Through the introduction of human interaction capabilities, the Resonant Chamber can also operate as a responsive system (Velikov et al. 2012). We are currently exploring the incorporation of multi-modal sensors such as Microsoft's Kinect to dynamically track locations of sources and receivers in order to calibrate performance spaces relative to audience position, number and spatial distribution. The control system might also be embedded with machine learning capabilities and priorities beyond acoustic optimization, inviting new forms of interactive play and live performance to develop. Alternatively, in a context of aural accessibility, individual hearing requirements can be addressed through the use of RFID receivers, tuning local aural conditions to the capacities of inhabitants.

The ecological paradigms of system/environment intertwinement, co-evolution and cybernetics inform not only the hybrid computational and physical method by which the Resonant Chamber system and its components is being developed, but also implicate an adaptive and open-ended role for the designer, both in terms of their relationship to the design process and also in terms of the constructed system's performance and goals. As such, the Resonant Chamber becomes a system continually able to be shaped through the interests of multiple agents and agendas, rather than limited by its initial perceived function and application.

ACKNOWLEDGMENTS

Project Team: Geoffrey Thün (lead), Kathy Velikov, Lisa Sauve, Wes McGee, Mary O'Malley, Colin Ripley, Adam Smith, Katie Wirtz, David Lieberman, Ian Ting, Lief Millar, Dr. Jerome Lynch, Devki Desai, Mike Kane. ARUP Acoustics Consulting Engineers: Raj Patel, Terence Caulkins, Dave Rife. Funding: University of Michigan Taubman College Research Through Making Grant, OVPR Small Projects Grant, and Center for Wireless Integrated MicroSystems; SSHRC Research Creation Grant.

REFERENCES

Bateson, G. (2000). Steps to an Ecology of Mind (1972). Chicago: University of Chicago Press.

Bergandi, D. (2000). Eco-cybernetics: The Ecology and Cybernetics of Missing Emergences. In Kybernetes 29 (7/8), 928–42.

Capra, F. (1999). Systems Theory and the New Paradigm. In Key Concepts in Critical Theory: Ecology (1988), ed. C. Merchant, 334–41. Amherst, NY: Humanity Books.

Corteel, E., K.-V. Nguyen, O. Warusfel, T. Caulkins, and R. Pellegrini. (2007). Objective and Subjective Comparison of Electrodynamic and MAP Loudspeakers for Wave Field Synthesis. In AES 30th International Conference.

Egan, M. D. (1988). Architectural Acoustics, 133. New York: McGraw-Hill.

Kay, J. (2002). On Complexity Theory, Energy, and Industrial Ecology. In Construction Ecology: Nature as the Basis for Green Buildings, ed. C. J. Kilbert, J. Sendzimir, and G. B. Guy, 72–103. New York: Spoon Press.

Kulper, A. (2011). Ecology without the Oikos: Banham, Dallegret, and the Morphological Context of Environmental Architecture. Field Journal 4 (1): 67–84.

Odum, E. P., and G. W. Barrett (2004). Fundamentals of Ecology, fifth edition. Various Locations: Brooks Cole.

Pask, G. (1969). The Architectural Relevance of Cybernetics. Architectural Design (September): 494–96.

Peters, B. (2010). Acoustic Performance as Design Driver: Sound Simulation and Parametric Modeling Using SmartGeometry. International Journal of Architectural Computing 8 (3): 337–58.

Peutz, V. M. A. (1978). The Variable Acoustics of the Espace de Projection of Ircam. In AES 59th International Conference.

Resch, R. (1972). The Topological Design of Structural and Architectural Systems. In AFIPS Conference Proceedings 42, National Computer Conference.

Ripley, C., G. Thun, and K. Velikov. (2012). Soft Goes Hard. In [bracket]—Goes Soft—Almanac 2, 34–43. Barcelona and New York: Actar.

Tachi, T. (2010a). Geometric Considerations for the Design of Rigid Origami Structures. Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium 2010, Shanghai.

Tachi, T. (2010b). Rigid-Foldable Thick Origami. Proceedings of 5OSME, Singapore.

Velikov, K., and G. Thün. (2012). Responsive Envelopes: Characteristics and Evolving Paradigms. In Design and Construction of High Performance Homes, ed. Franca Trubiano. London: Taylor and Francis.

Velikov, K., G. Thün, M. O'Malley, and C. Ripley. (2011). Toward Responsive Atmospheres: Prototype Exploration Through Material and Computational Systems. In ACADIA 2011 Integration Through Computation, Proceedings of the 31st Annual Conference, eds. J. M. Taron, V. Perlac, B. Kolarevic, and J. S. Johnson, 326–33.

Wolfe, C. (2010). What Is Posthumanism? Minneapolis: University of Minnesota Press.

Xiao, L., T. J. Cox, and M. R. Avis. (2004). Active Diffusers: Some Prototypes and 2D Measurements. Journal of Sound and Vibration 285: 321–39.

IMPLICIT FABRICATION, FABRICATION BEYOND CRAFT:
THE POTENTIAL OF TURING COMPLETENESS IN CONSTRUCTION

Jelle Feringa
Hyperbody TU Delft /
EZCT Architecture & Design Research

ABSTRACT

Over the last decades numeric control (NC) and robotics have become firmly grounded within architectural practice. While the hacking of CNC machinery, the development of production strategies specifically oriented toward architectural applications, and the development of robot end-effectors implementing new architectural fabrication processes are perceived as forms of craft, a salient contradiction surfaces. While robotics has essentially developed in the field of industrial automation, in architecture, NC is increasingly understood as a form of digital craft. This year's ACADIA conference is no exception; it refers to the increasing role of robotics in construction as robot craft. Is this a contradiction in terms? If so, what is the origin of this contradiction, and what consequences are implied? The state of the art in lights-out manufacturing (LOM) has plants operate unattended for a month, while on the other hand, handheld machining tools are being developed; the scope of NC has become so vast. The role of NC in fabrication is at a pivotal point in architecture; due to its widespread adoption, limitations of the approach have come within view. This article seeks to identify such limitations and bottlenecks and to develop a critical position for the coming decade of NC in manufacturing architecture.

figure 1

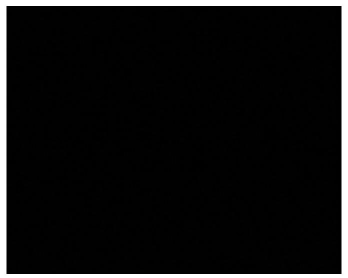

figure 2

1 INTRODUCTION

1.1 The Contradiction of Digital Craft

A renewed perspective is developing on the notion of the architect as master builder; the concept of an architect such as Jean Prouvé with the means of production resurfaces. Numeric control (NC, used here as a hypernym for both CNC and robotic production methods) is the critical technology that makes reexamining this concept worthwhile. The horizon of the impact of *"The linguistic turn of architectural production"*—to paraphrase Philippe Morel (Morel 2010)—on architectural practice is not yet in sight. Projects such as the Gantenbein of Gramazio & Kohler, practices such as Rok-Office and Snøhetta, and the emerging of specialized offices that bridge the fabrication gap such as designtoproduction and 1:One, convincingly demonstrate how a more central position within the construction process can be claimed—one where the architect ceases to be an interchangeable middleware between developer and contractor. Apart from the contradiction in terms of digital or robotic craft, the notion of craft seems to ignore its industrial potential, which is peculiar since the approach is rooted in industry. Given the potential for NC to renew the architectural profession, is framing NC as digital craft not Luddism, self-censorship? Or does the notion of craft merely point out that digital craft is an emerging field, or is it a consequence of an artisanal inclination toward manufacturing? The fact that NC is over half a century old argues against this, as do the economic implications of adoption of NC. The innovative FreeD handheld milling machine effectively explores the contradiction of digital craft; is it both craft and automation, or neither craft nor automation? Perhaps the contradiction—NC was conceived essentially as a form of automation—is conspicuous since these oppose some of the central concepts in computational architecture. I think of John Frazer's notion of an *"army of clerks"* (Frazer 1995); computing's diminishing costs are significant. The same holds progressively true for fabrication; the cost of running for NC is low in relation to its production capacity and potential returns.

The disparity in approaches toward NC ranges from handheld milling devices to the extreme of LOM where robots assemble similar robots, a rudimentary form of self-assembly. The central issue here is that the notion of craft fails to acknowledge precisely this linguistic turn of architectural production; the practical problem of production has been transmuted into an intellectual problem. The dilemma of production is no longer managing hordes of workers, nor is it the mechanization of work; it has become the more abstract and intellectual ability to express the construction of an artifact as a set of executable instructions—as code. A decade after architecture's linguistic turn, the moment is here to reflect on NC's original intentions and the merits and shortcomings of the development, and to look forward to which possibilities still lie ahead of us.

2 LIGHTS OUT MANUFACTURING

Roger Smith, CEO of General Motors in the 1980s, subject of Michael Moore's film Roger & Me, is the person who invented and evangelized the term "lights-out manufacturing." Since then, the manufacturing of robots has become a rudimentary form of self-assembly; robots have been assembling robots in an LOM approach since 2001. "At this moment, in one of FANUC's 40,000-square-foot factories near Mt. Fuji, robots are building other robots at a rate of about 50 per 24-hour shift and can run unsupervised for as long as 30 days at a time [author's emphasis]. When they stop, it's because there's no room to store the goods. Trucks haul off the new robots, the lights are cut, and the process begins anew. 'Not only is it lights-out,' says FANUC vice president Gary Zywiol, 'we turn off the air conditioning and heat too.' (Null and Caulfield 2003)

3 WORKFLOW

The artisanal inclination of digital craft can be explained partially by the highly involved process of moving from form toward fabrication. Three degrees of code are implied in what is best described as a canonical workflow in computational architecture. First, design intent is formalized in code that produces geometry.

figure 1

The scope of digital means of manufacturing. (Left) FreeD, developed by the responsive environment group at MIT: bot craft and automation, or neither craft nor automation? (Right) Lights-out manufacturing: FANUC plant near Fuji, Japan.

figure 2

A canonical computational architecture workflow.

Secondly, code is developed—custom fabrication processes require custom software to generate fabrication code (super-Tools, HAL, Kuka|prc, Lobster)—or existing tools (among the usual suspects are Mastercam, Robotmaster, RobotStudio, and RhinoCAM) are deployed for geometric interpretation and code generation for fabrication. Finally, the outcome of the geometric interpretation once again is code: a g-code dialect or robot code, such as RAPID for ABB robots or KRL for Kuka robots. In other words, moving toward fabrication is a highly indirect, transliteral, and cross-domain process.

3.1 Semantics

The circuitous and unidirectional nature of this approach is problematic for the semantic integrity of the generated fabrication code. If code is so central in manufacturing, shouldn't we pay more attention to the way fabrication instructions are structured? Clearly, semantics will not affect tooling precision, but it is important to realize that the limitations of production are increasingly framed as intellectual rather than mechanical limitations. I argue that such limitations are ever more embedded in the semantic structure of fabrication code. These programs represent fragments of a building, frustrating an integrated approach. Once fabrication code is generated from the model, the umbilical cord is cut off, with no way to adapt such fragments to late-breaking changes. These various fragments are described individually in an absolute coordinates system rather than in relative coordinates, which is semantically more meaningful, since this encodes how the fragments relate, allowing for online correction during production. This ability is especially important in relation to additive manufacturing processes, where material dispersal generally is less exacting than in subtractive processes.

The merits of such an approach were aptly demonstrated by dFAB (School of Architecture, Carnegie Mellon University) at the *"FABRICATE: Making Digital Architecture"* conference through the integration of computer vision (CV). Stacking the discrete elements one on top of another leads to buildup of tolerances and as such cannot be captured in an absolute coordinate system. When the stacking operations are described in a relative coordinate system, compensation for eventualities can be accommodated in the continued assembly. If this technique is already a prerequisite for the assembly of a folly with a diameter that equals the robot's reach, then the ability to generate fabrication code in a relational or relative manner is an essential technology for architecture.

figure 3

figure 4

figure 3

Remote sensing applied in the work of Gramazio & Kohler, ETH Zürich. (Left) Robust assembly by integrating computer vision at FABRICATE 2011. (Right) Flight assembly exposition opening at FRAC Centre, Orléans.

figure 4

Robotic welding of stiffeners incorporating force-feedback control, as developed by Jan Kranendonk.

3.2 Turing Completeness in Construction

Turing completeness is the ability of a computer to simulate the most primitive computer (while the ability to simulate the simplest computer equates to simulating the most complex); in practical terms, it implies the ability of conditional branching, the power to execute if and goto statements. As such, a CNC controller cannot be considered Turing complete, while a robot is. During production a robot has the ability to compute, to execute a software program, to sense its environment and respond to these signals, and to make decisions while the code is run. Turing completeness is essential in order to describe fabrication code at a higher, semantically more meaningful level than the mere motion instructions that a CNC controller is able to execute. What explains the merit of dFAB's demonstration is that the fabrication instructions are described in relation to one another, and the feedback is employed to adapt the instructions and as such explores Turing completeness in construction. An interesting application of the exploration of environmental feedback to develop robust and scalable construction methods is how force feedback is employed in welding ship hulls. Rather than assuming that a steel stiffener follows a perfectly straight line, the process has to be able to adapt to deviations. The robot is equipped with a force-feedback sensor; during the welding process, the tooltip locates both the horizontal plate and the stiffener; and given a measure of force feedback, the right welding spot is considered to be located. Rather than executing a stream of instructions, the robot is executing a program, a semantically meaningful construct other than the stream of explicit instructions. In order to move beyond craft, to develop original construction technologies that are both robust and scalable it is essential to embrace Turing completeness in construction. In an email exchange with the author, Jan Kranendonk argues that CAM operators routinely produce code for a hundred robotic welding hours within an hour. An important realization is that the economics of robotic imply that if you are able to program robots in a meaningful manner, you will.

On the other hand, the relatively modest cost of robots in view of their production potential and the ability for processes to run 24/7/365 robustly pave the way for construction processes that are of greater resolution or relatively inefficient, where the relative inefficiency is offset by the capacity to produce around the clock robustly. What kind of architectural design will give rise to a production process where a dozen robots run continuously for half a year? I am not arguing for robotic water dripping to form stalactites, but the possibilities that this perspective opens up are intriguing, while the cost involved is not prohibitive per se.

3.3 Limits of the Current Approach

The merits of numerical control have been well known for over half a century. Interestingly, the widespread adoption of computer-aided manufacturing (CAM) has brought attention to the limitations of the approach. The Foundation Louis Vuitton (FLV) project of Gehry CAM takes place at an unprecedented scale and level of complexity. The baroque project is a true building information modeling (BIM) tour-de-force with "over 100,000 versioned iterations of the BIM model, nearly 100Gigs of BIM data, 19,000 unique CNC-molded glass-reinforced concrete panels, 3500 unique CNC-molded curved glass panels." While these figures speak for themselves, never before had such an immensely detailed CAD model ever been developed. The project by itself is driving the French construction industry forward more than any other project in this decade, though the approach is not without issues. While the challenges of the project's adoption of CAM fabrication is unprecedented, so is the rumored ratio of architects to construction workers. Architecture seems light-years away from embracing LOM, but perhaps the approach offers a number of hints:

- In LOM, there is an extreme degree of integration.
- Robots poll the state of other robots on conveyor belts and are orchestrated to respond in a global, synchronized state of production. The process is local and deeply integrated. That is the opposite of how CAM is applied in architecture, where production is dispersed over a number of subcontractors, lacking both synchronicity and locality, and frustrating integration between the various phases of construction.
- This diaspora of production does not allow the fabrication code to be defined in a relational manner.
- Almost exclusively, the fabrication processes applied in computational architecture are driven by toolpaths. Toolpaths have become the lingua franca of CAM production, a unidirectional kitchen sink through which the project has to pass, where slowly but surely the horizon of what can be communicated effectively within this protocol comes into sight.
- Paradoxically, NC has reduced the margin of error to a problematic point, since the room left to correct eventualities has been proportionally reduced simultaneously. The margin for error is marginalized, paving the way for a tolerance rat race where the tolerance of component A inherits the tolerances of component B, since assembly is an unforgiving process in which error easily builds up.

4 TOOLPATHS

4.1 APT

There is no doubt that a huge leap was made with the formalization of production processes by NC. Conceptually the process is no different from when the concept of CAM arose, by the end of the 1950s at MIT. It is worth pointing out that APT (automatically programmed tools) is considered one of the first programming languages, invented shortly after Fortran, considered the first high-level programming language, and LISP (Bergin and Gibson 1996). (Interestingly enough, these three languages still are in widespread use today.) It is striking how CAM as a language has hardly evolved since, certainly when seen in the light of many great breakthroughs, revolutions, and evolutions in the development of programming languages. Douglas Ross, the inventor of APT, describes the design process of the APT language in detail in "Origins of the APT Language for Automatically Programmed Tools" (Ross 1978). What is striking is that to an extent, ongoing efforts to modernize machine tool programming in the STEP-NC effort share a similar ambition as early uses of the APT language. A central aspect of APT was its subroutine library. One can think of this library as a set of functions nowadays performed by CAM software. Ironically, rereading the ambitions of the APT languages (Ross 1978), in the early days of machine tool programming, such operations were described in a more high-level manner than what is common today. The STEP-NC effort follows a similar approach; machining operations—pocketing, drilling, roughening, contouring, tapping, swarfing—are associated to the CAD geometry. A library of subroutines then computes the toolpaths from this description—a considerable abstraction, since g-code is machine specific, while the STEP-NC approach is machine independent. So even while Ross took great care with regard to the semantic structure of the APT program and was deeply aware of the necessity of a clear semantic structure in order to achieve a high level of automation, his article starts with a wonderful and telling citation:

"(From THE NEW YORKER) Cambridge, Mass., Feb. 25, 1959. The Air Force announced today that it has a machine that can receive instructions in English, figure out how to make whatever is wanted, and teach other machines how to make it. An Air Force general said it will enable the United States to 'build a war machine that nobody would want to tackle.' Today it made an ashtray." (Ross 1978.)

Pointing out that the *"linguistic turn of architectural production,"* to paraphrase Philippe Morel (Morel 2010), has been insufficiently realized, the lack of semantic structure stalls the fulfillment of CAM in the context of architecture.

4.2 PostScript

John Warnock and Charles Geschke founded Adobe in 1982 and invented the page description language PostScript in 1984. In 1985 an interpreter for the PostScript language was developed by Adobe for the Apple LaserWriter. The coupled invention of PostScript and the LaserWriter gave way to the acronym WYSIWYG (what you see is what you get) and established desktop publishing. What

figure 5

figure 5
MIT ashtray.

figure 6

figure 7

made the approach revolutionary is that the technical complexities involved in printing are abstracted by the PostScript interpreter. The CAM equivalent is to load a CAD file tagged with machining operations into the CNC controller and produce it, while the interpreter on the controller deals with the cumbersome process of generating the toolpaths. This is the ambition of the STEP-NC (Minhat et al. 2009) effort and is a much-needed evolutionary step in machining. Though CAM and concepts not dissimilar to WYSIWYG printing were established 26 years ago, it has taken over 27 years to catch up; even when taking into account the considerable greater complexity of machining vs. printing, this is striking. I cannot stress enough that the differences in approaches here are first semantic, then technological. Ross had in mind a process that is closer to what PostScript achieved than what APT gave rise to. Machining code is regarded too much as data rather than as programming language. The close integration of the PostScript interpreter and the fact that PostScript is an adequate programming language led to experiments that suggest how manufacturing processes can be described in a more high-level manner.

5 POSSIBLE SOLUTIONS: AN OUTLOOK

5.1 Semantics in Fabrication Code

Just van Rossum—brother of Guido van Rossum, creator of the Python programming language—and Erik van Blokland together form LettError. A key project of the duo is the Beowulf typeface. The font is a small piece of software written in PostScript, as discussed in this interview:

Wired: You made the first "random" typeface, called Beowulf, by replacing the commands "lineto" and "curveto" in the PostScript code with your own command "freakto." The new command calls up a random generator that makes the character outlines irregular. When you created Beowulf, were you trying to prove something, or was it just a joke?

van Blokland: It was quite a joke. We were both into programming—or would you call it hacking? What came of that interest was a very cool-looking thing. We wanted to make a typeface that looked very smooth and rounded off, but instead it became spiky, with little pointy bits sticking out from the edges of each character in a most unpredictable way. And what's the most fun about Beowulf is that every time you print it, those spiky bits take on a slightly different appearance. (Spiekermann 1995)

What's of interest is that the design and its resulting products are self-contained in the Beowulf PostScript font, exploiting the elegant language/interpreter in a creative way. Interpreting PostScript equates to design production. The Beowulf font can be interpreted as a critical reflection of how design is encoded. The realization that PostScript is not merely a file format, but also a Turing-complete programming language is far-reaching. The production of unique characters is an impossibility in (tool)paths, though trivial to implement in a programming language.

5.2 Perspectives: Computer Vision

Is there an architectural analogy to the approach evoked by LettError? Implementing a design in a robot language like RAPID, where interpreting robot code on the controller equates to design production, offers perspective. A project of Hironori Yoshida (SIGGRAPH 2011) suggests such an approach and demonstrates the potential of remote sensing / computer vision for manufacturing procedures. The project exploits wood-grain patterns observed by a camera; imaging routines are applied to interpret g-code from it, mapping perception to production.

These procedures can be implemented in a high-level robot language with relative ease. Picking and placing wooden panels is a routine job as well. Therefore, such a process—even while yielding intricate artifacts—can be fully automated with relative ease, which essentially has to do with how the design was encoded. Certain processes are considerably easier to implement and will scale up in terms of production. Being able to recognize this early on in the design process increasingly is an essential ability, while geometric intricacy of the design is a diminishing factor. A product and process that exemplifies this idea is Bolefloor, developed by the Institute of Cybernetics at Tallinn University of Technology. The approach integrates remote sensing and applies advanced nesting algorithms. Data from a laser scanner is fed to nesting algorithms that both maximize lumber yield while adding value in terms of design. To paraphrase Kevin Kelly: *"in fact, in the technium, self-generated positive constraints are more than half the story; they are the main event"* (Kelly 2010).

figure 6

Applications of computer vision coincident in design and production. (Left) Hironori Yoshida: hybrid materiality explored by means of computer vision algorithms. (Right) Bolefloor: integration of advanced nesting algorithms and computer vision results in increased yield and adds design value.

figure 7

Stigmergy revolves around the idea of embedding construction instructions in the environment and reading the instructions from it.

An important realization is that increasingly the generation of fabrication code is a process that potentially is more costly and/or more time-consuming than executing it, which vindicates the relevance of processes that are described in higher abstractions such as remote sensing, which lead to an automated generation of production code.

Considerable production potential is left largely unexplored where this is not an issue of costly materials, nor man-hours, but the intellectual problem of developing fabrication approaches that explore the 24/7/365 production capacity of the machinery. This realization leads to a field of research that can be described as implicit fabrication, where fabrication code is implied from the design process. What approaches can be identified that exemplify the future of this angle of research?

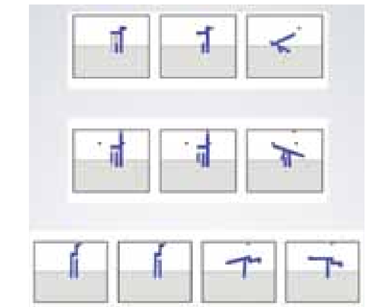

figure 8

6 PROCESSES OF IMPLICIT FABRICATION

An important step up the ladder of abstraction that computer vision provides is the idea of stigmergy. Stigmergy (Werfel 2006) revolves around the notion that work evokes work; a change in the environment embeds information encoding the continued development of that environment. As such, the process can be categorized as a form of autopoiesis. The action of assembling elements encodes instructions on how to further continue the assembling process; a subassembly becomes a symbol that implicates the further advancement of that assembly—a form of positive feedback.

Stigmergy does not equate self-assembly, nor is self-assembly necessarily a process that exemplifies implicit fabrication. "Implicit" here means that the generation of the code required for construction is implicit to the design process, whereas the self-assembly process often is explicitly defined.

6.1 Evolutionary Fabrication

The field of generative robotics offers a number of important ideas for the development of implicit fabrication. The grail of this emerging field is the concept of self-reproducing robots. An early and critical realization is the concept of embodiment (Funes and Pollack 1998). The research suggests that to evolve robots, simultaneous evolution of brain/body is required; otherwise one might end up with a body that cannot be controlled or a brain that cannot control its body. The concept of

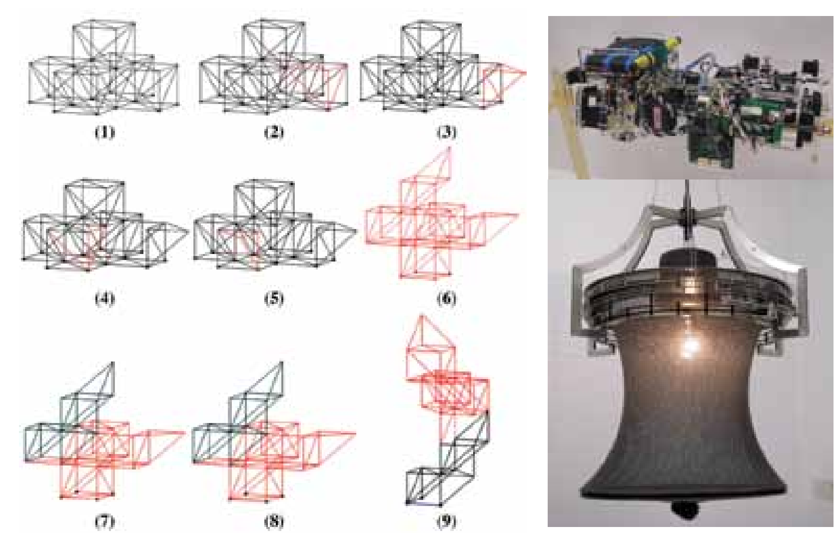

figure 9

figure 8

Simultaneous optimizing of design and assembly plan.

figure 9

The means of production and its consequent outcome have formed an indiscernible whole; fabrication is fully implicit to the artifact itself. (Left) Cornell Creative Machines Lab's robotically manipulatable structures. (Right) Sleeping Beauty, Nadine Sterk.

evolutionary fabrication emerged as a consequence of the idea of embodiment.

The pivotal realization is that *buildability* is a central concept in generative robotics. There is little meaning in the ability to produce design blueprints if these cannot be executed/self-reproduced.

"Approaching Fully Automated Design and Manufacture from this perspective requires a new formulation of Evolutionary Design, one that replaces descriptive blueprints with prescriptive assembly plans. In this approach, the formation of an object can no longer be taken for granted; we must realistically simulate not only the behavior of a finished object, but its entire assembly as well" (Rieffel 2006).

This is what Rieffel refers to as the *fabrication gap:* by merely specifying the form of an object, this approach leaves unanswered the vital question of formation. Rieffel's concept revolves around the idea of *situated development*. The assembly process is a part of the evaluation criteria of the fitness function for the evolutionary design process; how cleverly a structure is built is part of the evaluation criteria while the structure is being optimized. Evolutionary computing is used not only to arrive at a global optimum of a design given a set of constraints, but also, in this approach, to simultaneously generate an optimized assembly plan. Figure 8 demonstrates the potential of Rieffel's coupled approach; here the objective of the design is to maximize the shading area. The poetic term ontogenic scaffolding refers to the temporary elements used while assembling the structure. The dynamics of the assembly are simulated and subvert the assembly constraints. As a result, integrating and exploiting the assembly constraints bring about an interesting notion: where the unfinished structure is effectively used as a tool, raising the idea of self-referential fabrication; where the construction of a structure implies this unfinished structure as an essential means of production until it is completed, when it progressively ceases to be a tool. Nadine Sterk's lamp design Sleeping Beauty suggests such an approach. Can a pavilion be a tool for the production of that pavilion until it is completed and effectively stops being a tool? An approach developed by the Cornell Creative Machines Lab research on robotically manipulatable structures (Lobo, Hjelle, and Lipson 2009) moves toward such procedures, where robots continuously disassemble and reassemble a constant number of structural elements to facilitate an ever-changing architectural program, an approach inspired by metabolism.

Evolutionary fabrication holds great architectural potential. Both the design and building processes are simultaneously optimized, which inspires the realization that—much the way the evolution of a robot's body requires the evolution of its brain—exploiting the true capacity of robotics in architecture requires the simultaneous evolution of building and construction processes.

REFERENCES

Bergin, T. J., Jr., and R. G. Gibson, Jr., eds. (1996). History of Programming Languages—II. New York, NY: ACM.

Frazer, J. (1995). An Evolutionary Architecture. London: Architectural Association Publications.

Funes, P., and J. Pollack. (1998). Evolutionary Body Building: Adaptive Physical Designs for Robots. Artificial Life 4: 337–57.

Kelly, K. (2010). What Technology Wants. New York: Viking Adult.

Lobo, D., D. A. Hjelle, and H. Lipson. (2009). Reconfiguration Algorithms for Robotically Manipulatable Structures. In ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots, ReMAR 2009, 13–22.

Minhat, M., V. Vyatkin, X. Xu, S. Wong, and Z. Al-Bayaa. (2009). A Novel Open CNC Architecture Based on Step-NC Data Model and IEC 61499 Function Blocks. Robotics and Computer-Integrated Manufacturing 25(3): 560–69.

Morel, P. (2010). The Linguistic Turn of Architectural Production.

Null, C., and B. Caulfield. (2003). Fade to Black: The 1980s Vision of "Lights-Out" Manufacturing, Where Robots Do All the Work, Is a Dream No More. Business 2.0 (June 1). http://money.cnn.com/magazines/business2/business2_archive/2003/06/01/343371/index.htm.

Rieffel, J. (2006). Evolutionary Fabrication: The Co-Evolution of Form and Formation. PhD thesis, Brandeis University, Waltham, MA.

Ross, D. T. (1978). Origins of the Apt Language for Automatically Programmed Tools. SIGPLAN Notices 13(8): 61–99.

SIGGRAPH. (2011). Bridging Synthetic and Organic Materiality: Graded Transitions in Material Connections. In ACM SIGGRAPH 2011 Studio Talks, SIGGRAPH '11, 10:1. New York, NY: ACM.

Spiekermann, E. (1995). LettError. http://letterror.com.

Werfel, J. (2006). Anthills Built to Order: Automating Construction with Artificial Swarms. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA. AAI0810111.

THE FORBIDDEN SYMMETRIES

Rima Ajlouni
Texas Tech University

ABSTRACT

The emergence of quasiperiodic tiling theories in mathematics and material science is revealing a new class of symmetry that has never been accessible before. Due to their astounding visual and structural properties, quasiperiodic symmetries can be ideally suited for many applications in art and architecture, providing a rich source of geometry for exploring new forms, patterns, surfaces, and structures. However, since their discovery, the unique long-range order of quasiperiodic symmetries is still posing a perplexing puzzle. As rule-based systems, the ability to algorithmically generate these complicated symmetries can be instrumental in understanding and manipulating their geometry.

Recently, the discovery of quasiperiodic patterns in ancient Islamic architecture is providing a unique example of how ancient mathematics can inform our understanding of some basic theories in modern science. The latest investigations into these complex and chaotic formations is providing evidence to show that ancient designers, by using the most primitive tools (a compass and a straightedge), were able to resolve the complicated long-range principles of tenfold quasiperiodic formations.

Derived from these ancient principles, this paper presents a computational model to describe the long-range order of octagon-based quasiperiodic formations. The objective of the study is to design an algorithm for constructing large patches of octagon-based quasiperiodic formations. The proposed algorithm has proven to be successful in producing an infinite and defect-free covering of the two-dimensional plane.