ABSTRACT
Emerging technologies of design and production have largely changed the role of drawings within the contemporary design process from that of *design generators* to *design products*. As architectural design has shifted from an analog drawing-based paradigm to that of a computational model-based paradigm, the agency of the drawing as a critical and important form of design representation has greatly diminished. As our design tools have increasingly become computational and the production of our drawings have become predominantly automated, this paper examines the effects on the architectural discipline and attempts to catalog examples of how artists, designers, architects, and programmers have used rule-based techniques in the process of drawing as a critical act in their process. Furthermore, the paper presents the Drawing Codes project, an ongoing research and exhibition platform that critically investigates the intersection of code and drawing: how rules and constraints inform the ways architects document, analyze, represent, and design the built environment. The project features commissioned drawings by a range of contemporary architects and designers as a means of gathering a diverse set of perspectives on how computational techniques, but more importantly, *computational thinking*, can reexamine the role of architectural drawing as a creative and critical act.
INTRODUCTION

A history of architecture that dealt with the impact of drawing would need to explain two things: how architectural spaces arose out of the deployment of depthless designs, and how architectural space was drawn into depthless designs.

— Robin Evans, The Projective Cast

The relationship between drawing and architecture is foundational yet paradoxical. As Robin Evans suggests, architecture could be defined by the struggle between the inherently two-dimensional plane of the drawing and the three-dimensional reality of space. Architects must fold the complexities of construction, materiality, and perspectival view into flat drawings while at the same time unfolding the abstract rationality of the drawing into much more complex space. For hundreds of years, architects primarily conducted their work in the parallel space of projective drawings: plan, section, and axonometric. With the advent of computer-aided design (CAD) software in the last few decades of the 20th century, the fundamental act of drawing was relatively unchanged despite the new and different toolset. Only as digital modeling and, more recently, building information modeling (BIM) have become the dominant methods used to design architecture has architects’ relationship with drawing radically changed. After centuries of drawings being the essential mechanism to generate and critically engage architecture, we are at a moment where drawings might exist as only as an output of production. Within a model-based design paradigm, the construction of drawings has either been bypassed altogether or been outsourced to algorithms. Drawings could be viewed as an anachronistic byproduct of the legal requirements of building permits.

In this void, what is the agency of drawing within the discipline? Should architectural drawing be forgotten and left in the dustbin of history? Freed from the complex historical interrelationship between drawing and building described by Evans, should we simply create buildings without drawings as researchers such as Michael Silver have suggested (Silver 2006)? Or, might the very technologies that have enabled this freedom be re-examined for their potential to critically engage drawing as a generative and creative act?

Our research explores how emerging technologies of design and production have opened new ways to engage with traditional practices of architectural drawing. Instead of using computational techniques to automate the translation of models into drawings, we are interested in computation’s agency to generate drawings without models. Our research investigates how rule-based processes might yield new ways of thinking about both architectural design and representation.

This paper presents the Drawing Codes project, an ongoing research and exhibition platform that critically investigates the impact of computational design and production on architectural representation. The research has been pursued through the authors’ teaching, professional, and curatorial practices; however, this paper focuses initially on the larger conceptual and historical context of computational drawing and then specifically on the curatorial results of the inaugural Drawing Codes exhibition. The Drawing Codes exhibition features commissioned drawings by a range of contemporary architects and designers whose work explores the intersection of code and drawing: how rules and constraints inform the ways architects document, analyze, represent, and design the built environment. In doing so, these works begin to suggest renewed forms of agency and possibility for the architectural drawing in the computational era.

BACKGROUND

The Drawing Codes project was instigated not only by an interest in the agency of computational drawing, but also by a larger frustration with prevailing paradigms of computational design in architecture. In contemporary practice, computational approaches to design are often now associated with stylistic tropes of continuous differentiation, panelized surfaces, twisted towers, and the like. This tendency is perhaps best formalized in Patrik Schumacher’s notion of “Parametricism,” in which he calls for “a maximal emphasis on conspicuous differentiation” (Schumacher 2008). In many regards, this alignment between technique and stylistic outcome that Schumacher called for has now been firmly cemented; this is immediately evident in a simple Google Images search for “parametric design” or “digital fabrication,” which reveals how the initial novelty of computational form-making has
become mainstream within the last decade (Figure 2). Although Schumacher acknowledges that computational tools can be used to create non-Parametricist designs, he discounts this as a modernist resistance to complexity: “This is evidenced by the fact that late modernist architects are employing parametric tools in ways which result in the maintenance of a modernist aesthetics, i.e. using parametric modelling to inconspicuously absorb complexity” (Schumacher 2008). However, this presents a false dichotomy as one must seemingly choose to adopt Parametricism as style or remain modernists misusing computational tools.

This conundrum also resonates in the academy, as schools of architecture struggle to teach computational tools in a critical manner. The stylistic trap outlined above drives a vicious circle: students too often think that the only application for these tools is to create twisted towers or complex panelized surfaces, and that computational thinking has no other purpose within the architectural design process. This represents a failure of pedagogy, as students (and often faculty) equate a set of tools with a type of architecture. This project seeks to challenge the easy association of specific toolsets and processes with stylistic tropes as unproductive in that it forecloses broader conversations, such as how computational workflows relate to historical precedent, or how they can have transformative impact on architecture beyond a certain visual style. It asks questions like: How can we move beyond the cliché without rejecting the value of the tools? How can we move beyond the limiting construct of Parametricism to explore the remarkable range of contemporary architects engaging in diverse and innovative approaches to computation in their work?

PRECEDENTS
The initial step in this project has been to construct a computational design pedagogy that introduces computational thinking and techniques outside of the clichés of Parametricism. Specifically, we have focused on a range of case studies within the history of art and design that apply rule-based approaches to drawing. These case studies can be categorized into the following groups.

Proto-Computational Artists
There is a long history of code-based artistic drawing practices that span across the disciplines of art, architecture, and computation. In particular, several specific moments provide important precedent for our work and teaching. The first is a group of “proto-computational” artists in the mid-twentieth century who produced work that explored ideas important to computation, without actually using nascent electronic computational tools. This includes the work of Ellsworth Kelly, which incorporates qualities and techniques like randomness, probability, and pixelation.
For example, in *Cite* (1951), Kelly used random chance to reorganize cut-up segments of a painting into a new arrangement. Sol Lewitt’s work shares many of the same aesthetic qualities, but it also embraces a more process-driven approach to code. Many of Lewitt’s projects employ permutations, variable, rules, and code to constitute the generative act for a drawing, model, or building being just one artifact of that code. As Lewitt stated in two of his “Sentences on Conceptual Art”: “Irrational thoughts should be followed absolutely and logically;” and “For each work of art that becomes physical there are many variations that do not” (1969).

### Conceptual Programmers

Concurrent with this group of artists beginning to engage with computational processes was a set of scientists and programmers working with early computers who began to experiment with methods of producing drawings with digital pen plotters. These include artists such as Vera Molnar and Manfred Mohr, whose work is seminal in forging connections between the history of art and the emerging discipline of computer science. In describing her approach to using early computers in the 1960s and 70s to generate new variations on a theme, Molnar states, “This approach to the generation of pictures is not new; it had been applied long before computers were constructed. Making a series of pictures that were alike except for the variation of one parameter is not uncommon in the history of art... My computer-aided procedure is only a systemization of the traditional-classical approach” (Leavitt 1976).

### Generative Artists

Building upon the legacy of both the proto-computational artists and conceptual programmers is the work of a more recent generation of artists who have developed new languages such as Processing and constructed new machines to produce drawings. Often spanning the worlds of art, design, hacker culture, and DIY maker culture, this work represents a more synthetic hybrid of the artists who engage in code and programmers who engage in art. Artists such as Casey Reas built on the ethos of conceptual artists like Sol Lewitt by positing that software is art, and that the drawings of his work are artifacts that provide him “the opportunity to examine the state of the process at a precise time with extremely high fidelity” (Reas 2006). In works such as *Process 7*, the drawing is not a static thing, but rather a set of instructions through which an infinite number of artifacts could be extracted:

A rectangular surface filled with varying sizes of Element 1. Draw a line from the centers of elements that are touching. Set the value of the shortest possible line to black and the longest to white, with varying greys between. Draw the perimeter of each element as a white line and the center as a black dot. (Reas 2006)

### Coding Architects

Within the discipline of architecture, there is also a long history of engaging in rule-based, code-driven logics of form or spatial organization. One can look back to Alberti’s famous mid-fifteenth century mapping of Rome, the *Descriptio Urbis Romae*, in which he documented the city using a radial measuring device and codified these measurements into a table of numbers, as perhaps one of the first truly digital recordings of spatial information (Alberti 2007). As Mario Carpo states:

> Alberti tried to counter the failings of analog images by digitizing them, in the etymological sense: replacing pictures with a list of numbers and a set of computation instructions, or algorithms, designed to convert a visual image into a digital file and then recreate a copy of the original picture when needed. (Carpo 2011)

The relationship between code and drawing persists in contemporary conversations about architecture and representation. The early work of architects like Peter Eisenman and Daniel Libeskind, which consisted primarily of unbuilt projects pursued through analog drawing processes, is highly computational in its use of codes, procedures, and disciplinary conventions of representation like projection to produce innovative and compelling spatial conditions. These works show clear influences from artists such as Sol Lewitt in both their scripted process of construction as well as their rigorous application of Lewitt’s dictum that “irrational thoughts should be followed absolutely and logically.” In Eisenman’s “Notes on Conceptual Architecture,” itself a play on Lewitt’s “Sentences on Conceptual Art,” he probes the relationship between process (or the codes that delineate a concept) and the status of the object (Eisenman...
Kelly Bair discusses how this work, although rigorously and intensely pursued, maintains a sense of surprise and innovative discovery:

The drawing becomes a catapult for invention, leaving behind the expectations of convention, buildability, or financial burdens. The drawing is... a proto-document of something that is not quite finished—an act still underway in the eye of the author. (Bair 2012)

Although pre-digital, this work was enormously influential in conditioning today’s discourse on computation and its role in the design process. Recent years have seen a renewed interest in disciplinary questions of representation—and drawing in particular—perhaps in response to, or reaction against, the homogeneity produced by architecture’s parametric turn: both the clichés of Parametricism and the banality of drawings automated through BIM. While there has been an abundance of exhibitions, publications, and events celebrating and exploring representation in architecture, there has not been a focused and critical examination of the overlaps between computation and architectural drawing. The Drawing Codes project began with this premise, and the idea that focusing exclusively on representation would be a productive way to explore architecture’s relationship to code in a very direct way that avoided the pitfalls of stylistic tropes. The focus on drawing raises questions not only of how code can be used to generate architecture, but also to analyze and represent it.

**CURATORIAL PREMISE**

**Prompts**

The Drawing Codes project draws inspiration from these multiple legacies—proto-digital art, conceptual programming, generative art, and contemporary architecture’s grounding in the relationship between code and representation. To explore this territory, twenty-four contemporary designers and practitioners were invited and commissioned to produce a new drawing that responded to a series of prompts on the definitions of code within architecture. The curators explicitly decided to expand the definitions beyond code as solely a computational act to capture a more diverse understanding of how codes are perceived and deployed by architects. These prompts included:

- **Code as generative constraint.** Restrictive codes often govern what is permitted and what is prohibited. Examples of this include building codes, urban codes, zoning codes, accessibility codes, and energy codes. How can such constraints become generative, creating new opportunities for design and representation?

- **Code as language.** A code can be understood as a set of rules, conventions, and traditions of syntax and grammar that structure the communication of information. The discipline of architecture similarly has its own language of typologies, taxonomies, and classifications. How can drawing engage with such architectural languages?

- **Code as cipher.** Encoded or encrypted messages are intended to hide or conceal information. Likewise,
architectural geometries, forms, spaces, and assem-
bles are embedded with invisible organizational, social, political, or economic logics that may not be immediately evident. How can drawing engage with these latent meanings and messages?

• **Code as script.** A code can be understood as a script or a recipe: a set of instructions to be executed or performed by a computer, a robot, or (in the case of theater or film), an actor. Scripts often produce unexpected discrepancies between the intent of the code and how it is executed. How can drawing explore these open-ended processes that may not have a defined outcome?

**Rules**

In addition to the thematic prompts, the curators established a set of ground rules, or constraints, for the contributors. The intent was to provide consistency from one drawing to the next, to foreground the different responses to the prompts. The rules included:

• The drawing can be produced in any way: hand drawn, plotted, printed, produced by a robot or other machine.
• The drawing must be black and white.
• The drawing must be orthographic in projection. This includes planar, sectional, elevational, axonometric types of drawing; no perspectives.
• The drawing medium should be sized 25” x 25” so it can fit in a 27” x 27” frame.

**SELECTED WORKS**

Although initiated by a prescriptive and narrow brief, the range and diversity of the commissioned drawings is quite broad. Within the body of work, several tendencies or inclinations emerged, each of which suggests a unique interpretation of architecture’s relationship to code.

**Building Codes**

Several drawings in the exhibition engage issues related to building codes: the real codes of planning offices, preservation groups, permitting bodies, and regulatory agencies. Two of the drawings that best speak to these codes are those by Clark Thenhaus of Endemic Architecture and Dwayne Oyler and Jenny Wu of Oyler/Wu Collaborative. Thenhaus’s drawing, appropriately named *Then House No. 2* (Figure 3), constructs a sort of mutant Victorian house that intentionally misinterprets specific codes within the San Francisco Planning Department rules governing conventional elements of Victorian houses such as “bay windows, porticos, dormers, gables, fenestration, stairs, and turrets” while also adding in completely unconventional but permissible elements such as fat rolls, rubble stairs, and shaggy facades (Thenhaus 2017). In addition, the drawing is a mutant hybrid of different projections including perspective, axonometric, and elevation oblique. As Thenhaus states, “If ‘Drawing Codes’ were afforded some terminological freedom, we might re-consider how the literal planning and preservation codes of a given municipality are implemented through architectural drawings, or how drawing historically salient features or...
building elements intentionally ‘wrong’ might still maintain specific aesthetic codes, or how manipulating conventional drawing techniques can frustrate the legibility of an architectural form’ (Thenhaus 2017).

Oyler/Wu Collaborative’s drawing (Figure 4) also confronts building codes in their design for 3DS Culinary in Los Angeles. The project is a renovation within a 1924 bank building. However, a city code implemented well after the originally building was built requires more parking spaces than would be possible if a new mezzanine was created. In their words:

It appeared in the early stages of the design process that no additional square footage may be added, effectively preventing the proposed mezzanine space from being built. Further study of the Los Angeles Zoning CODE uncovered [Sec. 12.21 A 4(k)]: Fractional Space. When the application of these regulations results in the requirement of a fractional automobile parking space, any fraction up to and including one-half may be disregarded and any fraction over one-half shall be construed as requiring one automobile parking space. Based on this section of the Los Angeles Zoning CODE, the mezzanine was reduced very slightly to a floor area of 249 square feet and constructed in 2015. (Oyler/Wu Collaborative 2017)

Thus, through the clever application of parametric code, Oyler/Wu could slide their project into a restrictive building code.

Generative Catalogs of Difference
Several drawings in the show share an interest in organizing and visualizing a set of different experiences or configurations of geometry. David Gissen’s contribution, The Appearance of the Letters of the Hollywood Sign in the Smog (Figure 5), is an “experimental text [that] exists in a 5000-, 10,000- and 50,000-word versions, each of which simulate the progressively mutated appearance of the Hollywood Sign’s letters when seen through heavy smog and at various distances” (Gissen 2017). The drawing uses a code called the Pelli-Robson Contrast Sensitivity Chart used by ophthalmologists in determining eyesight in low contrast conditions when it is easy to mistake certain letters with others. Although the spelling of the word “Hollywood” is self-evident, Gissen’s drawing questions how our environment transforms the legibility of iconic architectural structures.

In Thom Faulders’s drawing, Blankspots: A Selection of Possible Adjacencies (Figure 6), he catalogs possible locations for artwork on the facade of the Wynwood Garage project in Miami. Located in Miami’s Wynwood Arts District, known for having one of the highest concentrations of street art in the world, Faulders’s facade integrates the idea that the building client will commission large murals across the facade over many years. These artworks may remain on the facade indefinitely or be sold as collector items. As such, the client and artists needed a catalog of possible configurations for potential new work. The drawing reveals different permutations and is organized

11 Erin Besler, 100%
12 Ron Rael and Virginia San Fratello, Hairline Drawing
from the smallest possible square footage to the most expansive.

Absurd Limits

Two drawings explore the potential of code-based drawing to produce unexpected, and potentially absurd, designs through an interpretation of Lewitt’s call for “irrational thoughts [to] be followed absolutely and logically” (Lewitt 1969). In Andrew Kovacs 20 Steps For Creating Beautiful Floor Plans Made Of Walls (Figure 7), the drawing is produced through a set of tutorial instructions in Photoshop. This drawing most clearly follows Lewitt’s approach to drawing in that not only is it based on a set of rules, but the results will differ based on who is performing the act of drawing. Each iteration of the drawing would be different based on the drawer’s subjective selection of “beautiful walls”. In many of Kovacs’s works, he uses an almost surrealistic collage technique to join found architectural elements such as entire plans or walls into new wholes. In an earlier work called Plan for a Porch 004, Homage to Durand, Kovacs reconfigures Durand’s catalog of porticos into a new labyrinthine porch. Kovacs’s projects construct new works from the sampling of pre-existing projects in a way that decontextualizes the source material and envisions a new life for it.

In The Walled City (10-Mile Version) (Figure 8) by Andrew Kudless, the drawing is a means by which to explore the concept of a wall that somehow utterly confuses the notion of sidedness and instead, through inflation and folding, become a city. Rather than a line that divides space, the wall becomes the entire space of the city. Using a combination of parametric modeling, computational physics simulation, and Google Search suggestions, the wall is grown and then populated with a crowdsourced interpretation of popular rooftop programs. Kudless states:

I want to push the algorithm beyond the abstract and explore possibilities at scale, even (or especially) absurd possibilities. If computation tools had produced a trap of cliché, how might I use them to generate the unexpected? (Kudless 2017).

Data-Driven Practices

Another theme that emerged in the exhibition is an interest in data, and how it can be used to generate compelling spatial constructions. Environment as Politics (Figure 9) by The Open Workshop mines demographic data from the 2016 presidential election to understand the role that spatial organization of built form may have on one’s politics. The mapping and visualization exercise explores the correlation between population density and political affiliation—testing the hypothesis that the environments we inhabit play a primary role in shaping our politics (The Open Workshop 2017).

Joris Komen’s drawing Sample XC184534 (Figure 10) builds upon a body of work looking at humanity’s relationship to nature, specifically through human/non-human interactions. In this case, the drawing is an example of
what Komen calls “two-dimensional taxidermy” — whereby the drawing replaces the stuffed animal, but nonetheless pursues a desire to capture and preserve (Komen 2017). The drawing spatializes the call of a bird called the Hartlaub Spurfowl to produce a three-dimensional sonograph, and it’s an example of using computational techniques of analysis to discern spatial patterns that may be latent within sonic environments.

**Friction Between the Abstract and the Real**

Several drawings within the exhibition employed automated, robotic processes in the production and translation of an image. These works engage in a rich dialog between the abstract and the real, in which machine-driven processes, typically celebrated for precision and accuracy, are used here to introduce error and chance. Erin Besler’s drawing 100% (Figure 11) is one of a series of drawings that build upon the transformation diagrams of Peter Eisenman’s House VI. This iteration is the fastest drawing—with the robot operating at 100% speed—and it explores the contingency between kinetic motion and fidelity of an image.

Ron Rael and Virginia San Fratello’s contribution Hairline Drawing (Figure 12) uses 3d printed deposition of bioplastic to depict Le Corbusier’s chapel in Ronchamp—which itself was a kind of self-critique of Corbusier’s earlier work inspired by the machine aesthetic. Unlike Besler’s drawing that evoked the materiality of the machine, Rael and San Fratello’s drawing explores the materiality of the drawing’s media (bioplastic) and its visual affect hovering between an abstract pointillist or pixelated image and a hirsute body.

**Conventions of Architectural Representation**

Several drawings, perhaps looking to examples like Eisenman and Libeskind, embrace the generative capacities of representational codes. 100 Walls, Seed 96 (Figure 13) by Adam Marcus explores the overlap between procedural design processes and perceptual multiplicity. One hundred arced walls are arrayed according to an algorithm that allows for both continuity and discontinuity, producing a labyrinthine space within and between the curved walls. This space is conveyed via plan oblique projection and rendered only as lines. The projection and line work yields an ambiguity between architecture and picture plane, collapsing foreground and background at moments, and allowing for new, ghostly figures to emerge within the field (Marcus 2017).

The 11-House / 1.8% (Figure 14) by Andrew Heumann is a perverse but serious reflection on labor and computation, using the conventions of representation that are embedded within Revit, the software package that arguably has the most impact on the production of contemporary architecture. On the left side are 11 randomly seeded iterations of the “Revit Basic Sample Project,” the default template included in the program. On the right side, these iterations have been stitched together in what Heumann calls a “digital Micromega” (Heumann 2017). The intent is to contrast the easy, predictable labor-saving features of the software, with a richer notion of computational automation that creates, challenges, estranges, and makes ambiguous, resulting in a drawing that demands the human labor of interpretation and reflection.

**CONCLUSION**

Although the origins of the Drawing Codes project lie in a very focused critique of contemporary architecture’s engagement with computation, the diverse range of responses to the prompt provides a compelling cut across the current state of the discipline. Within this considerable diversity of medium, aesthetic sensibility, and content, several common qualities emerge. First is the unsure link between code and outcome: glitches, bugs, accidents, anomalies, but also loopholes, deviations, variances, and departures that open new potentials for architectural design and representation. Second is a mature embrace of technology not as a fetishized endgame, but as an instrument employed synthetically in concert with other architectural “tools of the trade.” And finally, these drawings demonstrate how conventions of architectural representation remain fertile territory for invention and speculation.

**ACKNOWLEDGMENTS**

Exhibition Assistant: Gina Bugiada

Thanks to the following people for making the Drawing Codes exhibition possible: Jaime Austin, Stephen Beal, Mark Donohue, Nataly Gattegno, Jason Kelly Johnson, Jonathan Massey, Karina O’Neill, Amanda Schwerin, Dustin Smith, Justin Smith, Ingailill Wahlroos-Ritter, Sandhya Kochar; Maryann Wilkinson, Sharon Haar

**REFERENCES**


Bair, Kelly. “(Mis)Behaviors of Drawing”. In *Digital Aptitudes: 100th ACSA Annual Meeting Proceedings*. 2012.


Kelly, Ellsworth, “Cite (1951)” . San Francisco Museum of Modern Art, the Doris and Donald Fisher Collection at the San Francisco Museum of Modern Art, and promised gift of Helen and Charles Schwab.


**IMAGE CREDITS**

Figures 3-14: © by the respective listed participants of the Drawings Codes exhibition.

All other drawings and images by the authors.

*Adam Marcus* is an Associate Professor of Architecture at California College of the Arts in San Francisco, where he teaches design studios in design computation and digital fabrication and co-directs CCA’s Architectural Ecologies Lab. He has previously taught at Columbia University, the University of Minnesota, and the Architectural Association’s Visiting School Los Angeles. He directs Variable Projects, an award-winning design and research studio in Oakland, California. Adam is a graduate of Brown University and Columbia University’s Graduate School of Architecture, Planning and Preservation, and he currently serves on the Board of Directors for the Association for Computer-Aided Design in Architecture (ACADIA).

*Andrew Kudless* is a designer based in San Francisco where he is an Associate Professor at the California College of the Arts. In 2004, he founded Matsys, a design studio exploring the emergent relationships between architecture, engineering, biology, and computation. He holds a Master of Arts in Emergent Technologies and Design from the Architectural Association and a Master of Architecture from Tulane University. The work of Matsys has been exhibited internationally and is in the permanent collections of the San Francisco Museum of Modern Art, the Centre Pompidou in Paris, and the FRAC Centre in Orleans, France.