Machines for Making Architecture

SARA CONSTANTINO
Princeton University

EMMETT ZEIFMAN
Columbia University

Computational automation, which is today everywhere, can be abstracted from its technological manifestations and considered as a conceptual structure comprised of three elements: division of labor, systematization of rules, notation of instructions. By applying this logic of automation to two seemingly disparate architectural practices, this paper argues that they are unified by a primary concern with working methods, part of a broader shift in the discipline towards a critical understanding of the computational tools used to make architecture today. In a context of pervasive automation, the projects discussed here suggest a turn away from the use of automated tools for instrumental ends and towards work on the material and conceptual structures of computational automation itself.

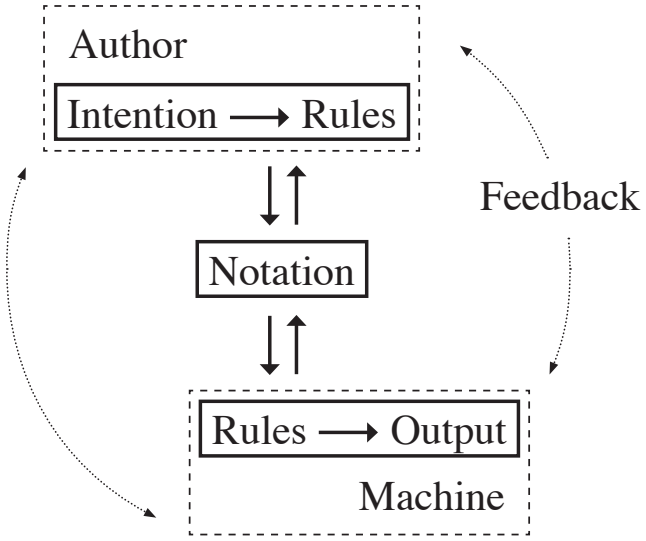


Figure 1: Schema of computational automation.

This paper was prompted by the more or less simultaneous emergence of two bodies of work, developed by young faculty at and around SCI-Arc, that appear almost antipodal in technique, form and affect. One trades in references to conceptual art and architectural history, quoting Marcel Duchamp and Eugène Viollet-le-Duc. At the junction discussed here, it involved simple, sequential geometric manipulations of found objects, represented through precise line drawings, monochrome white renderings or simple foam models, accompanied by deadpan narratives. The second body of work trades in computational power, data sets and intelligent agents. It is produced using scripts that are passed around SCI-Arc like alchemical secrets, generating complex, colorful aggregations of pixels and voxels, without any diagrammatic explication. In appearance, it tends more towards the Baroque than the post-minimal, and is generally displayed on wall-mounted monitors or printed in colored powder. One might make you feel bad for not having read Melville and the other for not knowing Python.[1] The first body of work was produced by First Office (Anna Neimark and Andrew Atwood) and the second by Kinch (M. Casey Rehm).[2]

The distinct aesthetics of this work reveal the common conditions of its production. Recalling the stylistic and discursive plurality of the Whites and Greys (and Silvers, etc.), which collectively signaled a shift out of modernism, these practices are representative of a broader movement away from the pursuit of new forms and programs and towards a critical understanding of the computational tools that are used to make architecture today.[3] What unifies the two practices, and many of their contemporaries, is a primary concern with working methods themselves.[4] Specifically, methods that, through a conceptual (and literal) engagement with computational automation, challenge conventions of architectural authorship and representation.

The projects discussed here can be related to three distinct, though increasingly intertwined, historical trajectories. The first is that of computation, which provides a continually evolving set of tools used in the design of architecture (as well as its construction and occupation). The second is that of the codification of forms of authorship and representation that structure the field of architecture itself. The third is that of modernist and postmodernist cultural production

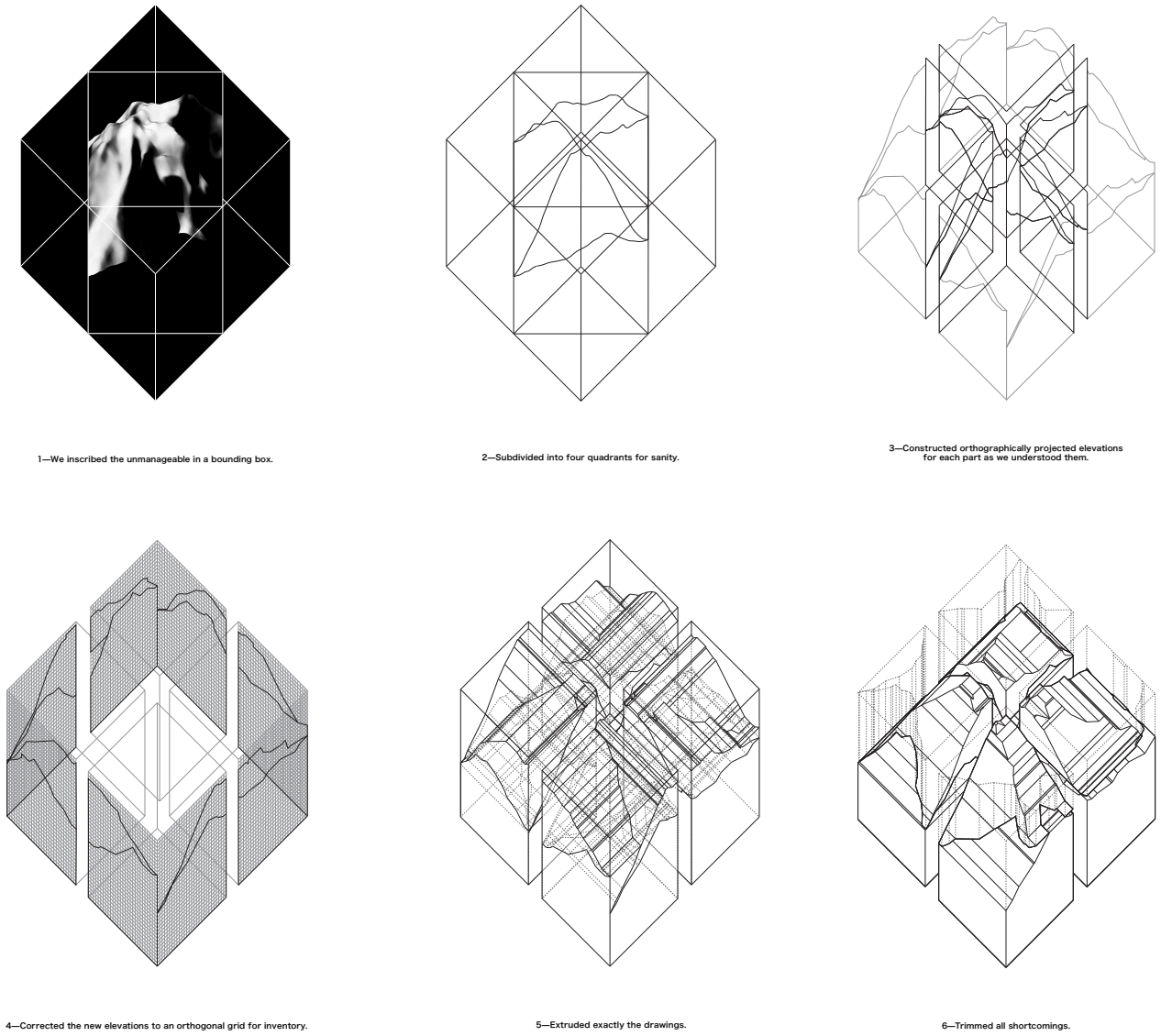


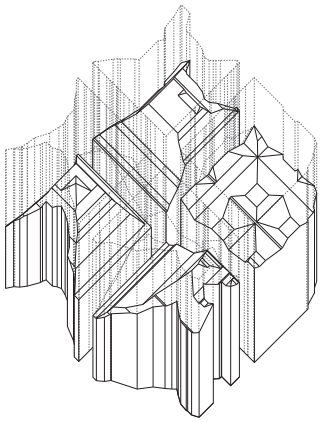
Figure 2: First Office, "How To Domesticate a Mountain," as published in *Perspecta 46: Error* (2013).

in which the "automatic" serves as a conceptual model of practice that challenges established institutions and forms. Abstracting automation from its specific technological manifestations in order to understand it as a more general conceptual framework allows parallels to be drawn between the two bodies of work discussed above, as well as with the varying contexts from which that work emerges.

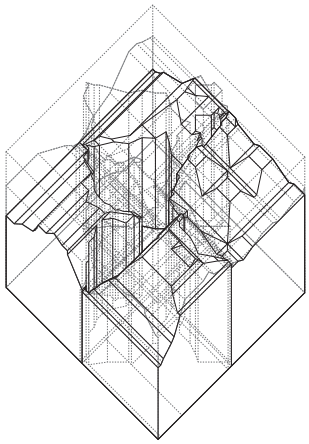
A SCHEMA OF COMPUTATIONAL AUTOMATION

Architectural practice, like life, is fully mediated by computation. A half century after the first experiments with computation in architecture, and nearly a quarter century since digital modelling softwares began to move into the mainstream of design practice, we might say—as has been claimed of capitalism[5]—that there is no outside of computation. It is no longer possible to conceive of a new work of architecture independently of the capabilities of computational tools.

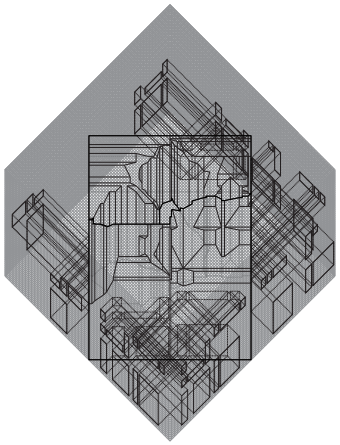
These tools transform labor through the automation of human tasks (manual and, increasingly, cognitive and affective) and the invention of new capacities (e.g. big data, machine learning). Earlier mechanical forms of automation are directly operated through physical engagement (pressing a button, pulling a lever, turning a dial), whereas computational automation cannot function without language to mediate between operator and machine, whether through the direct act of programming or through representational



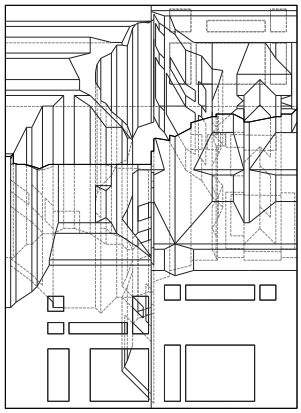
7—Projected the underbelly curves through a cube to remove all defects of character.



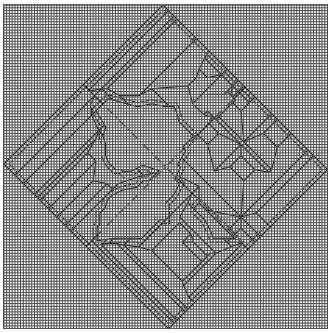
8—Rotated the willing quadrants 180 degrees.



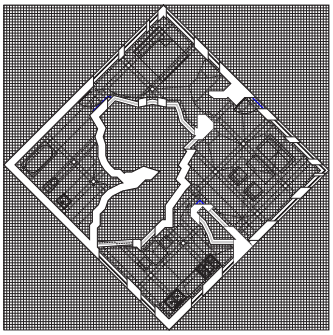
9—Projected apertures from the bounding diamond, and when we were wrong, promptly admitted it.



10—Called it a house only for the power to carry that out.



11—Turned the plan 45 degrees whenever possible.



12—Having had no spiritual awakening as the result of these steps, we nonetheless tried to carry this message to architects, and to practice these principles, as we furnished.

interfaces that in turn run programs. This intermediary step leaves material traces—software, graphic user interfaces, programming languages, data—that can be analyzed in their own right.

Outside of its specific technological manifestations, computational automation can be abstracted to a conceptual structure or schema. The use of game theory to solve strategic decision problems or the relationship between musical composition and performance are examples of practices that, analogically, can follow principles of automation.[6] This logic of automation comprises three basic features. First, labor is divided between author and executor—a system exterior to the author that is capable of following the author's instructions. For example, the division of labor between programmer (composer) and computer (performer). Second, the process to be executed is abstracted into rules. For example, the contents of a computer program (musical score). Third, the instructions for executing the rules are communicated by the author to the executor through a notational system. For example, programming languages (musical notation) have a specific grammar and conventions that are readable by both author (composer) and computer (performer). These three elements—division of labor, systematization of rules, notation of instructions—are the structural conditions that enable computational automation.

ARCHITECTURE AND CONSTRAINT

The clear relationship of this schema to the historical formation of the discipline of architecture allows for a broader view of the relationship between architecture and automation than that delimited by the direct use of computers or the term automation.[7] The disciplinary paradigm codified by Leon Battista Alberti in *On the Art of Building* is characterized by: a division of labor between the

architect, who defines the complete form of a project, and the builder, who exactly executes that design; the systematization of the design process through self-imposed disciplinary constraints (e.g. styles, proportional systems) and, as the profession developed, externally-imposed regulatory, material and market constraints; and the use of a notational system, orthographic drawing, which records and communicates design intent through conventions understood among parties in the design and construction process. While these disciplinary conditions have been continually challenged, including by computation,[8] the basic paradigm persists.

Rules-based structures in architecture prefigure, in some cases by centuries,[9] computational automation; in turn, computational automation has been readily adapted as an instrumental tool, with both geometrical and performance-based constraints formalized into design algorithms.[10] The disciplinary model defined by Alberti forms the basis for the literal automation of architectural design (as manifest in BIM software such as Revit, for instance[11]) in addition to defining those conditions of authorship, constraint and representation that today are investigated through the use of automation as a critical tool.

THE "AUTOMATIC"

Conceptual automation, manifest in varying takes on the "automatic," has been a recurring model of artistic practice throughout the twentieth century. Its appearance in art, music and other disciplines is coincident with a technological context of mechanical reproduction and, subsequently, computation.[12] There are varied manifestations of the impulse to defer authorship of the work through the "automatic": individually crafted objects replaced by mechanically reproducible ones (e.g. Marcel Duchamp's *Fountain*); the production of instructions to be executed by others, the notations of which supplant or stand next to the traditional art object (e.g. Sol LeWitt's *Wall Drawings*); indeterminacy pursued through strategies of arbitrary or random action, in which direct control over material is limited, often through the incorporation of external forces (gravity, entropy) into the procedure (e.g. Robert Smithson's *Asphalt Rundown*). To borrow LeWitt's phrase, with this type of work, "the idea becomes a machine that"—once a procedure is set in motion—"makes the art."[13]

Through each of these strategies, a distance is introduced between the artist and the production of the art object, as the labor of producing the object is taken out of the hands of the author. This distance shifts the framework of evaluation from one of individual talent and irreducible object to the system of production that relates author to object. In this relationship, the object now serves as an index of the process of its production. A primary function of such practices is disciplinary critique, whether through the framework of medium-specificity (in which the automatic actions reveal the structural conditions of the medium, i.e. what makes a painting a painting) or conceptual art (in which the automatic actions reveal the structural conditions of the field, i.e. what makes a work of art a work of art).

AUTOMATIC ARCHITECTURE

Architecture has participated in the emergence of these forms of critical practice through conceptual, and at times literal, techniques of automation, albeit within the context of its unique disciplinary structures. Of particular resonance are Peter Eisenman's houses (e.g. House II, 1969-70), which develop according to a logic of automation prior to the direct adoption of computational technology in the design process.[14] In Eisenman's work, the deferral of authorship comes not in the relationship of design to construction (a given condition of architecture), but in the design process itself, which is systematized through the procedural manipulation of abstract formal structures. The work appears to unfold as if automated, through rule-based transformations that produce a notational record of the project in the form of sequential axonometric diagrams, as well as indexical traces on the built form of the complete architectural object. This unfolding notational structure—a code—replaces the traditional orthographic set of plan, section and elevation as the basis by which the project is developed and evaluated, making the temporal structure of the design process itself explicit. The project can therefore be read as a critique of its own making—each step of which is represented and open to question.

Other projects extend this lineage. MOS's ongoing software experiments automate the aggregation of simple elements by subjecting them to the forces of "fake" physics environments within videogame engines. Like a Robert Smithson "pour," the entropic qualities of these aggregations makes clear the arbitrary (or indifferent[15]) origins of the final object. Automation precludes the deliberate arrangement of the aggregated elements by the architect, deferring authorship to an external computational system. The work appears in screenshots and animations,[16] media that privilege the representation of process over outcome and destabilize the result of the procedure, which is infinitely variable within the parameters of the software. This method of working represents a shift from the deliberate, linear sequence that unfolds in Eisenman's hand-drawn houses towards a "real-time," digital development of the project, which can be stopped, started and repeated with little effort.

TWO ARCHITECTURE MACHINES

Through a conceptual framework of automation, the work of First Office and Kinch can be related at a methodological level. Both practices produce scripts for the development of architecture that confine authorial intervention to specific moments in the design process. Rather than optimizing design performance with respect to external metrics of evaluation, automation serves for both practices as a means of revealing and questioning disciplinary conventions.

With the project *Mountain House* (2013), or "How to Domesticate a Mountain,"[17] First Office transforms a ready-made—a digital model of the surface of a mountain—into architecture by subjecting the object to the formalization of architectural representation. The complex surface of the mountain is flattened and re-projected until the geometry has been regulated in a manner analogous to the design of a building, the form of which must be sufficiently ruled that it can

Figure 3: Kinch (with P-A-T-T-E-R-N-S), details from *Oblicuo*, 2015.

be translated into the orthographic drawing format, communicated to a builder, and physically constructed. The project consists of an illustrated script—or set of instructions—elaborating sequential transformations that are carried out through common software functions that relate to conventions of orthographic projection. The design process itself is constrained to the sequential application of these commands.

The project repositions conventions of orthographic projection—marginalized as a design tool by the emergence of modelling software—as a means of deploying basic modelling tools in a structured manner in order to produce specific formal results. In this sense, the modelling is doubly automated. It represents a "default" in terms of the capacities of the software deployed, with little room for design decisions outside of the execution of simple Rhino commands. And these commands instantaneously complete acts of projection and intersection formerly only possible through the painstaking labor of descriptive geometry.

The conceptual automation of the design process establishes an apparently objective rigor, notated in ten sequential diagrams, which describe the development of the project, each step numbered as if

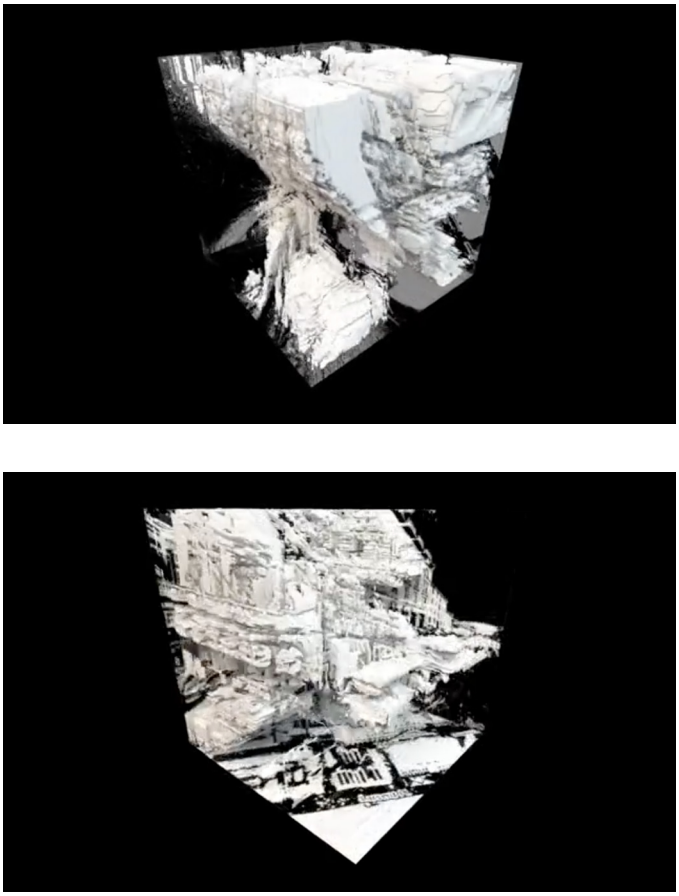


Figure 4: Kinch, screenshots of the *Street Vox* app, developed for SCI-Arc 2GAX design studio, taught with Ramiro Diaz-Granados, Marcelyn Gow and Florencia Pita, 2014.

it follows logically on the last. Each diagram is constructed in the same oblique view, constraining the infinitely variable camera positions and cutting planes that can be used to translate digital models into renderings and drawings. In sequence, the diagrams have the quality of animation frames, depicting a continuous process uninterrupted by authorial interventions. This determinism is undermined, however, by the captions that attend each step and make explicit the arbitrary nature of the process. At the conclusion of the oblique sequence, a plan of the to-this-point abstract model presents the "domestic" interior, complete with furniture. This plan reveals the ineffectiveness of the procedure in producing a house; the elaboration of the interior is not a part of the otherwise comprehensive set of instructions that constitute the representation of the project. The "house" is merely a suitable alibi for a didactic interrogation of architectural conventions, neatly encapsulated in a script that can be disseminated and repeated by others.

Kinch works with scripts as well—in the direct computational sense of lines of code—and these too represent design tools that can be disseminated to students. But where First Office produces a discursive pedagogical package—a lesson or instruction manual (or, a set of "reconstruction" documents, which allow for the design procedure to be replicated by others)—with Kinch the effect is inverted. Rather than sequential, annotated descriptions representing a conceptual approach to automation, single images or animations, with no exposition beyond the immediately perceptible forms they depict, present the effects of a literally automated computational system. To the viewer, this system is a "black box," functioning according to an inscrutable logic that cannot be fully communicated or replicated.

Oblicuo (2015) consists of a series of distorted images of six speculative projects for cultural institutions in the center of Budapest, designed by the firm P-A-T-T-E-R-N-S. Artificially intelligent agents progressively manipulate a photomontage of renderings of the projects, arrayed in an aerial drone photograph of the site, making individual decisions to alter pixels based on machine-vision criteria. Programmed to locally manipulate pixels based on relationships that are not legible to a human viewer, the work of the agents cannot be interpreted as a deliberate composition directed towards a comprehensible goal. It is not clear when the process of manipulation is finished or how such a determination would be made—the completion of the project can only be understood as an arbitrary intervention of the author into a process that might otherwise continue until the image is reduced to such entropic noise that no further transformation is possible.

In this sense, the project makes clear the deferral of authorship to the moment of inception (the programming of the procedure) and termination (the selection of the "complete" image), with all intermediate decisions regarding the manipulation of pixels carried out autonomously by the software. No additional material beyond the manipulated images is provided to the viewer; the project establishes no "objective" criteria for evaluating the transformation in terms other than its immediate effects. This injects a heightened level of subjectivity into the precisely scripted project, both in terms of the act of selection by the author and the act of interpretation by the viewer.

With *Oblicuo*, the agents work on images of architecture. This thematization of representation makes evident its central role in the context of computational technology, in which previously distinct hierarchies between representation and object have broken down. Work on architecture is now work on, or through, images,[18] which both represent models and exist in the same media as the models themselves—lines of code or binary contained in various file types and accessed through various softwares. To manipulate a pixel or vector in an image is little different, materially, than to manipulate a voxel or curve in a model; in any case, the computer screen is the material structure through which these objects are engaged. The projects discussed here are material investigations of the computational tools with which architecture is conceived and produced, which is to say that they are investigations of the representational media (programming languages, software interfaces) through which these tools are accessed.

The agents deployed in *Oblicuo* automate the material organization of digital files in ways that transform representations of buildings

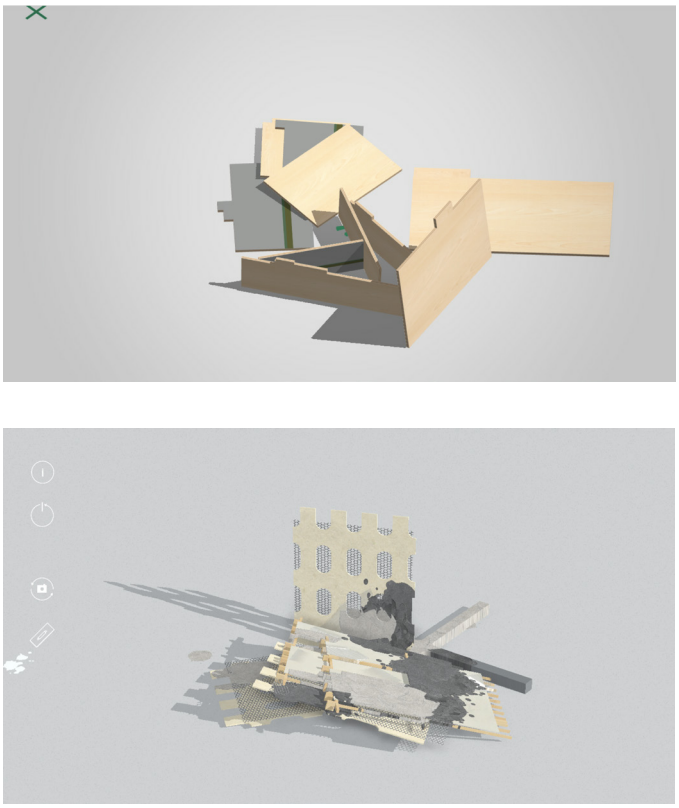


Figure 5: First Office (with Theo Triantafyllidis), screenshots of the *Blocks of blabla* (2016) and *City Render* (2017), apps.

(and sites) and, as such, suggest new approaches to manipulating the material organization of buildings themselves. The collapse of image and building is explored in other Kinch projects, including a piece of software written for mandatory use by students in a core studio at SCI-Arc in 2014. The script projects Google Street View images—arranged on three sides of a cube to correspond with the traditional orthographic composition of a building—through one another, producing a tool for instantaneously translating Google's vast library of "site context" data into an infinitely variable set of "contextual" architectural models.[19] The methodology parallels First Office's "Mountain": a "ready-made" digital object is selected, a script is applied to the object, profiles are orthographically projected and intersected to produce new objects until the author intervenes to terminate the procedure. That Kinch's project is scripted in a programming language and First Office's scripted in English prose is secondary to the fact that both pursue automated means of translating existing digital files into prospective building organizations through the manipulation of representational conventions.[20]

Recently, both practices have turned to explicitly interactive forms of design: First Office collaborating with Theo Triantafyllidis to produce apps that allow users to drop, stack and paint architectural elements, producing their own tentative architectural configurations, subject to the gravity of the simulated environment; and Kinch feeding surveillance footage of spaces onto the surfaces that define those spaces, providing occupants the opportunity to participate in the production of the physical environment that they interact with.

Recalling the development of conceptual art practices in parallel to early commercial computing in the 1960s and 1970s, from codes for serial operations (e.g. LeWitt) to entropic fields (e.g. Smithson) to interactive feedback mechanisms (e.g. Bruce Nauman), these experiments each explore the capacity for new computational tools to continually widen the gap between the authorship of a project and the direct production of its form.

CONCLUSION

Whether deployed towards rigid order or apparently random distribution, automation defers or eliminates the individuated choices typically made in the design of a work of architecture, prompting new questions in the evaluation of the objects that are produced. Directed towards new goals, requiring new media, and displacing authorship such that it is no longer coextensive with the composition of an architectural object, but is limited to specific moments of intervention, automation reveals underlying and often unquestioned disciplinary structures of control, constraint and communication.

In the essay "Ways About Error," Sean Keller puts forward a description of "automatism," drawn from the work of the philosopher Stanley Cavell.[21] Cavell suggests that for modernist artists, the dissolution of traditional genres and forms leads to the necessity of generating new structures to work within.[22] The coherence and criteria for evaluation once given by these external frameworks shifts to the individual production, on the part of each artist, of a systematic working method that allows for rigorous investigation of a medium through new "objective" criteria. A parallel might be drawn to a contemporary moment in which technological advancements, and a widening discursive field, suggest that, within architecture, anything (or everything) is possible. The design of a system of production that is encoded in representational media—which can be evaluated as material artifacts in their own right—enables work to be understood and evaluated at the level of method.In a pluralistic context, and in the pedagogical environments in which these projects are produced, such structures constrain techniques and makes evident the ways in which they are used. When confronted with few external limits, the production of a working method that is, at some level, machinic--or derived from the otherwise implicit constraints of the tools being used--curbs the blind adoption of conventions that have become naturalized (or automatic.)[23] In a context of pervasive automation, the projects discussed here suggest a turn away from the use of automated tools for instrumental ends and towards work on the material and conceptual structures of computational automation itself.

ENDNOTES

1. It is perhaps not surprising that two such projects should emerge in Los Angeles, where there has been a long-standing history of avant-garde architectural aspirations tending towards visual art and computational technology (e.g. Frank Gehry).
2. These and two other practices--Reimaging (Gabriel Friese-Briggs, Nicholas Pajerski and Brendan Shea) and Plethora Project (Jose Sanchez)-- participated in "A Brief Symposium on Automation and Architecture," organized and moderated by the authors at 2426 Set in Los Angeles on April 15, 2017.

3. This shift out of the "digital" culture of the 1990s and 2000s has been termed by some as the "postdigital." See, for instance, *Lluís Ortega's Total Designer: Authorship in the Architecture of the Postdigital Age* (New York, NY: Actar, 2017)

4. A spectrum of such work might include formlessfinder (Garrett Ricciardi and Julian Rose) or Milliøns (Zeina Koreitem and John May). The process recounted in Samuel Stewart-Halevy's "L'Auberge Espanol," *Project* 4 (Winter 2015), is a clear example.

5. See Mark Fisher's *Capitalist Realism: Is there no alternative?* (Winchester, UK: Zero, 2009), which attributes the quote "it is easier to imagine an end to the world than an end to capitalism" to both Fredric Jameson and Slavoj Žižek.

6. See our forthcoming essay, "Paragraphs on Automation," *Depot* 1 (2017).

7. Which was not widely used until the 1940s.

8. An argument pursued, for instance, by Mario Carpo in *The Alphabet and the Algorithm* (Cambridge, MA: MIT, 2011).

9. Among numerous examples, one might think of the work of Claude Perrault on the classical orders or Jean-Louis-Nicolas Durand on plan organization.

10. The most ambitious attempt at a contemporary theory of computational automation, Patrik Schumacher's parametricism, systematizes constraints into rule sets that purport to optimize architectural form in functional terms, while achieving a unifying style. See, *The Autopoiesis of Architecture: A New Framework for Architecture*, Volume 1 (London, UK: Wiley, 2011).

11. Which maximizes authorial control over the outcome of the design through highly-constrained modelling parameters--and the integrated production of orthographic drawings from comprehensive building models--that enable, in theory, all design conflicts to be fixed prior to construction.

12. Think, for example, of the more or less simultaneous invention of Fordist mass production and Marcel Duchamp's readymades in the mid-teens, or the advent of commercial computing and Sol LeWitt's rule-sets in the late 1960s.

13. Sol Lewitt, "Paragraphs on Conceptual Art," *Artforum* (Summer 1967).

14. Several authors have remarked that Eisenman's work anticipates computational automation. Alejandro Zaero-Polo describes Eisenman as "the first truly machinic architect," "replacing origin, presence, and author with arbitrariness, absence, and machinic behaviour." See "Peter Eisenman's Machine of Infinite Resistance," El Croquis 87 (1997). Greg Lynn argues that prior to introducing computation into his design work, Eisenman was "the computer," working "with the code and the iteration and the variables and the repetition." See *The Archaeology of the Digital* (Montreal, QC: Sternberg/CCA, 2013)

15. For an explicit take on the relationship between this work and that of artists of the 1960s and 70s, see Michael Meredith's "Indifference, Again" *Log* 39 (Winter 2017).

16. Screenshots are a form of automation as well, distilling the labor of drawing to the manipulation of software settings. For an extended take on the use of screenshots see Matthew Allen's "Screenshot Aesthetic" in *MOS Selected Works* (Princeton, NJ: Princeton, 2016).

17. This is a reading of the project as it appeared in *Perspecta* 46: Error (2013).

18. For an extended discussion of images in a computational context, see John May, "Everything is Already an Image" *Log* 40 (Spring/Summer 2017).

19. These experiments are also being pursued towards more conventional ends, as in the skyline design tool developed by KPF's Urban Interface group, which allows building massing to be generated through drawing profiles of buildings as seen from various locations in Google Earth. http://ui.kpf.com/blog/2016/2/2/skyline-design-tool.

20. In another convergence, subsequent iterations of the "Mountain" project (e.g Paranormal Panorama, 2014) move from the manipulation of digital models by orthographic projection to the production of rendered, panoramic images. These images are progressively abstracted into layers of pure color that can be communicated through construction drawings to a builder (or, properly, a painter) and fused into the architectural surface. The project is conceived through the representational logic of the rendering (the automatic production of the image through computational labor) and the construction document (the automatic production of the painting through physical labor).

21. Sean Keller, "Ways about Error," *Perspecta* 46: Error (2013), the same issue that features "How to Domesticate a Mountain."

22. See the section "Automatism" in Stanley Cavell, *The World Viewed* (Cambridge, MA: Harvard, 1971).

23. Though as strategies of automation proliferate, they too become conventional, as seen in the art historical examples referenced throughout.