Students begin this exercise in various ways. One way is to identify basic geometric shapes to create patterns. Another way is to begin folding paper – perhaps using basic types of origami patterns like the herringbone pattern. However a student chooses to begin, students experiment with shapes and patterns by copying and connecting, and outlining and tracing, to create patterns – with perhaps 100 shapes or cells making-up each pattern.[x] The pattern is lightly re-drawn on Bristol paper. The pattern on the paper is used to make decisions about where to fold and where to cut the paper to create tabs (no paper can be removed; the pattern governs folding and cutting). The orientation and spacing of the pattern lines are then modified (2D diagram) to create variation in the original pattern. The revised pattern diagram is lightly traced on a new sheet of paper; the paper is folded, cut and compared with the original. These steps are repeated indefinitely to arrive at a pattern that exhibits repetition and variation three-dimensionally along the continuous surface. Students explore and experiment with 2D and 3D pattern articulation relationships while exploring and experimenting with param… straints.

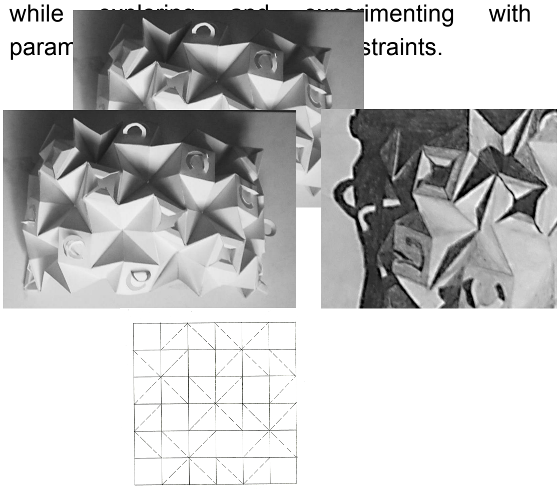

Fig. 3. Continuous pattern surface design: folding paper, 2D & 3D representation and design modeling, Bristol paper, pencil.
Left, ‘star & box’ pattern folding paper model. Right, detail rendering of model. Bottom left, diagram of fold pattern. WSU Arch 101 ’11, Instructor, John Abell. Student, Ting Zhang.

Continuous pattern façade design, planar and linear elements, 3D.

This exercise (Fig. 4) is viewed as an extension of the continuous folding paper exercise (above). This reinforces the concept of continuous associational relationships among linear and planer basswood sticks and sheets. Here, however, greater emphasis is on the interdependent nature of the linear and planar elements of the façade and the associational modeling of the parts in diagrams, orthographic drawings and physical models of the façade.

Façade diagrams play a central role in the design process. The particular focus is on the parametric function of the diagram as a representation of a continuous surface with nodes (ridges, valleys, surface areas between and points of intersection) that define what the façade system can and can’t do. The diagram becomes a critical vehicle with which to define and experiment with the façade as a performance envelope, a differential performance envelope marked by repetition and variation. As discussed earlier, this kind of thinking and modeling in architectural design is particularly useful for performance based form-finding. In practice, successful design decision-making hinges on leveraging material constraints to explore opportunities, particularly incremental variations in form finding and site configuration parameters to generate, evaluate, and revise alternatives.

The focus here is on the mutually influential relationship between the 2D façade pattern diagram and the 3D model of the façade: the diagram and the nodal system of planar cladding elements and linear framing elements. The pattern diagram lines which form ‘valleys’ and ‘ridges’ in the folding paper exercise become potential locations for linear framing elements in the façade exercise. Likewise, planar ‘slopes’ between valleys and ridges in the folding paper exercise become potential locations for cladding and openings for day lighting, and for views in or out.

The emphasis on computational design thinking here is balanced with traditional concerns for the layering and articulation of façade elements, its surfaces and systems, and the expression of craft veracity. Students create study diagrams and models to explore how a façade can filter, shape, and sculpt light, and how façade surfaces and filtering strategies can create transparency, opacity, luminosity, as well as ambiance, sidewalk-scape experience, and pleasure.

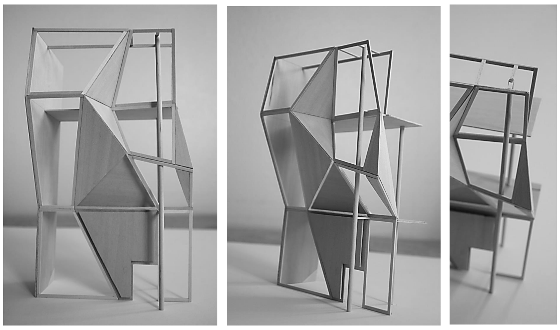

Fig. 4. Continuous pattern façade design, planar and linear elements, 3D, Bass wood model: repetition, variation, & transformation. Left, elevation. Center, side view. Right, top view detail. WSU Arch 103 ’11, Instructor, John Abell, Student, Ting Zhang.

Conclusion

The case examples discussed above are drawn from basic studio exercises emphasizing hand crafted computational design thinking. The exercises balance traditional principles and skill sets and computation design thinking and specifically parametric and algorithmic concepts. The examples reflect 2D and 3D associational tectonic pattern making and form finding goals with linear and planar material elements. These cases offer provisional answers to questions concerning how to balance traditional and non-traditional skill sets early in the design curriculum to better prepare students for design thinking and architecture design practice today.

Notes

[i] See, Kimbell, Lucy. “Rethinking Design Thinking, Part 1,” in *Design and Culture*, Berg Publishers: Volume 3, Number 3, November 2011.

[ii] Carnegie, Teena, A. M., “Design as Problem Solving,” in *Designing Texts, Teaching and Visual Communication*, edited by Eva Brumberger and Kathryn Northcut, Baywood Publishing Company, Inc. 2012. p 44.

[iii] Alhusban, A., *What does the architectural creative leap look like through a conceptual design phase in the underground architectural design studio?* Unpublished doctoral dissertation. Washington State University, Spokane, WA, 2012. p 133-134, 148-149.

[iv] Cross, Nigel. *Designerly Ways of Knowing*, Birkhäuser, Basel, 2007. p 53-58.

[v] Ching, *Design Drawing,* Wiley, New Jersey, 2010. p 326-328.

[vi] See, Reichlin, Bruno. “Controlling the Design Process: A Modernist Obsession” Daidalos, no 71, 1999. p 6-21.

[vii] See, Russo, Rhett. “Wild Cards and Stacked Decks,” in *Meander, Variegating Architecture*, edited by Ferda Kolatan and Jenney E. Sabin, Bentley Institute Press, Pennsylvania, 2010. Also see, Kolatan, Ferda. “Lines and Patterns” in *Meander,* 2010.

[viii] There are many examples. For a representative range see *Computational Design*, AD Reader, edited by Achim Menges and Sean Ahlquist, John Wiley and Sons, West Sussex, U.K., 2011.

[ix] Woodbury, Robert. *Parametric Design*, Routledge, 2010. p 8, 11, 34.

[x] Introduced to me by my colleague Taiji Miyasaka.