ARCHITECTURE REVISITS MATH AND SCIENCE: COMPUTATION IN A VISUAL THINKING PEDAGOGY

Robert Lee Brackett III, Adjunct Assistant Professor, rbrac131@pratt.edu
School of Architecture, Undergraduate Architecture, Pratt Institute, Brooklyn, NY, USA

1. ABSTRACT

This paper makes a case for the greater integration of computational logic and principles in core undergraduate architectural design courses as visual thinking pedagogy. Math and computation present abstract problems that may seem at odds with the real-world design concepts with which students are familiar. Because architecture students are typically strong visual thinkers, abstract mathematical language can be difficult to learn, but these concepts can be used as a pedagogical interface to support visual problem-solving in the design process. Building on the work of Christopher Alexander in *Notes on the Synthesis of Form* and *A Pattern Language*, the idea of "pattern languages" can be used to develop a curriculum that relies on math and computation to connect the visual and social systems at work in the design process. Design curricula can integrate computational thinking based on vector math, geometry, calculus, matrices, set theory, visual programming, and scripting to build students' computational literacy through visual problem-solving. George Stiny's "shape grammars" offer an intuitive analog method for introducing students to computational thinking through elements and rules in preparation for designing with digital tools.

The further we distance ourselves from the fundamental operations of mathematics and computation, the more we risk becoming obsolete in the process. Computer programs can automate modeling, analyzing, programming, reviewing, and even designing buildings. For now, that places the architect in a narrow domain of design and visual aesthetics, which will quickly be subsumed by machine algorithms deployed by software companies. These machine constructions operate at the social/cultural scale, a place suited for the critical position and service of architects. The education of an architect should therefore provide students with critical knowledge and skills that position them to define the parameters of automation and challenge the computer programmers with radical ideas, communicated in a shared language of mathematics that is both visual and abstract.

Keywords: Computational Thinking, Visual Thinking, Christopher Alexander, George Stiny, Rule-Based Systems, Shape Grammars, Pattern Language

2. INTRODUCTION

Computation and mathematics are the foundations of architecture's primary interfaces for thinking and communicating design. Drawing, modeling, rendering, analyzing, and constructing buildings rely on the integration of multiple fields of mathematics. If we distance ourselves from the fundamental operations of mathematics and computation in educating future architects, we risk becoming professionally obsolete, supplanted by computer algorithms that analyze site data to model building variations for population and environment simulations constructed from integrated construction catalogs. Design pedagogy can incorporate the rigor and structure of computational thinking into a visual thinking form-based design curriculum. By approaching design in 2D and 3D mediums as the definition and transformation of rule-based systems, formal variety and spatial expression are embedded with trackable design logic. Pratt Institute's first-year architectural design pedagogy builds from the abstract patterns written about by Christopher Alexander and the shape grammars of George Stiny to develop three specific trajectories in a B.Arch degree: (1) computation can be done visually with analog means that have digital corollaries for generating form and organizational systems of architecture; (2) building simple design skills and concepts step-by-step through a series of design exercises that present learning as a cycle of play, feedback, reflection, and integration that is not focused solely on fostering talent but rather on building individual competency through a flexible and accessible curriculum; (3) abstraction can be achieved visually in drawing and material systems that build on the foundation of two primary ways of designing, additive and subdivision systems, that can work as hybrid approaches to design architectural propositions. The education of an architect should provide critical knowledge and skills that empower one to define the parameters of automation and computerization. Architects must challenge the developers, engineers, and computer programmers with radical ideas communicated in a shared language of mathematics that is both visual and abstract. This paper aims to make a case for the greater integration of computational techniques, principles, theory, and logic in core undergraduate architectural pedagogies through visual thinking.

2.1 Math and Science in the Undergraduate Architecture Curriculum

What is the role and impact of math and science in the education of Bachelor of Architecture degree-seeking students in the twenty-first century? Reviewing the published curriculum tables for several National Architectural Accrediting Board (NAAB)-accredited art and design schools offering the Bachelor of Architecture and BFA degrees, such as Pratt Institute,[1] Massachusetts College of Art and Design (BFA),[2] California College of the Arts,[3] Parson School of Design (BFA),[4] Rhode Island School of Design,[5] and The Cooper Union,[6] reveals that there is a surprising lack of advanced math and science courses required of architecture students. The Cooper Union requires an Analytic Geometry and Calculus course, and Pratt offers Physics for Architects and Ecology for Architects, but no higher-level abstract mathematics, science, or computer science courses are required. This is similar in many smaller schools with an art and design focus. It is unclear to the author if this is a result of NAAB's dense curricular requirements for accreditation that rely on general education, a devaluing of math and science within the discipline, or a shift in focus from isolated courses to integrated learning, as many math and science subjects exist within the architecture curriculum. This void opens the potential for further research, as it suggests a potential gap in critical knowledge and skills required by twenty-first-century architecture students and practitioners to address problems that cannot be solved by compositional design and organizational speculation alone. Architects will need to develop a strong foundation in mathematical concepts that provide an interface with emerging computational strategies for designing, documenting, organizing, representing, and manipulating the complex forces at work in contemporary architecture. One approach to address this is by integrating core concepts from individual math, science, and computer sources into the studio pedagogical model to engage the students with practical and visual translations of abstract concepts.

2.2 Pedagogical Questions for Core Design Studios

Managing complex systems is a fundamental aspect of architectural practice today, but how does a student learn to think systematically about these abstract problems? Architecture must engage with the analytical aspects of science as a model for establishing knowledge and resisting the embedded assumptions in the tools we utilize. Architecture students can learn from the rigorous formal structures of mathematics to build and manage complex knowledge with computation-based strategies. The first year of an undergraduate architecture education should be about developing fundamental abstract design principles that gradually build a foundation for structured intuition using the visual and systematic organization of form, material, and space. This suggests that architecture education must teach a balance between formal and intuitive thinking and that students can position themselves as designers by working at this critical juncture.

3. DESIGN AS ABSTRACT PATTERNS—DIAGRAMS AND THE MATHEMATICS OF FORM

> The analysis of design problems is by no means obviously possible. There is a good deal of superstition among designers as to the deathly effect of analysis on their intuitions—with the unfortunate result that very few designers have tried to understand the process of design analytically. (Alexander 1964, 6)

Architecture can be thought of as an art of form, light, material, and culture, which could also be positioned as a science of geometries, physics, tectonics, environments, and systems. Architecture occupies a precarious territory that negotiates both our subjective experiences and objective control of our built environment. It forms the fabric of interaction and cooperation within our societies as well as the foundations of social and cultural practices. Buildings and cities (and the ideas embedded in their design) are visual and somatic experiences that give form to complex abstractions of spatial organization, movement, access, structure, material, environment, finances, and people. The overlap of these areas exists in the abstract patterns, or diagrams (Alexander 1964, Preface), that allow the quantitative to influence the qualitative, and vice versa, to produce a mathematics of form.

3.1 Christopher Alexander on Formal Design Systems

The dialogue between art and science, between intuition and logic, between form and function, has long been an ontological struggle of architecture. Christopher Alexander's seminal work on the design process, *Notes on the Synthesis of Form*, establishes a mode of design thinking that is critically responsive to this hybrid condition. Alexander defines the design process that considers both aesthetics and function, using a symbolic method that can be applied to specific problems and goals (Alexander 1964, 21). This is an initial positioning of design as a search through a space of possibilities accessible through a formal process, connecting it to scientific fields such as physics, biology, and computer science. These fields offer a structure to the search process to fit form with context, defining a position from which to design that challenges architectural assumptions with data, logic, and fact (Alexander 1964, Appendix 2). This process allows for novelty and variety to emerge from complexity. Alexander also recognizes that the increasing complexity of design problems makes the use of trial and error prohibitive as a design process. Alexander sets the stage for the problem of design as an objective, self-conscious process:

> Today functional problems are becoming less simple all the time. But designers rarely confess their inability to solve them. Instead, when a designer does not understand a problem clearly enough to find the order it really calls for, he falls back on some arbitrarily chosen formal order. The problem, because of its complexity, remains unsolved. (Alexander 1964, 1)

Alexander identifies the growing complexity in the modern design problem and looks to the formal systems of mathematics and scientific study as a method for developing future design thinking. Even in 1965, he was suggesting the integration of computation as a means for processing architectural design at the level of data, a far-removed abstraction from the sketch or compositional methods dominating since modernism emerged in the early twentieth century. The book defines a struggle to find a system that can provide a new way of thinking about design as an abstract process—one that includes analysis of part-to-part and part-to-whole relationships that can address problems of increasing complexity. Alexander (1964) works with mathematicians to develop rigorous formal systems based on graphs and probabilities, sets and matrices, and algebra and functions. A diagram is always at work in the definition and solution of the design problem, and some of the mathematical constructions begin to work diagrammatically to represent form and spatial organization in a drawing. The abstract diagram informs and resolves the mathematics of the equations into a visual format that provides the interface for communicating design as a representation of ideas and things.

3.2 Abstract Design Patterns

> The idea of a diagram, or pattern, is very simple. It is an abstract pattern of physical relationships which resolves a small system of interacting and conflicting forces, and is independent of all other forces, and of all other possible diagrams. The idea that it is possible to create such abstract relationships one at a time, and to create designs which are whole by fusing these relationships—this amazingly simple idea is, for me, the most important discovery of the book. (Alexander 1964, Preface)

While Alexander later shared his strong reservations about the complicated symbolic process outlined in *Notes on the Synthesis of Form* (1964, Preface), he was able to synthesize the rigorous thought exercise of the book into his concepts of design patterns. This is a term (“Design Patterns” or “Architecture Patterns”) often used in computer programming to define schemas of computer code for common computation problems, akin to building codes or graphic standards in architecture (The Open Group n.d.). There have been several attempts to define universal design patterns for architecture. Vitruvius’s *Ten Books on Architecture* documents several design patterns from and for antiquity but shares little with the conditions of contemporary urban technologies and design (Vitruvius 1960). Andrea Palladio's *Four Books on Architecture* and his subsequent body of work have inspired many contemporary designers and provide endless lecture slides. Palladio's design systems, specifically employed in his villas, introduced visual diagrams for designing plans and elements paired with written patterns for creating architecture (Palladio 2002). Le Corbusier's "Five Points" in *Towards a New Architecture* produced a new set of patterns for the twentieth century based on the technological and aesthetic developments of modernism (Le Corbusier 1985). Alexander's *A Pattern Language* was the next major evolution of his idea of the diagram, focusing on the abstraction and creation of patterns that could have a near-infinite recombinant structure to define a building design system. We could even look at these patterns (visual and descriptive diagrams) as architecture's version of pseudocode, a visual language for describing the abstract elements and relationships of a problem before the translation to software and hardware, or, in the case of architecture, drawings, models, and buildings.

3.3 Reflections on the Pedagogical Potential of Abstract Patterns

Alexander's (1964) work provides an interface between the slowly evolving process of vernacular design strategies (unself-conscious), the formal abstraction design principles of modernism, and an emerging territory of design research through formal systems and computation strategies (self-conscious). *Notes* presents a diagram of a village that resembles a tree graph but introduces an analytically and intuitively derived sketch of the ideal conditions of the problem (Alexander 1964, 153). While this may seem to drift back into a historically typological model of design, it offers an abstract structure for visually interfacing with decomposing and coordinating the elements of a design problem (Alexander 1964, Appendix 2). The dynamic dialogue between Alexander's unself-conscious and self-conscious design is further developed in *A Pattern Language,* where he analyzes and extracts design strategies, or patterns, as diagrams for implementing multiscalar architectural proposals that are simultaneously flexible and rigorous as an approach to formal, programmatic, structural, and social organization. He proposes an adaptable and complex way of developing the material spaces of architecture as an integrated whole, carefully considering the scale of each element (Alexander et al. 1977, xxxviii–xi). *A Pattern Language* also presents an approach to the social and cultural aspects of architectural design that provides critical guidance to students about the way we design for life, community, culture, and the future of the planet. Unfortunately, Alexander's diagrams are often analytically derived design patterns rooted in known conditions and require vast amounts of disciplinary knowledge to unpack and deploy. This makes them difficult material for teaching design, especially to students early in their architectural education. We should now ask the question, how can formal, creative, and visual design strategies be taught in a way that incorporates the rigor of mathematics and computation without the need to also be a mathematician or programmer?

4. SHAPE GRAMMARS—VISUAL THINKING AND COMPUTING

Alexander's diagrams provide visual representations of mathematical structures and historical contexts, but without significant historical knowledge of architecture, the diagrams do not provide a way of generating design. Another figure emerging from the dawn of the computer age to explore the relationship between math and design is George Stiny. His

2006 book, *Shape—Talking about Seeing and Doing*, documents Stiny's work with shapes since the 1970s as an interface for visual thinking. His work on computing with shapes and shape grammars provides a direct and visual connection between mathematics and design (Stiny 2006, 58, 319–54). Stiny examines design and architecture from the perspective of a scientist and mathematician searching for an underlying language, or interface, to design that is formal but not symbolic while also producing creative visual results. He speculates on where creativity comes from: "Another one of my big questions is how to be creative in design. My guess is that shapes—drawing them and seeing them—have a lot to do with the answer" (Stiny 2006, 6).

Any element capable of operating as a bit with different states can be used to do analog computations (Tegmark 2017, 61–71), which means a shape in a drawing can operate computationally and generate complexity from simple rules. Abstract structures of mathematics, such as matrices, vector transformations, and schemas, can operate visually without a layer of separating symbols to give them meaning. Schemas of shapes and transformation rules produce interactive visual systems with organizational and tectonic consequences. Algebra, sets, and Boolean logic provide the underlying formal construction, but drawing opens the interface as a learnable and teachable pedagogical system (Stiny 2006, 159–310). The first computers were people calculating outcomes of mathematical formulas for tabulation, so why can't we also calculate using shapes as visual elements interacting in a complex system to compute design outcomes? It is this calculating with shapes and further computing with schemas and rules that can define the overlap of design and math as a pedagogical interface.

Stiny presents four key metaphors about shapes that help bridge the gap between design and mathematics. He states that "(1) Design is Calculating; (2) Drawing is Calculating; (3) Design is Drawing; (4) Design is calculating when you don't know what you're going to see and do next" (Stiny 2006, 311–12). The last point is important, as both design and mathematics are a way to discover abstract structures and relationships that can have physical consequences in the world. Shape grammars can operate both analytically to understand designs of the past, and generatively to compute near limitless design possibilities. His work starts from the abstract visual system of thinking with shapes and works back into history to test the theories. Some of Stiny's initial work focused on analyzing Palladio's villas (1964) based on shape grammars (Stiny 2006, 341–54), which has led to further research and formalization of the process by Athanassios Economou and Thomas Grasl and Georgia Tech's Shape Computation Lab.[7] Palladio's treatise on architecture has had a significant influence on contemporary design pedagogies through the popularization of people like Peter Eisenman,[8] and shape grammars can help make the logic and spatial relationships accessible to any student who can draw and play in a visual medium.

Stiny's ideas also build from and advance the tradition of composition and abstraction established with the Bauhaus and other modern architects. Shapes—point, line, plane, and volume—are still the primary elements of spatial exploration and description, but the shape grammar introduced by Stiny reduces the reliance on aesthetic composition and metaphors. Shapes provide an interface between computation and mathematics, between analysis and synthetic creation. Drawings and models become a computational medium. Seeing and doing are a method of calculating with things instead of symbols (Stiny 2006, 14–15). Related to Alexander's extension of D'Arcy Thompson's idea of design in *Notes on the Synthesis of Form*, shapes as forms are a living diagram of forces (Alexander 1964, 21).[9]

Shapes are geometry; shapes have spatial relationships; and shapes can become architectural form, organization, and structure.

> I can calculate in design without saying much about designing. To begin with, I can analyze designs in a variety of ways. For example, I can look at their physical performance in mathematical models, or I can “rationalize” them—divide them up into components that I can manufacture and assemble. But analysis isn't the only use for calculating in design. I can be more synthetic. I can try functions or parametric representations to enhance my creativity. Equations define lines, curves, and surfaces that are used in pictures, buildings, and many other things that are admired today. (Stiny 2006, 13)

5. FORMAL SYSTEMS—PRATT ARCH 101 PEDAGOGY

The developments of Alexander’s and Stiny’s works provide the foundation of a pedagogy for teaching design in an open way that is not constrained by the rhetoric of the past and prepares students for the future of architecture as a visual and computational field with physical consequences. Pratt’s ARCH 101 syllabus states:

> Architecture is an art of form and material—it is also a social product that participates in economic, political, and cultural contexts. There is a language of architecture lodged in the space of the drawing that utilizes geometry, and the same language of architecture can have agency in the world. Materials, technologies, and systems of architecture are tools through which we can design buildings and environments. This studio will introduce design approaches to architecture students by way of rule-based visual systems that will gradually lead to questions of material production and architecture as a mode of cultural production.

The primary pedagogical interface for developing visual thinking draws from Stiny’s shape grammars by teaching the students to develop systems as schemas of elements and rules created in analog layered drawings. This method’s intuitive, structured, and analytical recursive feedback loop helps students learn at their pace and with their strengths. Stiny states, “Whenever I put a pencil to paper, I'm calculating with shapes or symbols, but there's nothing to code in a drawing, so I don’t have to use symbols in place of shapes to calculate” (Stiny 2006, 311). The geometry and construction of shapes as visual elements defined as collections of lines in a specific organization provides a syntax for visual thinking and designing. The propagation, manipulation, transformation, and organization of shapes through rule-based systems allows students to iteratively build complexity from simple elements (Stiny 2006, 194–95). We call these systems *Design Approaches*, and they are iterative, layered, resilient, and generally substrate independent, meaning they can operate in drawing or model, digital or analog, and at any scale. This independence from the medium is a crucial factor in a visual system’s ability to offer many design solutions at multiple scales as generative abstract systems (Alexander 1964, 135). From the foundation of visual elements and rules, we derive a pedagogy of design that can be taught systematically and with consistency, providing access to all students.

After completing several semesters under the initial shape grammars–based first-year pedagogy developed by Duks Koschitz (2013–18), we undertook a research and cataloging project to extract useful lessons from what the students were learning. From our observations, we developed an ARCH 100 Pedagogy Manual[10] to clarify the Design Exercises and document the interface of teaching design as a formal and creative process. What follows is a brief description of the first-year design pedagogy based on computation through visual thinking. This system continued to run through 2019 and has now transitioned to a new coordinator with a different pedagogical model, though several successful aspects outlined below remain an active part of the curriculum.

5.1 Design as Visual Thinking—Additive and Subdivision Systems

The semester is divided into three parts: (1) a dense six-week technique learning phase of Design Exercises, which increase in complexity but repeat common themes of adding parts and dividing wholes with shapes; (2) a one- to two-week midterm curation phase in which students analyze and reflect on what they are learning to organize their work into two Design Approaches that include 2D and 3D elements as coherent visual systems; and (3) a six-week final project during which they test their understanding of elements and rules as a way of designing a hybrid visual system as a proto-architectural proposal on an abstract site. During this process, students learn to frame and reframe their design decisions and strategies within the two primary contexts of additive and subdivision systems that work with shapes and rules.

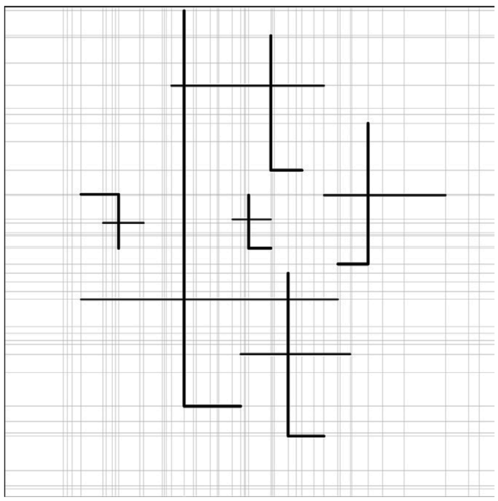
Additive systems work directly with shapes and transformations to build complexity. Students work with many layers of vellum to design and calculate by seeing and doing. They also develop material translations (models) of their discoveries in drawings focused on the visual systems rather than representation. A shape is any group of points, lines, and/or planes and can change and adapt as the layers of the drawing develop. The results produce densely layered fields of visual systems with emergent and embedded forms for further exploration as spatial relationships. This is adapted from Stiny's schemas of shapes and rules, but the algebra and Boolean math are all handled through the direct interface with the drawing or model rather than at a high degree of symbolic abstraction. The outcomes are abstract, but the process of design is concrete and direct. The pedagogy is also rapidly iterative, and inquiry-based, as students are asked to perform written descriptions and formal analyses of their designs weekly.

> The transformations that make rules work move shapes around, turn them over, and make them bigger and smaller. They're operations on shapes that change them into geometrically similar ones. They distribute over the Boolean operations, and may include, for example: Translation, Rotation, Reflection, Scale. (Stiny 2006, 194)

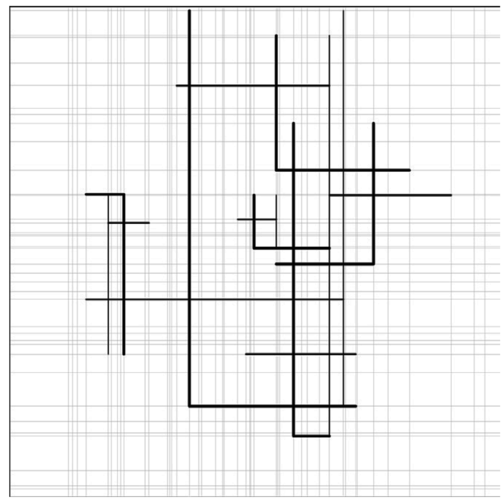
ARCH 101 - FALL

DESIGN EXERCISE 01 · ADDING LINES IN 2D & 2.5D

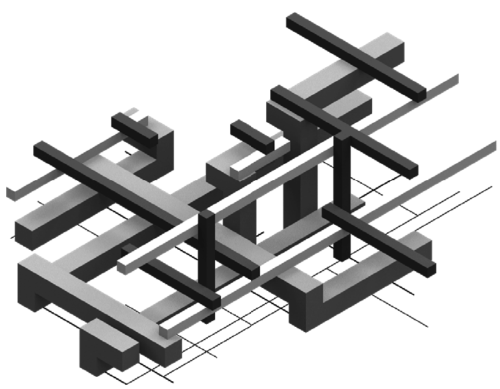
DE01



· Extract [L]-Shapes; Long stem Vertical
· Cross Vertical leg of [L] with a Horizontal line
· Flipping [7] Shapes



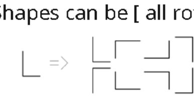
· Extend the Vertical leg of [L] Shapes
· Offset the Vertical leg of [L] Shapes to the internal end point of the [L]. Draw with a lighter line-weight



· 2-D shapes separated into three layers based on size (Small, Medium, Large)
· Resolve material conflicts by layering / stacking
· Make tectonic connections between shapes
· Cross link layers
(rotation is a possible operation for depth)

On Shapes:

· Define shapes on visual terms:
[L-shapes, zig-zags, 7-shapes, U-shapes]
· Define shapes on procedural terms:
[draw line 3 space right, 7 space up or down]
[offset lines 3 spaces, connect 1 set of end pts]
· Shapes can be [all rotations] & [all mirrorings]



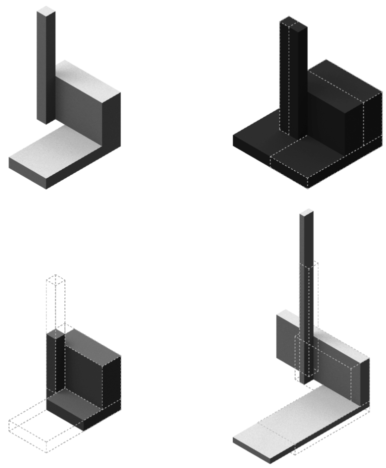
· Shapes Overlap at multiple scales
· Shapes are modified by the environment (grid)

Adding Lines · Students are learning:

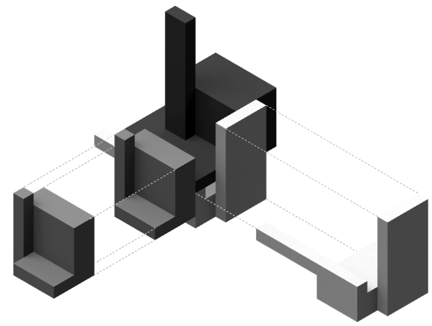
· to separate visual information onto layers
· to identify and create visual relationships
· to create part to whole relationships through the repetition of parts (elements)
· to identify characteristics of shapes (elements)
· to create variation by modifying a shape's (element's) characteristics (point & line)
· to draw with hierarchy through line-weight
· to develop rule based systems based on shapes (elements) made of points and lines within and environment (the grid)

4

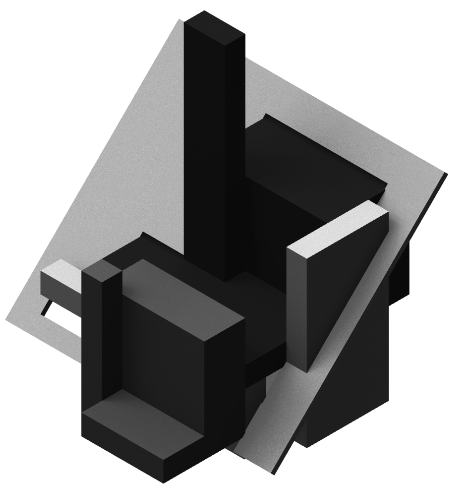
Figure 1: ARCH 100 Pedagogy Manual—Additive Shapes Exercises (Robert Lee Brackett III)



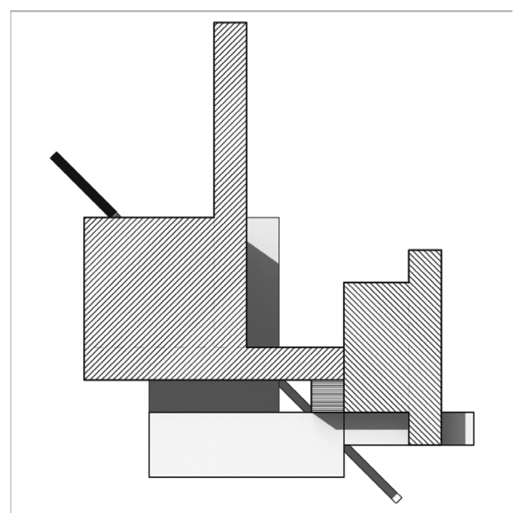
Base component of modular MDF extended into three variations by stretching and sliding



Exploring color and orientation for tectonic fit assembly



Aggregation interaction with plane proto-site. Object outline projected through plane from top view to create wrapped relationship



Section (including material difference and shadow projections optional)

Figure 2: ARCH 100 Pedagogy Manual—Additive Volumes Exercise (Robert Lee Brackett III)

Subdivision systems operate differently from additive systems, with shapes forming from the relationships between closed regions subdivided into recursive parts. Again, layers of vellum are used to allow for rapid iteration and design development while seeing and doing. This is difficult to learn on a computer, where the shape elements lose their flexible visual relationships and transformational freedom. Computers turn the shapes into symbols, which impede direct visual manipulations as a way of thinking, but with practice, this can be learned. Shapes offer a visual interface to computing, since they can be translated into symbols for a computer. Stiny explores thinking and doing with the hands and eyes to learn creativity and develop a visual interface to the initial abstract schemas that define computational processes (2006, 6–8). We remain analog for these exercises because we are not teaching students how to write computer code, but how to think about design computationally. When we start to introduce more material tectonics, the boundary between additive and subdivision systems is challenged by the material translation. The artifacts of the process can become quite visually rich and rigorously controlled while still producing a high degree of variability and individual design research. These exercises derive from Stiny's research into Chinese lattice designs, turning an analytical process into a synthetic process.

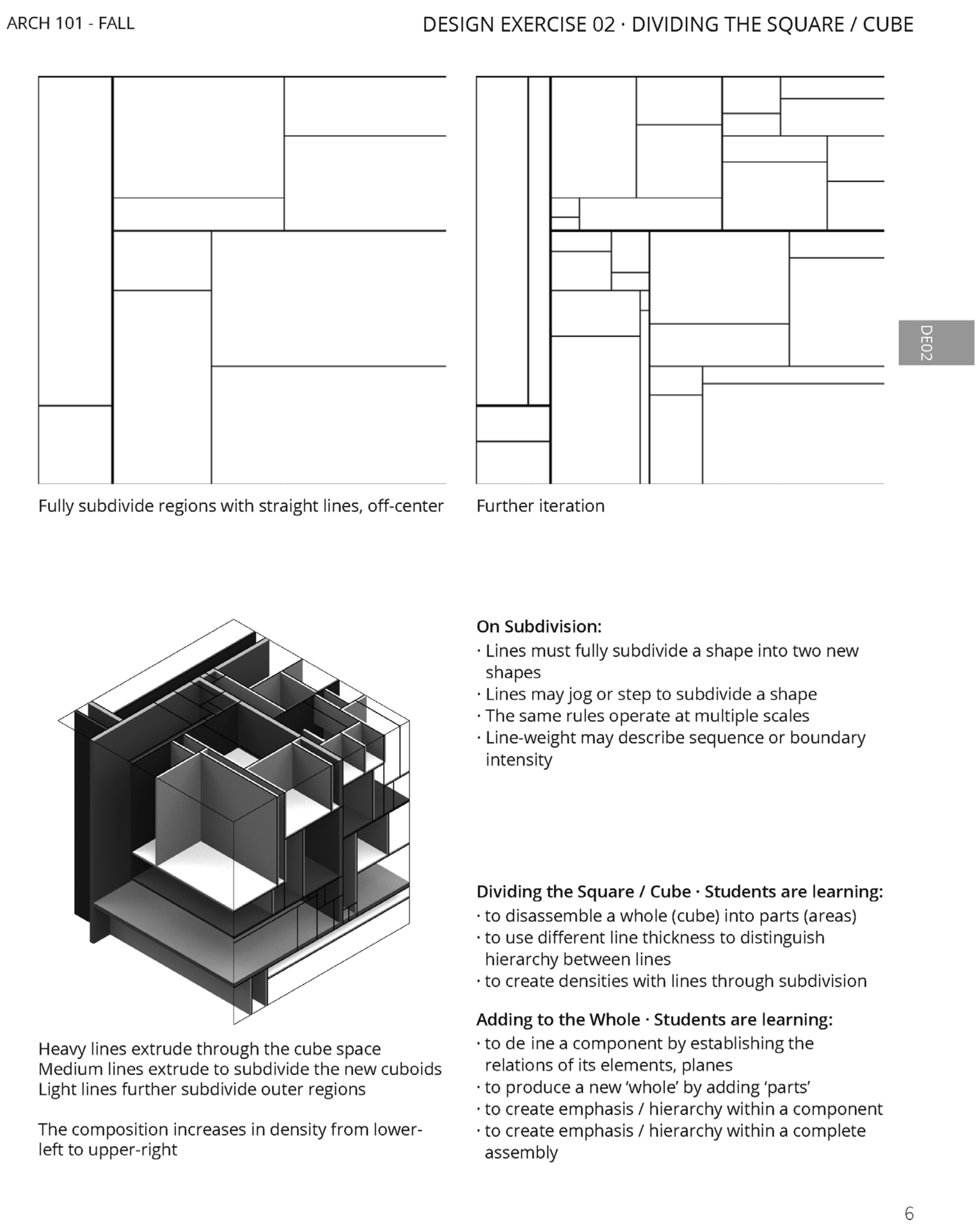


Figure 3: ARCH 100 Pedagogy Manual—Subdivision Exercises (Robert Lee Brackett III)

Each subdivision is made in the same way: attach an appropriately sized stick between two edges of a previously constructed triangle or quadrilateral or pentagon so that it does not cross previously inserted pieces. Each stage of the construction is stable; each stage follows the same rules. Indeed, the steps in the ice-ray lattice generation given in figure 5 could well comprise the frames in a motion picture of the artisan creating his design! (Stiny 1977, 97)

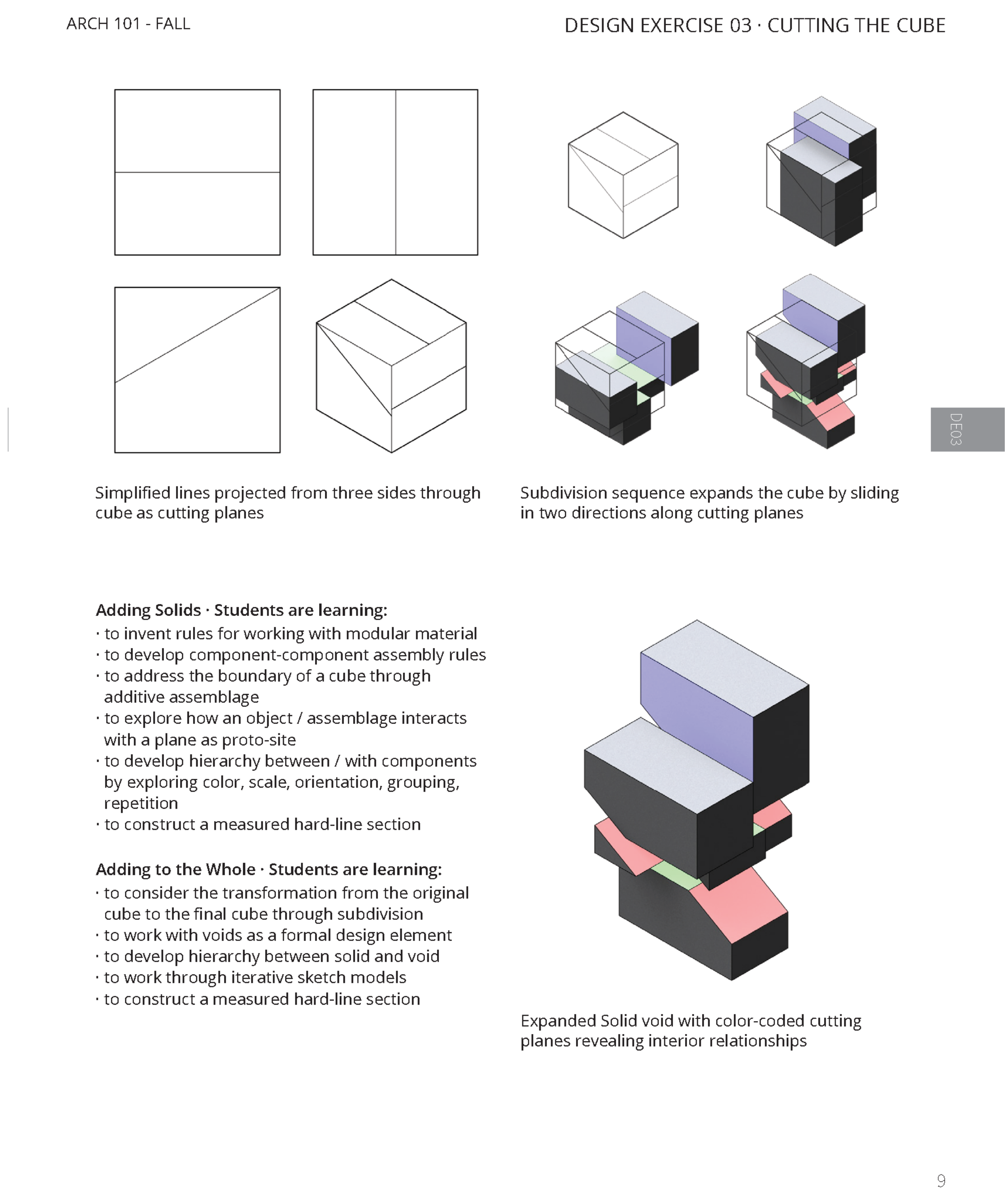


Figure 4: ARCH 100 Pedagogy Manual—Dividing a Cube Exercise (Robert Lee Brackett III)

5.2 Design as Visual Thinking—Hybrid Design Approaches

The Design Exercises continue to introduce new fundamental principles of geometry, tectonics, organization, and ways of designing in 2D and 3D based on elements and rules. After the first eight weeks, a student should be able to define two Design Approaches as visual systems that operate via schemas of formal relationships generated by shapes and rules of transformation in 2D and 3D. The remainder of the semester is spent discovering and defining the interface between these two design approaches and testing the results as a proto-architecture of forms, space, and material. This pedagogical model further builds on Stiny's investigation of thinking with shapes as a creative medium based on visual computation.

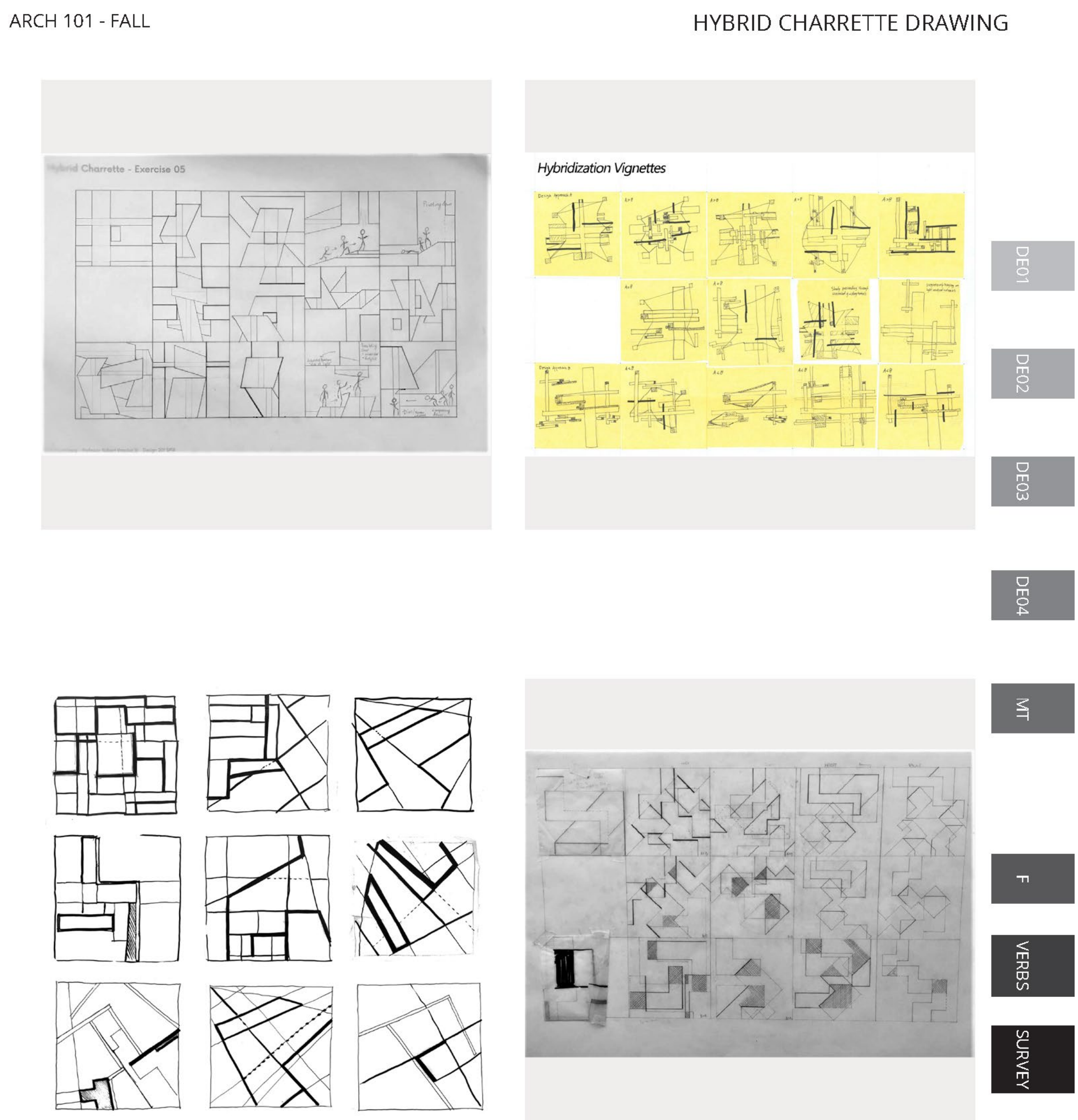


Figure 5: ARCH100 Pedagogy Manual—Hybrid Visual Systems (Robert Lee Brackett III)

> There's creativity in combining shapes and in dividing them. But the one without the other is just reciting by rote, merely counting out. It's all memory when shapes are divided in advance, but otherwise, everything is always new. No one took any notice of this. Maybe the difference between sets and shapes in calculating—between identity and embedding—is too subtle. Or perhaps rigor and formality don't work. I'm less technical now, and as informal as I can be. The message is the same, and I don't want it to be missed. It's all about seeing—there are no units; shapes fuse and divide when I calculate. (Stiny 2006, 53)

This hybridization process is open and nonlinear. Because a visual system is interactive and defined with shapes and rules, the students explore multiple simultaneous methods for interfacing the design approaches in models and drawings. A drawing-based design charrette that rapidly forces the two systems to intersect, swap shapes and rules, and resolve complexity by searching for spatial relationships is an effective system for generating hybrids. This is performed in parallel with the creation of small sketch models that bring a tangible spatial dimension to the 2D drawing systems. Models are simultaneous translations from 2D to 3D with unique formal-material speculations that can inform 2D systems. Neither the drawings nor the models should be representational analogs to each other, but instead, discrete visual artifacts of a system at play. A challenge we have faced when onboarding new faculty is that they tend to treat the hybrid process as linear and continuous, which maintains high visual integrity but loses opportunities for invention. Here we can draw on the conclusions of Stiny as he states, "Shape grammars treat [shapes] as spatial objects; they require no special parsing of [shapes] into fixed [parts]. Spatial ambiguities are thus allowed, as given compositional units in [shapes] can be recombined and decomposed in different ways" (2006, 53).

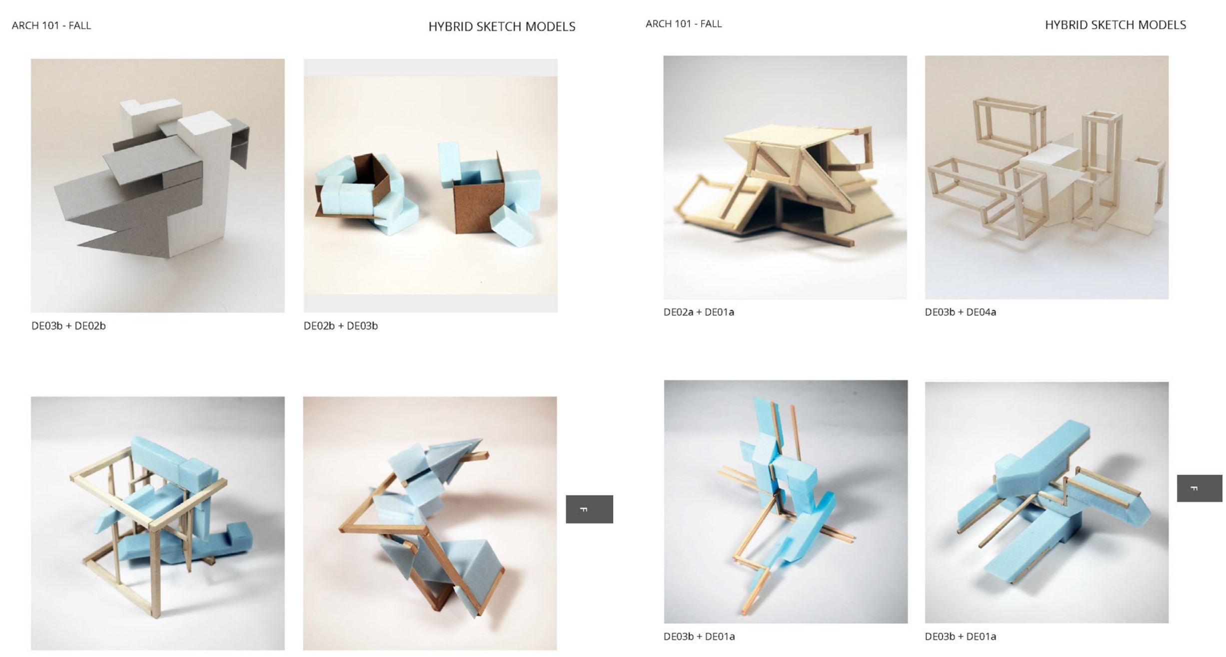


Figure 6: ARCH 100 Pedagogy Manual—Hybrid Models (Robert Lee Brackett III)

While the process may be open and full of ambiguity, the interface of elements and rules provides a rigor that allows students and faculty to access the design decisions within the project and incorporate feedback iteratively. Since the medium and material palette are limited, the results can be discussed in an objective framework while still offering a wide range of formal and aesthetic invention. This helps students play an active role in the design feedback process and better interpret the discussion of their work and the work of their peers. The proto-architectures remain abstract and malleable, yet highly controlled and specific. The lessons prepare students to integrate more complex architectural considerations within a computationally driven process as interrelated systems of form, space, structure, and site.

6. CONCLUSIONS AND FUTURE WORK

The pedagogical position presented here integrates key concepts from math and science, specifically concerning computation, at a first-year undergraduate level. Working from "shape grammars" composed by schemas of discrete shapes and rules, this pedagogical approach seeks to integrate computational thinking and formal systems of design discovery into the learning processes. In this approach, students learn to operate in a structured visual context, working toward rigorous systems through a responsive feedback loop of abstract drawings and models with their spatial consequences. Students are engaged in a process of feedback and reflection that treats design decisions as data points available for critical integration. This prepares students to engage more complex computational concepts during their studies using an analytical, process-driven approach. At the same time, students must understand how even objective representational systems are embedded with implicit social and cultural values that have real-world outcomes and that constitute a politics of space.

A new way of thinking about design as a scientific process that builds visual aesthetics on top of a formal structure of mathematical relationships emerged from the work of Christopher Alexander during the 1960s. The shift from analytical abstraction to abstract process opens new ways of thinking about design that integrate computational thinking with visual thinking. Methods such as George Stiny's "shape grammars" offer a rigorous and accessible way to introduce undergraduate architecture students to core concepts of mathematics that support a field rapidly shifting from analog invention to a digital space of computation-based exploration. Teaching design as visual systems developed iteratively through the interaction of elements and rules shifts the evaluation of design work from a focus on preexisting talent to a system of rigorous and repeatable techniques layered with critical thinking. Design can be taught as a learnable set of skills rather than as something a person is naturally good or bad at based on previous experience. This helps individual students find their own design approaches that produce consistent and diverse outcomes leveraging the shared mediums of visual thinking and abstract systems. This approach to design thinking could be introduced earlier in the education process to help develop methods of learning and problem-based thinking that integrate math and science into design education.

7. ACKNOWLEDGMENTS

I would like to thank and acknowledge Duks Koschitz, PhD; Dipl. Ing.; Associate Professor in Design and Technology; Director, Center for Design Research in Architecture at Pratt Institute, for introducing shape grammars, visual thinking, and analog computational rule-based systems to our first-year pedagogy. I would also like to thank Sacha Frey, Adjunct Assistant Professor of Humanities and Media Studies at Pratt, for being a teaching partner in the development of this pedagogy in her role as faculty and coordinator of the Architecture Writing program at Pratt. She has also been generous in her editing and feedback for this paper.

8. REFERENCES

Alexander, Christopher. 1964. *Notes on the Synthesis of Form.* Cambridge, MA: Harvard University Press.

Alexander, Christopher, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. 1977. *A Pattern Language: Towns, Buildings, Construction.* New York: Oxford University Press.

Le Corbusier. 1985. *Towards a New Architecture.* Translated by Frederick Etchells. New York: Dover.

Palladio, Andrea. 2002. *The Four Books on Architecture.* Translated by Robert Tavernor and Richard Schofield. Cambridge, MA: MIT Press.

Stiny, George. 1977. “Ice-Ray: A Note on the Generation of Chinese Lattice Designs.” *Environment and Planning B: Planning and Design* 4, no. 1: 89–98.

———. 2006. *Shape: Talking about Seeing and Doing.* Cambridge, MA: MIT Press.

Tegmark, Max. 2017. *Life 3.0: Being Human in the Age of Artificial Intelligence.* New York: Vintage Books.

The Open Group. n.d. *Architecture Patterns.* Accessed March 12, 2020. https://pubs.opengroup.org/architecture/togaf8-doc/arch/chap28.html.

Vitruvius. 1960. *Vitruvius: The Ten Books on Architecture (Bks. I–X).* Translated by Morris Hicky Morgan. New York: Dover.

[1] Pratt Institute B. Arch, https://www.pratt.edu/academics/architecture/ug-dept-architecture/arch-b-arch/.
[2] Massachusetts College of Art and Design (MCAD) B.Arch, http://academic-catalog.massart.edu/content.php?filter%5B27%5D=EDAD&filter%5B29%5D=&filter%5Bcourse_type%5D=-1&filter%5Bkeyword%5D=&filter%5B32%5D=1&filter%5Bcpage%5D=1&cur_cat_oid=9&expand=&navoid=241&search_database=Filter&filter%5Bexact_match%5D=1#acalog_template_course_filter.
[3] California College of Art B.Arch, https://www.cca.edu/architecture/barch/#section-curriculum.
[4] Parsons BFA, https://www.newschool.edu/parsons/bfa-architecture-design/?show=program-curriculum.
[5] Rhode Island School of Design B.Arch, https://www.risd.edu/academics/architecture/undergraduate/.
[6] The Cooper Union B.Arch, http://cooper.edu/architecture/curriculum/bachelor.
[7] https://shape.design.gatech.edu/Research/Projects/2018_Palladio/index.html.
[8] This can be seen in Peter Eisenman’s work in his exhibition and book *Palladio Virtuel*.
[9] Alexander references D’Arcy Wentworth Thompson, *On Growth and Form*, 2nd ed. (Cambridge: Cambridge University Press, 1959), 16.
[10] The ARCH 100 Pedagogy Manual was developed by Robert Lee Brackett III, Duks Koschitz, Tiffany Pun, and Mandy Xe in 2017–2018 as part of Pratt Institute’s design research lab, Center for Design Research in Architecture (d.r.a.).