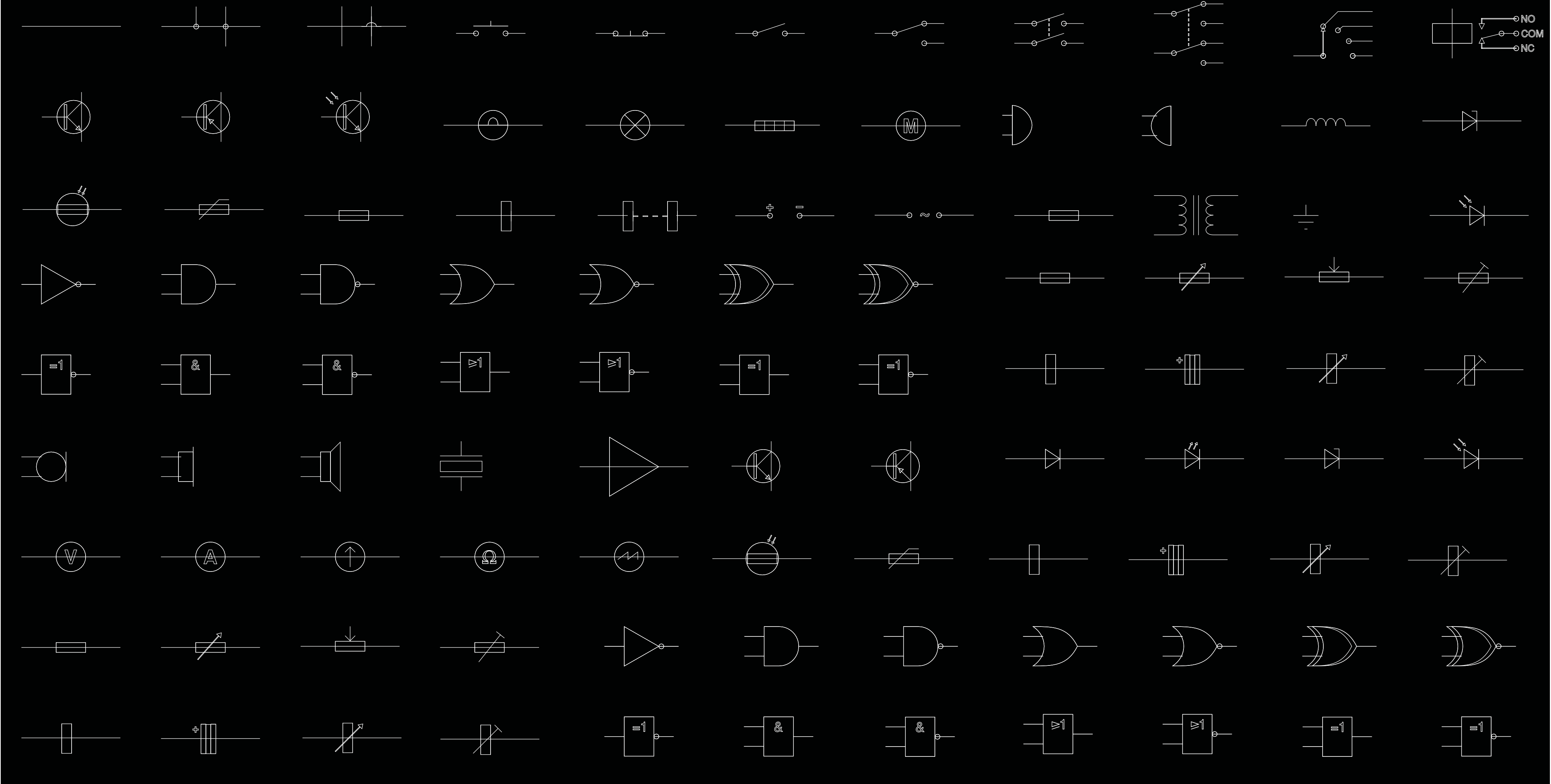
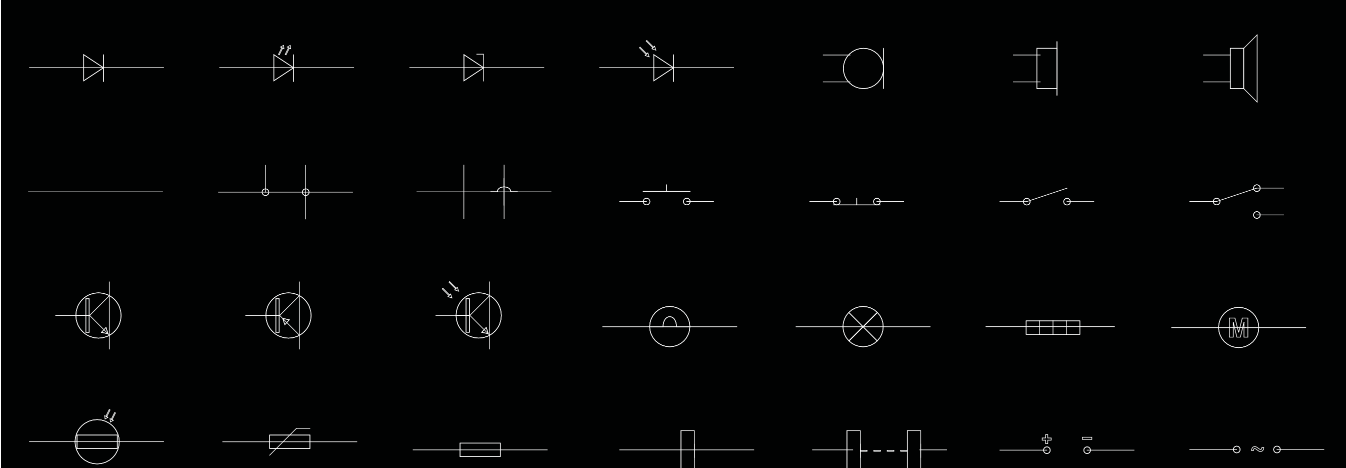




APPLIANCE ARCHITECTURE

in the INVIIBLE COLLEGE

a pedagogical text

[blank for printing]

FEBRUARY 3rd 2011

KEYWORDS:
appliance, computation,
interactive, particaportory,
responsive, robotic

APPLIANCE ARCHITECTURE
in the INVISIBLE COLLEGE
a pedagogical text

JONATHAN LORNE GRINHAM

THESIS SUBMITTED TO THE FACULTY OF THE VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE
IN
ARCHITECTURE

ROBERT J. DUNAY

KIHONG KU

ROBERT P. SCHUBERT

[blank for printing]

APLLIANCE ARCHITECTURE
in the
INVISIBLE COLLEGE
a pedigogical text

JONATHAN LORNE GRINHAM

ABSTRACT

This thesis presents a pedagogical framework for understanding dynamic Parametricism within the new media culture. As indicated by the title, 'Appliance Architecture in the Invisible College: a Pedagogical Text', this paper will serve two purposes. First, appliance architecture will construct the theoretical framework that will provide the context for the four case studies presented within this thesis: an interview with Rob Ley, designer of the Reef Project; the design and development of the Eclipsis Screen for the Solar Decathlon house, Lumenhaus; the development of an architectural robotics design laboratory, Prototyping in Architectural Robotics for Technology-enriched Education (PARTeE); and workshop > no.1, a physical computing workshop held at the College of Architecture + Urban Studies (CAUS). Second, the invisible college will serve as a pedagogical framework for teaching dynamic Parametricism within appliance architecture. The invisible college will explore the emergent design typologies developed through the PARTeE laboratory's first year and will culminate in the application of the teaching methodologies used for the physical computing workshop.

The following serves to establish the architectural discourse within which 'Appliance Architecture in the Invisible College' is embedded. In the broadest sense, this discourse is that of kinetic architecture. The word 'kinetic' is used to denote motion, or the act or process of changing position of over time, where time is the unit of measurement or relativity. The 'appliance' is defined as any consumer object or assembly with embedded intelligence; it does not shy away from the modern connotation of objects such as a coffee maker, refrigerator or iPod. The appliance as an assembly, therefore, presents a part-to-whole relationship that is understood through GWF Hegel's *organic unity*, which states: 'everything that exists stands in correlation, and this correlation is the veritable nature of every existence. The existent thing in this way has no being in its own, but only in something else' just as the whole would not be what it is but for the existence of its parts, so the parts would not be what they are but for the existence of the whole' (Leddy, 1991). It is this part-to-whole relationship which provides an understanding of the emergent typologies which structure the foundation for learning within the invisible college.

TABLE OF CONTENTS

LIST OF FIGURES

[blank for printing]

INTRODUCTION

In his book 'Cognitive Surplus: Creativity and Generosity in a Connected Age,' a study of the new media culture, Clay Shirky, Assistant Arts Professor at New York University's Interactive Telecommunications Program (ITP), presents an anecdote that illustrates a paradigm shift within our modern culture. Shirky recounts a friend's story of his four-year-old daughter suddenly rising to her feet mid-movie and beginning a vigorous search behind their television screen. The friend, from his own childhood experience, assumed the child was searching behind the screen for the people she was seeing on-screen. When asked, 'what are you doing?' the child responded, 'looking for the mouse.' This anecdote is used by Shirky to represent the new media culture in which we live, a culture that, in many ways, is more perceivable to a four-year then it is to previous generations. Shirky states, 'Here's something four-year olds know: a screen without a mouse is missing something. Here's something else they know: media that's targeted at you but doesn't include you may not be worth sitting still for' (Shirky, 2010). The four-year-old protagonist represents a societal and generational shift from a culture of media consumers to a culture of media producers. More importantly, the four-year-old represents the inescapable future of a culture whose members expect malleable, interactive and user-oriented environments. Skirky's story establishes the tone for the following work, which readily accepts that architecture currently exist within the new media culture.

THE ZEITGEIST / GENIUS LOCI PARADIGM

Since the late 20th century, studies of new media have created a vast lexicon. Researchers have studied new media from film, to television, to video games, to the Internet. The effect of these new media, and specifically the use of interactive technologies, has allowed the consumer to become a producer of innovative and powerful media content (Jerkins, Purushotma, Clinton, Weigel, & Robison, 2006). The new media culture's communication technologies enable and facilitate user-to-user interactivity, as well as interactivity between user and information. David Marshal, Chair of the Department of Communication Studies at Northeastern University in Boston, describes the new media culture by saying, 'These cultures, in their dynamic relationship with products, networks, hardware, software and practices are constantly changing in sometimes profound and sometimes banal ways' (Marshall, 2010). Architecture, through its design processes, its adoption of computer software and its formation of global design communities, has become a nodal point in the complex network of information exchange within the new media culture.

Peter Eisenman states, quite dejectedly, that contemporary architecture is subject to these new medias. He describes the new media culture as the *zeitgeist*, or spirit of the time. In the first point of his *Six Point Plan*, 'Architecture in a media culture,' Eisenman states:

> If architecture is a form of media it is a weak one. To combat the hegemony of the media, architecture has had to resort to more and more spectacular imaging. Shapes generated through digital processes become both built icons that have no meaning but also only refer to their own internal processes (Petrunia, 2008)

Eisenman expresses that architecture's adoption of the new media culture is a result of the 'lateness,' which is a term borrowed from Edward Said's book, 'On Late Style.' Here, Eisenman sees that architecture has fallen to a state of lateness in reference to Modernism, which he also describes as the consumer driven society of 'Late Capitalism.' Lateness describes a period of style which has essentially stalled; there is no longer an opportunity for 'newness,' resulting in a period within which designers and artists search for a new style or 'ism.' The result, in Eisenman's opinion, is the false emergence of media driven architectural 'ism' as the spirit of the time—here media is the use of the computer and the resultant digital imagery, surfaces and algorithmic designs. Eisenman believes that architecture is still in the 'Late Period' and has falsely adopted digital processes as a new style of design. He argues that as a result of the adoption of process oriented digital design, architecture is in a battle between the spirit of the time—*zeitgeist*—and the spirit of a place—*genius loci*. These processes that embody the *zeitgeist* have ultimately left architecture without a sense of place and a relationship to the socioeconomic and cultural particularities of the people it serves. (Eisenman, Lateness and the Crisis of Modernity, 2010). Instead, the *genius loci* has been replaced with designs that are self-referential to their own design processes. The resulting designs are subject to the 'Whorfain effect', which characterizes designs that are dictated by the language of the computer tools they use and result in the mannerism of 'hi-tech' (Terzidis, 2006). The self-referential style of these designs describe their perpetual stagnation; without a spatial reference they remain as late style, continuing to design within the philosophies of Modernism, De-constructivism and the Avant-grade. Eisenman, as an out-spoken philosopher of architecture, provides a substantiated view. However, to fully understand the polarity between *zeitgeist* and *genius loci*, the other side must first be presented.

Patrick Schumacher, partner at Zaha Hadid Architects (ZHA) and Codirector of the Digital Research Lab (DRL) at the Architecture Association (AA), London, presents a much different and equally subjective counterpoint. In his Parametricist Manifesto, Schumacher states:

> Contemporary avant-garde architecture is addressing the demand for an increased level of articulated complexity by means of retooling its methods on the basis of parametric design systems. The contemporary architectural style that has achieved pervasive hegemony within the contemporary architectural avant-garde can be best understood as a research program based upon the

> parametric paradigm. We propose to call this style: Parametricism…Parametricism is the great new style after modernism. Postmodernism and Deconstructivism have been transitional episodes that ushered in this new, long wave of research and innovation. (Schumacher, Parametricism as Style - Parametricist Manifesto, 2008)

Schumacher believes that design has passed the cusp—a term used by Eisenman in his book, 'Ten Canonical Buildings,' to describe the point at which design has passed the brink of lateness and stands without a stylist reference—and has situated itself in a new style, Parametricism (Eisenman, Ten Canonical Buildings 1950-2000, 2008). This new style is a response to the complexities of a 'post Fordian-Society' that has required a 'retooling' of its methods toward parametric design systems (Schumacher, Parametricism as Style - Parametricist Manifesto, 2008). Parametric is defined here as computer software and scripts capable of 'precise formulation and execution of intricate correlations between elements and subsystems' (Schumacher, Parametricism as Style - Parametricist Manifesto, 2008). As a result of these capacities, Schumacher proposes five agendas or subsystems within Parametricism: [1] Inter-articulation of sub-systems: correlation of differentiated systems within a whole; [2] Parametric Accentuation: amplification of differentiated systems; [3] Parametric Figuration: multiple readings within differentiated systems; [4] Parametric Responsiveness: adaptive /responsive differentiated systems; [5] Parametric Urbanism: differentiated systems at the scale of urbanism (Schumacher, Parametricism as Style - Parametricist Manifesto, 2008).

Schumacher's subassemblies of differentiated systems present a static Parametricism. Designs developed through these tools reside in dynamic virtual landscapes—parametric platforms that allow for the use of virtual time; the result, however is architecture that is in a frozen state and void of time. The resulting residue—the processes-based presentation and marketing to clients— is a log of evolution of a design versioning in relation to a given set of parameters (Terzidis, 2006). In this way, Parametricism is a means to an end, a system for articulating and applying the 'continuous differentiation' of the contemporary state of Avant-garde design and could rightly be considered 'hi-tech' (Schumacher, Parametricism as Style - Parametricist Manifesto, 2008) (Terzidis, 2006). However, Schumacher's fourth agenda presents an opportunity for new media culture to bridge Eissenman's *zeitgeist* /*genius loci* polarity. He States:

> We propose that urban and architectural (interior) environments can be designed with an inbuilt kinetic capacity that allows those environments to reconfigure and adapt themselves in response to the prevalent patterns of use and occupation. The real time registration of use-patterns produces the parameters that drive the real time kinetic adaptation process. Cumulative registration of use patterns result in semi-permanent morphological transformations. The built environment acquires responsive agency at different time scales. (Schumacher, Parametricism as Style - Parametricist Manifesto, 2008)

The proposition of a real-time, environmentally adaptive architecture presents a 'dynamic' Parametricism that allows for a new understanding of the spirit of architecture. Dynamic Parametricism proposes that new media allows architecture to explicitly describe both the spirit of the time—*zeitgeist*—and the spirit of place—*genius loci*—through a dynamic, emotive and evocative architecture. These architectures describe environments which are capable of referencing, in real-time, the social/cultural/environmental spirit of a place through the use of the interactive technologies of the new media culture.

In his book 'Towards a New Architecture,' Le Corbusier states that, 'Architecture is a thing of art, a phenomenon of the emotions, lying outside the questions of construction and beyond them' (Corbusier, 1931). It is this 'phenomenon of the emotions' that dynamic Parametricism is able to capture. Through the evaluation of new media cultures, Eisenman's polarity is brought into a balance. The question then becomes: how do we produce 'phenomenon of the emotions' through dynamic Parametricism?

APPLIANCE ARCHITECTURE

To evaluate this question, we must take a step back and understand the framework of dynamic Parametricism. First, we will adopt new nomenclature that will more effectively define emotive architectures. If one were to seek a definition for this term within the community whose thoughts and research it investigates, an immense number of naming conventions could result in an accurate description of its primary content. Various communities would associate its content with 'Digital Design', 'Parametric Architecture', 'Architectural Robotics', 'Interactive Architecture', 'Kinetic Architecture', 'Physical Computing', 'Participant Design', 'Algorithmic Design' or 'Responsive Architecture', etc., all of which may properly focus on this emerging area. Therefore, in order to avoid an etymological debate that has potential to deviate back to Socratic philosophy, a thematic of 'Appliance Architecture' will be adopted from Brian Boigen and Stanford Kwinter's 1991 studio brief, *Manual for 5 appliances in the Alphabetical City: A Pedagogical Text.*

Boigen and Kwinter's manual not only serves to establish a nomenclature for this work, but their studio brief also constructs a framework that allows for understanding of the 'phenomenon of the emotions.' In doing so, Boigen and Kwinter introduce the appliance:

> A piece of architecture, like an object, may be defined not by how it appears but by practices: those it partakes of and those that take place within it. Conceive, then, of the object of architectural practice (the appliance) as embodied not in a design in the traditional sense, but in a program, a set of procedures, a narrative, a catastrophic event, a mobilization, a complex new sense, affect, or emotion, a corporal technique, a collective hallucination, a dangerous scientific theory. (Boigen & Kwinter, 1991).

The appliance—those architectures which have an embedded intelligence at any scale— is the theoretical construct that carries with it a much more profound capacity for locating an architectural condition than its modern connotation as consumer good suggests.

The appliance, as expressed by Boigen and Kwinter, has produced phenomenal changes in the formal, spatial and social theories of architecture. These phenomena can be understood at multiple scales and periods of time. However, the full effect of the 'phenomenal' can most clearly be illustrated by the emerging architectures at the turn of the 20th century. The phenomenal shift can first be seen in Chicago and Manhattan in the late 1880s with Elisha Otis' invention of the elevator in 1852. Here, we see the appliance allow for a new architecture lexicon—the skyscraper. We may understand the elevator as an appliance architecture, the embedded intelligence and servomechanism within its minimal profile, paired with the new open spaces of the building's steel frames, allowed architecture to rise ad infinitum, ushering in the modern metropolis. As Rem Koolhaus states in his book 'Delirious New York,' 'The elevator is the ultimate self-fulfilling prophecy: the further it goes up, the more undesirable the circumstances it leaves behind' (Koolhaus, 1994) As buildings and their occupants rose, the deep spans of the building's open floor plans required new spatial conditioning. Rem Koolhaus highlights the 1900s shift in perception, he quotes Theodore Starrett's proposal for a 100-story building, 'Another interesting feature is the made to order climate we shall have. When we shall have at last reached the ideal construction, we shall have perfect control of the atmosphere, so that there will be no need of going to Florida in the winter or to Canada in the summer' (Koolhaus, 1994). The addition of the centralized heating, ventilation and air-conditioning (HVAC) systems and the elevator depict the utilitarian and subversive nature of the appliance that allowed for the emergence of the modern metropolis. Subsequently, the refrigerator, paired with the automobile, allowed for the suburban sprawl in the post-war American era. Here the refrigerator allowed for the domestic household to see a new suburban landscape, each individual's need for harvestable land was reduced, ushering in new social and architectural lexicon—the supermarket. These three examples highlight the ability of appliance architectures to radically shape the environment and social construct that we live within, yet they are imperceptible and uncelebrated in our modern lives. The human mind is incapable of grasping the complex social, economical and political networks that are the result of pouring a cold glass of orange juice in the morning. In this way appliance architecture seeks to understand how these subassemblies produce radical change.

Boigen and Kwinter see the appliance as an exploration of the 'mindless.' The 'mindless,' in this case, is the social and the phenomenal design that exists beyond the physicality of the appliance. Through the use of diagrams—what will become the five appliances and what we will understand to be directly analogous to the algorithms of computational logic found within this thesis—the idea of the 'mindless' is explored, where the goal is not to escape the metaphysical, but rather to be implanted within the modern context of the appliance. Kwinter and Boigen state:

> We think these acts, these functional diagrams, as machines, after Le Corbusier's "machine for living," yet we call them appliances to signal their late-modern rehabilitation as trivialized consumer objects. And yet it is in the very readymadeness and superficiality of these objects that we may discover what is radical about them: each bears within itself an abstract mechanism for producing political and social transformations at even the minutest scales of existence. (Boigen & Kwinter, 1991).

Although easily perceived as negative or opposing, the use of appliance here is heterodoxic. The appliance is, on one level, utilitarian, ubiquitous and subversive, capable of empowering its user to affect their local climate/condition without submitting themselves to the physicality of the appliance itself. It is this subversive quality that also allows for the evaluation of the 'mindless.' The appliance in our modern day has become virtually invisible, yet it produces social and phenomenal changes. By understanding appliance through its diagram, the subversive qualities of the appliance can be brought to the surface—teasing out the 'phenomenon of the emotions' that resides past the physicality of the appliance.

The diagrams, or the five appliances, within Boigen and Kwinter's Alphabetical City are concurrent investigations meant to produce and relay social, aesthetic, and political desire (Boigen & Kwinter, 1991). As a whole, the Alphabetical City is an amalgamation focused on the relationships of each desire to one another. By examining the feedback loop or conversation between relays of desire as responsive actors (meaning each diagram both pushes and pulls, or gives and takes data), the subversive, intangible emotions of an 'appliance existence' may be brought to light (Boigen & Kwinter, 1991). The appliance becomes the responsive, adaptive and emergent logic of the Alphabetical City. Therefore, the appliance is a method or media through which desire is transposed into an environment (either explicitly or implicitly). Michael Fox, a leader in the field of interactive architecture, explains the importance of this relationship of a relay to the creation of responsive architecture when he states, "a truly interactive system is a multiple-loop system in which one enters into a conversation: a continual and constructive information exchange" (Fox & Kemp, 2009). The diagram becomes a tool by which these desires, phenomenal design that exists beyond the physicality of the appliance, are deciphered. Kwinter will later produce a more concise definition of the diagram in his forward *The Judo of Cold Combustion* for Reiser and Umemoto's 'Atlas of Novel Tectonics:'

> But what exactly, after all, is a diagram? The diagram is an invisible matrix, a set of instructions, that underlies—and most importantly, organizes—the expression of features in any material construct. The diagram is the reservoir of potential that lies at once active and stored within an object or environment (or in every aggregate or section of these). It determines which features (or affects) are expressed and which are saved. It is, in short, the motor of matter, the modulus that controls what it does. (Reiser, 2006)

This understanding of the diagram is a direct analog for understanding the algorithmic logic found within appliances' embedded intelligence. In the case of the five appliances, the diagram serves as a tool for understanding the subversive logic which resides within the appliance. Within algorithmic design or dynamic Parametricism, the algorithm becomes the invisible matrix, which produces what the user reads as the phenomenal. The appliance in this study describes interactive, responsive and adaptive systems that employ circuits, processors, servomechanisms and sensors and that recognize the computer as an 'associate'. However, the framework of the study understands the physicality of these assemblies to be secondary, or subverted, to the emotive capacity of the collective whole. The physical appliance becomes a means by which to relay the phenomenological.

To understand how algorithmic logic is capable of diagramming the 'phenomena of the emotions' within appliance architecture, we must look toward the roots of new media as it relates to architecture.

The Logic of Appliance Architecture

In his 1969 article, *Towards a Theory of Architecture Machine*, Nicholas Negroponte, the founder of MIT's Media Lab, asked, "Can a machine deduce responses from a host of environmental data?" (1969). This question and others developed in the Media Lab sought to realize the machine and its algorithms as a partner or 'associate' to its human counterpart, ultimately developing 'humanism through machines' (Negroponte, 1970). In order for the machine to be an associate, Negroponte describes five subassemblies; the primary mechanism, which is the heuristic or learning mechanism, can be understood as a 'relay.' This mechanism allows for the constant feedback loop between the human and the machine (a computer, processor or any assembly of circuits which constitutes intelligence). As we look at the work of Kostas Terzids, an algorithmic design theorist, the 'relay' or conversation will explore 'humanism through machines' by means of the radical 'otherness' of the machine (Terzidis, 2006).

In his book 'Algorithmic Architecture,' Terzids explores the relationship between the algorithm and the human mind, which he defines as 'otherness.' However, this 'otherness' is described as an ambiguous contradiction that allows for rationalization of the algorithm as the potential for humanism within design:

> One might argue that the algorithmic procedures of the machine still remain fundamentally different from the way we think. But there again, a closer examination reveals a more ambiguous situation. For our mind follows rules in order to avoid the excessive familiarity that might otherwise defuse the originally of the creative endeavor, and these rules are usually as constraining as the algorithmic procedures run by the computers. In other words, the otherness that Terzidis attributes to the machine is also present in ourselves, in the apparent opposition between the creative impulse and the set of rules that enable us to control it (Terzidis, 2006).

'Otherness' is used throughout Terzidis' work as a theoretical framework that facilitates the understanding of the relationship of the algorithm within the design process. He notes the entomology of the word 'design;' in Greek the word is translated to mean 'nearly,' 'almost', 'about,' or 'approximately' (Terzidis, 2006). From this translation, he extrapolates that design is a means of capturing the elusive. Therefore 'otherness,' the relationship of algorithmic logic to human logic, is a way of capturing the elusive. Much like 'mindless' in the Alphabetical City, 'otherness' seeks to understand the emotive quality of algorithmic logic. This understanding is gained through a bottom-up system. Within bottom-up systems, rules or logic at one scale of complexity produce systems that exhibit behavior complexity that lies beyond the original logic; these systems are also defined as emergent (Shirky, 2008). As Terzidis states, "even though, physically, computers may appear to be a set of mindless connection, at the information level they are only a means of channeling mathematical and logical procedures" (Terzidis, 2006).

This elusiveness is the agent that allows for the understanding of humanism in appliance architecture. The 'otherness', or elusiveness, is a system operating outside the human comprehension while still provoking feelings and empathy upon the inanimate. This idea is described by Masahiro Mori's 'uncanny valley.' Humans will empathize with an anthropomorphic design which presents human qualities and emotions up to certain point, and then typically at the moment when an object is on the brink of being perceived as other or same (i.e. human life), the observer will respond with repulsion (Parkes, 2008). Although Mori's 'uncanny valley' is more aptly suited to media studies related to film and video games, it does provide a proof of the importance of 'otherness' through falsification of the same. Within Appliance architecture, 'otherness' takes on many types of logic: the algorithmic logic, the mechanical logic, and the behavioral or material logic. To understand how 'otherness' is relayed and transcends the physicality of the appliance we must address the subassemblies of appliance architecture

The Subassemblies of Appliance Architecture

The new media culture and economies of scale have provided a rich soil to cultivate highly complex interactive architectural systems within appliance architecture. With this complexity, there is a need to fully comprehend the subassemblies embedded within these systems. These elements produce the capacity for an appliance to receive sensory data, process the data and perform a given kinetic motion. In order to understand how these criteria are relayed, we must understand the logic system or agents used to generate a bottom-up system that produces 'otherness'. We must also ask how these systems move past response systems into interactive system, which allows users to effectively augment a system's logic by using a series of feedback loops, or relays, between the user and

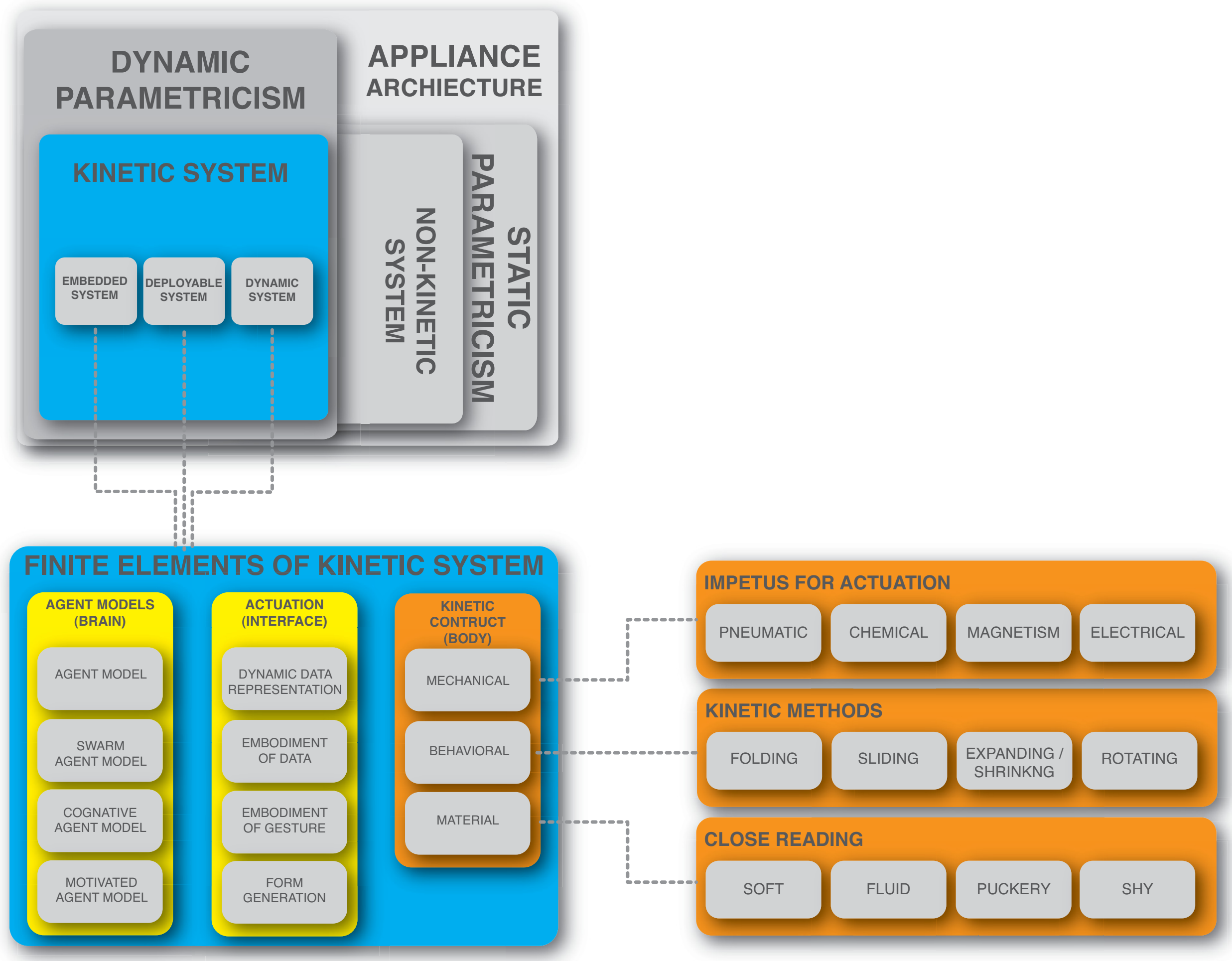


Figure 1 diagram of the subassemblies of appliance architecture.

the appliance architecture. Although appliance architecture (and dynamic Parametricism) is in its youth, the spectrum of its capabilities is vast. The appliance provides the understanding that these systems are a subassembly within a building system. Scales of design can range from toys to wall assemblies to roofing systems to entire buildings. The appliance's ability to condition environments presents a bifurcated potential within architecture research; sense-data implemented in these kinetic structures can dynamically affect a building system's environment conditioning and energy consumption. The majority of studies in architectural sustainability are focused on integrated building systems, such as HVAC systems, which means there is great potential in the study of alternative spatial conditioning systems (Fox & Kemp, 2009). However, in order to research within a finite scope, issues of sustainability and environmental response will have to be explored relative to sense-data potential and will only be alluded to within each appliance.

The research of appliance architecture within this body of work focuses only on those systems which can be defined as kinetic, or as having physical motion. Although heavily embedded within the new media culture and a society that calls for rich emergence in all media types, appliance architecture seeks to ensure a physicality.

The appliance's physical relationship to building systems is understood within three kinetic system typologies, which are defined by Michael Fox:

Embedded Kinetic Structures

> Embedded kinetic structures are systems that exist within a larger architectural whole in a fixed location. The primary function of this type of kinetic structure is to control the larger architectural system or building, in response to change (Fox & Kemp, 2009).

Few structural systems have been designed within this category. Examples of embedded kinetic structures include: autonomous HVAC systems and potential examples range from Diller Scofidio + Renfro's Blur Pavilion for the Swiss National Expo in 2002, Kas Oosterhuis's [ONL] Muscle NSA exhibited in Paris France in 2003 and the theoretical work of Tristan Sterk at the Office for Robotic Media & Bureau for Responsive Architecture [ORMBRA].

Dynamic Kinetic Structures

> Dynamic kinetic structures are clearly the most recognizable category of kinetic systems in architecture. This typology also exists within a larger architectural whole but acts independently with respect to control of the larger context (Fox & Kemp, 2009).

Examples of dynamic kinetic structures include: Santiago Calatrava's Brise Soleil at the Milwaukee Art Museums in Milwaukee, Wisconsin, Jean Nouvel's Arab World Institute in Paris, France and Mark Goulthrope of dECOi's HypoSurface.

Deployable Kinetic Structures

> [Deployable kinetic structures] typically exist in a temporary location and are easily transportable. These kinetic structures posses the inherent capability to be constructed and deconstructed in reverse (Fox & Kemp, 2009).

Examples of deployable kinetic structures can be understood in context of art's instillations, such as Andrea Zittel's Homestead Units and in the context of consumer products such as, camping tents and Airstream trailers

Having understood the potential appliance's scale of application within a building system, we can now evaluate these systems at more finite scale. Motion constructs, as described by Amanda Parkes of the Tangible Media Group at Massachusetts Institute of Technology's (MIT) Media Lab, are the physical and tangible architectural elements of an interactive system. These elements are the physical perception of the appliance and are classified as:

Mechanical

The physical and spatial design of how the motion is created; what Fox describes as the "means," or impetus for action. Examples include: pneumatics, chemicals, magnetism and electrical systems. (Fox & Kemp, 2009) (Parkes, 2008).

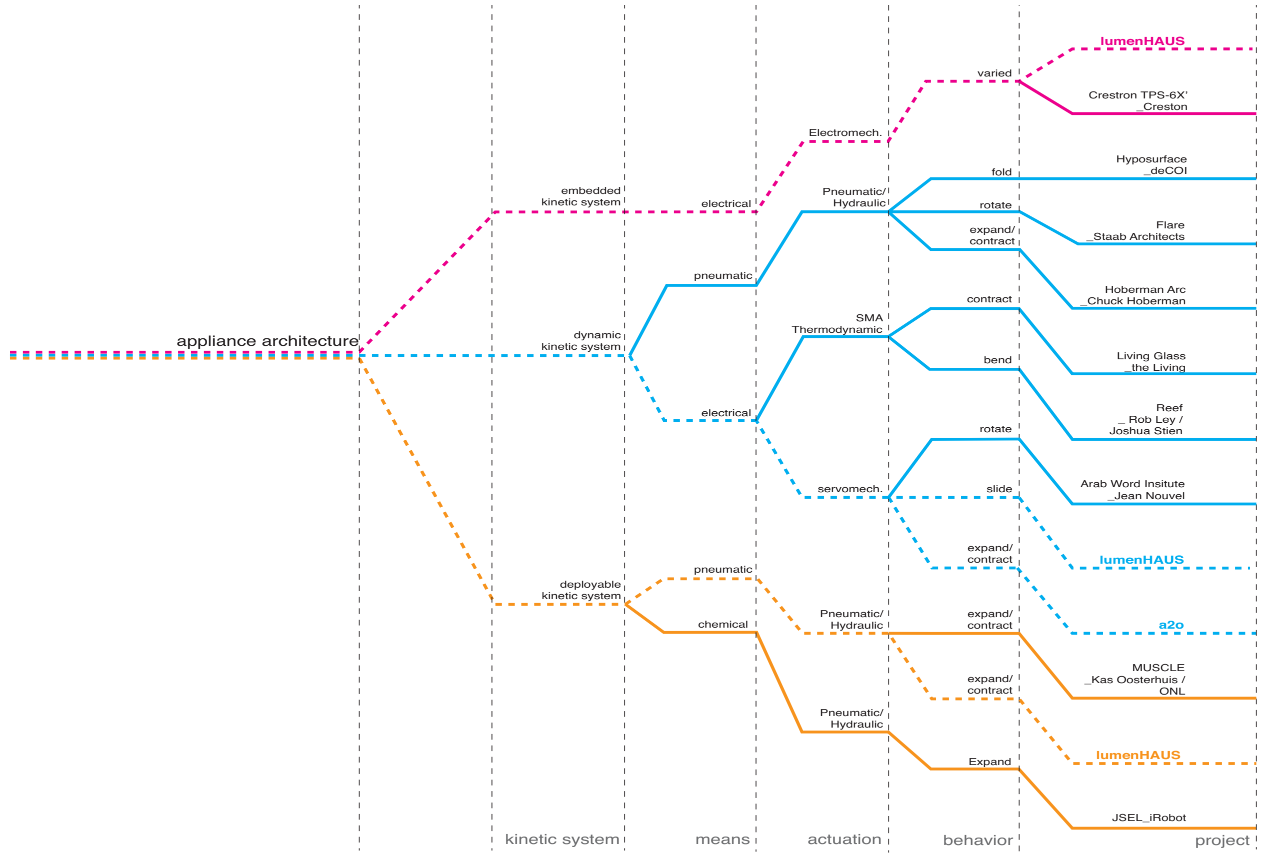


Figure 2 phylogenetic diagram of selected works within appliance architecture. The diagram shows the phylogenesis of the Lumenhaus and a2o case studies as well as the frequency of kinetic systems within appliance architecture.

Behavioral

The tectonic structure of motion; what Fox describes as the "way," or performance of the kinetic system. Examples include: folding, sliding, expanding / contracting and transforming (Fox & Kemp, 2009)(Parkes, 2008).

Material

The physical qualities of a medium in which the kinetic motion is actuated. This medium affects the perceived nature or 'close reading' of the motion (Parkes, 2008). Examples include, fluid, soft, saturated and puckery.

Lastly, we explore the behavioral logic or 'otherness' of the appliance architecture. The behavioral logic is the digital "memory" stored on the processing unit of appliance architecture system. The behavior is designed under the context of Negroponte's desire to explore "humanism through machines". This logic does not simply describe the motion construct's reasoning for a response or physical relay, but rather the pieces / rules / logic through which the state of ‘otherness’ or feeling emerges– feeling, in this case, describing the emotional response of a user toward the architecture, i.e. empathy (Frens, Djajadningrat, & Overbeeke, 2003). The understanding of behavioral complexity is examined through ‘agent models’ which illustrate the subassemblies of systems able ‘to perceive their environment through sensory-data, reason about the data and affect the perceived environment’ (Maher & Merrick). Mary-Lou Maher and Kythryn Merrick describe three levels of complexity through their study of virtual environments and avatars in Second Life:

Agent Model

> An agent is a system that perceives its environment through sensors, reasons about its sensory input using some characteristic reasoning and acts upon the environment through effectors. (Maher & Merrick)

Agent models are the basic units of responsive environments. As an appliance, they could be understood as a simple occupancy sensor for a lighting system. An occupancy sensor within a room consisting of multiple sensors is not aware of the neighboring sensor, nor the status of the light. Therefore, one sensor produces a single response; the sensor detects a body, it sends a constant ‘on’ signal regardless of whether the light is on already or not.

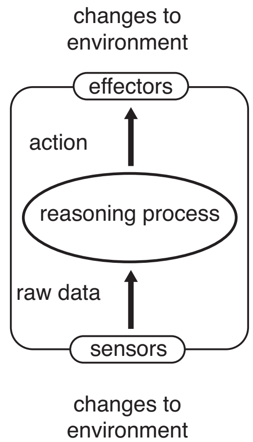
Swarm Agent Model

> Swarm intelligence is the property of a system whereby the collective behaviors of unsophisticated agents interacting locally with their environment cause coherent functional global patterns to emerge. (Maher & Merrick)

Swarm logic is one level higher in terms of complexity. Typically we understand swarms in relation to animal life: a swarm of bees, a flock of birds, or a school of fish. Swarm logic can be understood by considering an ant colony. For the most part, ants are a de-centralized community; however, they exhibit a collective intelligence. Ants communicate through pheromones, or chemical scents. If an ant finds food, it returns to the colony and leaves a pheromone trail along the way. Ants within the colony can then follow the trail to the source of food. Subsequent ants leave more trails that intensify the ‘smell’ for the next ant, up to the point when the food has been depleted, at which point no additional pheromones are left and the trail dissipates. What emerges is collective intelligence and patterning that leads straight to the desired source.

Cognitive Agent Model

> Like swarm agents, cognitive agents function as members of a society in which each agent controls exactly one object from the virtual world. Unlike swarm agents, cognitive agents sense the global state of their environment, that is, all the



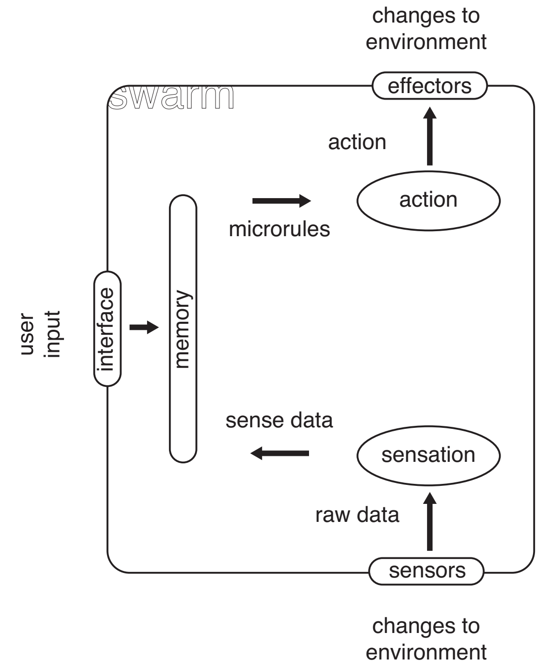
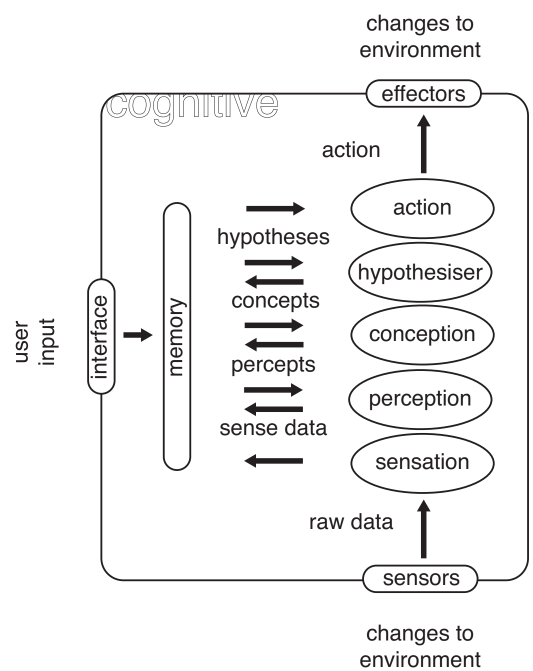


Figure 3 top left, agent model diagram adapted from Maher and Merrick.

Figure 4 top right, swarm agent model diagram adapted from Maher and Merrick.



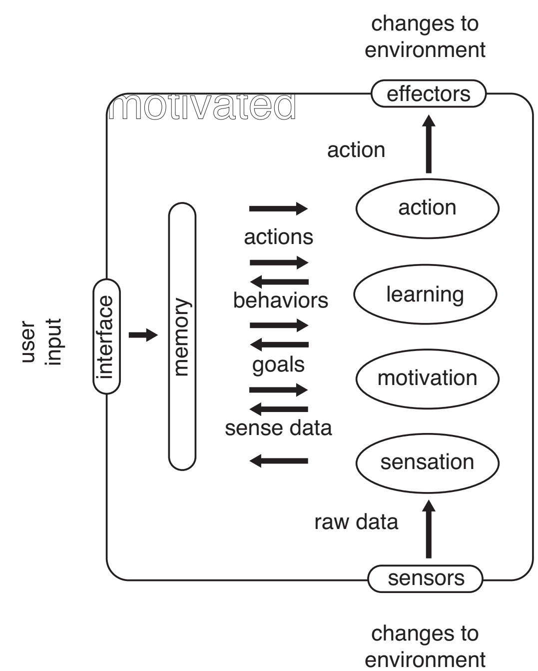


Figure 5 bottom left, cognitive agent model diagram adapted from Maher and Merrick.

Figure 6 bottom right, motivated agent model diagram adapted from Maher and Merrick.

> agents that fall within the bounds of the society. (Maher & Merrick)

Cognitive agent models are able to sense data in the environment, perceive or recognize patterns, understand a concept or rule of thumb, and perform an action from a hypothesis. Cognitive models are analogous to an intelligent traffic signal. These signals are able to sense how many cars are passing through the streets below. As more cars pass below, the signal can recognize that it is 'rush hour,' and apply a rule of thumb, reduce the amount of time a light stays red, in turn reducing the amount of traffic that backs up.

Motivated Agent Model

> A motivated agent model has the potential to achieve behavioral complexity without the need for domain specific rules. Motivation is that which gives purpose and direction to behaviors and is the driver that arouses an organism to action towards a desired goal… In place of influences, these agents are motivated to generate their own goals by identifying interesting events in their environment. In addition, they are motivated to solve their goals and encapsulate the knowledge acquired while solving goals a new behavior (Maher & Merrick)

Motivated agent model are by far the most complex and are on the verge of qualifying as artificial intelligence. These models can be understood as having a self-training mechanism that is similar in process to how a dog is trained through positive or negative heuristics. In the case of the dog, it learns by recognizing patterns; for example when the owner makes the sound 'shake' and the dog puts its paw in the owner's extended hand, the dog receives a treat. At first, this is what is described in motivated agent models as a 'rare-event' (Maher & Merrick). The 'rare-event' of getting a treat motivates the dog to learn how to get a treat again and therefore the dog will learn through positive reinforcement how to shake. The dog also exhibits the ability to encapsulate knowledge. Now that the dog knows that the treat is reinforcing its action, it will be motivated to repeat other actions in the future if a treat is received.

Conclusion

The question that must be asked is how and moreover, why is there a 'new re-emergence' of appliance architecture in relationship to those systems emerging in the 1960s? Unfortunately, as described by Tristan d'Estrée Sterk of ORAMBRA, the early study and development of interactive architecture struggled to find its foundation due to the architects' inability to construct the computational and structural systems needed to realize the vast complexity of interactive architecture. He states:

> It is from within these developments that feedback became a tool for use within architecture. As a mechanism, architects discovered that feedback could be incorporated directly into buildings via the use of responsive systems and that these systems would enable spaces and people to enter into a dynamic relationship. With this shift the tradition of modernism within architecture was slowly eroded, until in the mid 1960's, responsive systems became a favorite topic of the discipline. Unfortunately this movement was short lived. It came to an end in the mid 1970's as architects struggled to build the computational and structural systems needed to implement their new architectures. By the 1980's the idea for using responsive systems within buildings had completely transferred from architecture into the domain of engineering (Sterk, 2003).

As explained by Sterk, the study found residency in the fields of mechanical, electrical and structural engineering. The next section will explore the new media cultures as a means to this 're-emergence.'

THE NEW MEDIA CULTURE

In his article, Confronting the Challenges of Participatory Culture: Media Education for the 21st Century, Henry Jenkins provides this data:

> According to a 2005 study conducted by the Pew Internet and American Life project, more than one-half of all American teens—and 57 percent of teens who use the Internet—could be considered media creators. For the purpose of the study, a media creator is someone who created a blog or webpage, posted original artwork, photography, stories or videos online or remixed online content into their own new creations. Most have done two or more of these activities. One-third of teens share what they create online with others, 22 percent have their own websites, 19 percent blog, and 19 percent remix online content. (Jerkins, Purushotma, Clinton, Weigel, & Robison, 2006)

This data represents the new media culture in which appliance architecture resides. Within the new media culture, a major shift toward the decentralization of knowledge changes both the perception of who produces content and who possess authoritative view on content.   The result of this shift will be evaluated for its ability to support a re-emergence of fields of study related to appliance architecture and serve as a pedagogical framework for the invisible college discussed in the next section.

Clay Shirky describes 'Cognitive Surplus' as the resulting potential of free time generated by television media in a postindustrial age. The shift of the American life to the suburbs and the advent of the television in the late 1940s ushered in a media driven 'catharsis'. The result was an America (and world) with an abundant amount of free time spent consuming television – Shirky's modern 'Cognitive Surplus' (CS). In today's new media culture, the use of free time, or surplus, has shifted from a culture of media consumer to a culture of media producer whose new media moguls are communities such as, Facebook, YouTube and Twitter. This shift can be described, and will be referred to in this text, by many pseudonyms including: 'participatory culture,' 'user-generated content,' 'the people formally known as audience,' 'collective intelligence' and 'communities of practice.' Shirky measures this surplus in human potential for production relative to time. He evaluates the scale of this potential through a mathematical translations of the time people have spent editing Wikipedia in its entire existence – approximately one hundred million hours of human thought – to the amount of annual time spent watching Television. The annual consumption of television is two hundred billion hours, which translates to two thousand Wikipedia projects annually (Shirky, 2010). The consideration of CS presents two major points. First, there has been a substantial decentralization of knowledge from top-down owner/producers to bottom-up user/producers. Second, de-centralization questions who the experts are

(Shirky, 2010). For example, if one Wikipedia page is edited by thousands of peers in real-time, does that make it a more reliable source of information then a peer-reviewed article in a popular journal? But why does this matter to architecture?

'Cognitive Surplus,' or rather the resulting production of the new media culture, provides a bifurcated framework for the understanding of 'Appliance Architecture in the Invisible College: a Pedagogical Text'. First, CS provides one of the primary means by which architecture and design has seen a re-emergence of the subassemblies of appliance architecture. Second, CS provides the pedagogical means for teaching within the invisible college.

Participatory culture models reveal a new community of user-generated content that has appeared in the last two decades thanks to the emergence of the Internet and dramatically faster and more complex networking capabilities. Participatory cultures are described by Henry Jenkins in his article, Confronting the Challenges of Participatory Culture: Media Education for the 21st Century:

> A participatory culture is a culture with relatively low barriers to artistic expression and civic engagement, strong support for creating and sharing one's creations, and some type of informal mentorship whereby what is known by the most experienced is passed along to novices. A participatory culture is also one in which members believe their contributions matter, and feel some degree of social connection with one another (at the least they care what other people think about what they have created). (Jerkins, Purushotma, Clinton, Weigel, & Robison, 2006)

These communities describe intrinsically motivated participants / users who freely publish and contribute to a collective body of knowledge through their own desire for autonomy and competence (Shirky, 2010). The question remains, why is this important?

Over the last decade, design has seen a re-emergence of responsive / interactive architecture. Participatory cultures provide an outlet for learning and collaboration for those engaging in the research of themes related to appliance architecture. This can be understood through the programming software Processing. In 2001 Ben Fry and Casey Reas released Processing, a programming language and development environment for the visual arts. Fry and Reas developed the programming software while at MIT's Media Lab under the direction of John Maeda. The programming software uses the graphic capacity of Java programming with simplified and new features geared toward students, artists and design professionals. The importance of Processing is not necessarily that it is an easy to use and powerful programming language developed specifically for the visual arts (it is), but rather, that it is an open source software platform. Processing itself is the result of other open source software that provided guidance and components for Processing (Reas & Fry, 2007). When it was released Processing also unveiled the framework for an collaborative community. When the software launched, it brought with it an network geared toward creating a community of connected users and producers. This online community included tutorials, code banks, forums and 'wiki' pages where like-minded programmers of all skill levels could come together and share ideas and codes and continue to develop the Processing environment. Fry and Reas describe the Processing website as a 'communication hub' and describe other Processing-based websites' willingness to share source codes. Jared Tarbell's of Complexification.net states, 'Opening one's code is a beneficial practice for both the programmer and the community. I appreciate modifications and extensions of these algorithms(Reas & Fry, 2007). This willingness to share provides a rich soil for learning. Designers seeking to learn Processing are not required to create immediately, nor do they need to seek out a centralized learning point. Rather, they can begin were others have left off, learn from their peers and build upon them.

Open source software like Processing provides the necessary framework for a participatory culture. This framework is described by Elinor Ostrom's 'Mechanisms of Joint Governance of Commonly Accessible Resources.' Simply stated, by providing a free good that allows users themselves to access the source code, open source software produces a low hurdle for participation. Users are motivated to use the free code and make changes where the code is weak or to develop functions further. Because it is free, subsequent versions or refinement of code are markedly unsustainable for private sale (Shirky, 2010). Nowhere is this better explained then through the development of Arduino.

Processing, through an open source licensing, gave birth to two new platforms: Wiring in 2003 and Arduino in 2005. Both platforms were designed for artists to learn how to program microcontrollers—small computers with a single, integrated circuit containing a processor core, memory and programmable input/output peripherals. These platforms are an example of the means by which we see a re-emergence of responsive architecture (I have chosen to focus on Arduino due to its popularity in academic communities including the Architecture Association in London, UCLA's undergraduate studies in the Design | Media Arts program and New York University's graduate ITP program—all of which have produced valuable online communities that have contributed to this study. Arduino is also used extensively in PARTeE's prototyping).

Massimo Banzi and David Cuartielles founded Arduino in 2005 seeking to develop an open source, inexpensive prototyping system based on the processing language. Much like Processing, Arduino launched with an online community where users can contribute to, and borrow from, their peers. Arduino provided a platform for development that paired Do It Yourself (DIY) users with academics and professionals around the world.

These networks also provide a secondary aspect of open source software, known as 'off-the-shelf' hardware. The open source platform of Arduino combined with 'off-the-shelf' hardware allows for user-created coding 'libraries' for simple actions. The rotation of an 'off-the-shelf' servomechanism by 180 degrees can be accomplished with pulse width modulation [PWM], which requires relatively simple code. However, if one desires to establish a feedback loop between the servomechanism's controller and the Arduino microprocessor, this can become much more complicated – 800 lines of code more complicated. However, through Arduino's online community, tens of thousands of DIY users can access code 'libraries' for 'off-the-shelf' parts. These user generated 'libraries' offer multiple functions that can be called from a single line, for example myServo.Write (thisServo, 180). Where 'thisServo' is the digital output pin that is used to communicate with the servomechanism and 180 is the amount of rotation in degrees. Through this simple code beginners can begin program prototypes within minutes.

The use of 'off-the-shelf' hardware and circuits offers a similar framework to the use of open source software. In this case, the hardware is front-loaded with information. The use of 'off-the-shelf' hardware ensures consistency of products, specifications and results. More importantly, like Processing, Wiring and Arduino, the suppliers of 'off-the-shelf' hardware have produced information-rich online participatory cultures. Venders such as, Sparkfun.com, Parrallax.com, Robotshop.com and Allelectronics.com provide their consumers with outlets to become information producers. Product pages provide areas for product reviews and forums where consumers can find valuable codes, circuits diagram and links to experiments using the specific products. Sites like Sparkfun.com also include tutorials by staff members and customers. Programs such as Fritzing, an open source platform, allow users to design circuits through computer software that includes libraries of 'off the shelf' hardware. Users can import parts, design and share their project through Fritzing.

Arduino and 'off the shelf' hardware are only a small part of the re-emergence of responsive architecture. As stated earlier, the algorithmic design and software platforms of Parametricism contributed the greatest body of work toward this re-emergence. However, the development of Processing and Arduino, along with their substantial online communities stands to provide an example of the strength of the pedagogical framework for the invisible college

A BRIEF CASE STUDY

To investigate the scope of participatory cultures, an interview was conducted with Hiroshi Jacobs, founder of RevitCity.com, an online community designed for the free exchange of knowledge and digital material for Autodesk Revit. The website hosts forums where students and professionals can ask questions to the community and share ideas and processes. The site also hosts Revit-specific 'object,' 3D geometries or parametric families, job posting and related news.

The site represents the incredible growth of online communities within the architecture and design practices. It is a popular resource for users of Revit and is featured on Autodesk's website as an external resource for Revit. Its founding in 2003 also represents data and growth patterns that could be correlated with similar communities such as Processing.

Revit is a 3D modeling environment for building information modeling (BIM). BIM provides professionals in the field of architecture, engineering and construction (AEC) the ability to produce real-time, information-rich models considered to be designed in 4D (potential 'Nth'D). Revit supports 3D descriptive geometry that carries vast amounts of data about the object it represents in the real-built environment, including the objects construction phasing (time). Revit also represent the commercial AEC side of Parametricism. Users of Revit can establish parametric relationships between building objects and data. Users can design 'families' which are user-friendly parametric design versions of building objects. This allows for real-time adjustment within the 3D environment that proliferates through all aspects of the design documentation. A change to a parametric object within the 3D environment will change that objects geometry and information data in plans, elevations, sections, details and schedules in real-time.

The following interview was conducted with Hiroshi Jacobs, founder of RevitCity.com, on January 6th 2011.

H: Hiroshi Jacobs
J: Jonathan Grinham

J: *Can you explain the original impetus for developing RevitCity?*

H: I came up with the idea of starting RevitCity in 2002 as a senior at Tulane University after using Revit during the summer at an internship and in my studio projects. I was frustrated at the general lack of community support and content for Revit on the internet. When we started RevitCity, we thought the Revit community needed a website where people could communicate and upload/download content interactively and in real-time. There was one other community at the time (Revit Users Group International - RUGI) but despite having a very active forum, it was not interactive on the content side. We also thought that through advertisements we could eventually recoup the cost of our time and expenses in developing and maintaining RevitCity. If there was enough growth, some profit was also a possibility.

J: *How was the website environment developed and how did it evolve. i.e. what methods were used to establish and evolve an online community?*

H: We developed the website from scratch with some ideas about what a community of Revit users might need. I believe we started with a forum, a way to upload content, and a way to download content. In late 2006 we overhauled the design of the website and added all the additional features you see today.

J: *What methods were used to market RevitCity to members and industry members/sponsors?*

H: We started by uploading all the content I had personally created and the Autodesk created content. To launch the website we invited the members of RUGI to join RevitCity and also worked heavily on Search Engine Optimization to ensure that search engines would find RevitCity.

J: *What is the rational for a member based community?*

H: I believe member-based communities, especially related to technology, are very good at ensuring that advice given to novice members of the community is timely, accurate and valuable. Additionally, having many voices ensures more content and more responses to questions and comments. Despite the possibility for bad content or bad advice from a few, the vast majority

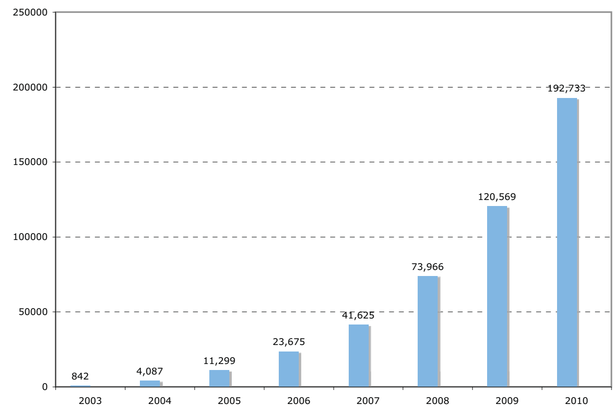


Figure 7 top, RevitCity.com total yearly membership.

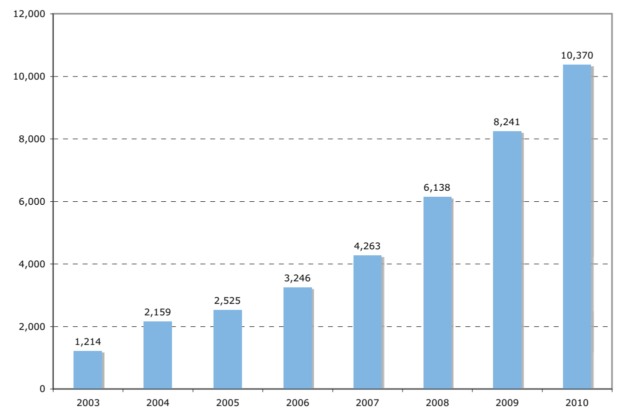


Figure 8 center, RevitCity.com total yearly forum posts.

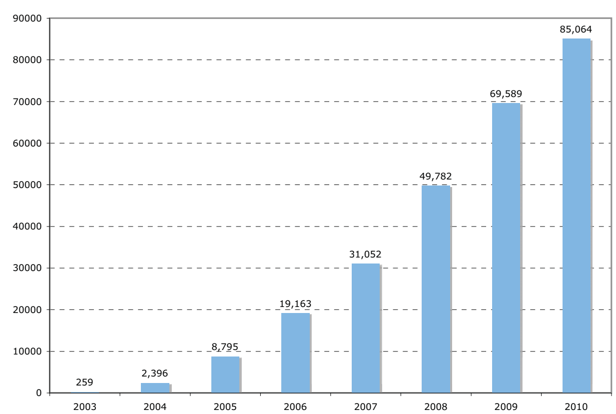


Figure 9 bottom, RevitCity.com total yearly hosted 3D objects.

of people are there for good reasons and willing to help. In my mind, the good outweighs the bad by far.

J: *What is the basis for membership ranking?*

Membership ranking is based on user contributions to the community. There are specific point values assigned to different types of contributions. The exact point values can be found in the FAQ section on RevitCity.

[END]

Hirohi's use of a membership-based community highlights an important motivation for user-generated content within online communities. Clay Shirky states, 'social motivations reinforce the personal ones; our new communications networks encourage membership and sharing, both of which are good in and of themselves, and they also provide support for autonomy and competence' (2010). The use of membership provides a rich intrinsic motivation for sharing. Membership approach to problem solving produces a shifting from, 'I did it' to 'we did it', resulting in direct social feedback and a sense of connectedness (Shirky, 2010). In RevitCity's case, membership ranking also serves as a way of substantiating user-generated content. Membership ranking is a peer-reviewed system that not only produces trust within a community, but also provides extrinsic motivation to share, thereby increasing user content. This increase is evidenced by RevitCity's growth.

The interview and the adjacent data shows the proliferation of online communities within the architecture and design practice. Within seven years, RevitCity has established 190,000 members who actively contribute a yearly average of 12,115 new forum posts and 1,300 new objects. This data serves to show the importance of user-generated content and knowledge, and begs the questions of how designers will collaborate in the future?

THE INVISIBLE COLLEGE

In 1465, two scientists named Robert Boyle and Robert Hooke and architect Christopher Wren established an academic institution that was unlike anything of its day. The institution was, in fact, not an institution, but rather a sort of 17th century proxy for our modern day online forum. The three men had established what they referred to as an 'Invisible College.' At the time, Boyle, Hooke and Wren were all located away from their homes in Oxford. The three members of this newly formed group sought to establish a peer-reviewed system of experiments in alchemy, a field of study which, at that time, was a closed system of experiments and findings without any documentation. As a result of their distance, the three scientists' only means of communication were written mail and an occasional physical meeting. These forms of communication insisted that all experiments were fully communicated and ensured experiments could be repeated for verification through falsification, stating, *Nullis in Verba*, 'Believe nothing from mere words' (Shirky, 2010). The three men's work ushered in some of the founding principles of the scientific method, moving alchemy to the science of chemistry we know today (with Robert Boyle considered to be the father of modern chemistry). Shirky states, 'by insisting on accuracy and transparency, and by sharing their assumptions and working methods with one another, the collegians had access to the group's collective knowledge and constituted a collaborative circle'(Shirky, 2010). This anecdotal example of the 'Invisible College' introduces the framework for learning the subassemblies of appliance architecture.

The invisible college provides an analysis of a pedagogical approach for learning the subassemblies of appliance architecture. As stated above, rich participatory cultures created by, and designed for, the fields of algorithmic design, physical computing and robotics have emerged in the last decade. Much like the relationship of the appliance as the 'site' for new, responsive architecture, new media cultures are the 'site' for our new invisible college. This new site requires multidisciplinary collaboration between designer, behavioral scientists, engineers, programmers and material specialists. However, in the case of the invisible college, any number of these participants can be a member of an online community. Therefore, the invisible college introduces a new educational lexicon of posting, googling and hacking.

The invisible College is not a pedagogical revolution. The importance of faculty to direct education is still relevant, if not more important. The invisible college is a pedagogy which explores the subassemblies, or low-level understanding of appliance architecture. As a part of a larger system, it fails to address the philosophy and critical theory of architectures that it represents. Also, the critical mass of knowledge communities produces a potential unchartable sea of information. Much like a design studio, academics do not dictate a project's program or practice; rather, they guide and shape its path. The de-centralizing of information requires that academics guide students through what Jerkins calls the 'transparency problem'—the inability for learners to perceive the truth or error of new media the are consuming (Jerkins, Purushotma, Clinton, Weigel, & Robison, 2006).

A bottom-up approach toward design and research is adopted. Like all forms of learning, appliance architecture requires small steps. The invisible college strives for emergent research typologies, which can be understood in relation to complexity versus simplicity:

> Complexity is a term used to denote the length of a description of a system or the amount of time required to create a system. While complexity may be a characteristic of many natural systems or processes, within the field of design the study of complexity is associated with artificial, synthetic, and human-made systems…There is often confusion between simplicity and significance. While simplicity may imply lack of sophistication, it also suggests abstraction, clarity, unpretentiousness, austerity and straightforwardness. In contrast, complexity is often regarded by theorists as indicative of sophistication, novelty, uniqueness, originality, and advancement. (Terzidis, 2006)

Within the study of appliance architecture, complex systems may be developed requiring little to no coding relying on sophisticated material and behavioral logic, such as Ned Kahn's wind powered facades. On the other hand, complex—used here to describe the length of code—programming may emerge to describe the same diagram, such as Mark Goulthrope's Hyposurface. Complexity and simplicity can be evaluated through the use of source code. Source code allows for inexperienced programmers to understand complexity at multiple levels by utilizing tools that they may not fully comprehend; however this allows for a learning method through mimicry— almost a trial by fire method.

FIVE POINTS OF THE INVISIBLE COLLEGE

The following points are an initial framework for the study of appliance architecture in the invisible college and should be understood as reflecting a moment in time within a constantly growing and evolving research. The points serve to provide bottom-up guidelines to the invisible college

Hacking: Learning Through Mimicry

Online participatory communities invite and exist through sharing. We will adopt a new type of learning—hacking. Hacking is commonly thought to be the illegal access of online information. In the invisible college, it is invited. So much so that Arduino has an entire directory on its webpage devoted to the hacking and is aptly titled 'hacking.' When a user clicks on this page and is redirected to its content, the page is renamed 'Extending and Developing Arduino.' Arduino is an open source software, therefore by inviting its users to hack its code and by making the software free, it motivates users to improve its code while removing the ability for capital gain from hacking. The invisible college (and related online communities) invites the sharing and borrowing of code as long as the source is properly credited if publicly released. In this way, logic statements or complex reactions can be borrowed from other developers and modified. The modifications allow for an understanding and learning of the code and an eventual ownership of the principles that are being developed.

Hacking may also refine the scope of a study; if a student wishes to accomplish a certain behavior, he or she can 'google' code for similar precedents. The source code found serves as a means of guiding the study instead of producing completely original code that may not produce a desired behavior. This methodology can result emergent typologies, what we will understand as learning scenarios. Each addition or subtraction to an existing code produces an evolution or de-evolution of a system, or a completely new species all together. Similar to the phylogenetic building studies conducted in the Foreign Office of Architects (FOA), the invisible college uses typology studies in order to correlate and understand the complexities of the subassemblies of appliance architecture.

Hacking also describes the act of crude design. The verb 'to hack' is define by Webster's Dictionary, as 'to cut or sever with repeated irregular or unskillful blows or to cut or shape by or as if by crude or ruthless stroke.' This definition allows for a more complex understanding of hacking as it relates to the pedagogical approach of the invisible college. Hacking, or unskillful manipulation, of code is invited. Hacking in this sense can provide gestural improvisational, if not accidental design. As Rem Koolhaus states, 'I think one of the important evolutions is that we no longer feel compulsively the need to argue, or to justify things on a kind of rational level. We are much more willing to admit that certain things are completely instinctive and others are really intellectual.' The computer sciences seek to produce concise and bug-free code. The invisible college, while concerned about proper structure and syntax at one level, is more attuned to the behavioral logic of a system. If a code produces a given response, it is successful, and can then be understood and debugged.

Prototyping also requires hacking. The proper use of tools produces effective results; however the invisible college suggests that one rethink, manipulate and hack the tool. Emergent design typologies can require crude and mechanistic proofs of concept. Like coding, if a physical construct produces a given response, it is successful, and can then be understood and debugged

Off The Shelf

'Off-the-shelf' hardware provide a low hurdle to participation and do not require the proverbial 'reinventing the wheel.' The use of circuits, transistors, and servomechanism that can be considered 'off-the-shelf,' increases the availability of online user-generated 'libraries' and source code for a designer. The use of libraries reduces the need for complex code and allows beginning level programmers a lower barrier to entry. Factory-made parts also ensure that a minimum level of quality and reliability are built into the mechanical behavior of a system. By using parts that are standard within DIY communities, we can insure that errors or unexpected results can be checked against user experiences.

'Off-the-shelf' parts aid physical hacking. The use of standardized 'off-the-shelf' hardware allows for a re-imagining of the tool, and the use of controlled subassemblies provides a control group by which experiments in 're-tooling' can be explored.

Simulation is Realization

Computational models allow for the representation of geometric descriptions of objects, which can carry a host of mathematical relationships. The use of parametric software allows designers to simulate architectures that statically or dynamically respond to given parameters and local conditions. These models are virtual landscapes that can analyze and simulate the complexity that arises from the aggregation of continuous differentiation. The invisible college establishes that simulation through Parametricism is realization. The same algorithmic logic that is used to describe nodes within virtual models can be proportionally applied to real-time dynamic systems. (The reverse, incidentally, is not always true. Beauty, for example, should be realized in kinetic systems even if they become non kinetic. For example, Jean Nouvel's World Arab Institute is no longer functional, but its parts still combine to produce a woven texture that evokes iconology of the Arabic culture.)

The universality of mathematics allows for the logic of one system to be translated from virtual environments to physical computing systems. Furthermore, the use of software such as Grasshopper allows for a direct translation. Through plug-ins including LiftArchitect's Firefly and internet hosted data networks such as Pechube, multi-loop systems can be established between once static parametric simulations and built environments using the same definitions that establish virtual designs.

Digital Versioning

Algorithmic logic also allows for digital fabrication and prototyping. The same intelligence applied to emotive systems constituting 'otherness' can be translated into the materiality and tectonic relationships of appliance architecture. Algorithmic design allows for design versioning, either through iterative processors or iterative parts within a whole. The evaluation of place-sensitive design criteria within the new media culture allows for 'mass customization,' as well as an understanding of the novel tectonics necessary for 'design for the masses.'

[Inter]disciplinary Blind Spot

Architects are spatial consultants in the invisible college. The complex subassemblies of appliance architecture require new synergies between computer science/engineering, mechanical/electrical engineering, behavior sciences and material sciences. The architect's role is to understand the capacity of each of these fields to generate space. For this reason, the architect must also posses the ability to manipulate, edit and construct elements of the subassemblies of appliance architecture.

Conclusion

The invisible college may be considered novel, but the new media culture's ability to both birth and foster appliance architecture cannot be denied. Much like the quickly evolving communities of participatory cultures, the invisible college necessitates adaptation and collaboration; as Clay Shirky states, 'the technology will continue to improve, and the population will continue to grow, but change in the direction of more participation has already happened. What matters most now is our imaginations' (2010).

interview with **ROB LEY** of the **REEF PROJECT**

Figure 10 previous pages, image of the Reef project at the Taubman Museum of Art in Roanoke, Va.

Figure 11 left, image of aggregation of 'fin' assemblies with Nitinol connection.

REEF PROJECT

There are two types of beauty, free beauty (pulchritudo vaga) and merely accessory beauty (pulchritudo adhaerens). Free beauty does not presuppose a concept of what the object is [meant] to be. Accessory beauty does presuppose such a concept, as well as the object's perfection in terms of that concept. The free kinds of beauty are called (self-subsistent) beauties of this or that thing. The other kind of beauty is accessory to a concept (i.e., it is conditioned beauty) and as such is attributed to objects that fall under the concept of a particular purpose (Kant, 1987)

The Reef series examines the role of architectural skins through emergent material technologies and behavior logic. The project was designed by Rob Ley, a principal at Urbana, and Josh Stien, a principal at Radical Craft, and was commissioned by the Store Front for Art and Architecture in New York City.

Ley and Stein's design and implementation of the Reef series express the potential of appliance architecture and the necessary research synergies for its creation—I use creation loosely and in reference to Ley's cat/dog analogy found below.

The design began as a bottom-up study inspired by a fascination with a new material technology, *Nitinol* (Nickel Titanium) or Muscle Wire, a heat sensitive shape memory alloy (SMA). The application of heat or electric current to the *Nitinol* alloy causes the wire's cross section to increase (expand), in turn constricting its length by about 5%. Ley and Stein conceived of an instillation that consist of 600 fiberglass 'fins' that are actuated by the *Nitinol* wire. The 'fin's' motion and geometry could be considered a result of the wires' limited constriction. Although the 5% constriction provides limited motion, that motion is amplified when applied to a larger surface, and in a sense the wire acts to shorten the arch segment of a circle tangential to the fin. The result is a fluid, amorphous movement similar to grass blowing in the wind or human fingers being opened and closed.

New synergies were necessary for the design and production of the Reef. Ley will later describe the relationship with Dynalloy, the manufacture of *Nitinol*, as a complete 'pledge' of support. This type of committed support is necessary in the research of advanced biometric material such as SMA. Ley and Stein also employed the expertise of Pylon Technical, a custom software developer. Pylon Technical's relationship is a rare synergy. Pylon Technical's ability to envision the needs and limitations of the project, while also producing a software—designed in Max/MSP, a visual programming language designed for music multimedia,—that could produce, and adapt to, the behavioral logic, is hard to find. Pylon Technical' experience also provided a new understanding of the Reef's behavior. The Reef was originally conceived of as 600 independently actuated fins; however, this design was too cumbersome and expensive. Instead, Pylon Technical envisioned the design as a series of pixels or cluster of fins. In this way, the field consisted of 48 responsive pixels that would actuate 12 fins in unison.

The Reef's form and dynamic space packing resulted in a woven fabric in which the boundaries of each pixel could not be determined. The fins were parametrically designed in MEL (Maya Embedded Language), a programming language for Autodesk's 3D modeling environment, Maya. Each of the 600 fins had a unique geometry that could be understood as a type within a family that was in reference to its neighboring 'brothers and sisters.' The use of CNC laser cutting allowed for the fabrication of each of the unique geometries. This use of design versioning allowed for each piece to embody a slightly unique behavior in relation to its neighbors, who were receiving the same amount of actuation. The result was an emergent and amorphous behavior that is both subtle and provocative.

Pylon Technical's interaction also allowed for the evolution of behavioral logic. The original software developed for the instillation at the Store Front for Art and Architecture was an elegant system that allowed Ley and Stein to prescribe and orchestrate the Reef's movement. The 48 pixels were unrolled within the Max/MSP environment and mapped to a two-dimensional environment. The system was a literal paint-by-number orchestration. Ley and Stein could select each pixel and tell it how long and when to actuate—imagine playing the piano and each key is a pixel of the Reef. Patterns could then be stored and looped on top of one another—now add the rest of the orchestra. The result is complex, understated and emergent orchestration of movement.

For the instillation at the Taubman Museum of Art in Roanoke, Virginia, the designers added a new layer of complexity in the form of a responsive behavior. For this instillation, Pylon Technical developed a motion tracking software (also designed in Max/MSP with Java modules). Pylon Technical used a monochromatic video image from camera placed above and away from Reef. The camera's image was fed to the Max/MSP environment, which tracked the motion of visitors by comparing the image frame-by-frame. The visitor proximity to the reef could also be determined by re-mapping the spherical distortion of the video camera into a two-dimensional coordinate system within the Max/MSP environment. As visitors passed by, the adjacent pixels would be actuated—extended out. The two-dimensional coordinate system allowed for the visitors' proximity to be proportionally mapped to the scale of the fin's motion. The adapted program also carried traces of its past. If no visitors were present for a given period of time, the Reef would return to the preprogrammed response of the orchestration software—what Ley will later describe as a cat-like interaction.

Figure 12 image of pixel clusters. Each colored tag represents a single fin within a uniformally actuated pixel. Image by Spencer Lee

The following interview was conducted on February 28th of 2010 at Mill Mountain Coffee in Roanoke, Virginia. The PARTeE team had the opportunity to meet with Rob Ley while assisting with the instillation of the Reef project at the Taubman Museum.

Rob Ley is the principal at Urbana, an architecture, design and fabrication studio in Los Angeles California. Urbana's work explores themes in generative design, responsive environments, emergent technologies and cross-contamination of research.

R: Rob Ley of Urbana
J: Jonathan Grinham
N: Negar Kalantar

J:
The question I have, in order to frame the conversation, is do you see the whole project as top down or bottom up?

R:
I'd say definitely bottom up.

J:
In that case can you elaborate about the initial studies in terms of the technology going into it [Reef] in terms of fabrication and materiality?

R:
Well, I think probably bottom up, in that my sense of top down would be a traditional idea like someone like a Frank Gehry way of designing where you have an overarching form and then you have a team of people that work on rationalizing the geometry—figuring out can we afford this, if we change this can we afford it even better. How do we get curvy shapes out of flat panels? It's actually kind of interesting top down approach, but they know what the thing is going to look like from day one and then they spend all the time trying figure out how to make it happen. Verses what we did, why I would call it bottom up, as you a saw in the [lecture], **when you are starting with technologies or materials in the beginning point and you are just trying to get little things to dance or do things, we didn't have any particular idea of what the thing would look like in the end.** We knew it would be about aggregation and so in some cases that are really fluid, it can be anything, right. Humans, organism are aggregations of billions of cells. You can make anything with a cell if you have enough of them.

It is really nice if something beautiful, but I am not sure if you have to start with making it beautiful first. I think you can inject it with behavioral quality, something you want it to be verses something you want it to look like. Because making it look like something is easy, that's kind of boring. I've had a lot of experience with 3Dsoftware, basically, every 3D software I know and I've used quite a bit. But I just got quite bored making shapes all the time. It is more interesting to have something with either an intelligence or in this case, this project, something that is more of a behavior. **It looks or moves like an animal—a little bit. So that when it turns into a psychological issue with people, how do they respond to it? I think it becomes something that is hard to do with form, because form just becomes beautiful- either beautiful or ugly there's not that much of a difference.** Scale can be awesome and form can be beautiful, but what else can architecture have? It can have more than then. So I think that is what we are trying to do.

J:
Can you go into some of the technologies that lead you to that [where you are], in terms of the Nitinol and fiberglass?

R:
One thing I like about materials it that when you start to study them, particularly biological materials you see that bones will get more dense at the site where the stress is. We started looking at composites. One of the interesting things about composites is, besides just saying it's fiberglass or it's carbon fiber, you can, by how many layers you put in one area verses another, you can have a shape change it's strength all through out the shape. So when we came across the Nitinol two things happened. One, we saw in all the little toys, some of the ones we showed last night, this really cheap twelve dollar little toys could be really—you would stair at them like you would stare at fire—you know how some people stare at fire because it's moving in a certain way? And I think we just became committed to trying to change the scale of the application into something architectural. We went and met with the company that makes it called, Dynalloy, and just sat in their board room and showed them these little models we made and

they got really excited. They were basically so sick of making just he little tiny things that when I said 'oh were are going to make buildings out of this stuff' the owner loved the idea, because he loved the idea of selling 200 times as much of his material, because building are really big. But, the engineers were excited to work on something beside little novelty toys. They pledged support, saying we'll help you guys with the engineering if you need it and we'll give you material if you need it. So that kind of started the project.

That's the mechanical part of the technology. The electronic part, that's something—I understand mechanical things pretty well, but electronic not at all. I have to still ask [Pylon Technical] how we are supposed to wire things I get confused between parallel and serial sometimes. So that about- even though it's very simple—**I think that's one area where I think you have to ask people to help, but it is really important to understand the logic behind what you want it to do, because those people tend to not be able to - it is very rare when you find someone technically proficient and creative and can suggest possibilities**. [Pylon Technical] is very good with at a lot of the things he does and he's actually one of the exceptions where once he gets into he can have an idea of maybe what happens. I think we deal with, at least with the electrical part of the technology, I think we deal a little bit more with the philosophy behind it. Why it's going to do something a certain way? What's the difference between intelligence and behavior? And how do you get something electronic to exhibit the difference between those two things?

J:
Can you elaborate a little bit more about the collaboration with [Pylon Technical], and maybe the feedback loop, not only between you and [Pylon Technical] and the actual development programming, but also the feedback for the Reef project between New York [Store Front for Art and Architecture] and here [Taubman Museum of Art] and the change from a prescribed movement to a responsive system? What was the feedback to get to there and the collaboration with [Pylon Technical] to get to that point?

R:
Ok I don't know if this will answer your question or not, but I think before we started working with [Pylon Technical] we had this idea that the thing would be directly responsive, every component would be like a pixel on a screen and you could have complete control over each pixel. We started realizing, working with him, he started telling us certain limitations that the hardware was going to have, that the software was going to have, based on what our budget was. **As always the limitations are one of the best things that can happen to you design wise.** When you have a huge budget you can do anything. It's kind of not as fun? Long story short what we started realizing was the thing didn't always respond in the way we wanted it to - for a number of reasons. One, there would be glitches or we weren't able to have every fin be on it's own pixel- we would have to cluster them together. And sometimes there would be unexpected delays and one of the things that came out of that, in terms of feedback, is the idea of responsiveness, or something being interactive doesn't need to be immediately responsive. Maybe it shouldn't be predictable necessarily in the way it is responsive

The best way I can [explain it]. **It would be the differences between some people are really into dogs, and some people are really into cats. Dogs, when you walk in the door, you know what they are going to do. If you have been gone all day they are going to come and jump on you and want to be around you. Whereas cats kind of look at you and go back to what they are doing because they have their own thing going on. Sometimes they want to be loved but a lot of time they could care less if you are gone, or if you come back. In a way we like the sassiness of that model.** Where the things could respond to you, but not in the way you would get bored with it in a couple minutes.

One thing you start [realizing] - maybe you are in the process – we've been pretty aware of a lot of other interactive projects - people who are working around the world. And they always kind of do the same thing. Honestly, they are kind of doing what ours is doing on Thursday, which is there is always this…. Have you guys ever read Benjamin? The philosopher, sort of about the awareness- well I wont get to deep – let's look at it another way. **When a child is two years old if you put it in front of a mirror they don't know it is themselves**

Figure 13 image of physical coordinate system. In order to map the spherical distortion of the video camera a matrix of 'Xs' was physically mapped on the floor of the instillation. The video of these nodes was then remapped to a Cartesian grid within Max / MSP to provide a measurable coordinate system. Image by Spencer Lee

looking back. And there is a certain point developmentally in a human mind when they realize, ok that's me and there is this fundamental shift. And all of the interactive projects we have come across are always stuck in that mode. At that moment you figure it out—they behave like a mirror. There's this guy named Daniel Rozen, this guy who makes these mechanical mirrors. It's really compelling, you should look at his work, they are pieces of trash or spheres that go in and out. It's really cool because at that point, it's an abstract mirror, which I think is really compelling. But it literally just does, it's feed of a video camera that looks at you and recreates your face back again. But really that's what most [interactive pieces do]. I think the best part of his, is the pixel, the novelty of the pixel, whether it is a chrome sphere that goes in and out or a garbage- there's one that is just pieces of trash. That's the clever part, not the face, *because that's pretty dumb.*

I think what we'll want to work on, it won't happen on this show so much because you can see [Pylon Technical] is still working on getting everything connected. But it's like everything, once everything is working the way you want it to work then you can start to explore other things. So, things like how it behaves, **why it would behave differently for a child verses a grown up or why someone with a blue shirt would have a different response then someone in a yellow shirt? Totally arbitrary - so the thing could become racist - it could not like you for reasons you don't even know and then you would start to feel jealous or insecure.** That's just a funny little thing, but it's something architecture has never done before, which is to elicit other emotions in you.

N:
If you had a chance to redesign, what part would you modify?

R:
I think we're probably going to go two different directions. One is going to be making something that is smaller, that can fit into a crate and be shipped somewhere else and that's just a pragmatic thing. We can't keep flying around and installing. It just takes a whole week out of your life. It would be nice to have something that could show up somewhere and kind of unfold and then operate. And the other side of it is I would like to do something that is more spatial. I still see this [Reef] as big object more then something architectural.

It's the way a Richard Sera sculpture is kind of architectural, just because it's big enough, but realistically it is still just a sculpture, it is still just an object. That's what that is—It's still just kind of an object.

I think the next step; the canopy model I started talking about on the way over here, is one way of trying to create something that is more spatial in the way it operates. So the relationship between the person and it is an architectural one not like an artist, like in terms of a piece of art where you stand next to it and look at

J:
You spoke about the grants, the secondary applications, such as sustainability. In the context of what we [PARTeE] are working on, do you see an educational aspect to this for children?

R:
[Laughing] What do you mean for children, is your guys' grant to educate kids?

J:
It is, the grant is geared toward collaboration with teachers that will use them as an educational tool in their curriculum - with grade school and high school teachers. So there is that aspect, how do we use them with multiple ages? So how do these things relate to teaching a grade-schooler, specifically, about science and math?

R:
You are still younger then I am, but lets put it this way. You guys were, what, teenagers when the internet came out? Imagine being three years old and you are using the internet when you are three or even younger, right? **There is an entire wave of people growing up right now that have gotten used to things being interactive. They get what they want. You click on a link it shows up, it changes, if you want to go somewhere else, that happens. If you get bored of that you go play Xbox for a while. There is an expectation level that these people, who are going to be adults pretty soon, have that maybe architecture needs to ready for that.** Like I said last night, it's really boring for architecture to just be a cave—it is still just a cave. You got warm when you walked inside, that's kind of nice and people don't steal your stuff as easily, but it has been like that for hundreds of years, it's getting a bit, maybe time for it.

There's a photo, you guys should look at this, there is a photo of the Villa Savoye with this Ford Model 'T' parked in front or Model 'A' – the second car that was made. It is bizarre because it is one of the only times I have ever seen a photo where the building is more contemporary –has higher technology and is more kind of on the pulse of something current then a car is—and it's never been like that ever since. Some how cars became really technologically advanced and building stayed exactly the same. So I think, maybe, there hasn't been a need for it, but if the public at large, maybe, sees the potential for building that are much more responsive —whatever that might mean—I think that's probably really good. So I think that's probably a big part of it, is their expectation and seeing buildings as performative devices, I don't want to say machines, but performative spaces that have to do stuff.

N:
At the scale of architecture should it respond to the environment or the user? Because at this scale it reacts to the user. What about at the scale architecture? I think we cannot apply it to the whole architecture, maybe some part, what is your idea in architecture, is it shading device, is it a canopy?

R:
I think probably, to understand your question; I think the answer would be change what you expect the building to do. **If you see it more as an interface to the environment - there is a filter between you and the world – the environment. That becomes a lot more interesting.** Even that space in the museum. In the summer time people probably don't like to sit by the windows because it is just hot? But in the wintertime it's nice to be in there because it is nice and warm. So we want the thing to keep changing actually, if you think about it, we are always opening windows and closing doors. You want your interface to update and change. So I would see it, **if you look at is as an interface sometimes you want the architecture to just go away entirely, other times you may want it to come back and there maybe gradients in between.**

N:
Most of your designs, even the wall, that you did for a private client that was a small office or home it changed according to the needs of the client, or users. But it doesn't react to the environment, most of the are inside, some component inside the building, are you going to expand to the outside of the building?

R:
Yeah, absolutely, you know the thing that is consistent with all those projects is they are still just one system. So rather then having four different systems in four separate things, it is generally an investigation in how do you create one system that has enough gradient in it be able to do all the different things it needs to? So it is just a way of minimizing, what the architecture is in terms of content, but maximizing the degree of flexibility it has with a singular system.

If you are asking me if it can respond to the environment I suppose it can. But I think it seems like the question is what is more important? I don't see it as one or the other. I am saying it is like you are here and the environment is here and something is in between and it's really just what side; do you see architecture as an object to look at or do you see it as something that is between you and the sun or you and the rain or you and bears outside or people that want to rob from you? It is just an interface.

I have my reservation against the environmental movement actually. Were we talking about this yesterday? **What do you do? When you make a building that can use twenty percent less energy then what do you do? Make it twenty-two less? Or why should people suffer, or their life style get worse so that the environment should get worse at a slower rate? I think it is a flawed system**. There is this guy; if you really want to read an interesting book there is a guy [William] Mcdonough, Craddle to Grave… Craddle, Craddle to Craddle. That is how it should be. Basically, he makes a comment in it that says that it is a flawed system when the way you want to fix things is to do things less bad. It just won't work. It works for a while but then your population growth is greater, exponentially, then the amount of energy you are reducing with that method. So it is flawed. You get a little bit of a return, say the whole world gets a little bit of return for a little while, then people - China and India - keeps doubling. So the problem gets worse again, so it is really like rethink entirely, like looking for a third way of doing it. So for example, one of the examples he had, which I think is kind of a clever one, is if we walked outside and I took my coffee cup and just threw it on the ground, you guys would probably look at me like I'm a jerk, right, because we are taught that you don't litter. One his examples were what if you make cups that not only was biodegradable, but inside the cup it had little flower seeds all over it. To where you would encourage people to throw the trash on the ground, because it would dissolve in a couple days and it would turn into flowers. Or you chuck it out the window of your car when you are driving. It is silly, but it does kind of point to something. First it sounds absurd because you have been trained for, decades, not to do that. It really has to do with what the material is and how it is constituted. Or even, think of it this way, what if your house was solar? Right, I'm not saying all these things are bad. Lets say you are off the grid and your house was sustainable you could leave your lights on all the time and you could leave hot water running all the time—right? Image, not ever turning your shower off just because you don't need to. Right, it's the water, it's the sun heating the water and it is a pump circulating it back around again. You never turn your lights off in your house. That seems odd, really bizarre to use, but there is no reason that can't be the case as long as things are set up in a particular way, your life still doesn't get worse.

J:
you made a really nice statement about, when you were discussing the roll of the designer and architect when it comes to these things [green design] because they get so lost. Your were mentioning such a highly engineered architecture - twenty percent to twenty two percent - then you have this mass replicatable piece of architecture, then what is the roll of the architect, where does the architect project the human aspect of it, or beauty.

R:
You guys work on the solar house? I think it is completely reasonable and acceptable to throw solar cells and windmills on buildings and say this is how we are going to solve things, but I think that it is a way of using devices and technology to solve a problem rather then other more interesting ways. Like, thermal, heat chimney effect, right, this thing, hot air rises, imagine just basing a new architecture on a more firm grasp of physics and a formal vocabulary that understands current flows and the kind of beautiful cities you could develop based on that.

Or we went up and talked with [a 4th year studio] for a little while and they were working on these Japanese metabolist projects, and they had these things kind of floating in the water and I was just thinking why don't you push the city down into the water and put a ring - do you guys ever see how they build bridges, how they put the pylons in the water - do you know how they do that? They take this huge cylinder and they push it down into the water and pump all of the water out so that the guys can- basically the river is flowing all around them and it's a dry cylinder so they can do all the work. Imagine doing that with a city and pushing it out so that you have this huge thermal mass, which is the water around it, so that you have, you know water is a great thermal buffer. You could have that and then biological energy sources, like algae, all around the parameter. It completely changes the form of a city in a really cool way. That's way more interesting then making bigger windmills and making bigger solar cells, or hoping scientist can squeeze three more percent of power out of silicon wafers, that's really boring. That's why I am against is, a little bit. Because I think we are putting all of our eggs into technology and we are hoping scientist can save us. I think that is too much responsibility for those guys, they can't do it, they cant do it fast enough.

What else?

[END]

Figure 14 image of aggregation of 'fin' assemblies depicting the unique geometry of individual motion constructs


lumenHAUS


CÎROC
ULTRA PREMIUM VODKA
AMERICAN EAGLE OUTFITTERS
THE OFFICIAL
VODKA OF
NEW YEAR'S EVE

Figure 15 previous page, image of south elevation of the LumenHAUS on the National Mall in Washington, DC. Image by Dept. of Energy Solar Decathlon.

Figure 16 left, aerial image of 2009 Solar Decathlon Village on the National Mall in Washington, DC. Image by Dept. of Energy Solar Decathlon

Figure 17 center, image of LumenHAUS on Time Square in New York, NY. Image by Rober Dunay

Figure 18 right, aerial image of LumenHAUS at the 2010 European Union Solar Decathlon in Madrid, Spain. Image by Dept. of Energy Solar Decathlon.

LUMENHAUS + ECLIPSIS SCREEN

"if the architecture that excites today had to do with investigation qualities of translucency, screening, and the creation of overlapping spaces and visions, this might indicate a sociological trend more than a formal one: with the dissolution of traditional space-defining elements, we are becoming more sensitive in perceiving subtle indications of territorial definition..." (Maki, 2008)

The Lumenhaus project presents a unique opportunity to understand appliance architecture at multiple scales and subassemblies. Designed for the Department of Energy's 2009 Solar Decathlon in Washington, DC and over-all winner of the European Union's 2010 Solar Decathlon in Madrid, Spain, the Lumenhaus represents an architecture that seamlessly embodies the three subassemblies of Michael Fox's kinetic system typologies. The project approaches energy conservation and sustainability through a narrative of responsive architecture. Designed under the precedent of Mies Van Der Rohe's Farnsworth house, the Lumenhaus challenges formal norms of energy efficiency and sustainability by seeking to produce an environmentally responsive glass pavilion.

The following section explores the Lumenhaus through Michael Fox's kinetics system typologies. The research focuses on this author's intimate relationship with the parametric design of the shading screens for the Eclipsis System.

DEPLOYABLE KINETIC SYSTEM

To understand deployable kinetic systems in relation to the Lumenhaus, we must look to the *Parti* established by Mies Van Der Rohe's Farnsworth house. Mies is widely considered the master of Modernism and lived by the motto of 'less is more.' The material composition of his open floor plans is considered to by a style of 'skin and bones' consisting of exposed industrial steel and plate glass. These elements are the core design that allows the Lumenhaus to become a deployable architecture.

The house incorporates an innovative transportation system allowing for an all-most completely 'plug-and-play' architecture that requires minimal site work. The 2009 house uses a structurally incorporated double-drop lowboy which was conceived and implement in Virginia Tech's 2005 Solar House, and which allows the house to travel with sufficient overhead and under-carriage clearance while maintaining a voluminous mass to design within. The chassis serves as both the floor of house and a structural bridge that connects to both a removable rear axle assembly and a removable gooseneck connection to the cab. The result is a movable bridge that spans inboard of the gooseneck and the rear wheel assembly. For normal static site use, the steel structure of the house is designed as an inhabitable Vierendeel truss supported by pile jacks at each column. For transportation, the Solar Team, with engineers at ARUP, envisioned removable diagonal bracing that works to reduce deflection and vibrations during travel. The result is a fully deployable architecture with reusable transportation connections.

The true effectiveness of the house at becoming a deployable kinetic system can be measured in the thousands of miles it has traveled. To this date, the house has traveled from its home in Blacksburg, Virginia to Washington DC, New York City, Madrid Spain, Chicago, Illinois and Plano Illinois where it now sits next to its 1951 inspiration, the Farnsworth house.

Figure 19 image of LumenHAUS transportation system. Image by Jim Stroup


VirginiaTech
Invent the Future
92





DYNAMIC KINETIC SYSTEM

The Vierendeel truss of the dynamic transportation system results in, or is the result of, an open pavilion design. Returning back to Meis's Farnsworth House, the Lumenhaus sought to produce an architecture of contradiction—an energy efficient glass house. To make this possible, a sliding, four-layer wall system, called the Eclipsis System, was designed for the north and south facades.

The Eclipsis System consists of a four-layer dynamic kinetic system. The layers of this system can be manipulated in order to alter the energy balance of the building and to optimize human thermal comfort, economy of operation, or a balance of the two. The first two layers, which are the stainless steel shade screen and the Aerogel-filled, polycarbonate insulating panels, are fully operable autonomous systems able to slide laterally through the use of precision brushless servomotors and a belt driven track. Aerogel is a lightweight and semi-translucent expanded silica with high thermal properties, and the panels that are filled with it offer a thermal resistance of R24. It achieves this through the use of two layers of two-inch Aerogel-filled polycarbonate with a two-inch air gap in between. The air gap subsequently allows for the illumination of the Aerogel-filled polycarbonate by means of embed RBG LEDs. The shifting hues of the LEDs produce a dynamic light-scape throughout the space, able to respond and actively effect the users physiological well-being (Simeonova, 2010). (These layers are linked into the house's embedded kinetic system and will be explored in the following section.) The third and fourth layers, interior curtains and aluminum sliding glass door assembly are user-operated, adjustable systems. The Eclipsis system is deployed as four separately articulated zones on the east of and west of both the north and south façade. Although each of the four layers has an essentially binary, or 'on/off,' response to weather and climate conditioning, the combination of each layer and zone of the Eclipsis System allows for a graduated response to a multitude of environmental criteria.

A second layer of dynamic kinetic systems is applied to the interior space of the house. Designed under the motto 'a house larger than itself,' the Lumenhaus integrates dynamic and adaptable furniture at all levels, with its hide-away power outlets, sophisticated and multi-function cabinetry that is able to generate space, completely mobile counter tops, and transformable furniture. The result is a house with a small footprint that can make big moves—the adaptable furniture allows the house to be reconfigured for a magnitude of user needs and social requirements.

Figure 20 top, image of LumenHAUS's southern façade with open polycarbonate panels and closed shade screens. Image by Jim Stroup

Figure 21 center, image of LumenHAUS's southern façade with open polycarbonate panels and open shade screens. Image by Jim Stroup

Figure 22 bottom-left, interior image of closed dynamic bedroom core providing an open plan. Image by Jim Stroup

Figure 23 bottom-center, interior image of open dynamic bedroom core providing closed and private space as well as an exposed entertainment system. Image by Jim Stroup

Figure 24 bottom-right, interior image of open dynamic bedroom core with exposed storage. Image by Jim Stroup

EMBEDDED KINETIC SYSTEM

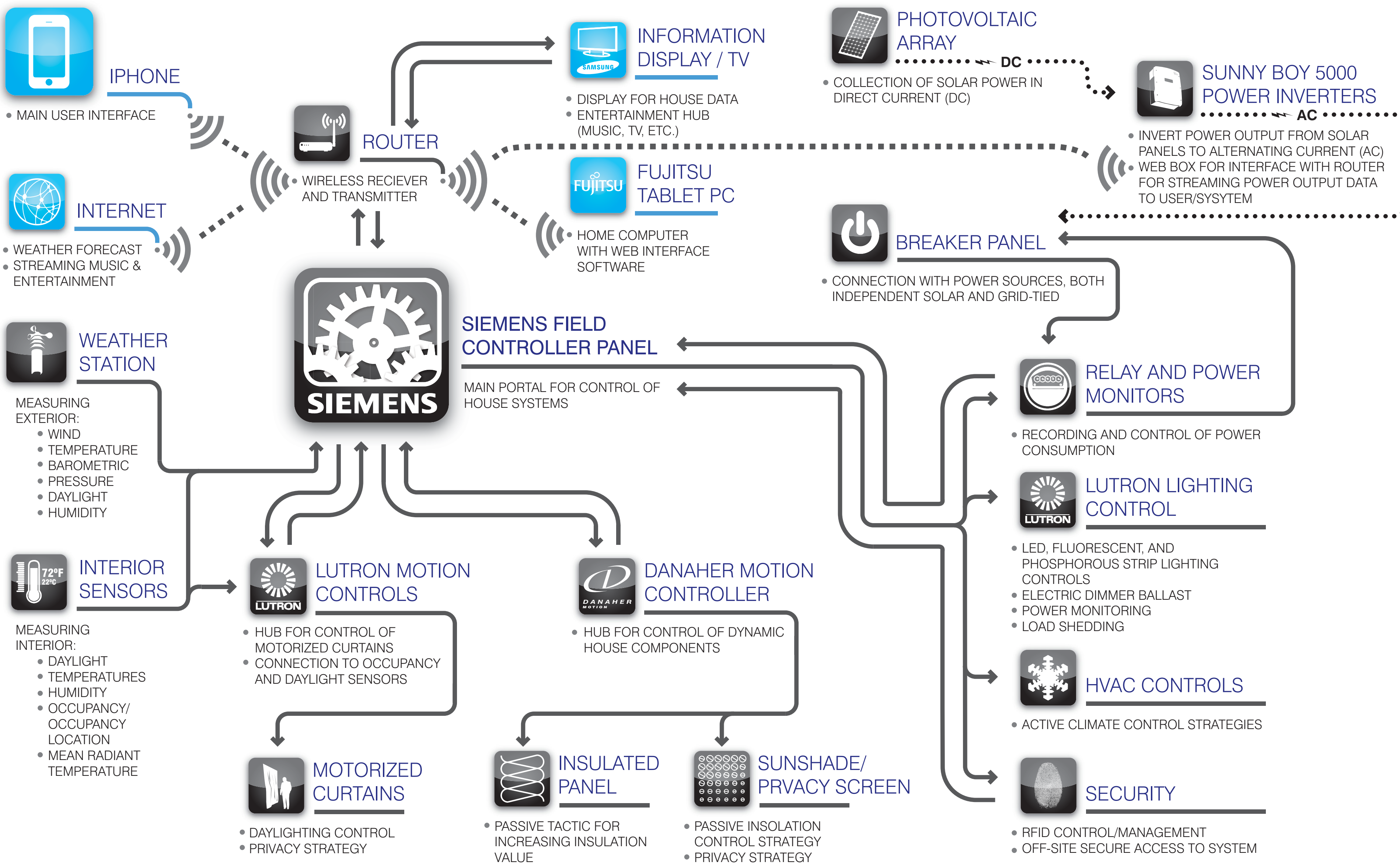
The Lumenhaus' greatest potential lies in its ability to produces a fully embedded kinetic system. The house is designed around a fully embedded computer input /output logic system. Through a system of interior climate sensors, thermal flux sensors, and an outdoor weather station, which can cross-reference outdoor temperature, humidity, wind speed and wind direction against interior climates and internet hosted weather data, the house is able to actively adapt to daily weather conditions as well as geographic climate zones. The house's ability to respond to a multitude of climate zones is another example of its deployable kinetic capacity. The weather station also serves as a protective measure; if wind speeds reach levels higher than a specified maximum wind speed, the shading and insulating panels, which sit outboard on the east and west when fully open, are closed to prevent any damage from high winds loads.

Through the use of eQuest and Trace 700 software, the Lumenhaus team was able to produce a schematic energy model for each layer of the Eclipsis System. This model established a set of rules or logic, which describe preset comfort condition for a given set of sensory data. For example, a series of studies showed that the shade screens allow 40% of direct solar radiation to penetrate into the interior space; therefore, the system has a prescribed response of closing the southern shade screens from 10:00am to 8:00pm during peak cooling seasons.

The centralized computer logic also allows for an adaptive photovoltaic system (another Dynamic Kinetic System). Through the use of six computer controlled linear actuators, the photovoltaic array can actively change its angle relative to the sun's position. The photovoltaic array pivots about its east/west axis on the south side allowing for the array to sit low during the high-sun summer months and rotates to a more perpendicular position to catch the low sun of the winter months.

The house also has the option of user override. The use of an iPad application designed specifically for the Lumenhaus by Virginia Tech computer science students provides a human computer interface between the embedded kinetic system and the occupant. The interactive application provides occupants with the ability to override a given comfort setting, as well as a way to actively monitor energy usage and indoor and outdoor environmental conditions.

Figure 25 diagram of LumenHAUS' embedded kinetic system. Diagram by LumenHAUS Team


IPHONE
MAIN USER INTERFACE
ROUTER
WIRELESS RECIEVER AND TRANSMITTER
INFORMATION DISPLAY / TV
DISPLAY FOR HOUSE DATA
ENTERTAINMENT HUB (MUSIC, TV, ETC.)
PHOTOVOLTAIC ARRAY
DC
COLLECTION OF SOLAR POWER IN DIRECT CURRENT (DC)
SUNNY BOY 5000 POWER INVERTERS
AC
INVERT POWER OUTPUT FROM SOLAR PANELS TO ALTERNATING CURRENT (AC)
WEB BOX FOR INTERFACE WITH ROUTER FOR STREAMING POWER OUTPUT DATA TO USER/SYSYTEM
INTERNET
WEATHER FORECAST
STREAMING MUSIC & ENTERTAINMENT
FUJITSU TABLET PC
HOME COMPUTER WITH WEB INTERFACE SOFTWARE
BREAKER PANEL
CONNECTION WITH POWER SOURCES, BOTH INDEPENDENT SOLAR AND GRID-TIED
SIEMENS
SIEMENS FIELD CONTROLLER PANEL
MAIN PORTAL FOR CONTROL OF HOUSE SYSTEMS
WEATHER STATION
MEASURING EXTERIOR:
WIND
TEMPERATURE
BAROMETRIC
PRESSURE
DAYLIGHT
HUMIDITY
RELAY AND POWER MONITORS
RECORDING AND CONTROL OF POWER CONSUMPTION
LUTRON LIGHTING CONTROL
LED, FLUORESCENT, AND PHOSPHOROUS STRIP LIGHTING CONTROLS
ELECTRIC DIMMER BALLAST
POWER MONITORING
LOAD SHEDDING
INTERIOR SENSORS
MEASURING INTERIOR:
DAYLIGHT
TEMPERATURES
HUMIDITY
OCCUPANCY/ OCCUPANCY LOCATION
MEAN RADIANT TEMPERATURE
LUTRON MOTION CONTROLS
HUB FOR CONTROL OF MOTORIZED CURTAINS
CONNECTION TO OCCUPANCY AND DAYLIGHT SENSORS
DANAHER MOTION CONTROLLER
HUB FOR CONTROL OF DYNAMIC HOUSE COMPONENTS
HVAC CONTROLS
ACTIVE CLIMATE CONTROL STRATEGIES
MOTORIZED CURTAINS
DAYLIGHTING CONTROL
PRIVACY STRATEGY
INSULATED PANEL
PASSIVE TACTIC FOR INCREASING INSULATION VALUE
SUNSHADE/ PRVACY SCREEN
PASSIVE INSOLATION CONTROL STRATEGY
PRIVACY STRATEGY
SECURITY
RFID CONTROL/MANAGEMENT
OFF-SITE SECURE ACCESS TO SYSTEM

THE ECLIPSIS SHADE SCREENS:
a Subassembly of Appliance Architecture

The design of the Eclipsis shade screens provides a study in appliance architecture's ability to evoke social dynamics, environment condition and the phenomenal. As stated earlier, the screens are one layer of a multilayer responsive architecture. In this way, they become a subassembly within a large responsive whole. The screens are a form of static Parametricism which is based on multiple static criteria, including maximum solar radiation, transposed spatial occupancy, cross ventilation and physiological well being.

From a pedagogical evaluation, the screens are a prime example of 'simulation is realization.' The tools set and algorithmic logic designed for the screens present a stepping-stone to a more complex understanding of appliance architecture found in this work. This is not to say the screens do not warrant recognition as in fact they do, and they have received it. The screens became a marketing identity for the Lumenhaus and a dynamic facet leading to the house being awarded first place in the category of architecture in the 2010 EU Solar Decathlon.

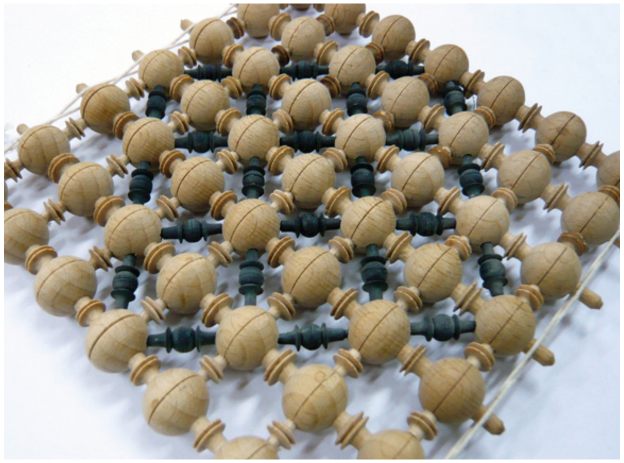
The screens are also a lesson in the power of simplicity to evoke complexity. When beginning the design, this author entered into a project that was already well into the design phase. The result was a controlled set of design criteria; however, these controls allowed for a freedom from spurious thought—design can be aided by the removal of external complexities. The original criteria called for a system of five panels. Three southern panels composed of a sliding two-bay western panel, a sliding single-bay eastern panel and stationary single-bay central/core panel. The north façade called for a sliding two-bay western panel and sliding single-bay eastern panel. Each panel was also limited in section. Based on the existing design and Solar Decathlon guidelines, the panels were limited to two-and-a-half-inch section, which would also account for the necessary structural depth of the tube steel frame. The final and most essential criteria were materiality and fabrication. The Lumenhaus team had established a strong partnership with A. Zahner Corp., an industry leader and innovator in sheet metal construction. The result was a design focused on the material properties of stainless steel and the digital fabrication technologies available through A. Zahner (which was possibly the only area without a constraint).

The original goals of the work were pragmatic – keep the sun off the façade of a building and reduce heat transfer. Precedents include shutters and screens that have been in use for centuries.

Mashrabiya: Architectural Precedent

As stated by Robert Dunay, a faculty leader for the Lumenhaus, the screens' precedents are a pragmatic and centuries-old technology: the shutter. He describes the primary precedent, the mashrabiya:

Figure 26 night Image of Eclipsis System on the National Mall in Washington, DC. Image by Dept. of Energy Solar Decathlon



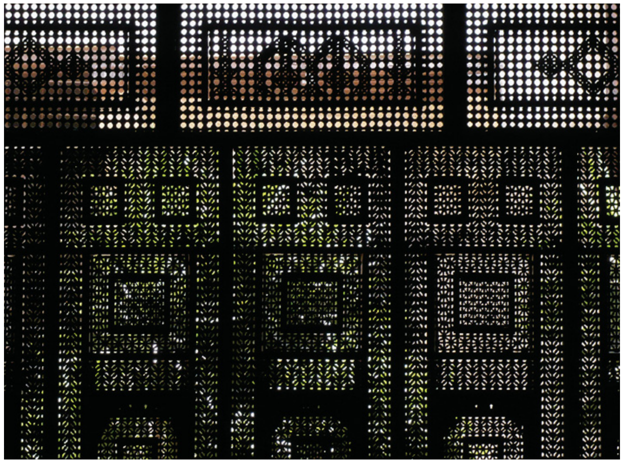


Figure 27 left, detail of Mashrabiya. Image care of Robert Dunay

Figure 28 center, detail of Mashrabiya. Image care of Robert Dunay

Figure 29 right, interior elevation of Mashrabiya. Image care of Robert Dunay

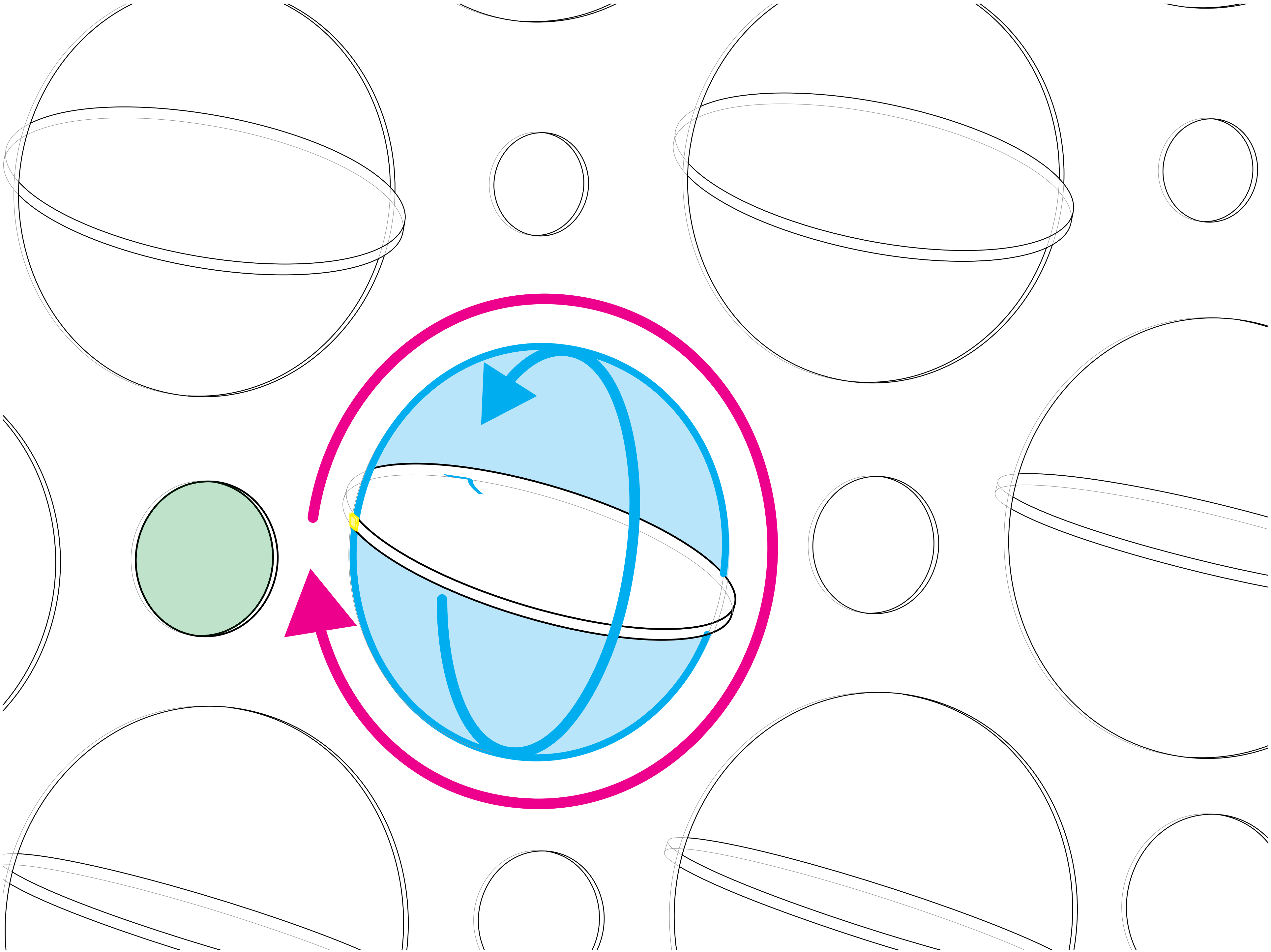
> Mashrabiya is an element of traditional Arabic architecture used since the middle ages up to the mid-twentieth century. A screen separating inside and outside, the construction is usually turned wood spools made of beech or mahogany. As the primary openings in dwellings, it allows a strategic interface between private and public space. The root of the name derives from the word overlook or to observe. Simultaneously providing privacy and openness, it allows the women of the house to be shielded from view but have a connection to the street. The wood screen gives shade and protection from the hot summer sun while allowing the cool air from the street to flow through. Gradations of openings increasing from bottom to top cause the draft to move faster above one's head providing significant air movement without uncomfortable drafts. The system combines environmental response with cultural value meeting five primary criteria: blocks the summer sun; controls the passage of light; controls airflow; reduces air temperature; and maintains privacy and security. (Dunay, Schubert, Wheeler, & Grinham, 2010)

The social/cultural/technical implications of this centuries-old building component inspired the development of an exterior wall system for a new type of dwelling. The imprint of the craftsman, the simple industrialized process of the lathe, the pragmatic response to climate, the unique spatial condition provided by sunlight, and the cultural value that can be read in the building component all became conceptual criteria for the development of a new architectural product. Thus, the historical concept of integrating technology and architecture finds expression in new materials and processes.

Eclipsis System: Shutter Screen

The primary function of the outermost layer of the shutter screen is sun control. The initial design by the Lumenhaus team called for a double layer of laser cut panels with an offset pattern that blocks summer sun, but admits winter light. RhinoScript—a Visual Basics (Vb) coding environment designed for, and embedded in, the Rhino 3D modeling environment—allowed for the development of variable criteria to produce a pattern that could be cut by laser or water jet.

Additional design criteria introduced to the design include security, privacy, view and the development of a particular quality of sunlight. The team's original design was amorphous and based on the Lindenmayer system [L system], which is an algorithm used to model organic growth structures and transformation of plant life, as well as the geometric growth of crystals. This gave a vital image to the industrialized process of the screen. However, fabrication issues and the difficulty of reconciling competing criteria proved difficult. Though the design was dynamic and visually stimulating, its performance, particularly regarding sun shading, was not satisfactory. In addition, the double wall proved problematic regarding weight, assembly and maintenance, causing alternatives to be explored.
The design research was re-focused, emphasiz-



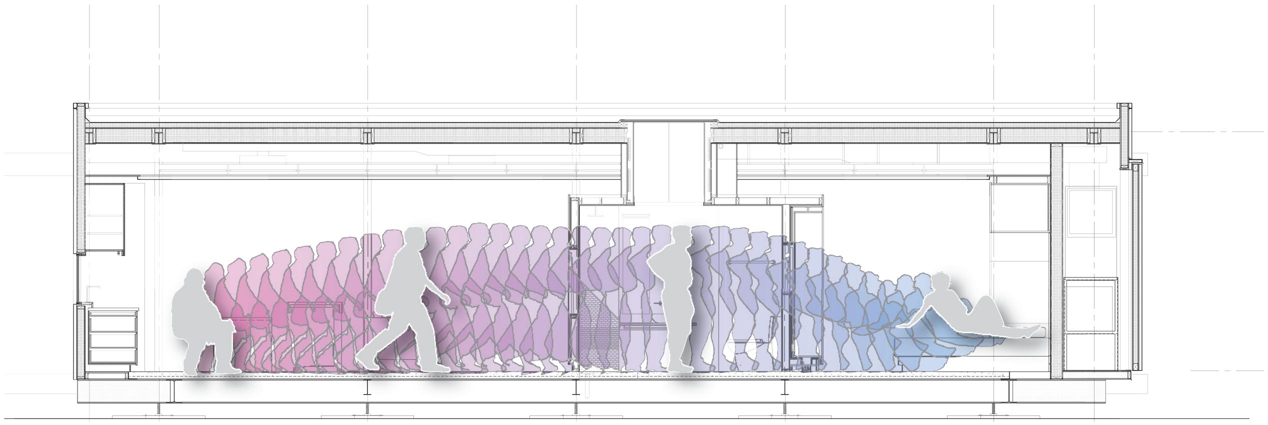
ing fabrication processes and material availability. A simple, repetitive, circular pattern was chosen that could be easily fabricated and met the multiple performance criteria. The use of circular geometry allowed for axial symmetry. By populating the screen with simple geometries with infinite orientations, a more complex understanding of the part-to-whole relationship of the shading pattern was established. Prototypes were developed and tested; at first, small panels were produced and later full-scale mock-ups.

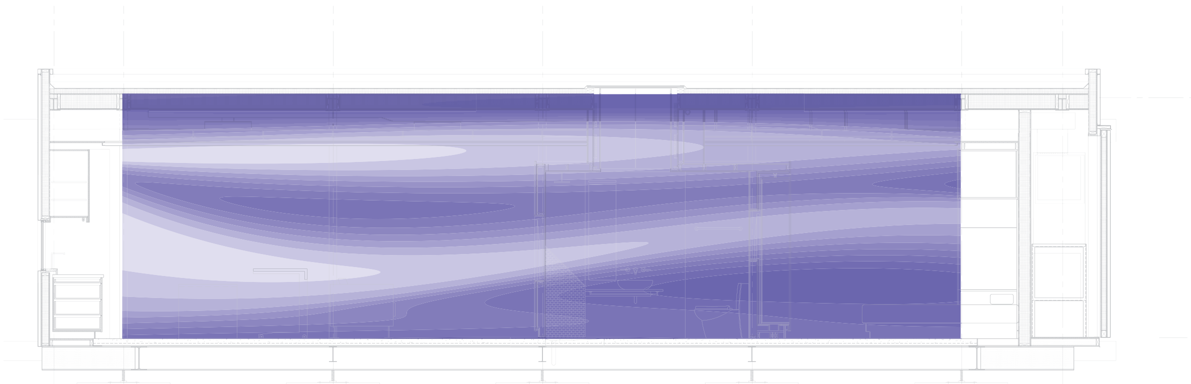
An innovative aspect of the new approach is the circular geometry of laser-cut holes with tabs folded at calculated degrees. This system allows a four-fold role: to keep the summer sun off the façade, to offer degrees of privacy while providing views to the outside, to break sunlight into fractals that intensify and enrich the space, and to permit cross ventilation. The folded tabs have four variables: the diameter of the circular cut, the planar orientation of the tab to the surface, the degree of tab rotation perpendicular to the surface, and the thickness of the tab material. These variables are articulated to block and bounce sunlight and to create views. For example, the bedroom tabs are only folded 10 to 30 degrees on a vertical axis favoring east orientation. This causes the rising sun to strike the backside of the tabs and illuminate the surface of the screen wall while blocking direct views into the space. In the dining room, strategic tabs are fully folded (90°) on a horizontal axis to create a direct view outside from dining height while blocking direct sunlight. As a research platform, the Eclipsis System's algorithmic logic was designed for mass customization. Shading, refracted light, privacy and ventilation were woven through a complex definition designed within Grasshopper, which is an open source graphic algorithm editor developed in conjunction with Rhinoceros. Grasshopper, along with Visual Basics programming language, allowed for the development of subtle complexity with extensive design versioning. The logic of each tab is simple; the complexity emerges from the swarm logic of the part-to-whole relationship

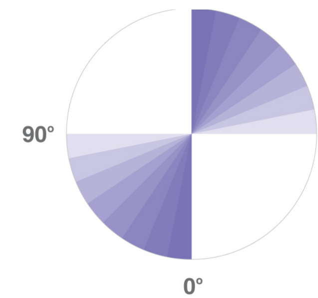
Programmatic Requirements

The shutter screen was designed for a specific situation, but it is scalable and can be used in many different sizes and combinations. When used in the open pavilion type floor plan, the program is used to facilitate clear views and privacy. The relationship of open/closed was diagramed through the narrative of a typical morning routine. The occupant begins their day in the privacy of the bedroom; upon rising, the tabs are open to allow views and light. The user then proceeds through core of the house and finally sits at the table for breakfast, where once again there is a clear view outside. This path and the resulting views both inward and outward produced a playful social interaction. When viewed from the outside, onlookers are only provided a view of seated occupants and the midsection of standing occupants. What emerges is a play on social interaction and communication. Viewers can only see body language and hand gestures of the standing occupants, resulting in a type of social interpolation.

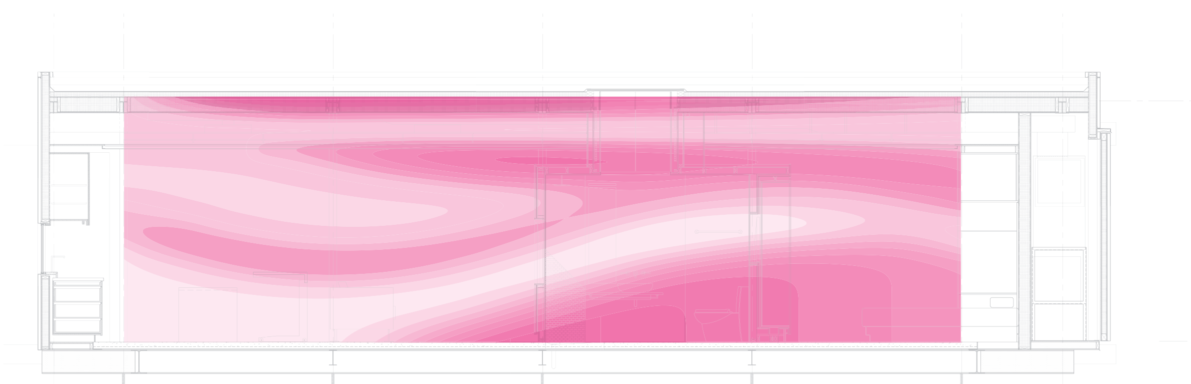
Figure 30 axonometric detail of shade screen diagramming material criteria designed within Grasshopper: blue, disk diameter and open rotation perpendicular to screen surface: green, secondary hole diameter: magenta, polar rotation parallel to screen surface: yellow, variable 'tab' thickness.

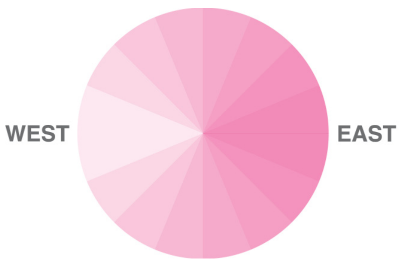






90°
0°





WEST
EAST

Figure 31 top, morphing diagram of programmed privacy and views during a typical morning. Magenta describes areas of view, cyan describes areas of privacy.

Figure 32 center, topographic diagram used by the Grasshopper definition to describe perpendicular rotation of disk in relation to programmatic elements of the south façade. Higher point (white) describe open view conditions where as lower (blue) describe areas of privacy.

Figure 33 bottom, topographic diagram used by the Grasshopper definition to describe planar rotation of disk in relation to captured and refracted light of the south façade. Higher point (white) describe rotation intended to capture western light where as lower (magenta) describe rotation intended to capture eastern light.

Time

The effectiveness of the screen to shade while providing dynamic light is an integral aspect of the project. The algorithm allows for refracted light to shift with the sun while also producing a smooth transition between lighting conditions.

Gradient

The algorithm's design allows for the customization of proportional relationship of the scale of opening and perpendicular rotation. In the case of this dwelling, the disk's maximum diameter of 2.5 inches represents a manageable, manipulatable unit. However, in large-scale applications, the scale of the screen could have much larger possibilities.

North v. South

Original versions of the screen patterning used a UV mapping system (x,y) to establish a Cartesian grid of points on which the circles were placed. Although this system provided a smooth visual morphing of the tabs' geometries along east /west axis of the façade, it resulted in a heavily stratified texture. Instead, a diagonal grid (diagrid) was adopted. The diagrid allowed for two new criteria. First, the diagrid allowed for a tighter 'packing' of circles that increased views and light transmittance. Second, and more importantly, it allowed for the weaving of a secondary condition: a pure, non-tabbed, cut hole. The addition of a secondary vocabulary provided an improved reading of the duality of the use of screens on the north and south facade. On the southern façade, the pure cut was woven into areas requiring more views and less privacy, such as the living/kitchen space. On the north façade where shading was not required, the pure hole cut was applied to all areas, and was erased from the bedroom area for privacy, producing a functional ornamentation pattern.

Override

The complexity of the algorithm allows for a new understanding of imagery. Through the use of image processing, the Eclipses System has the capacity for versioning that integrates environmental reasoning with style identity. The algorithm provides the rational for a physical gradient to be applied to any media (even image) that the user chooses.

Materiality

Prototypes were developed and tested—at first small panels were produced and later full-scale mock-ups. Prototyping, along with industry collaborators, provided a key understanding of feasibility and longevity studies. A Zahner was present throughout the entirety of the design development and ultimately fabricated the final panel system. The use of prototyping identified four major considerations: assembly methods, fabrication tools, material gauge and finish.

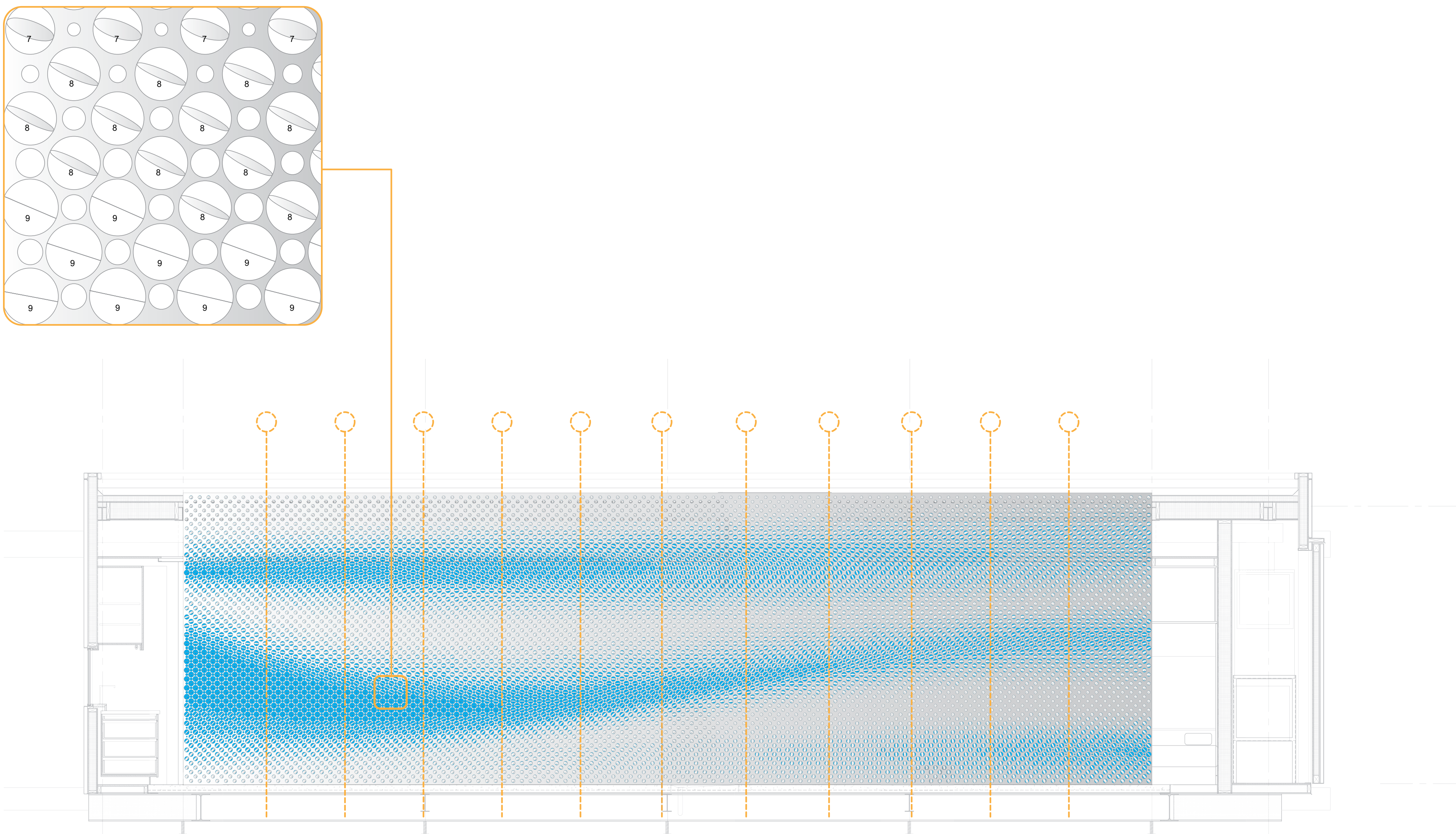

7 7 7 7
8 8 8
8 8 8 8
8 8 8
9 9 8 8
9 9 9
9 9 9 9

Figure 34 elevation drawing of southern façade with detail.

Assembly Method

The assembly methodology established many of the fabrication considerations. Given the complex nature of the circular tabs' rotations and the limitations of "off-the-shelf" CNC programming, it was determined that the discs would be rotated by hand. This presented many design stipulations. First, a controlled set of nine rotation increments was established, ranging from ten degrees to ninety degrees. This allowed for simple calibration "wedges" to be produced. Team members first bent the disk out and then calibrated the angle of rotation based on the corresponding wedge. In order to understand the polar rotation of each disk, an easy to read map of rotation degrees and polarity was needed. First attempts at spreadsheet style mapping and scaled drawings were found to lack a one-to-one legibility. A "paint by numbers" approach was adopted. Each disk would have its corresponding angle etched on the exterior surface. Etchings provided two streams of information: first, the degree of rotation, and second, the rotation's polarity—the direction the disk rotated perpendicular to the surface. Team members were informed to push in on the enscribed number, ensuring that the tabs properly rotate about the polar axis.

The use of an etched numeric system provides a new level of legibility to the consumer. At first encounter, the screens appear only as a graphic representation. However, upon a closer, more detailed viewing of the screens, visitors are able to see the logic and controlled engineering of the screens. By understanding each tab as a predetermined rotation, the effectiveness of the craft is translated to the user.

Fabrication Tool

Three computer numerically controlled (CNC) methods of steel fabrication were explored. The first method, die cutting, was evaluated based on a fabricator's in-house operations and extensive application of die cut panel systems. Die cutting was ultimately dismissed due to the limited variations of circle patterning (six diameters) and the inability to punch concentric arches with a material tab. The second method, water jet cutting, also an in-house operation, provided infinite geometric variations; however the water jet produced a large kerf, reducing the stability of the tab and resulting in a less visually compelling design along with a heavy "halo" effect in direct sunlight. Both die cutting and water jet cutting also proved to have limited capacity with regards to the "paint by numbers" system described above. Although both methods could incorporate a system of symbols to denote angle, each symbol would become a complete cut and thereby detract from the simplicity of the circle and ultimately appear as unintentional markings. CNC laser cutting was chosen for the process' ability for tightly controlled cuts, unlimited geometric patterning and laser etching capacity.

Material

As an exposed exterior shading system, the Eclipses screen system naturally lends itself toward a corrosion resistant material. It was determined that sixteen-gauge stainless steel sheet would provide the proper rigidity to prevent oil canning, given the extensive perforations. The sixteen-gauge section also provided an ideal average thickness for resisting (moment) loads at the tab rotation points. Through multiple prototypes, it was determined that a variable tab thickness was required to sustain tab rotation beyond sixty degrees. This adjustment was allowed within the Grasshopper definition by reducing the arc length for circles with a rotation angle greater than sixty degrees.

Finish

The full-scale mock-up designed for an exhibition at the Taubman Museum of Art in Roanoke, Virginia, provided immediate feedback on the exterior finish of the shading system. Although unfinished stainless steel provided the highest reflectivity, it also provided a large amount of glare. In order to reduce glare, a medium angel hair finish was abraded to the stainless steel.

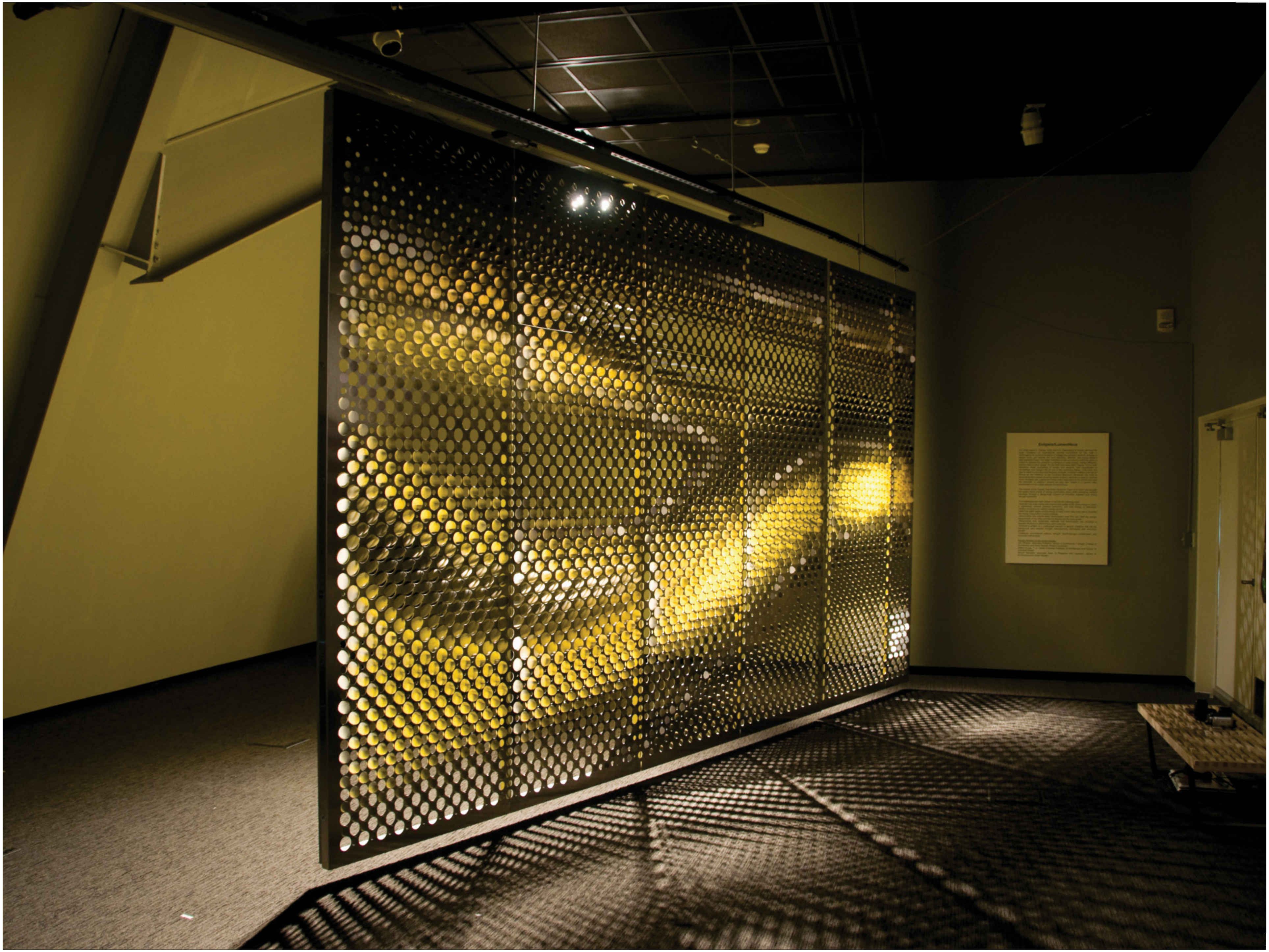
The interior powder-coated finish provides a new platform for efficiency and comfort. A consideration of the human factors and color perception allows for increased energy efficiency. By using a cooler color such as blue, the user's perception of skin sensations shift, allowing for lowered cooling loads within the space (Simeonova, 2010). In addition, a dynamic quality of movement is achieved depending on the intensity and angle of the sun and the position of the viewer, from outside, the color is revealed in staccato-like moments as one walks by. From inside, the color presents itself as an enclosing wall, partially transparent yet giving a sense of security and privacy.

Testing

For the purpose of doing a computer thermal simulation, the physical characteristics of the panel had to be determined. A representative 40-inch by 27-inch panel was fabricated and placed in an opaque box with its interior painted flat-black. This box was placed outside and oriented facing due south.

In order to determine the characteristics of the perforated screen, visible transmittance of the screen measurements were made using a combination of a light meter and a camera. These measurements were performed every hour from 7:30 AM to 06:30 PM on May 30, 2009. A series of illuminance readings were taken during this twelve-hour period. One reading was taken in front of the box in a vertical position and another was taken in front of the camera lens located inside the box. Spurious light was masked with black polyethylene plastic so that the measurements taken only represented the influence of the perforated sunscreen. These values were calculated as the ratio of the two positions. To accompany these measurements, interior photographic images of the perforated screen was taken by a camera equipped with an intervalometer and an image was recorded every 10 minutes for the duration of the analysis.

Figure 35 image of full-scale instillation at the Taubman Museum of Art in Roanoke, Va.



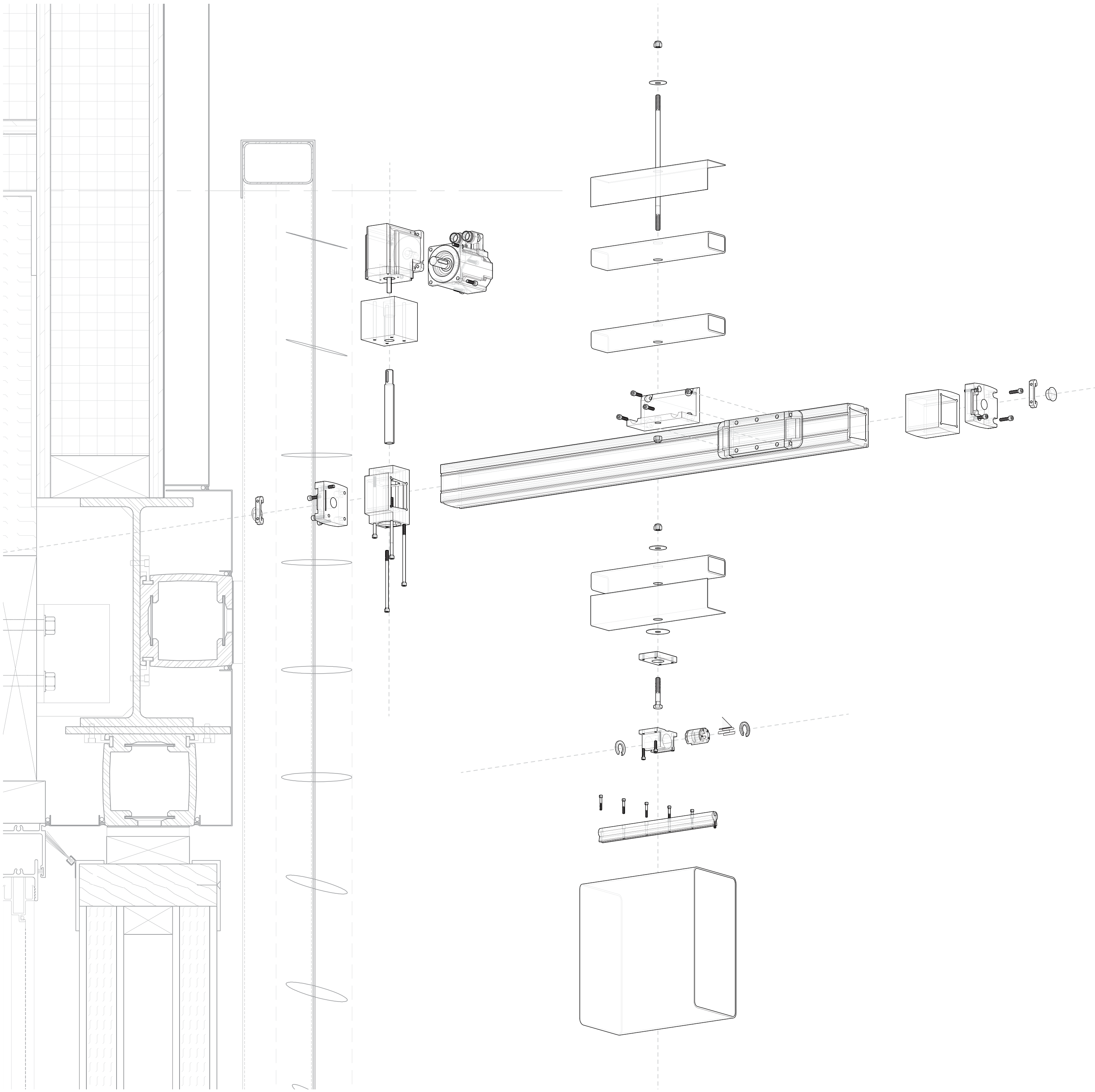


Figure 36 left, technical drawings of screen assemblies.

Figure 37 right-left, image of disks being rotated by hand and calibrated using a simple ‘wedge’ that corresponds with a laser etched numeric system. Image by Allison Ransom

Figure 38 right-top-right, image of tube steel frame construction.

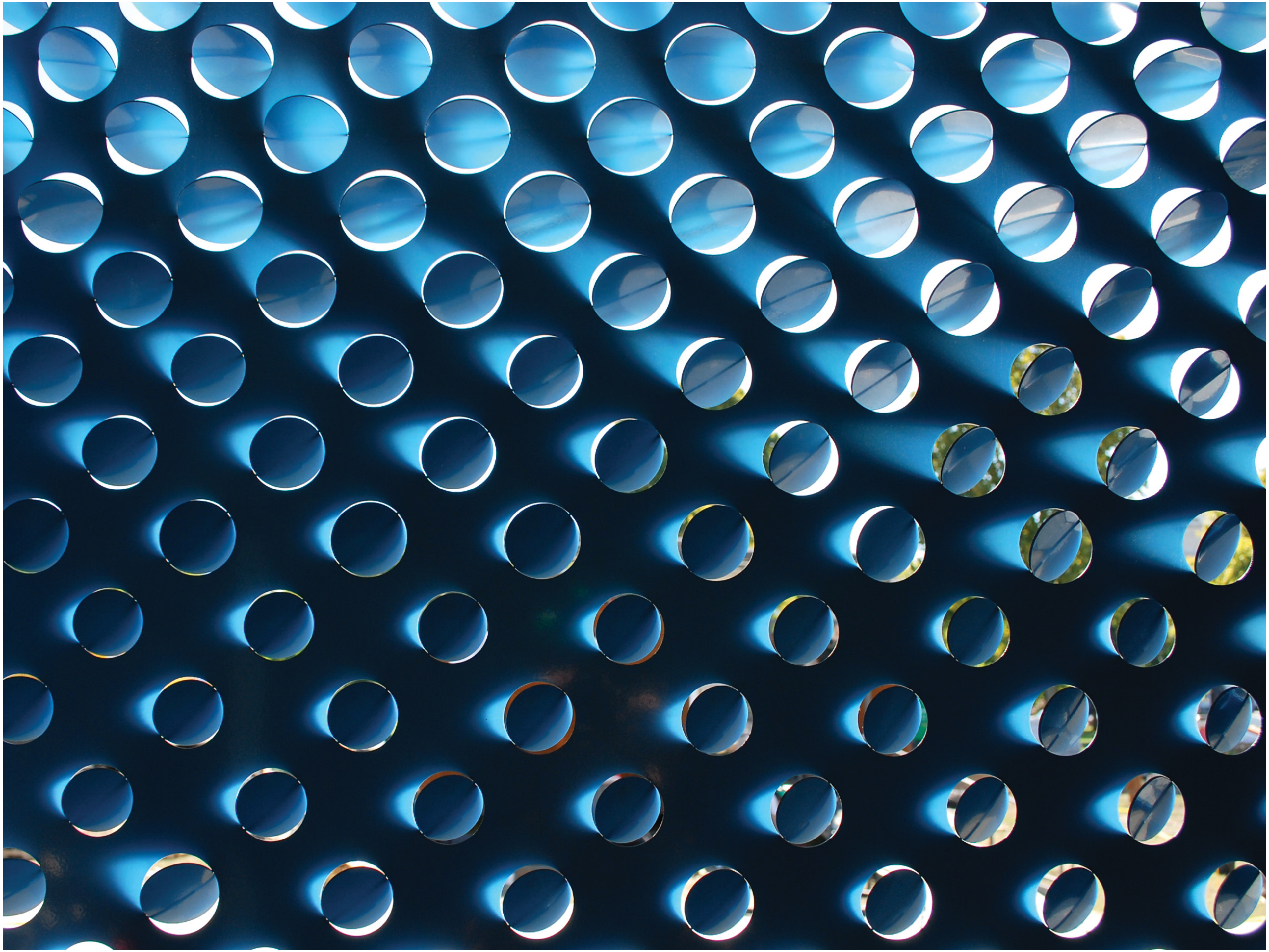
Figure 39 right-center-right, image of panal assembly using solid aluminium rivets.

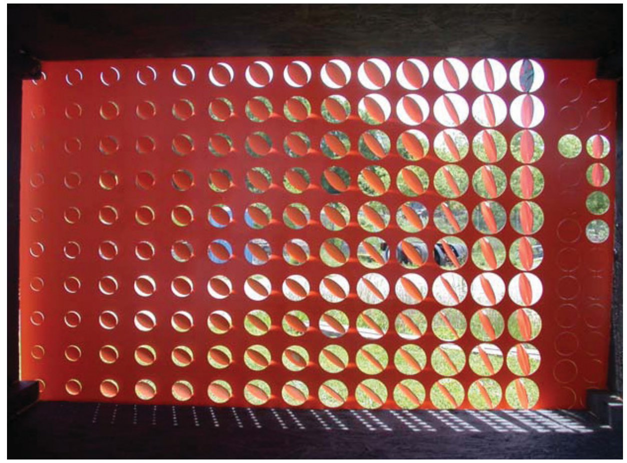
Figure 40 right-bottom-right, image of the shade screen’s ‘barn raising.’ Image by Allison Ransom

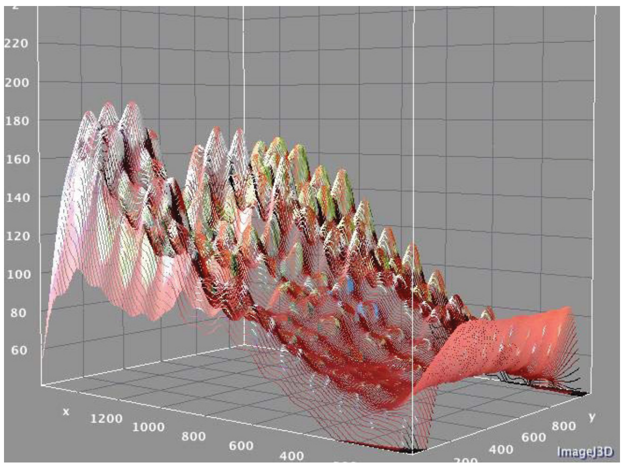
Figure 41 next page left, exterior image depicting the fluid, morphing and gradiated texture of the screens disks.

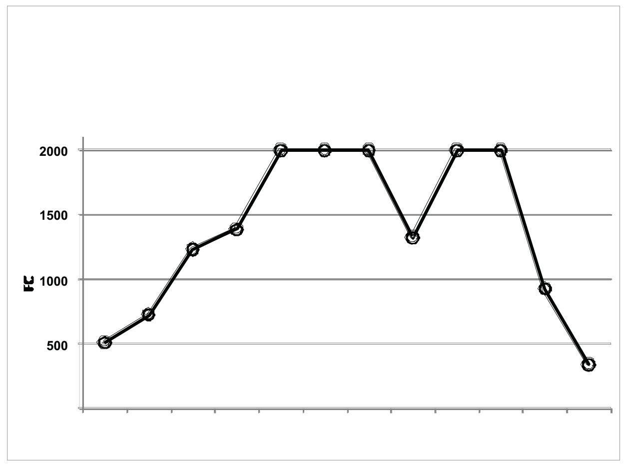
Figure 42 next page right, interior image of powder-coated surface depicting the flu-id, vector-like, transition of refracted light.

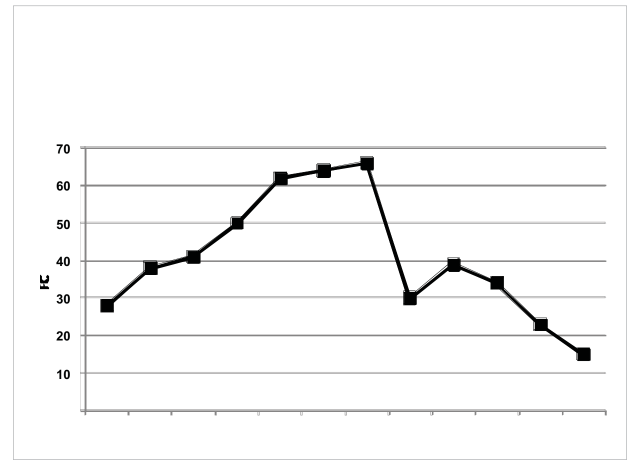

5
WRIGHT






220
200
180
160
140
120
100
80
60
x
1200
1000
800
600
400
y
800
600
400
ImageJ3D


2000
1500
FC 1000
500


70
60
50
FC 40
30
20
10

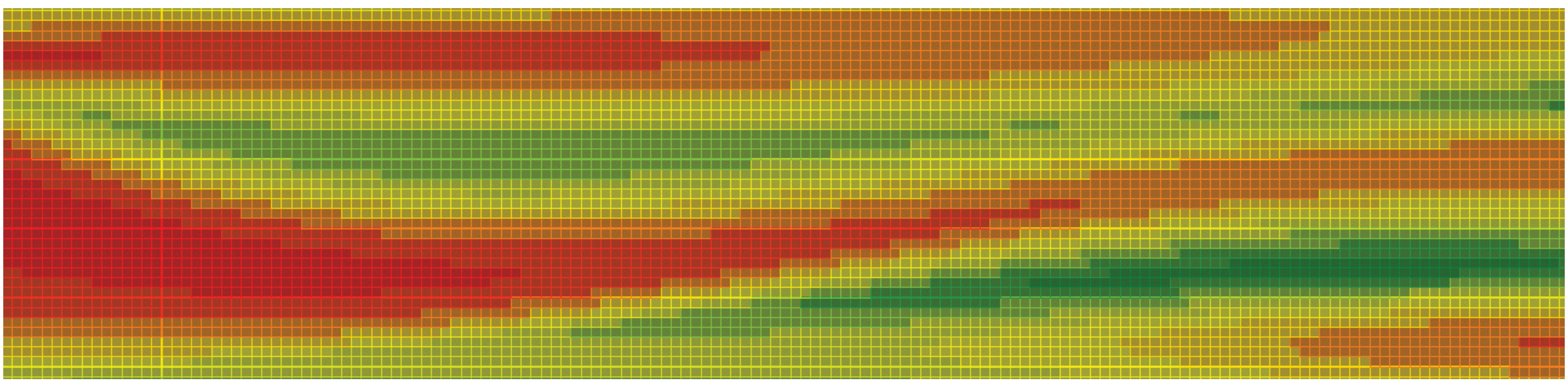


Figure 43 top-left, image of illuminance testing rig. Image by Robert Schubert

Figure 44 top-center, interior image of screens depicting light intensity. Image by Robert Schubert

Figure 45 center-left, 3D plot of pixel density histogram produced in ImageJ. Diagram by Robert Schubert

Figure 46 center-center, diagram of illuminance in front of the box in vertical position. Diagram by Robert Schubert

Figure 47 center-right, diagram of illuminance on the floor located in the center of the box. Diagram by Robert Schubert

Figure 48 bottom, histogram gernerated in Grasshoper representing projected illuminance, red depicting high illuminance, yellow depicting moderate illuminance, green depicting low illuminance

In addition, the photographic images were analyzed using the open source software program, ImageJ, which was developed by the National Institute of Health. ImageJ was used to determine pixel value statistics of the photographic images of the perforated screen. The program was also used to create density histograms and line profile plots indicating the variable transmission characteristics of the shading panel over time.

The result of the ImageJ analysis indicated an average solar admittance of 40 percent. This number was also used to ensure that subsequent design versions maintained a close relationship to the data being used in the energy model. The Grasshopper definition also allowed for the production of a histogram, which depicts the visible transmittance for each tab and provided an average transmittance for the southern façade that could be referenced to the 40 percent value. The histogram was produced through the following algorithm: $(2\pi(a/2)-((\pi a(a-a * (r_p/100))$, where a is the diameter of each circle and r_p is the rotation perpendicular to the screen surface. In other words, by calculating the total area of the circle minus the area of ellipse produced by the perpendicular tab, a histogram was constructed for the direct solar radiation for each disk.

Conclusion

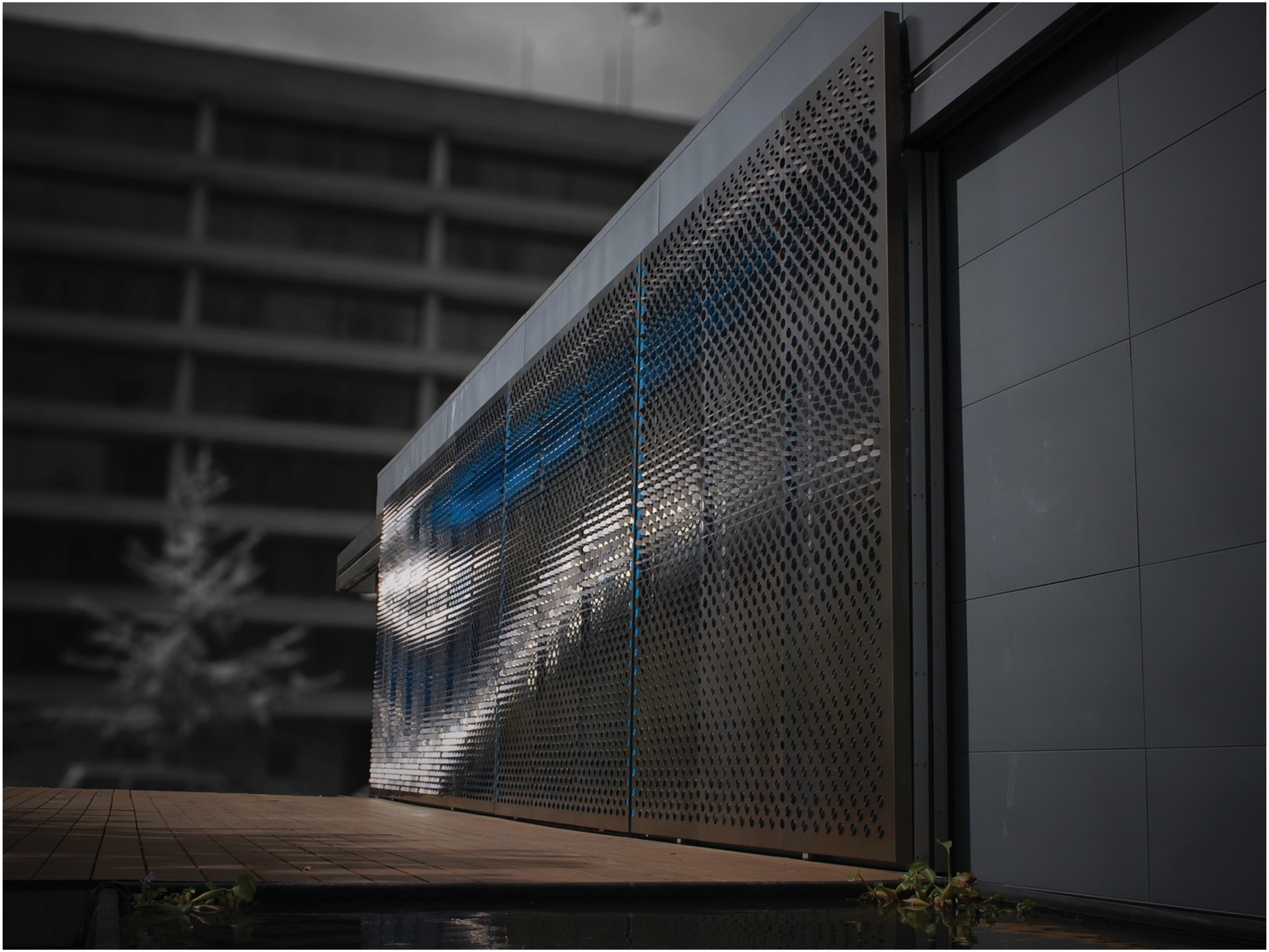
Lumenhaus provided a dynamic and rare opportunity to experience and research responsive kinetic systems at multiple scales of a building. The Eclipsis screens provided an example of the subassemblies of appliance architecture. As a technology and material study, they provided an example of the complex behavior and material logic that resides within appliance architecture. These material systems allowed for the development of parametric versioning that enabled the expression of place. At the same time, subtlety of pattern and user-defined design evoked curiosity and social interaction. The screens became a new media through which environmental, social and phenomenal criteria were prescribed. The same logic developed through the Eclipsis Screens will be jumping-off point for the studies developed in the next section.

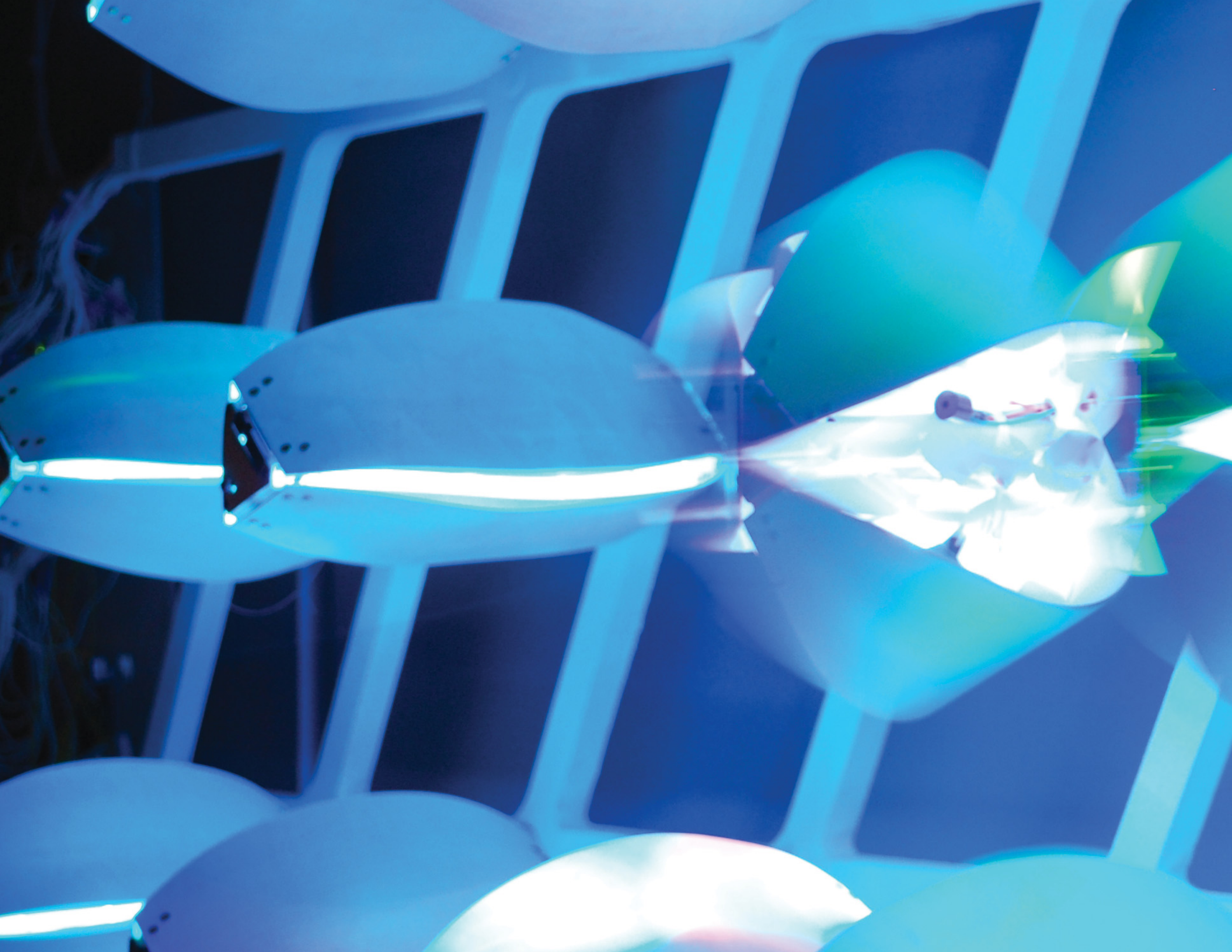
Figure 49 left, interior image of kitchen / living space. Image by Jim Stroup

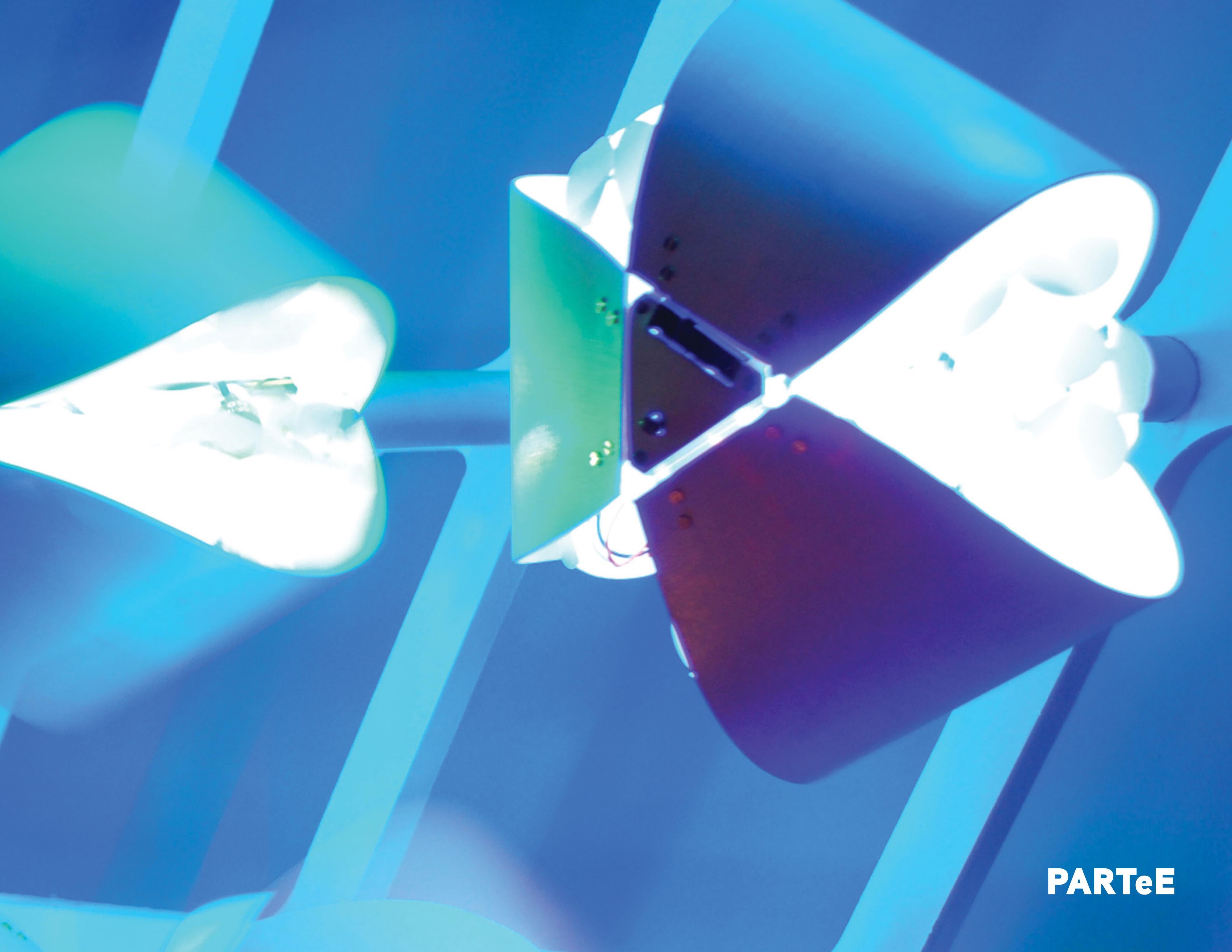
Figure 50 right, interior image of bedroom space. Image by Jim Stroup

Figure 51 next page, left, exterior montage of southern façade.

Figure 52 next page, right, exterior image of northern façade depicting the woven texture produced by the secondary holes.







PARTeE

Figure 53 previous page, time-lapse image of 'a2o.'

PARTEE LAB

"We are all transistors, in the literal sense. People always think they are in the world, but they never realize that they are the world"
- Karlheinz Stockhausen

The following series track the work of PARTeE (Prototyping in Architectural Robotics for Technology enriched Education), an interdisciplinary design laboratory approach that explores the potential of prototyping, architecture, robotics at the Virginia Polytechnic Institute and State University (Virginia Tech). The lab was established by professor Kihong Ku in the fall 2009 to investigate designs that integrate computationally-driven physical kinetic systems and components into buildings and spaces to meet the changing human needs. In order to engage in these fields, it is essential to address the aesthetic, social, and psychological human issues as integral parts of the design. A synergy between architecture, engineering, computer, behavioral, and material sciences is needed to achieve the design of interactive spaces. This synergy brings new forms of expression to architects, but it demands novel strategies that require a new, interdisciplinary generation of designers, engineers, and builders that can collaborate and exchange knowledge. In one way, the following studies show the development of concepts and designs related to appliance architecture. In another way, they are a pictorial and retrospective description of the emergent typologies developed by PARTeE that provide the proof of concept for the invisible college.

[Note: The evolution of bottom-up typologies is evaluated through this author's experiences during the PARTeE laboratory. Prior to the lab, this author possessed an intermediate skill level in relation to Visual Basics programming and advanced skill level in relation to Grasshopper. His understanding of physical computing and Arduino coding was limited. In this way, this author serves as test subject for the invisible college.]

EVOLUTION

The initial design prompt, Passage, was used as an ambiguous or blurred program within which emergent typologies in the field of architectural robotics could be explored. The brief was taken from a prompt provided by Kostas Terzidis in his book, 'Algorithms for Visual Design Using the Processing Language.' Kostas states:

> A passage is a movement from one place to another (as by going by, through, over or across). While a passage signifies a process of flow, transition and movement, it also implies the existence of a barrier, an obstruction, or an impediment. A passage is about the notion of a path, road, channel, trench, alley, or route, yet it is also about a cut, gash, incision, slash, slice or slit on a barriers. In architecture, passages are typically addressed through doors that connect rooms. A door is a movable structure used to close off an entrance, typically consisting of a panel that swings on hinges or that slides or rotates. (Terzidis, 2006)

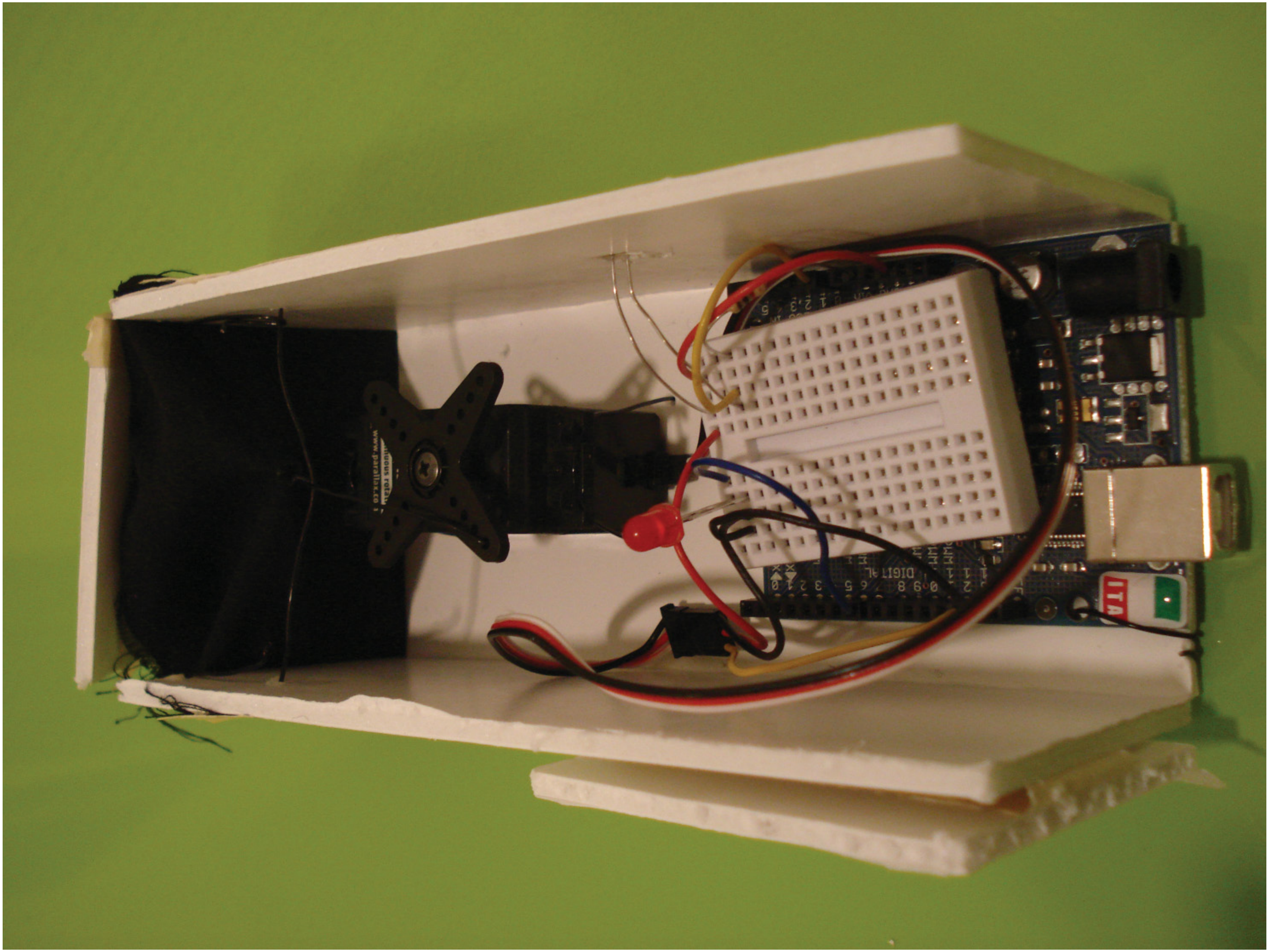
Just a Box

Just a Box senses light levels in space using a photocell, which is a variable resistor that changes its resistance in relation to light exposure. The level of light falling upon the photocell becomes analogous to the shadows of a body in space, and acts as an occupancy sensor. This data is processed through simple Arduino code. The Arduino's input peripherals read 10-bit analog values that range from 0-1024. In the case of Just a Box, if the value is greater than 500, the process is nullified, or turned off, meaning either that it does not detect a body in space, or there is a large light reading. If the value is less than 500, the value is translated into a rotational degree using a mapping function. This value provides a rotational angle and speed, as well as the delay between the LED's on/off values. The prototype implemented a continuous servomotor, which required the use of pulsewidth modulation (PWM). This allowed for a translation of the PWM into a literal pulse, producing a sense of anticipation, anxiety or jitteriness by the box as the body neared the prototype.

[Pulsewidth Modulation, or PWM, is a technique for producing smooth analog results through digital means. Digital signals that switch between on and off are used to create a square wave. This on-off pattern can simulate voltages in between full "on" (5 Volts) and "off" (0 Volts) by changing the portion of time that the signal spends on versus the time that the signal spends off. The duration of time that the signal is on or off is referred to as the pulsewidth. To produce varying analog values, the code changes or modulates the pulse width.]

The use of continuous servomotors over standard servomotors highlights the need for 'off-the-shelf' parts with low barriers to entry. For an entry-level designer, the use of PWM can be complicated and foreign, resulting in lost time. As mentioned early, standard servomotors can be controlled through Arduino libraries, which reduces the need for extensive understanding of core physical computing principles. However, this 'trial by fire' method does allow for early learning of more complex coding and computing principles.

Figure 54 image of Just a Box.



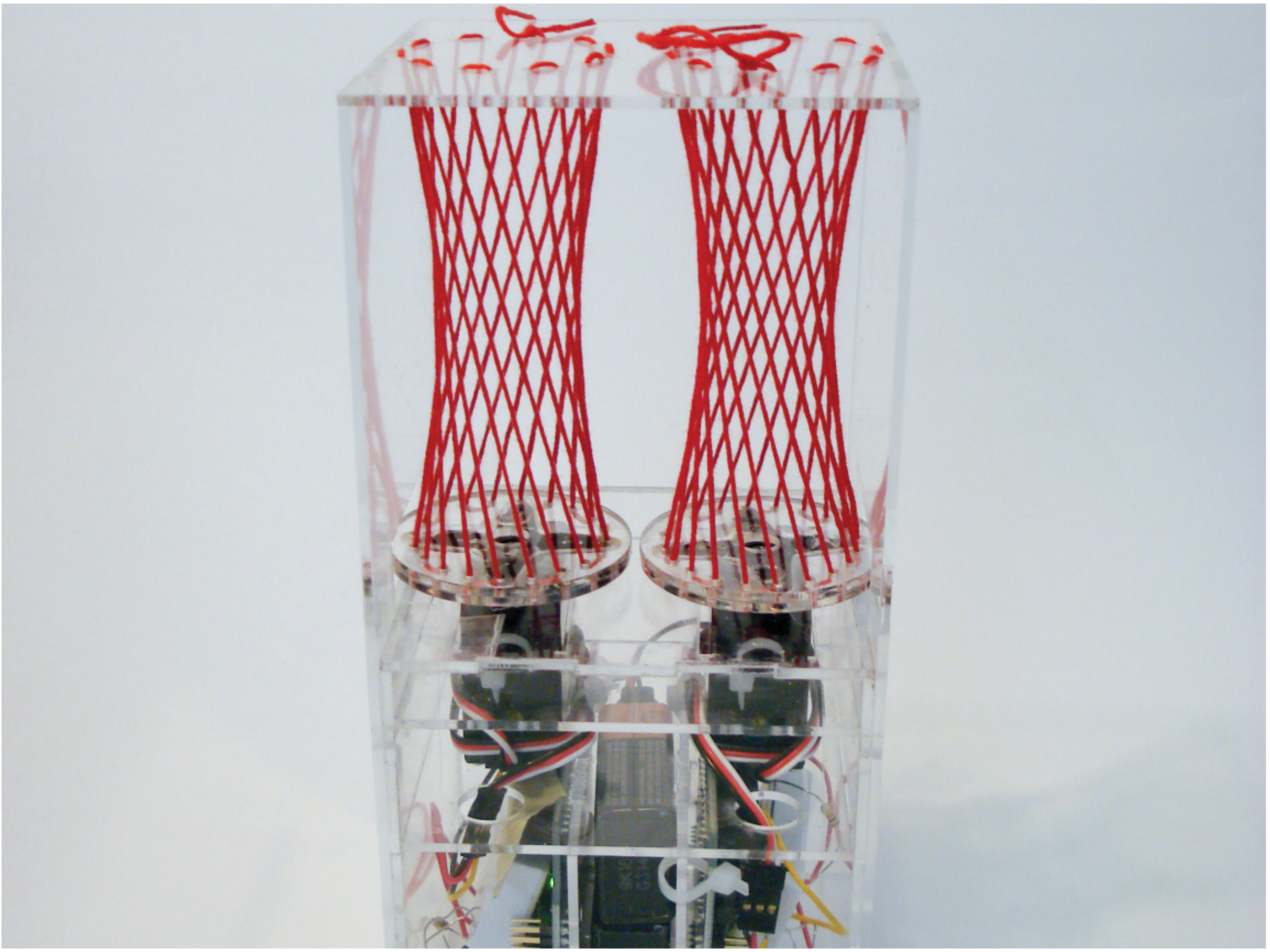


Figure 55 top, image of Tubes.

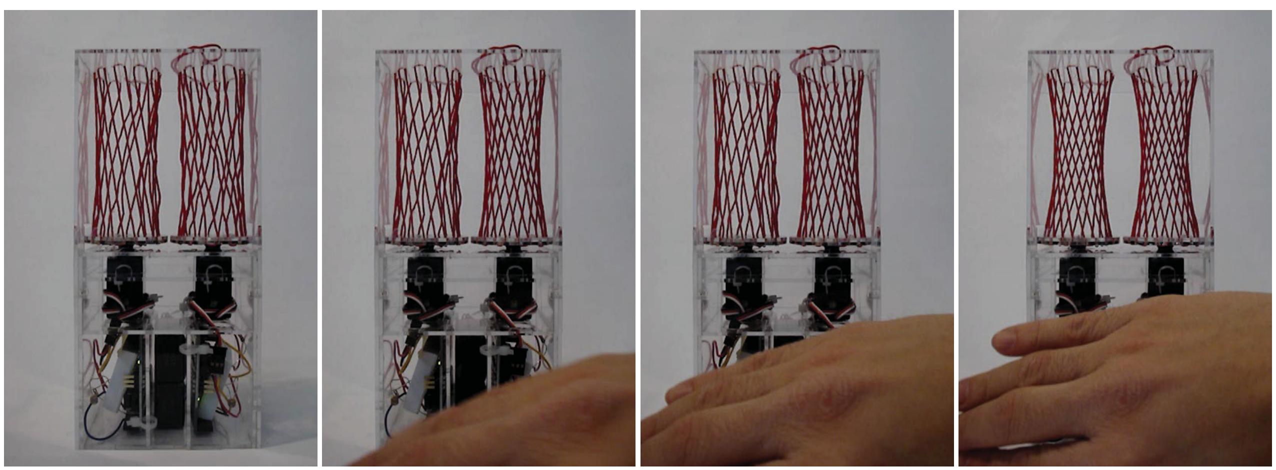


Figure 56 bottom, image series showing the rotation of Tubes when light is removed and the subsequent gill-like opening.

Tubes

Tubes sense light levels in space using two photocells. This data is then run through the Arduino microprocessor and is correlated with the adjacent servomotor. Tubes are understood in the context of opaque fabric tubes arrayed across the interior of curtain wall system—much like vertical blinds. Each tube responds to localized light levels falling on its photocell. When light levels are low, the tube's base rotates against a pinned position at the top of the tube. This rotation twists the fabric cylinder into two conic sections resulting in a gill-like opening between neighboring tubes. The photocell also measures disturbances in light levels such as the shadow cast by passers-by. In this way, the tubes produce isolated views into the space during sunny periods of the day.

The motion construct of Tubes is designed to explore space-packing and module systems on which multiple configurations can be arranged. The design of a plug-and-play product will become a continual typology throughout the work of PARTeE.

Diaphragm

Diaphragm is an additional configuration to the plug-and-play system developed for Tubes. Although similar code is incorporated, in this case photocells are placed on the east and west sides of the object. The code reads the analog signal from each sensor through a loop operation and compares the east and west values. If the east value is greater, for example, the code will call the east and west servomotor to rotate counterclockwise, tensioning the east wire and slacking the west wire. This operation causes the box's vertices to collapse and subsequently move the east vertices to the light.

Diaphragm can be understood at multiple scales. The first scale is reminiscent of a gill similar to what is seen in Tubes. The opening and closing of the Diaphragm is used to mitigate light levels and views. The second scale of design evaluates the possibility for the movement of a fully embedded building structure. Imagined as a building envelope that can flex in order to reduce solar gains falling on its southern façade and adaptively change the exposure of a photovoltaic array on its roof surface, Diaphragm results in a dynamic eave condition.

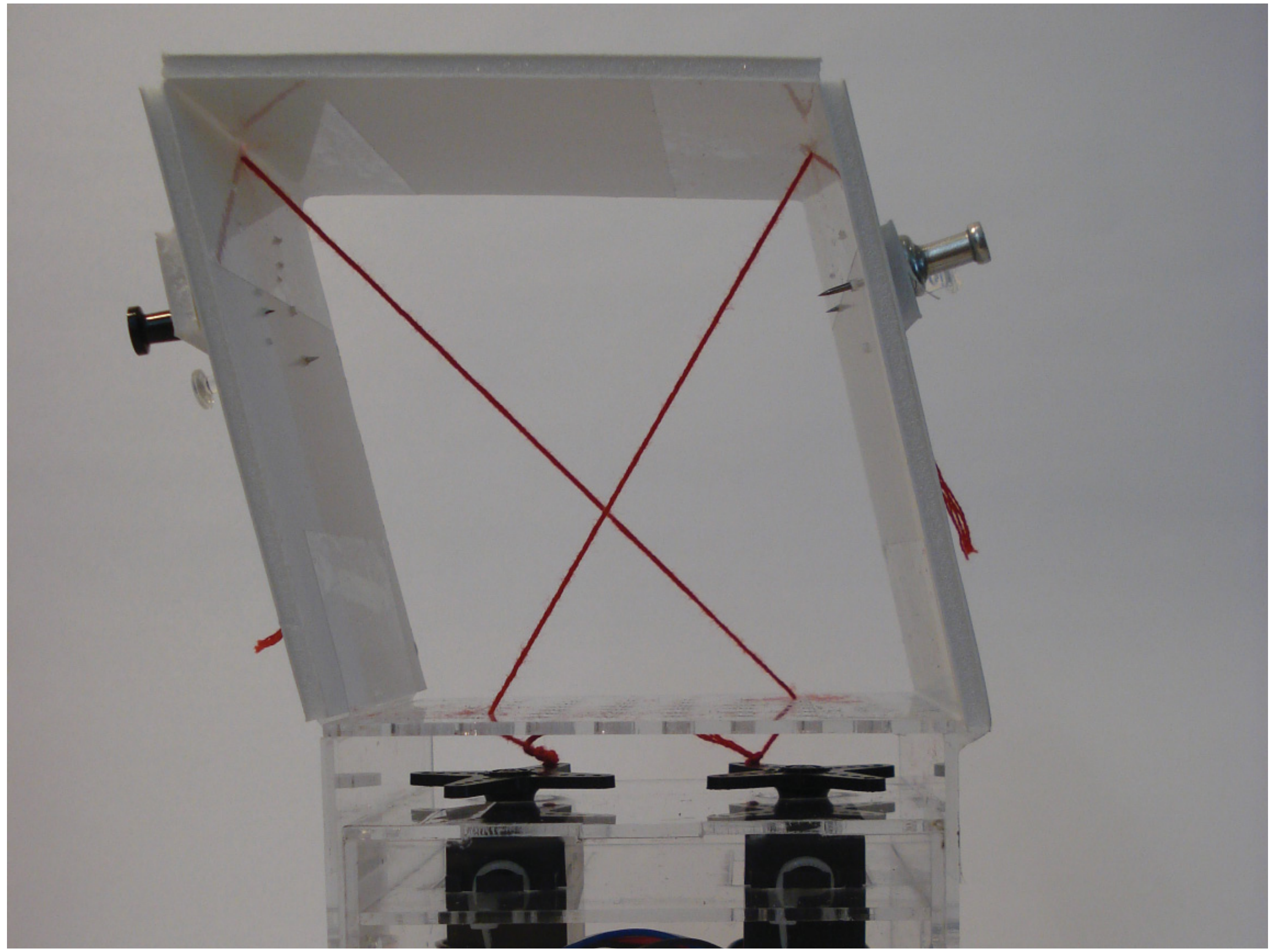


Figure 57 top, image of Diaphramgm.

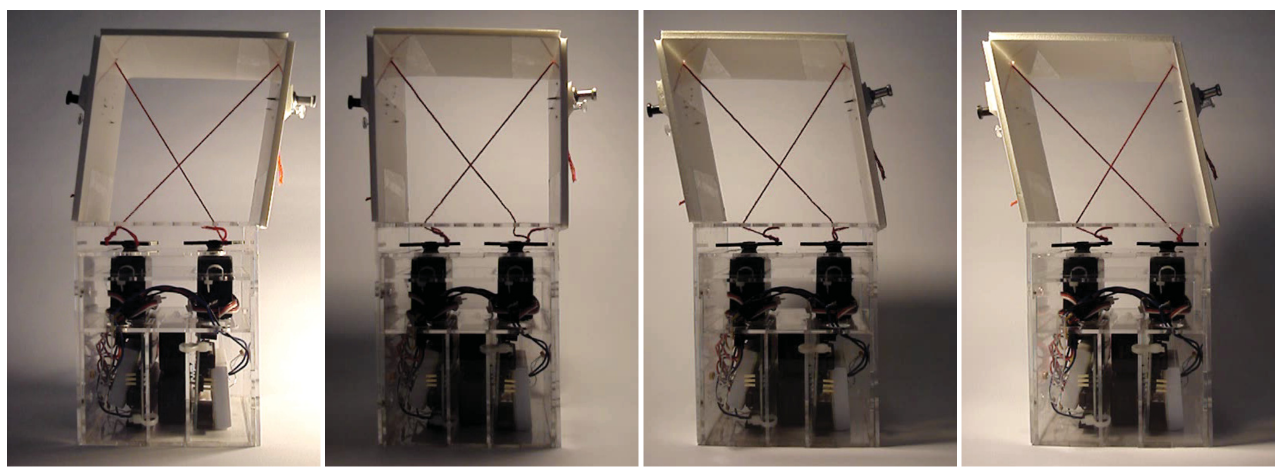


Figure 58 bottom, image series showing the shifting of Diaphragm toward the light source.

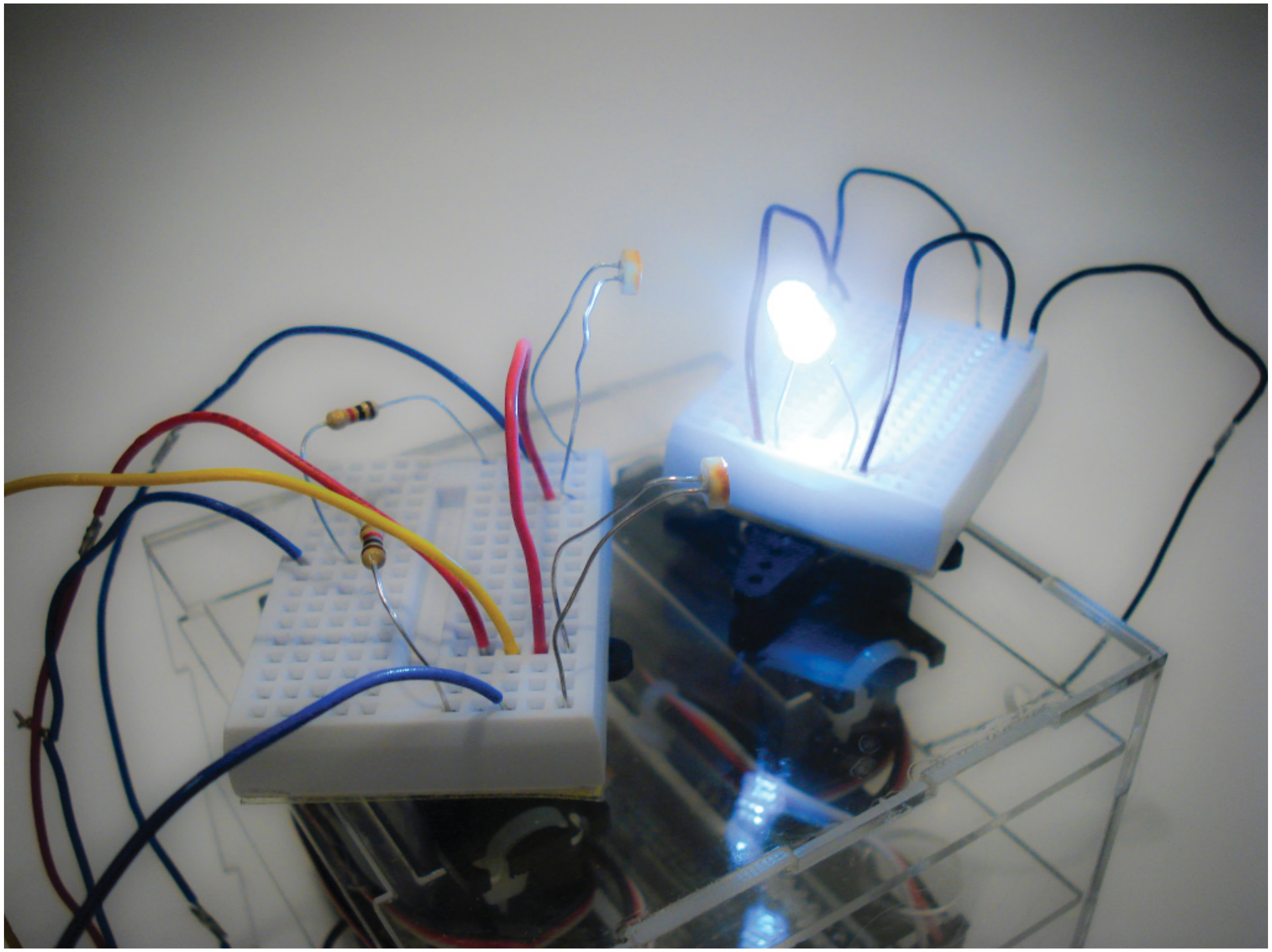
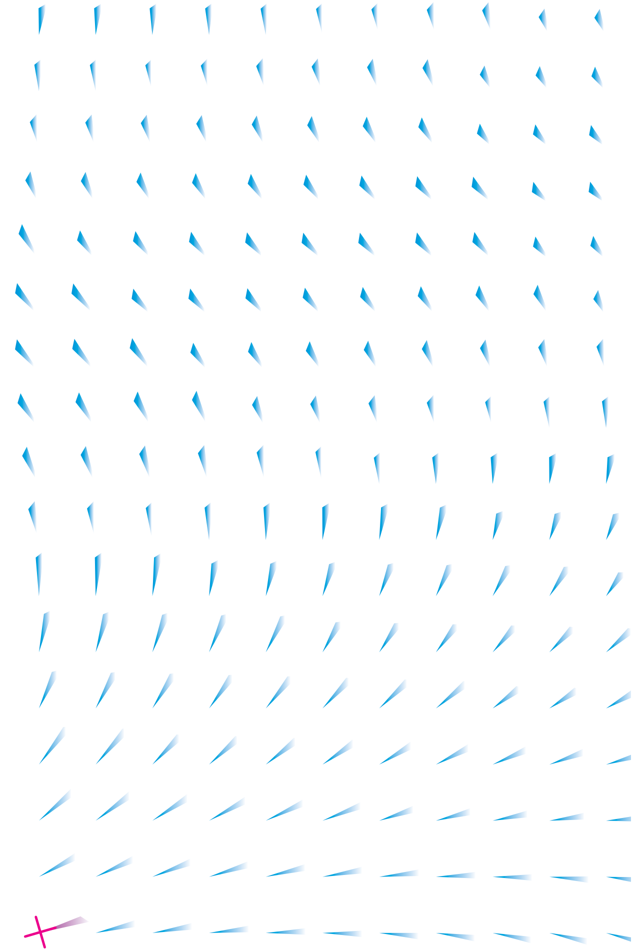
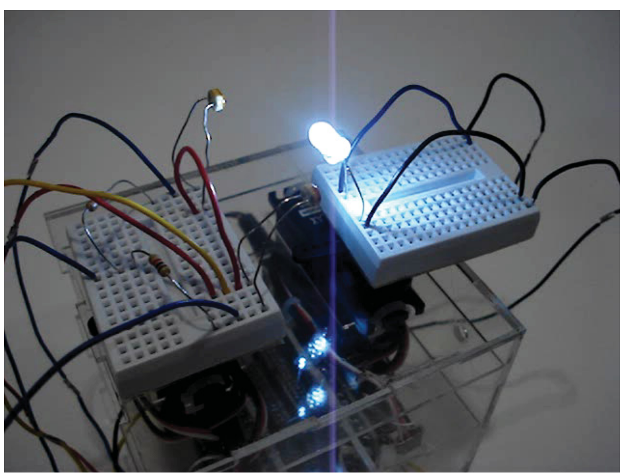
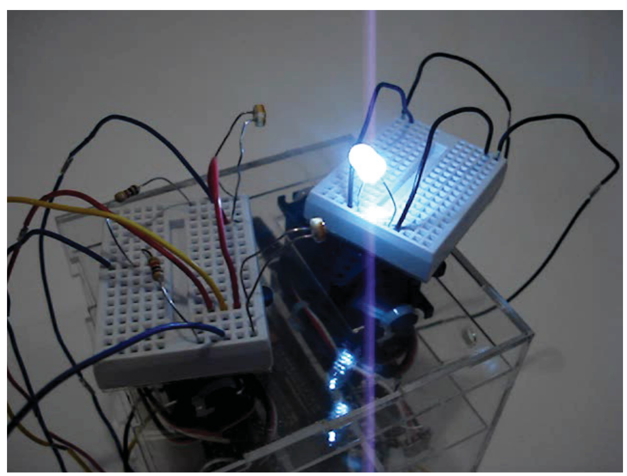
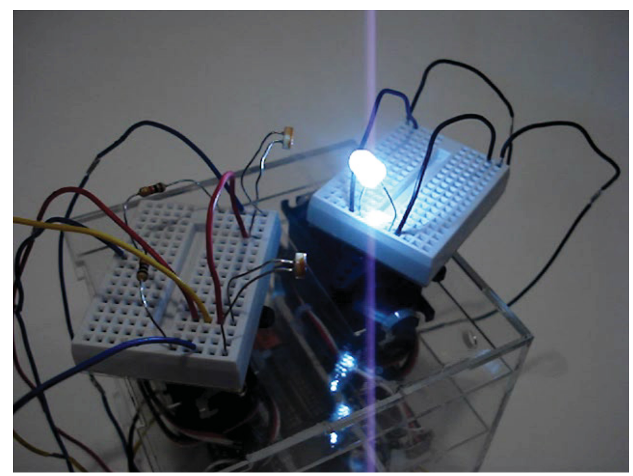






Figure 59 top-left, image of Tom + Jerry.

Figure 60 top-right, diagram of potential emergent pattern by an aggregation of Toms responding to random rotation of a single Jerry.

Figure 61 bottom, image series showing the response of Tom (left) to Jerry (right).

Tom + Jerry

Tom + Jerry explores material feedback loops, emergent patterning, growth and erasure. The initiator (Jerry) is comprised of a single Arduino board, LED and servomotor. The Arduino code produces a random string of numbers composed of 1's and 2's through a loop statement where each subsequent value of 'x' in the loop is manipulated by the algorithm: x = (x/3)%2. The motor rotates counter-clockwise by 20 degrees if the code generates a 1, and clockwise 20 degrees for a 2. Therefore if the string is 1,1,1,2,1 the servo will rotate 60 degrees counter-clockwise then back 20 degrees in the clockwise direction and then 20 degree counter-clockwise back to 60 degrees. This produces a random radial location of an LED, which is always 'on'. Subsequently, the responder (Tom) is comprised of a single Arduino board, two photocells and a servomotor. Tom senses the light levels of the left (A1) and right (B1) photocells. The Arduino code then compares these two values. If A1 is larger than B1 and the previous A1 value (A2), then the motor rotates counter-clockwise, or towards the LED light, and vise versa for a higher B1 value. The result is a physical translation of algorithmic logic into a secondary physical behavior.

Although the code and construct are quit simple—if not crude, Tom + Jerry provides a platform for studies of emergent patterning. If we are to understand this relationship as a larger matrix of Tom and Jerry's, or better, a series of Toms, which also have an LED on the opposite side of their photocell, we can start to see the impact of physical translations of rule sets and the possible erosion of motion or emergence of greater motion. In this way, the initial Jerry signal could be read by the first Tom (T1); T1's LED could then be read by T2, whose LED is read by T3 and so on. This physical translation, specifically the amount of rotational movement expressed in light levels, could produce a possible loss of translation, resulting in T100 expressing no movement at all. This emergent patterning expresses a great potential within appliance architecture and dynamic Parametricism to exhibit behavioral logic that could not be computed through 3D virtual environments.

Orchid

Orchid explores phystologic responses. The Orchid is controlled through the use of one Arduino microprocessor, four photocells and four standard rotation servomotors. The code maps the photocells analog signal between 0 and 10 (this is a way of averaging analog values or reducing the difference between photocell readings, i.e. reducing significant figures). The values of each photocell are then cross-referenced, and if the value of one cell is greater than the other three values, the value is mapped between 0 and 180 degrees. The corresponding servomotor rotates, moving the orchid to the light. If a 'greatest' value cannot be determined, the four values are averaged and the four motors produce linear compressive movement in proportion to the amount of light. When the value is less then 20% (i.e. there is no light), the Orchid goes into a 'screen saver' mode and performs a dance based on a similar algorithm to the randomization algorithm used for Tom + Jerry. The algorithms produce a string of 0's ,1's, 2's and 3's, which are then mapped proportionally to a rotational degree, producing a seemingly random dance. The algorithms are:

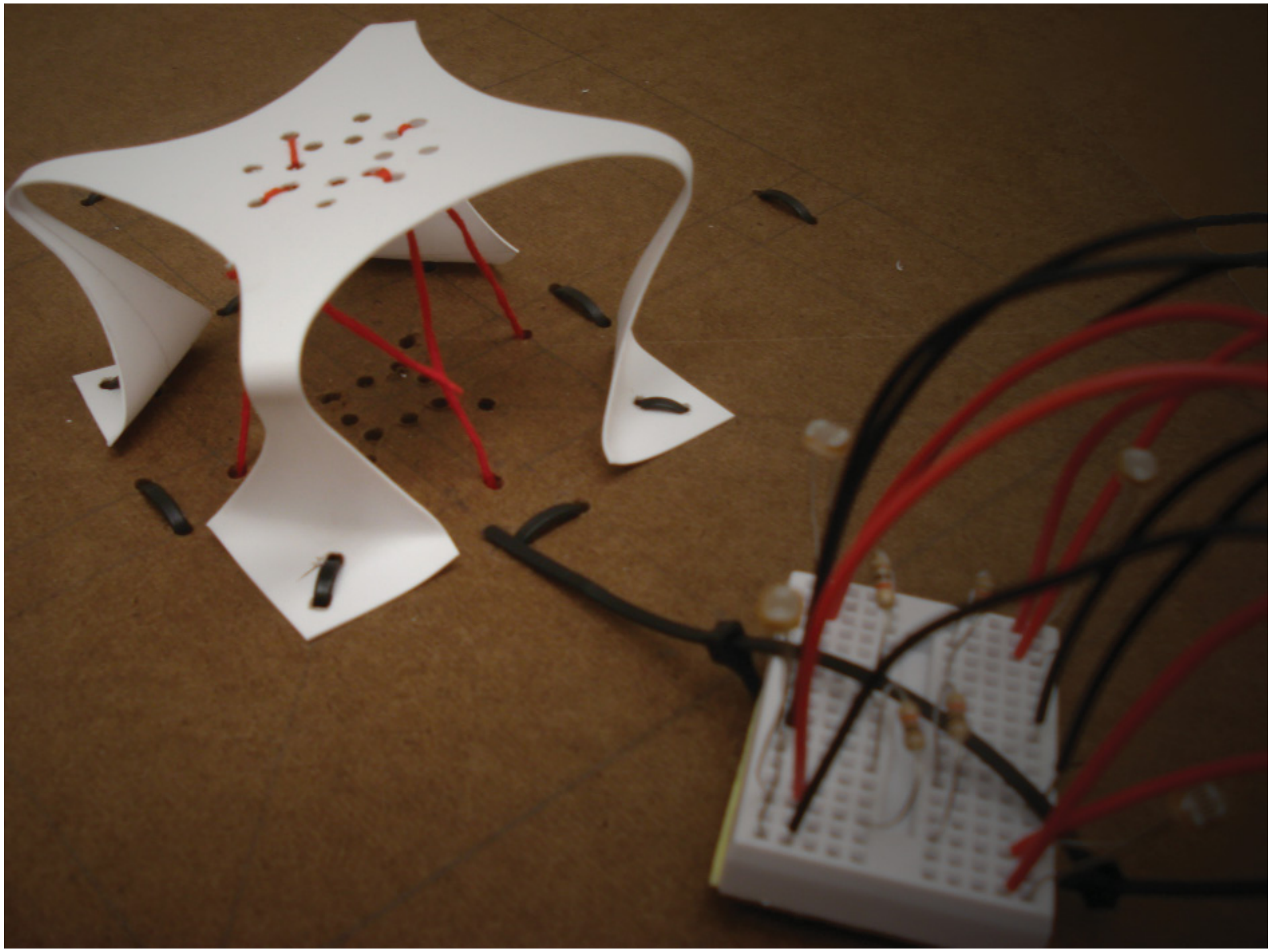
x=i%4
123012301230
x=(i+1)%4
012301230123
x=(i+2)%4
301230123012

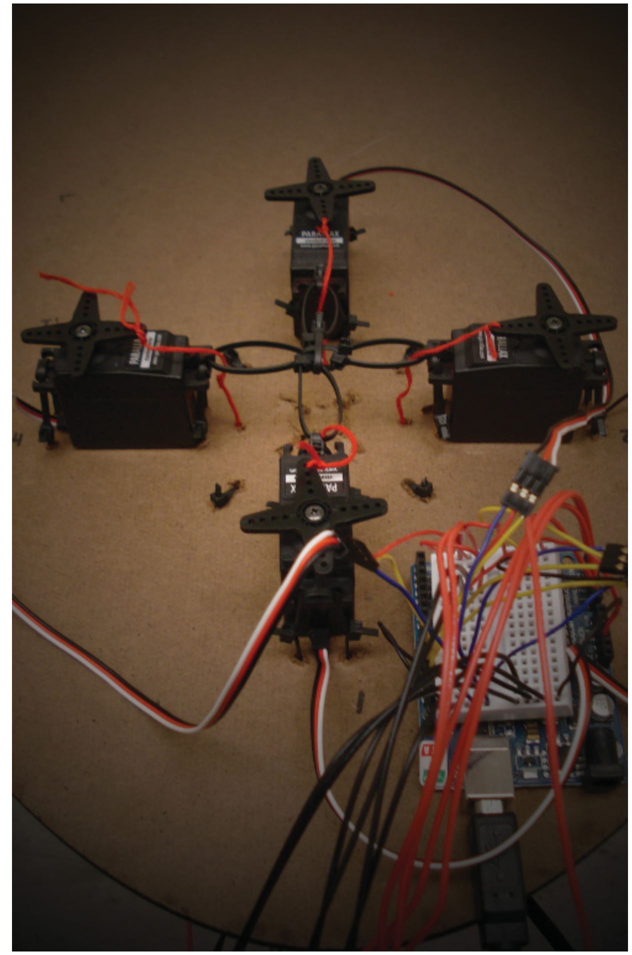
The Orchid revisits similar themes of light-responsive environments as the previous prototypes. However, Orchid also evaluates two other criteria. First, dynamic form is produced through malleable motion construct (HDPE) and multi axial forces. The high density polyethylene (HDPE) allows for live hinges and compound surface geometries in response to no uniform forces. Second, Orchid explores what Mary-Lou Maher and Kythryn Merrick described as a 'rare occurrence,' which in this case is low light levels. The screen saver mode is a way of inviting response. The piece is no longer reactive, but rather it becomes interactive in the sense that it produces a movement that is without an easily perceived actor or cause.

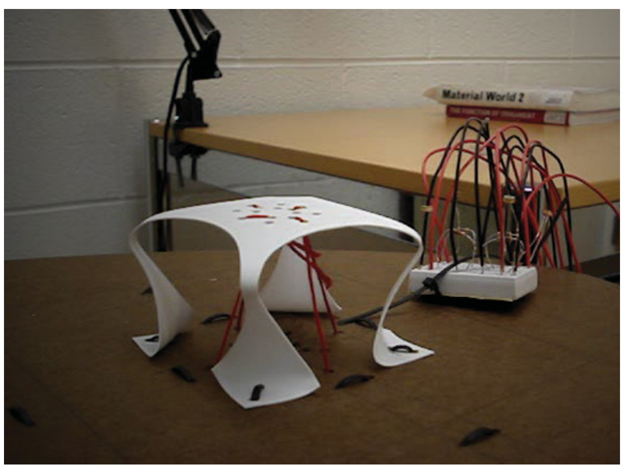
Figure 62 top-left, images of Orchid's motion construct and photocell sensors (4).

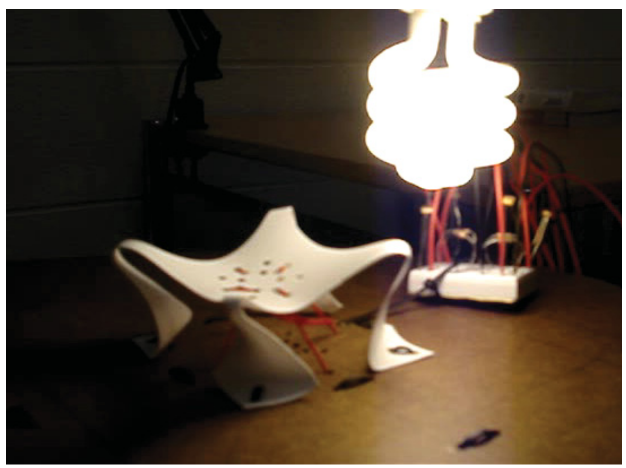
Figure 63 top-right, image of servomechanisms and Arduino

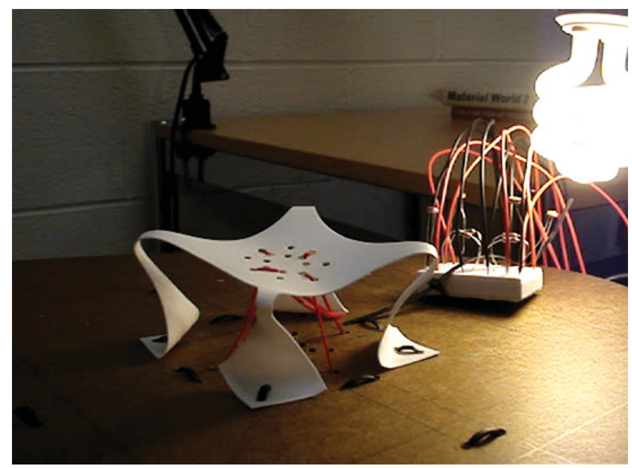
Figure 64 bottom, image series depicting the Orchid in a static state, then shifting away from the source of light.

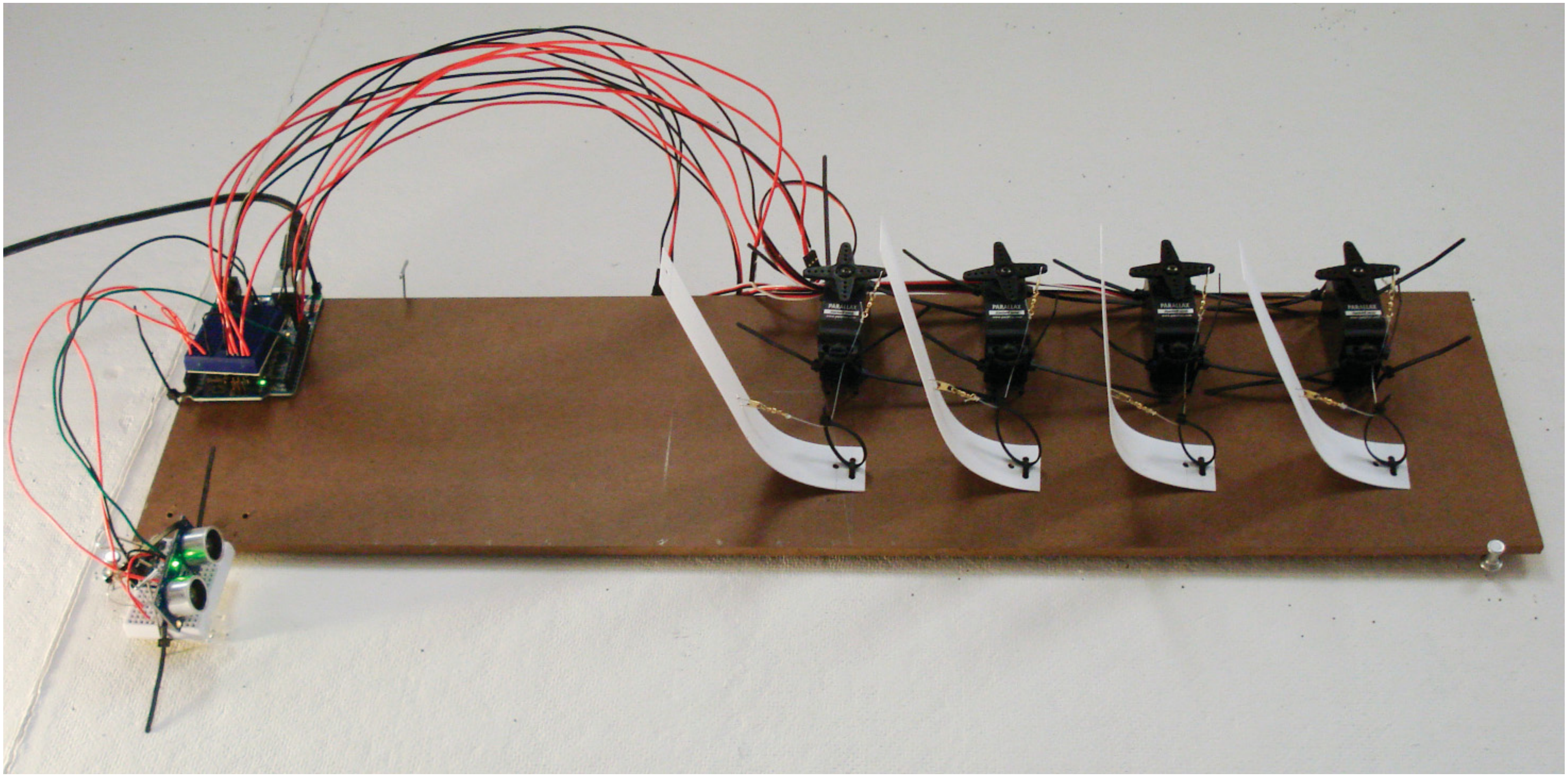


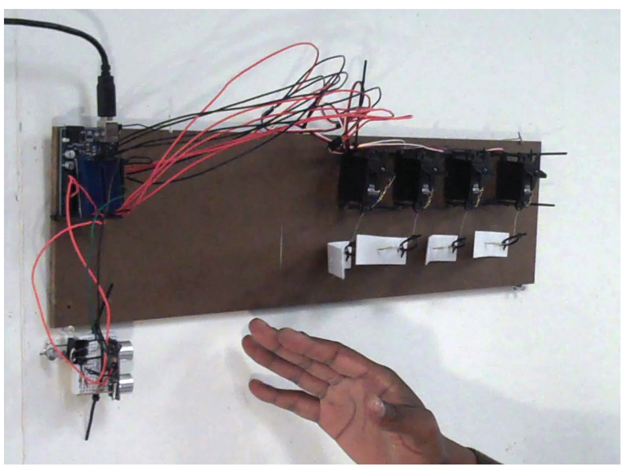


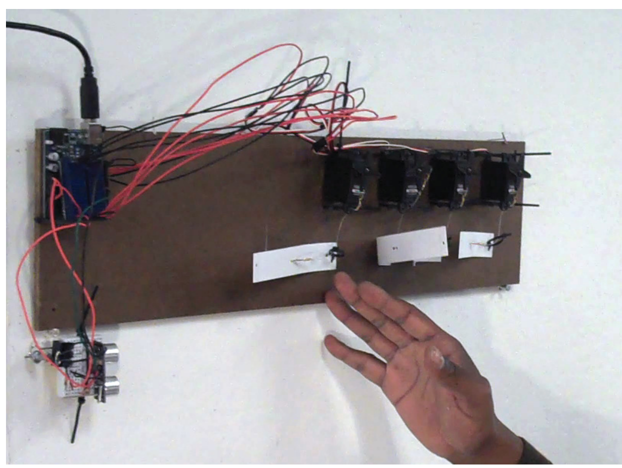

Material World 2




Material World 2


PARALLAX





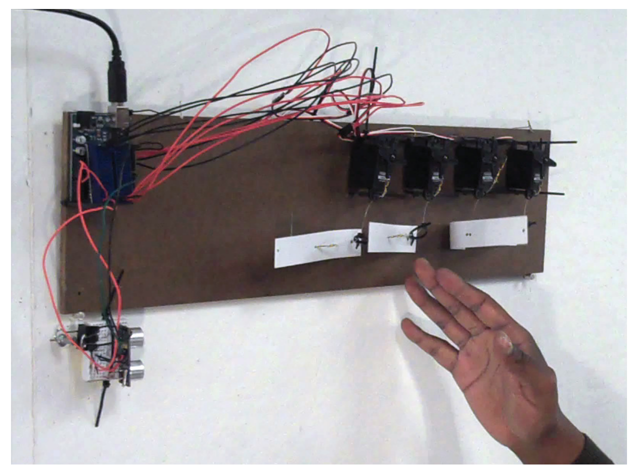


Figure 65 top, image of Grass opening.

Figure 66 bottom, image series of blades of Grass actuating (opening) in response to the detection of a body in the adjacent zone.

Grass

Grass explores the actor and actions found within Rob Ley and Joshua Stein's Reef series. Grass implements an ultrasonic 'Ping' range finder, Arduino microprocessor, a photocell and four servomotors. The use of a 'Ping' range finder provides an easy to use interface. First, the sensor is built with a fully incorporated circuit board that requires no additional parts. Second, like most 'off-the-shelf' hardware, the commonly used range finder can be coded through the use of an Arduino library. The range finder is capable of detecting an actor (in this case a hand) within a range of 0 to 150 centimeters. The distance is then mapped through a series of condition statements to four zones along a linear axis that corresponds to adjacent blades of HDPE grass. When the code detects an actor within a given zone, it calls the adjacent servomotor to rotate—pulling the grass toward the actor. The response is a simple on / off response; the grass remains 'open' or extended until the actor is removed from the zone. If no actor can be detected, the code's structure reverts back to the detection of light level. Lower light levels produce an 'opening' response inversely proportional to the amount of light in the space. When the code detects light levels below 10% a screen saver mode is triggered which produces a movement similar to grass blowing in the wind. This behavior is produced through time delays between on/off signals. By overlapping the on period for neighboring blades of grass, the movement can be expressed as a sin wave in sections.

Time becomes an important measurement of motion. Within appliance architecture, geometric descriptors, such as length, are measured through time or gradients of on/off signals that result in motion.

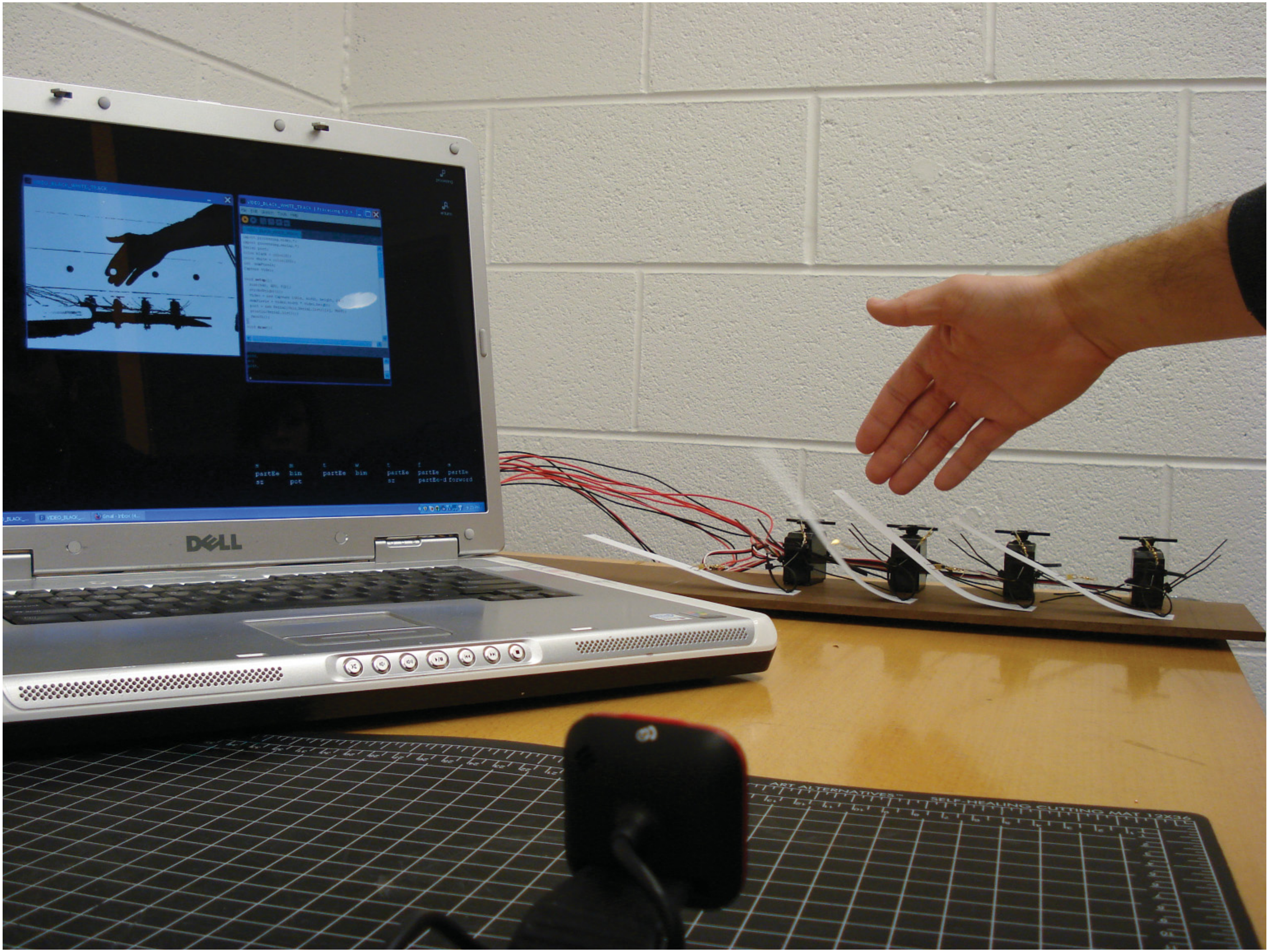

DELL
ART ALTERNATIVES
SELF-HEALING CUTTING MAT 12X36

Figure 67 image of GrassTwo. Image depicts the 'optical button' interface designed in Processing (left), the 'off-the-shelf' web camera (front) and the actuation of GrassTWO when the optical button is triggered (right).

GrassTWO

GrassTWO is a subsequent study of the previous prototype. Originally, the prototype sought to explore the logic and behavior described above through a two-dimensional matrix of grass blades. A series of tests were conducted using two range finders arranged perpendicularly to one another at an equal distance along the 'x' and 'y' axes of a horizontal plane. These experiments showed that the range finders were an inefficient system for locating an actor within a two-dimensional field. The ultrasonic 'ping' produces a signal with a limited width, approximately 20 degrees off center in either direction. The limited width provides a small square area of approximately 30 centimeters in which the 'x' and 'y' sensor could both detect an actor. These results necessitated the development of a new methodology.

To respond to this need, a gestural interface was adopted. The new interface is what Amanda Parkes describes as 'actuation as embodiment of gesture,' referring to those systems which capture motion directly from the language of the human body (Parkes, 2008). GrassTWO uses video optics to locate an actor within a two-dimensional field by means of an optical button and is a primary example of the use of 'hacking' as a learning technique within the invisible college.

Information gained during the instillation of the Reef project at the Taubman Museum provided a foundation for the use of the optical button. Pylon Technical's motion tracking software used a more complex version of a commonly found motion tracking code that is similar to low-level codes that can be found on the Processing website under the 'Learning' section. The 'Learning' section provides user-generated source codes along with tutorials and explanations of the low-level processes working behind the code. Although the site provides many valuable codes that include a frame-comparison code similar to the Reef software, one singular source code did not provide what was needed to code the optical button. Instead a series of source codes where 'hacked' together, along with original coding, to produce a working code. The three primary codes included a video import code, a black and white threshold code for still images, and a data export code provided by Arduino.

The optical button interface samples pixels generated by an inexpensive 'off-the-shelf' web camera and captures the image using VDIG, a Quicktime plug-in. The Processing code uses a video capture library to import the live image and the code then samples the brightness of each pixel of the captured video. This value (between 0-255) is then referenced against a programmer-defined threshold of 127. If the pixel's value is greater than the threshold value, the pixel is turned white and if the value is less than the threshold, the pixel is turned black, creating a black and white image. At the same time, a two-dimensional array, or matrix, samples the black or white values at specific, programmer-defined pixels (in this case, a 1 by 4 array). The value at each pixel is then 'written' as a series of strings and communicated through the computer's USB serial port. At the other end of the USB cable, the Arduino microprocessor then reads the streaming values and stores them in an array through a loop sequence. These values are assigned to a corresponding servomotor. If the servomotor receives a value of '0' or black, meaning a body is within the area of the corresponding pixel, the servomotor rotates a defined angle, thus opening the corresponding blade of grass.

The coding of the optical button not only provides an example of the importance of 'hacking' as a pedagogical tool, it also provides an example of emergent typologies. The use of simple design prompts can lead to the development of more complex systems due to the internal complexity of behavior and interactive logics. The result is an emergent investigation into the subassemblies of appliance architecture.

Call Your Friends

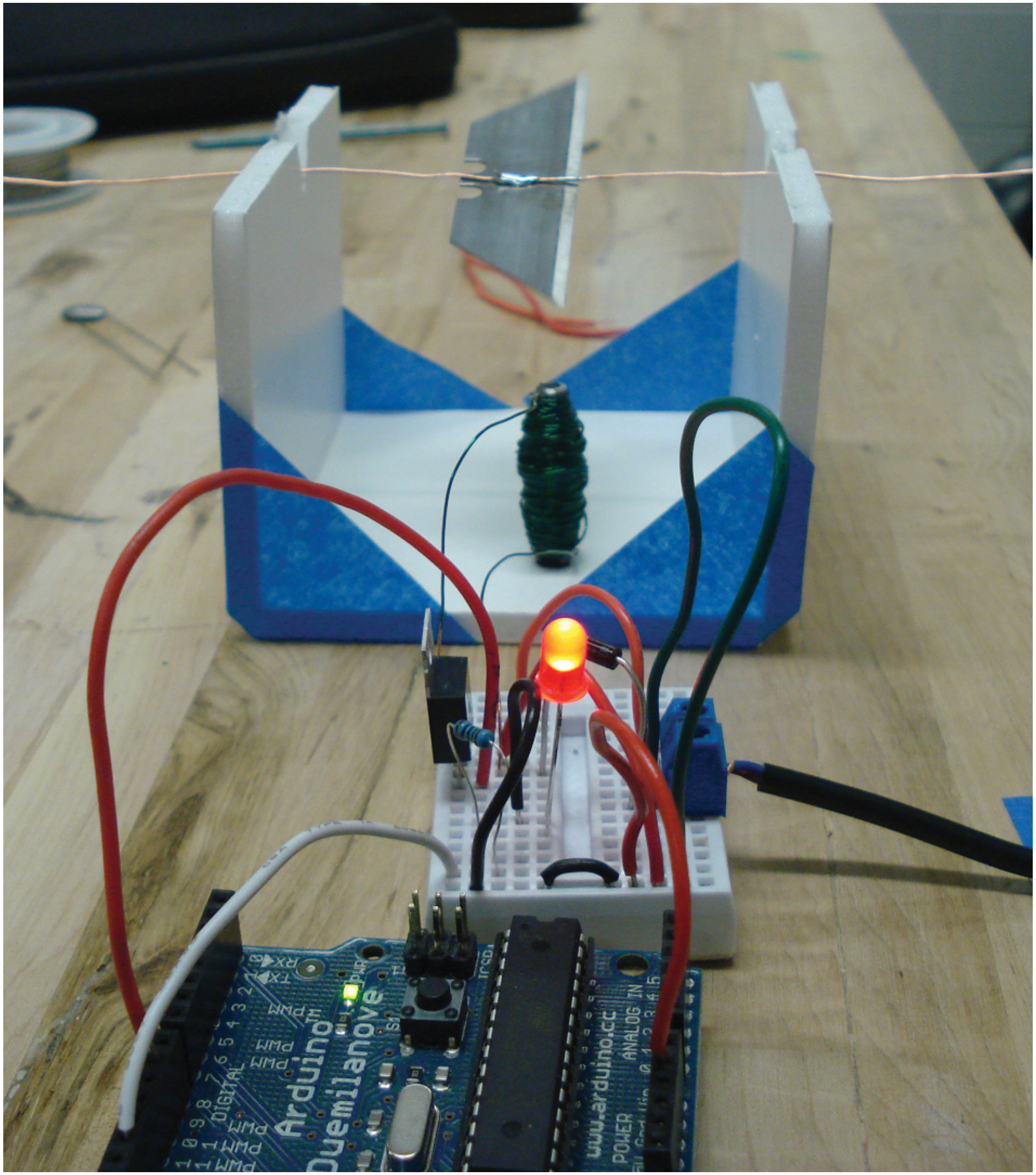
Call Your Friends explores the capacity for actuation through electromagnetic fields. The initial concept was to suspend a metallic disk about a ball and socket joint. This disk holds four earthen magnets in position around its circumference at the 12,3,6 and 9 o'clock positions. Four electromagnets are placed in direct alignment an inch and half below. The movement of the disk is accomplished through a push-pull effect of the electromagnets' polarity which is controlled through the use of an 'H' bridge that allows the Arduino microprocessor to reverse the direction and magnitude of electrical currents though PWM. Each positive face of the earthen magnet is placed in the line of the electromagnets. By placing a negative current at the 12 o'clock electromagnet and a positive current at the 3,6 and 9 o'clock electromagnets, the disk is pulled by the 12 while being pushed by the 3,6 and 9, resulting in a tilting motion. This motion is visually similar to the static tilt of the Eclipsis Screen, which is accomplished through the combination of perpendicular and planar rotations. Intermediate positions such as 1:30 o'clock are accomplished by placing a negative current on the 12 and 3 electromagnets and a positive current on the 6 and 9 electromagnets. Research has revealed promising potential: energy consumption of each electromagnet could be as low as 20 milliamps at 6 volts (.012 watts), which is effectively the same energy consumed by single 5mm 'cool white' LED.

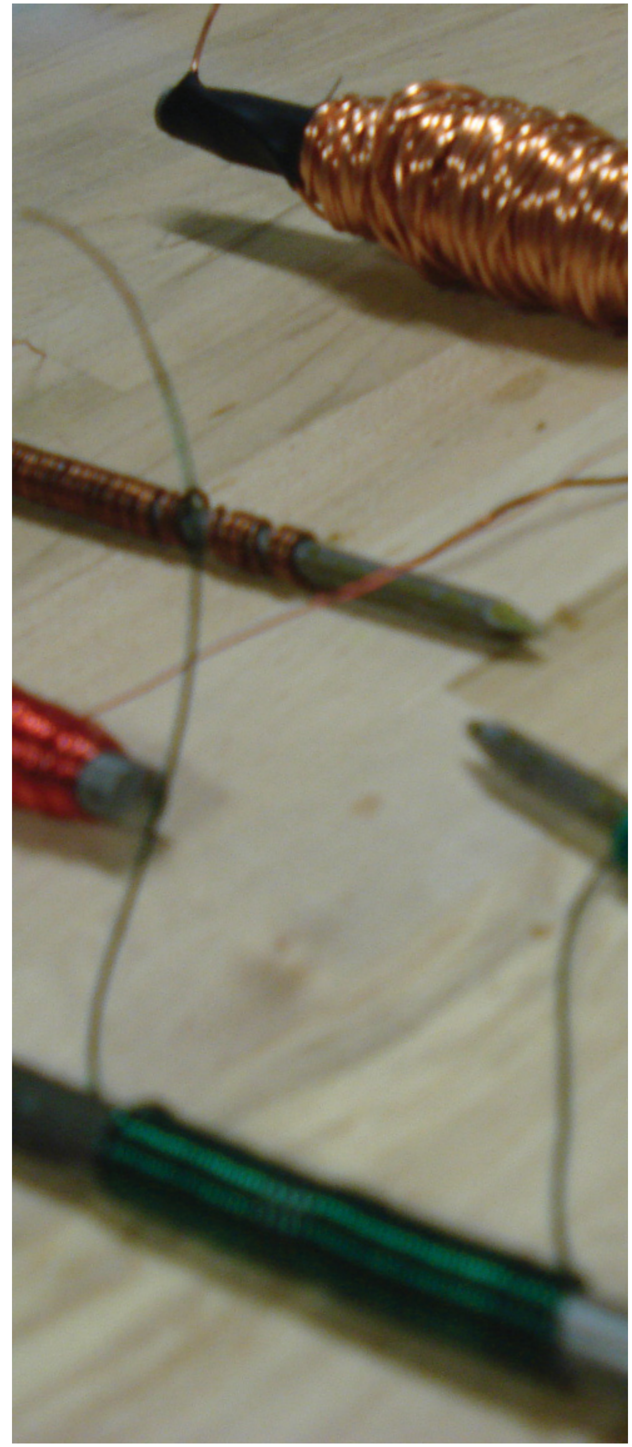
Although a very promising design, only simple prototypes could be developed. Ultimately, the use of electromagnets provided entry barriers that were too high to overcome due to the fact that electromagnets requiring precise engineering. 'Off-the-shelf' magnets, while strong and easy to actuate, are designed as torus geometries with polar ends in close proximity, which impedes the ability to change the magnets' polarity. Call Your Friends required a linear electromagnet, which is not a standard consumer item.

The experiment however, did show the potential for interdisciplinary research. Researchers in the fields of physics, mechanical engineering and electrical engineering lent their time and expertise to the prototype, revealing the necessity for synergies across disciplines.

Figure 68 left, image of simple construct for Call Your Friends. Construct consisted of box cutter blade that pivots about a central axis. When the electro magnet is turn 'on' (indicated by the red LED). The blade tilts toward the electromagnet, but does not come in contact with the electromagnet.

Figure 69 right, image of various DIY electromagnets.


Arduino
Duemilanove
www.arduino.cc




Microsoft

FLOWer

Biology, and more specifically biomimicry, provides a fundamental natural history and concept inspiration for appliance architecture. An organism is successful partly because it uses the minimum amount of material to make its structure and partly because it can then optimize its reaction to the local environment. The more of its environment that it can control and utilize for energy gain, the more successful the organism will be. In the case of FLOWer, the functional typology of a light-mitigating surface was first established. The design concept was inspired by the Christmas tree worm—a cone-shaped worm found on tropical coral reefs that retracts into its burrow in response to the slightest touch or shadow.

Behavioral logic is the digital memory stored on the processing unit of an appliance architecture system. This logic does not simply describe the rational for the motion construct's physicality, but rather it produces a construct intended to evoke an emotional response to the appliance from its user. The understanding of behavioral complexity is examined through agent models that describe systems able to perceive their environment through sensory-data, reason about the data and affect the perceived environment (Maher & Merrick, 2005). In the case of FLOWer, a swarm intelligence model is used to describe the basic response of the system. Swarm intelligence produces collective behaviors of unsophisticated agents interacting locally within their environment, causing coherent, functional global patterns to emerge (Maher & Merrick, 2005).

Once a response and action can be evoked from a system, a new level of architectural design emerges. The development of parametric models through Grasshopper allows for simulation and prototyping. Sensory data can be read by the model through the microprocessor and then modeled in real time. Parametric

Figure 70 image of FLOWer.

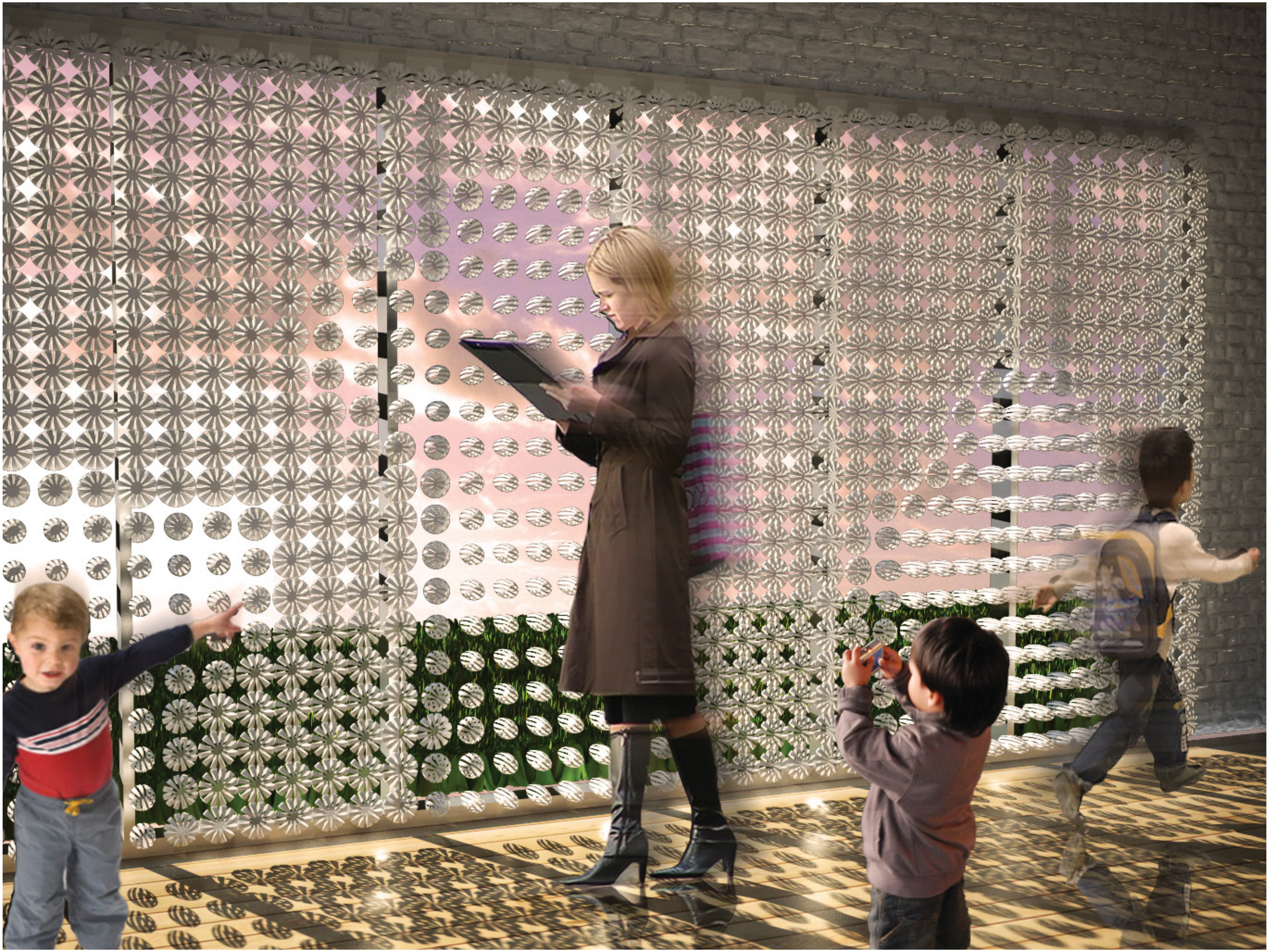
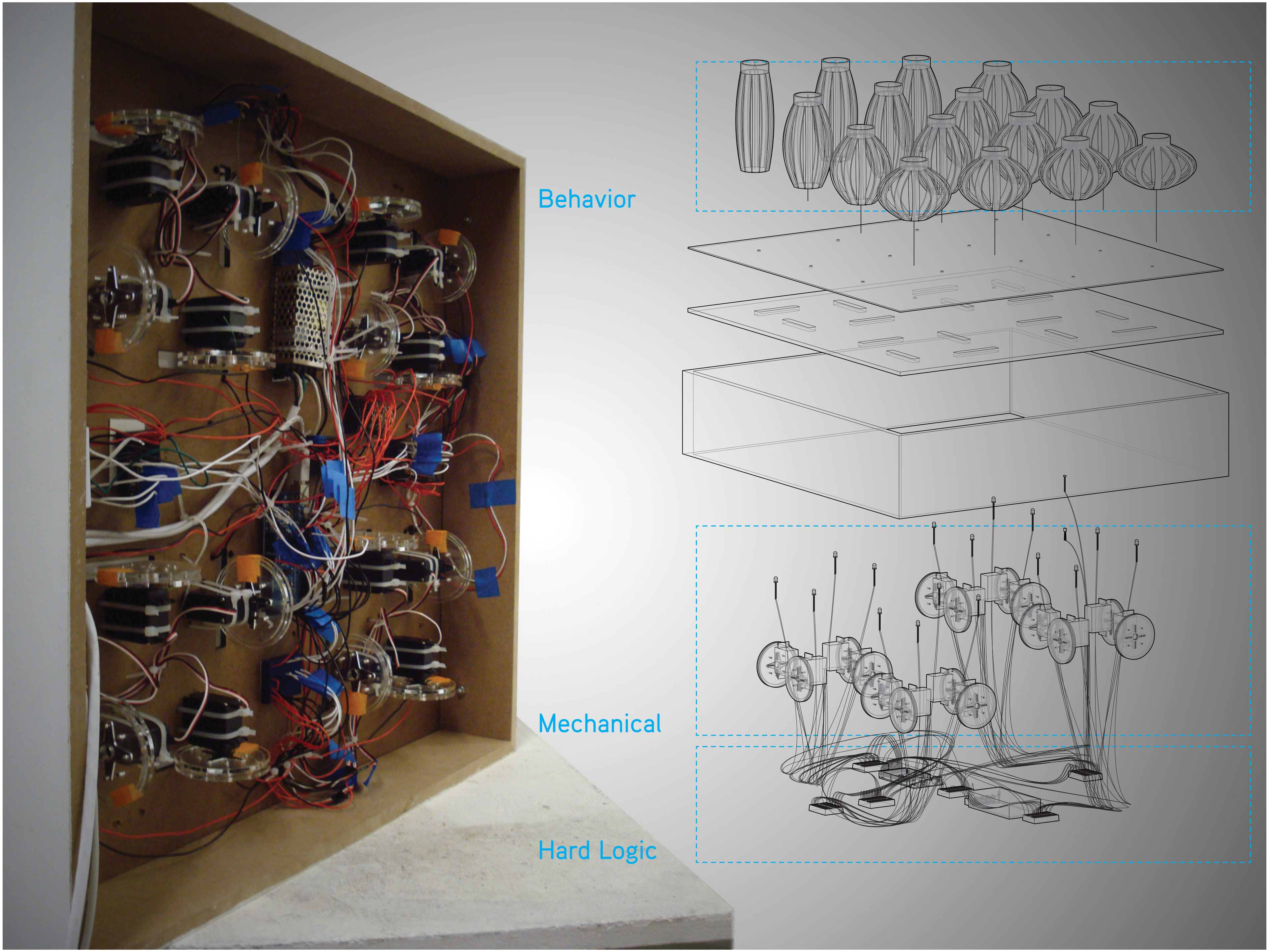


Figure 71 montage of FLOWer's architectural scenario. The image illustrates the potential of an aggregation of FLOWer units to produce a fluid, real-time fenestration for its users.

and kinetic relationships can be explored in real time, allowing designers the ability to virtually test designs using onsite sensors with real time data. In the case of FLOWer, the inherent physical quality of elasticity and memory of the felt fabric used in its motion construct, along with a described geometry, produced by CNC laser cutting produces an almost unforeseeable physical emergence. FLOWer no longer resides within the rigidity of computer software. Instead the physical motion and timing of the agents produce an amiable, lifelike reaction that could only be described through the physicality of the felt fabric.

FLOWer: Architectural Scenarios

FLOWer is designed under the primary scenario of a light-mitigating surface that reduces the solar heat gains falling on a building's façade. As a building envelope assembly, the individual agents respond through a swam-like logic to localized photocells that measure the light levels falling on each unit. Throughout the day, the units of FLOWer blossom with increased light. In the low morning light and then again at night, the units are extended out as cylindrical tubes by the elasticity and memory of the felt fabric and the minimal area of the elongated cylinder produces a porous surface condition allowing for views outward as well as views inward. As light levels increase, FLOWer's mechanical behavior contracts the felt cylinders so as to produce a blossoming affect that increases the cylinder's surface area. The result is a more opaque surface condition that shades the building's façade. The described swarm logic allows


Behavior
Mechanical
Hard Logic

for the emergence of complex patterning in direct relation to isolated solar exposure and shadowing from adjacent geometries. Shadows cast by surrounding urban conditions result in the extension of the units in the adjacent portions of the façade, while solar exposure actuates the remaining surface. This phenomenon produces a dynamically changing façade that is free from a linear-preprogrammed design. Patterns develop and morph slowly over days, weeks, and entire seasons.

FLOWer seeks to understand appliance architecture as an interface between users and their environment. The optical button code developed in GrassTWO provides gestural interface for the users of FLOWer. In this scenario, the user may consciously or unconsciously orchestrate the fluid-shifting fenestration. The user's silhouette is impressed upon FLOWer, as localized units expand outwardly when their corresponding 'button' is actuated. The surface reduces solar gains while also producing an isolated view out for the user. The result in an ornamental patterning that emerges and disappears as the building users do.

The described interaction questions how we interface with appliance architecture. In this scenario, the interaction is described as orchestration, and much like a symphony conductor, the user employs body motion to activate isolated of parts of larger whole. By removing a tactile one-to-one interaction, the use of a gestural interface evokes an empathetic response—even at its lowest level. The ability for the unit to sense and react to a user evokes a feeling of intel-

Figure 72 image and exploded axonometric of FLOWer's motion construct. Hard logic is the physical space required by the embedded intelligence processors, mechanical is the physical actuation of the logic and behavior is the emergent motion described through the material properties of the motion construct.

ligence or awareness on the part of the units, which results in its human counterparts perceiving a level of anthropomorphism, or what Kostas Terzids described as 'otherness' (Parkes, 2008) (Terzidis, 2006). The removal of a tactile interface allows for the reading of proactive interaction and yet FLOWer also explores the benefits of tactile interface.

The use of interactive design subassemblies required that their utility be transparent and malleable. Much of the media culture within which appliance architecture resides presents data and information through digital mediums such as screens. Through its materiality, architecture has, and invites, a more complex one-to-one interface.

Kinetic memory allows users to physically train the motion of FLOWer. Felt fabric provides a soft, malleable, medium that encourages tactile interaction. Through a series of flex sensors, an action placed on a unit, such as compression, can be mapped proportionally to a rotational degree in the servomotors of the field. This produces a one-on-one replay of the action by the other units of FLOWer. The kinetic memory is analogous to the simple act closing the blinds. Although FLOWer's complexity may be understood as complicated by a user, kinetic memory is a form of user override. If a user wishes to allow more light into a space, they can extend the units of FLOWer by simply pulling out a unit until the desire light level is produced by all the units within the array. Interactions can also be stored in the memory of the Arduino microprocessor. FLOWer allows the user to record an action placed on one of its units in a given period of time. That motion can then be replayed across the entire field. The user may also manipulate the motion in relation to time. The motion itself can be sped up or slowed down. More complexity can be added by manipulating the period of time between the motions of each unit resulting in an emergent patterning that can be described as fluid, pulsing, wave-like or bubbling.

FLOWer also has the potential for thermal conditioning. Subsequent versions could be designed with a thermal mass. The current version uses a temperature sensor to elicit respons-

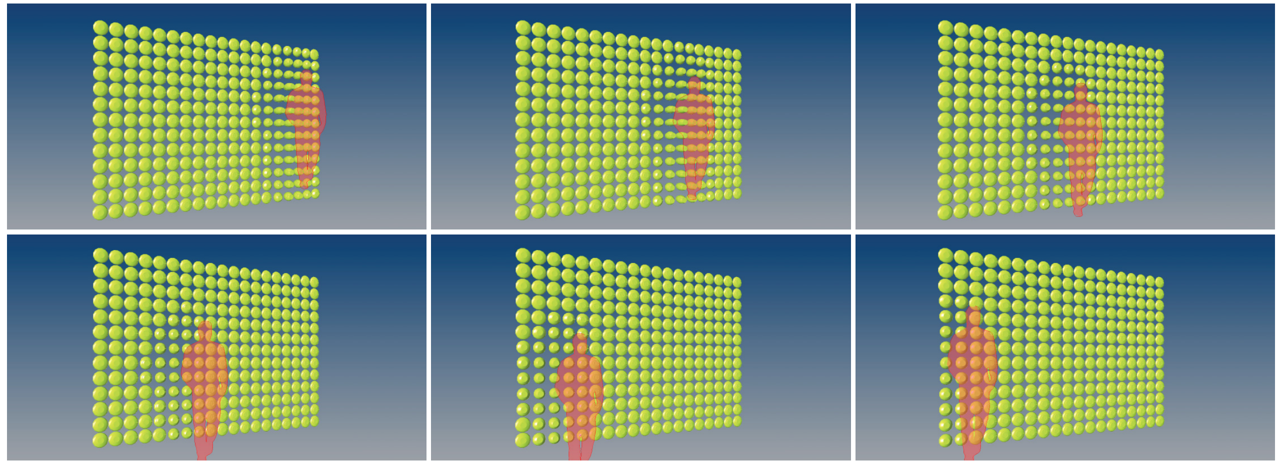


Figure 73 top, images series depicting simulation through Grasshopper. Images describe the potential application of FLOWer within a built environment. A 4 by 4 section of the simulation was also used to actuate the units of FLOWer by a virtual 'actor' within the Rhino 3D environment.

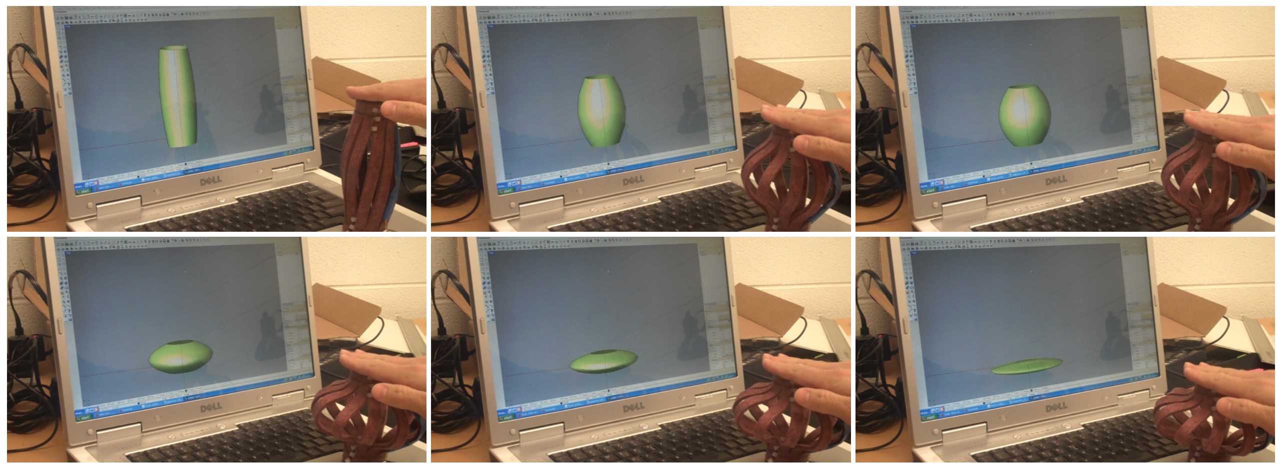


Figure 74 center, images series depicting virtual simulation of real-time sensory date. Data collected by a flex sensor through the Arduino microprocessor is mapped to a virtual unit within the Rhino 3D environment. Simulation of sensory date shows the potential of real-time virtual prototyping and design verisioning.

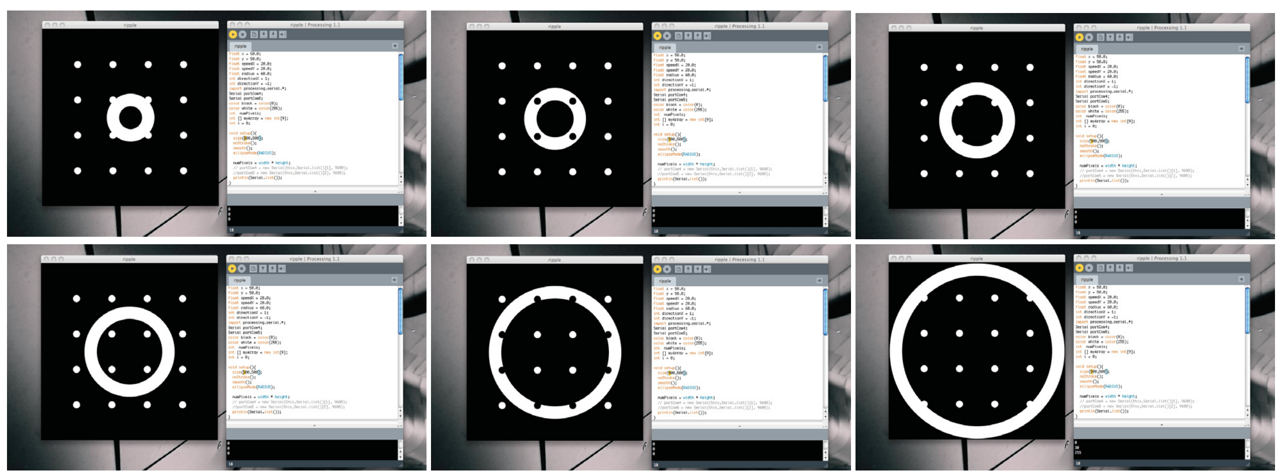
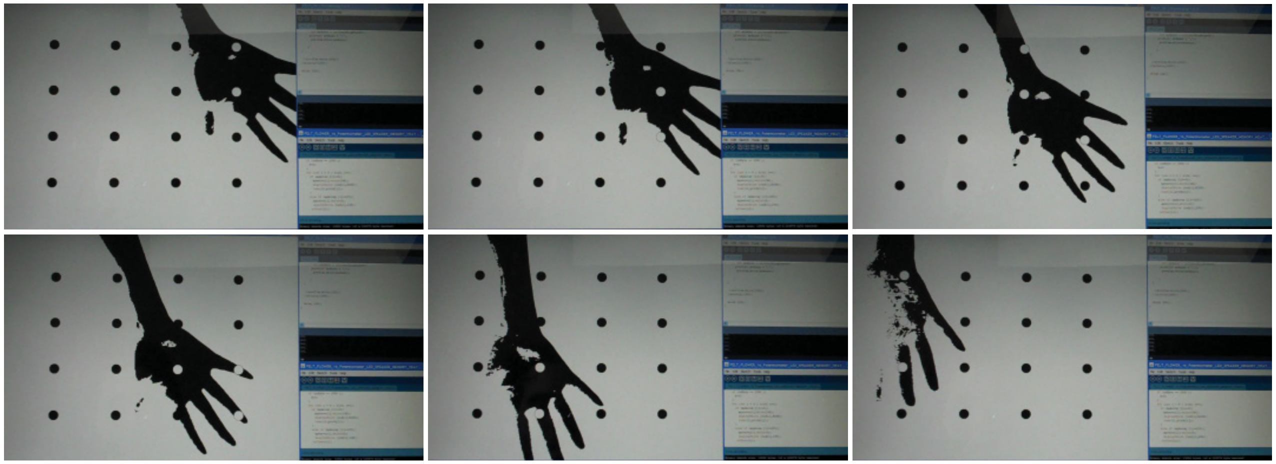
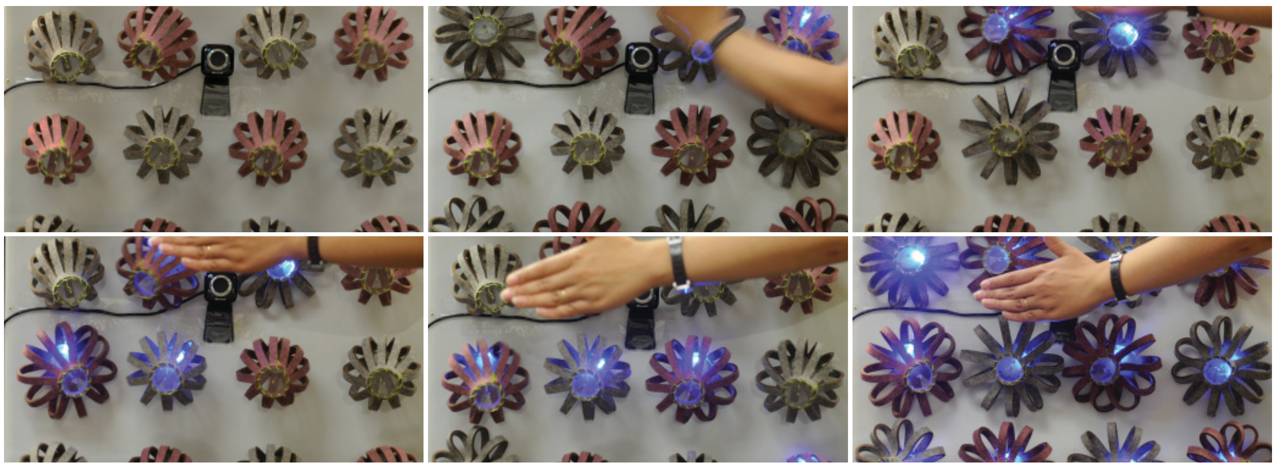


Figure 75 bottom, images series depicting the simple graphic interface within Processing. In this scenario the scaling of a circle actuates the digital buttons, which represent a unit of FLOWer. The simple black and white animation results in a wave / ripple pattern across the surface of FLOWer. The speed and magnitude of the wave can be adjusted based on the speed of the animation.





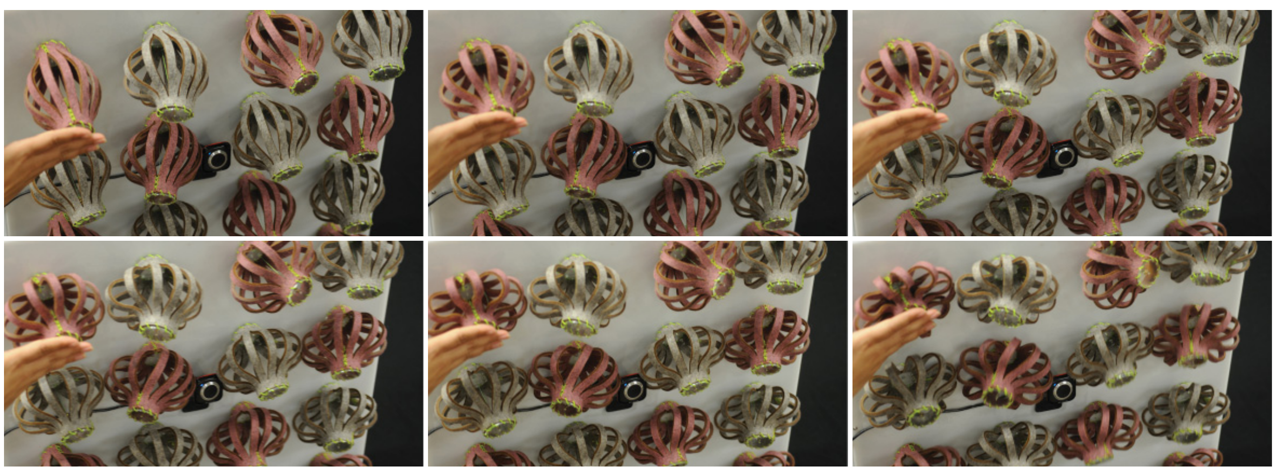


Figure 76 top, image series depicting the 'optical button' interface in Processing. The processing code maps the contrast and shadow in the live video feed to a black and white image. If a body is detected over a pixel within the given two-dimensional array the button turns white and sends an 'on' signal to the Arduino microprocessor.

Figure 77 center, images series depicting the reaction of FLOWer's units. In this scenario the corresponding units compress and LEDs are turned on when an 'on' signal is communicated by the Processing interface.

Figure 78 bottom, images series depicting the kinetic memory. The amount of compressional force (both magnitude and speed) placed on a receiver unit is mapped proportionally to a rotational degree for each of the other units of FLOWer.

es. Through the use of these sensors, FLOWer can respond to the flow of energy through a space. If energy is flowing out of a space—if warm energy is being pulled from an interior space to a colder exterior condition—FLOWer's units could compress, nesting together to produce a thermal barrier with a potentially high R-value. When a space requires cooling and cross ventilation, the units could extend, producing a porous, breathable, surface.

FLOWer also has the capacity to entertain. Mark Goulthrope's Hyposurface, a facetted surface able to articulate complex shapes through hundreds of hydraulic actuators, is described as one of the most important developments in signage technology (Goulthrope, 2006). Given its ability to shape highly articulated forms, Hyposurface is capable of three-dimensionally representing a given graphic input. Like Hyposurface, FLOWer allows for a graphic override. Using simple monochromic graphics developed in Processing, FLOWer is capable of displaying multiple levels of animated information across the building façade. As stated, in appliance architecture, time is a material description. The use of simple black and white animations can produce topographic patterns across the array of units using code similar to the programming developed for the optical button. These simple animations are low-level learning blocks within the Processing community, yet they are capable of producing sophisticated material translations.

Conclusion

FLOWer provides a medium through which many behavioral logics can be studied. The project represents the culmination of the emergent typologies developed through the bottom-up pedagogy of the PARTeE lab and was used to establish the framework of the invisible college. However, as a building systems application, FLOWer served as a proof-of-concept. The following project was designed to simplify, refine and edit these concepts into a real-world working model.

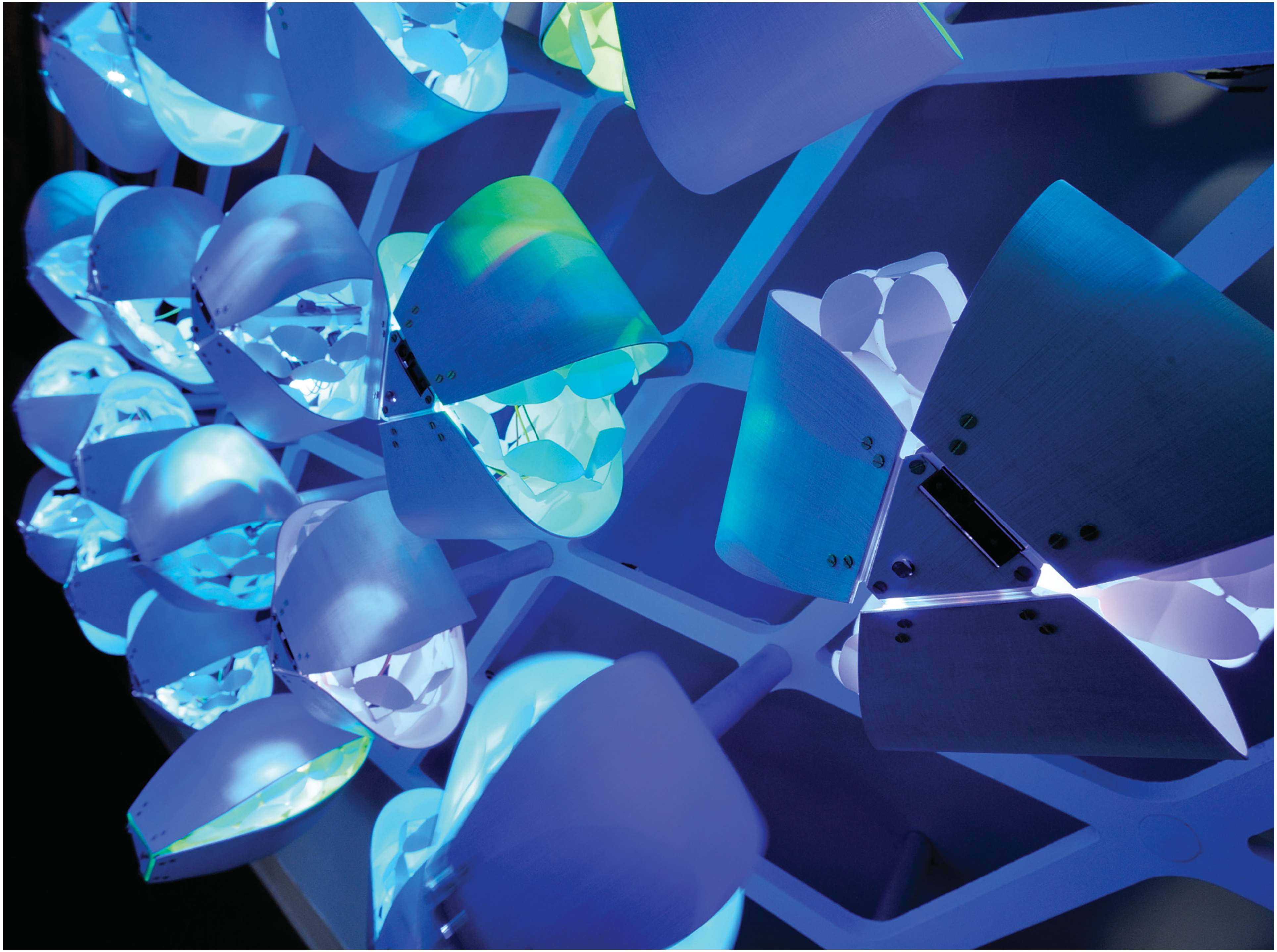


Figure 79 previous page, image of wires cluster.

Figure 80 night image of 'a2o.' Image by Jim Stroup

Figure 81 next page, left, image of gradiated response of a2o's units to localized photocells.

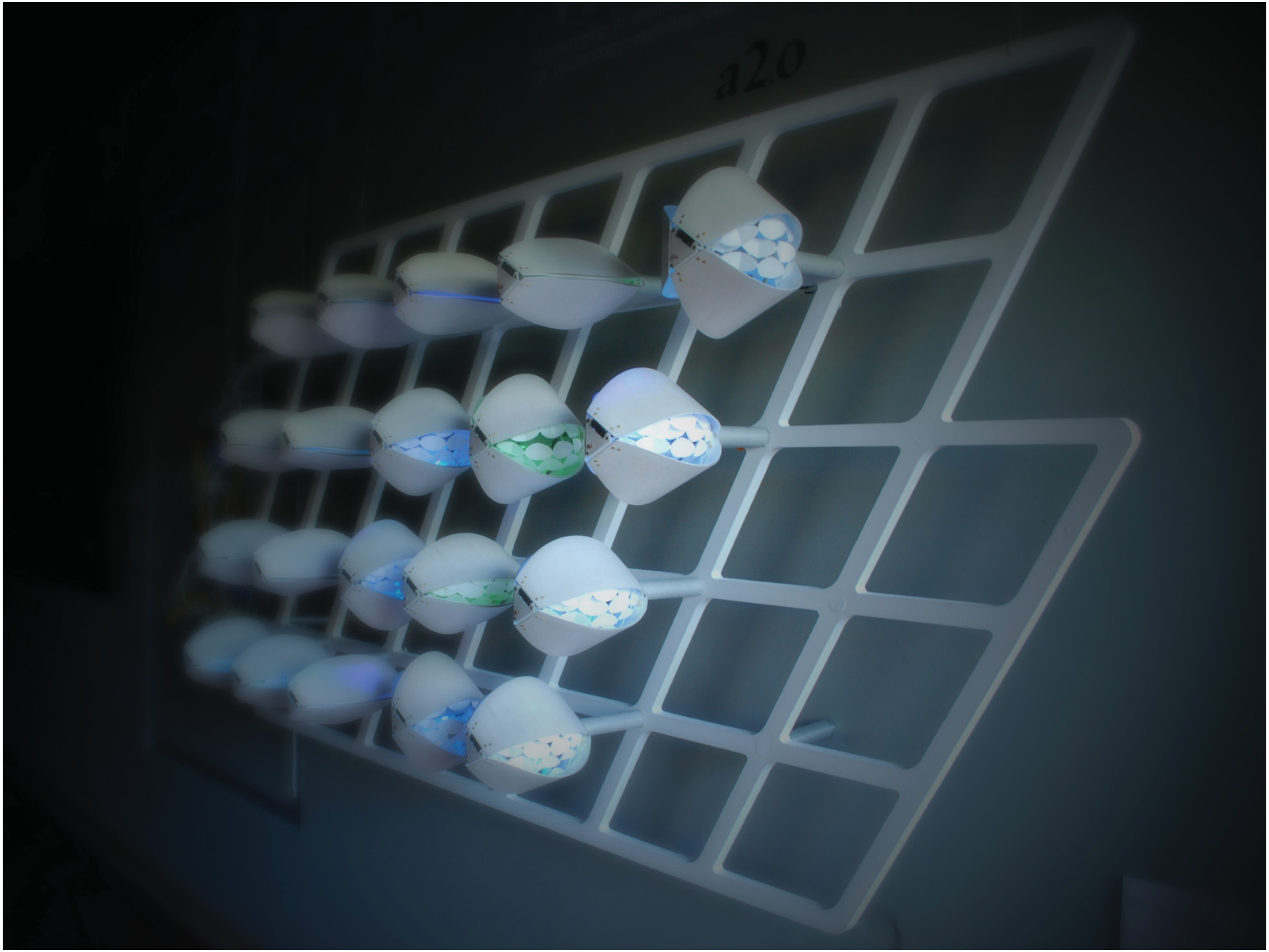
Figure 82 next page, right, image of interaction of a2o's units through a gestural interface.

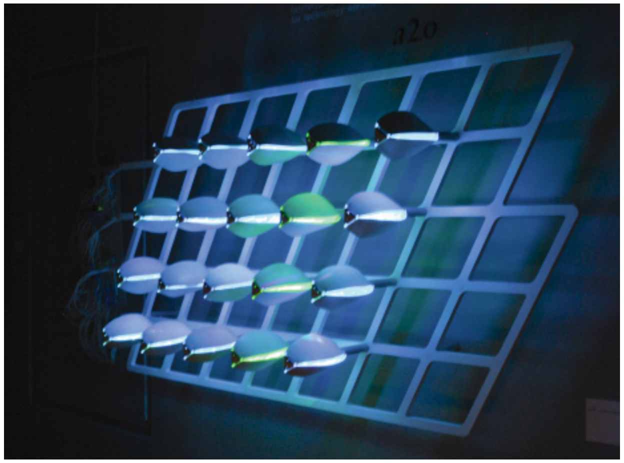
Audrey Twenty (a2o)

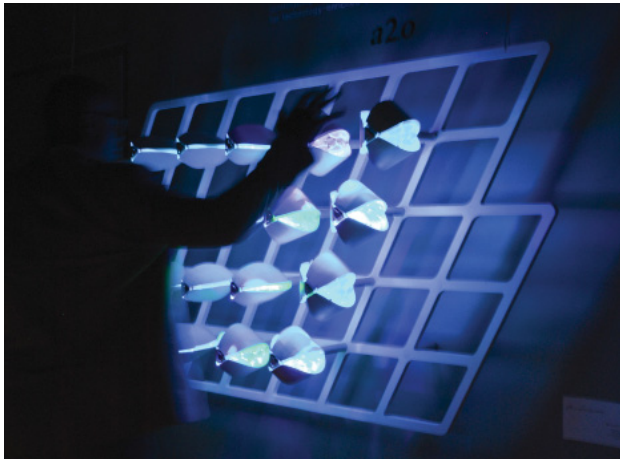
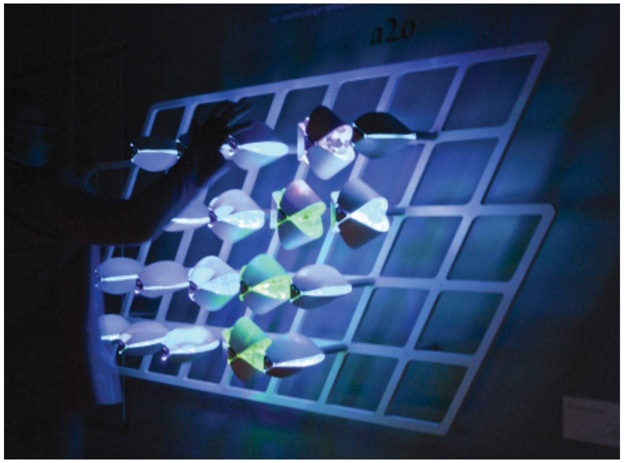
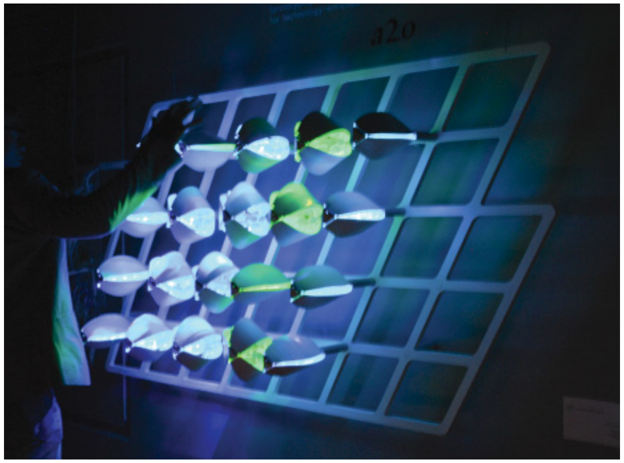
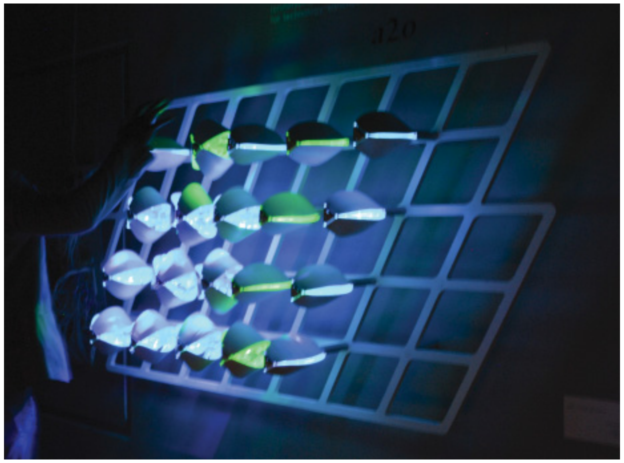
a2o [ey- too- oh], aptly named after her 'Little Shop of Horrors' doppelganger Audrey 2, is a responsive interface commissioned by Virginia Tech's School of Visual Art's Experiential Gallery in downtown Blacksburg, Virginia.

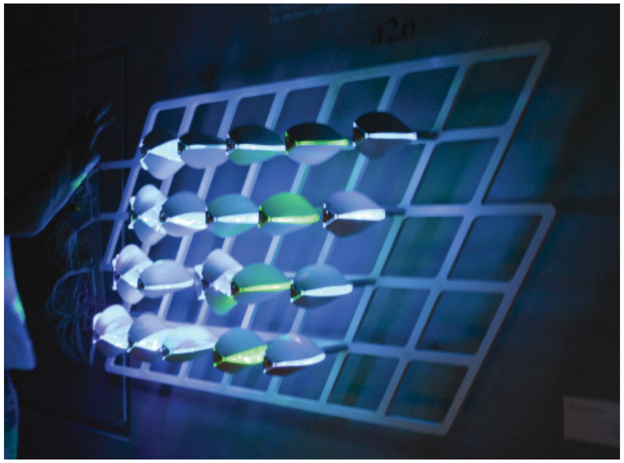
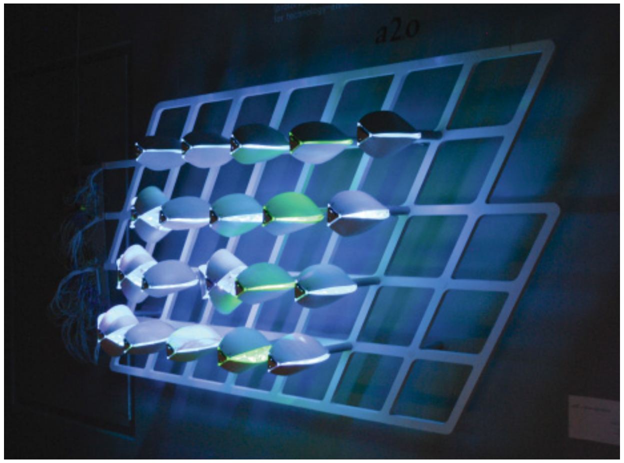
Similar to FLOWer, a2o was designed under the thematic of a sun-shading surface assembly. FLOWer's material and behavioral logic provided many key points that required new approaches and methodologies. Although thematically a shading device FLOWer's the motion construct, the servomechanism for movement, were housed within an opaque frame, impeding its ability to be understood as a permeable wall assembly. These parts and their connections (especially the felt fabric) were also unreliable due to fatigue. Having been commission for instillation at a gallery, a2o required that the motion construct be simple, for this reason an of 'off-the-shelf' linear actuator was used to provide structured motion. Furthermore, although the optical button is an elegant and simple solution, it had two conflicting issues. First, light levels within the gallery space are in constant flux, therefore requiring a variable threshold value. Second, the optical button needed a personal computer and serial connection to the prototype, which consequently required that the gallery staff be able to operate the Processing software if power were lost or complications arose. The scale of the Arduino microprocessors allowed for a de-centralization of computing processes. The ability to embed intelligence into a physical construct is a major facet of the capacity of the appliance; therefore a more autonomous, self-contained processing configuration was desired. These criteria provided a conditioning that would also give rise to a more real-world, autonomous architecture.

Within appliance architecture, the appliance nomenclature denotes a consumer good or product. In order to establish a simple, self-contained system, a plug-and-play nodal design was adopted. A nodal design allowed a2o to have


a2.0



a small profile in elevation as well as a singular physical connection to the system's framework. The use of a nodal system also shifted the understanding of a2o from an architecture that required a host building to an architectural product that was an expandable piece/part system. The team envisioned a bottom-up design for the motion and logic constructs within a top-down processing configuration. Each unit contains dedicated sensors [infrared range finder and photocell] and dedicated actuators [linear actuator, RGB LED, and Piezo buzzer]. Sensory data collected by each individual unit is relayed to a master controller—in this case an Arduino microprocessor—which controlled a pixel of five units. The master controller would then describe an action for the individual units to perform in direct relationship to the sensory data it collected. If the master controller recognized a specific set of data—in this case no data—it could then describe a top-down response for all of the units within its pixel to perform. This logic structure, described as cellular automaton, allows for the piece/part system to be expanded infinitely as each pixel within the system becomes a subassembly within the subsequent pixel.

a2o: Architectural Scenario

A field of extended a2o units is arrayed across a building's southern curtain wall. The morning sun pears around the adjacent building and each unit slowly compresses, blossoming proportionally to the amount of light it senses. The result is an emergent texture driven by the light and shadows of the surrounding environment. As the day continues, the units compress to a fully open state, nesting within one anther to produce an opaque surface, fully shading the space within. The building occupants arrive and the façade takes on a new aesthetic - the building becomes a living *poché*. As the user(s) circulate through the space, their silhouettes are impressed upon the field as fenestrations moving through the façade. The individual units extend out from their contracted state when their localized distance sensor perceives a user(s) in the space. The result is an isolated view for the user(s) while the adjacent space remains shaded. As the sun falls, the units have completed their daily task and return to an extended state. Yet their work is not finished. At night, the a2o units once again seek out their human counter parts. However, in this case, when a user is present, a2o compresses and illuminates, providing the user with light and privacy. a2o also has a logic of its own. As the day slows and its nightly activities subside, a2o recognizes, rather, feels bored. In this situation, if a2o has not sensed a user(s) in the space for a given period of time, it may display a topographic image across its surface, providing a range of information or possibly replaying the more exciting events of the day.

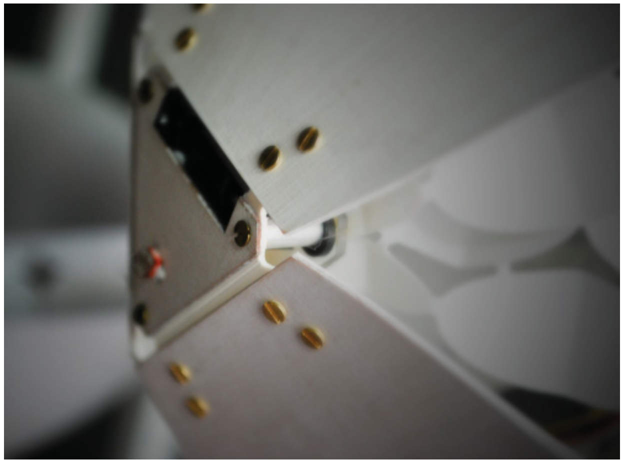
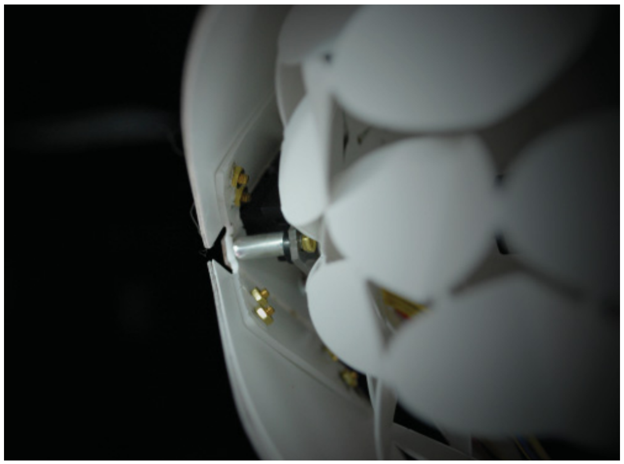
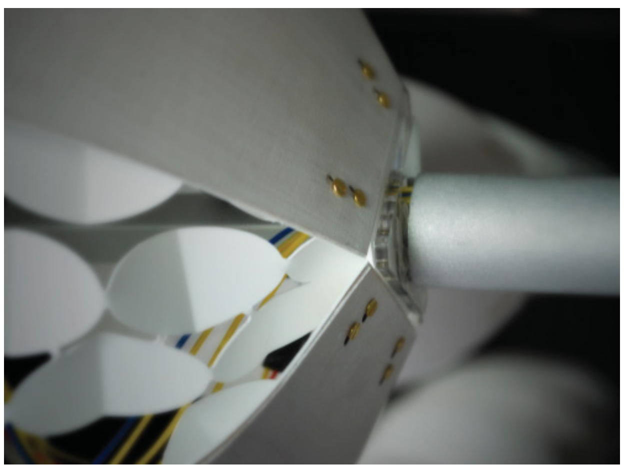
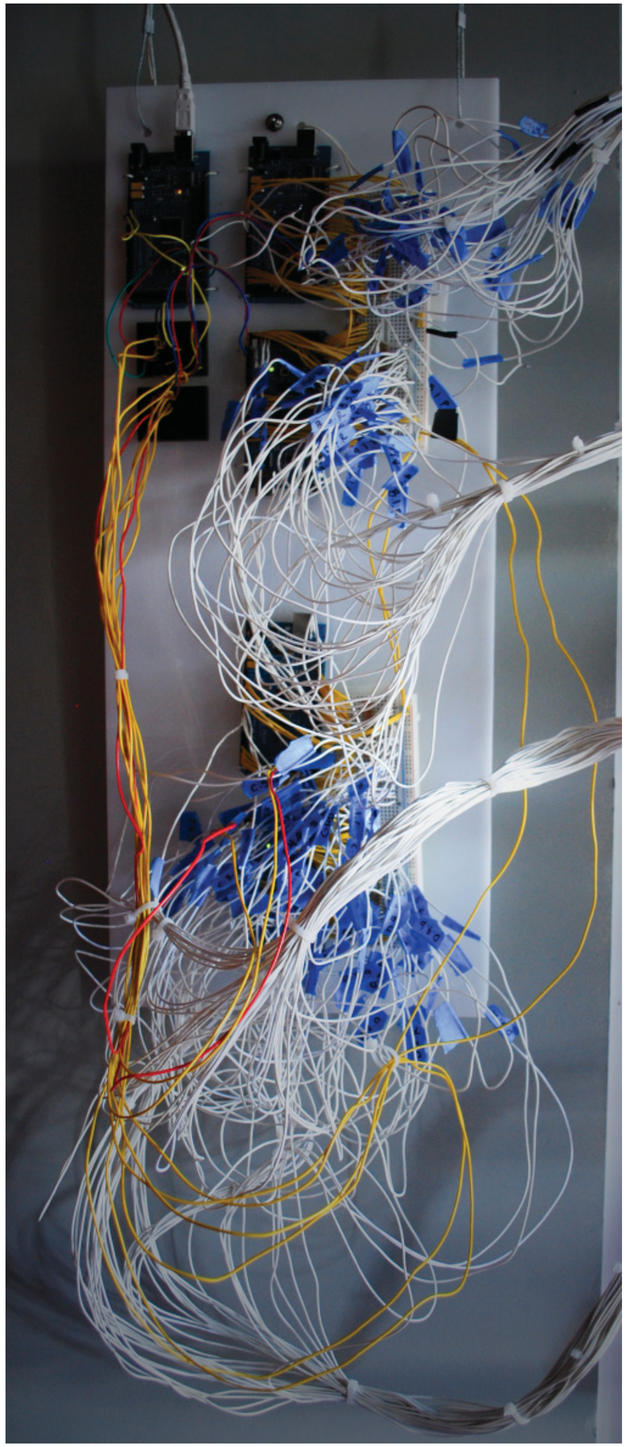
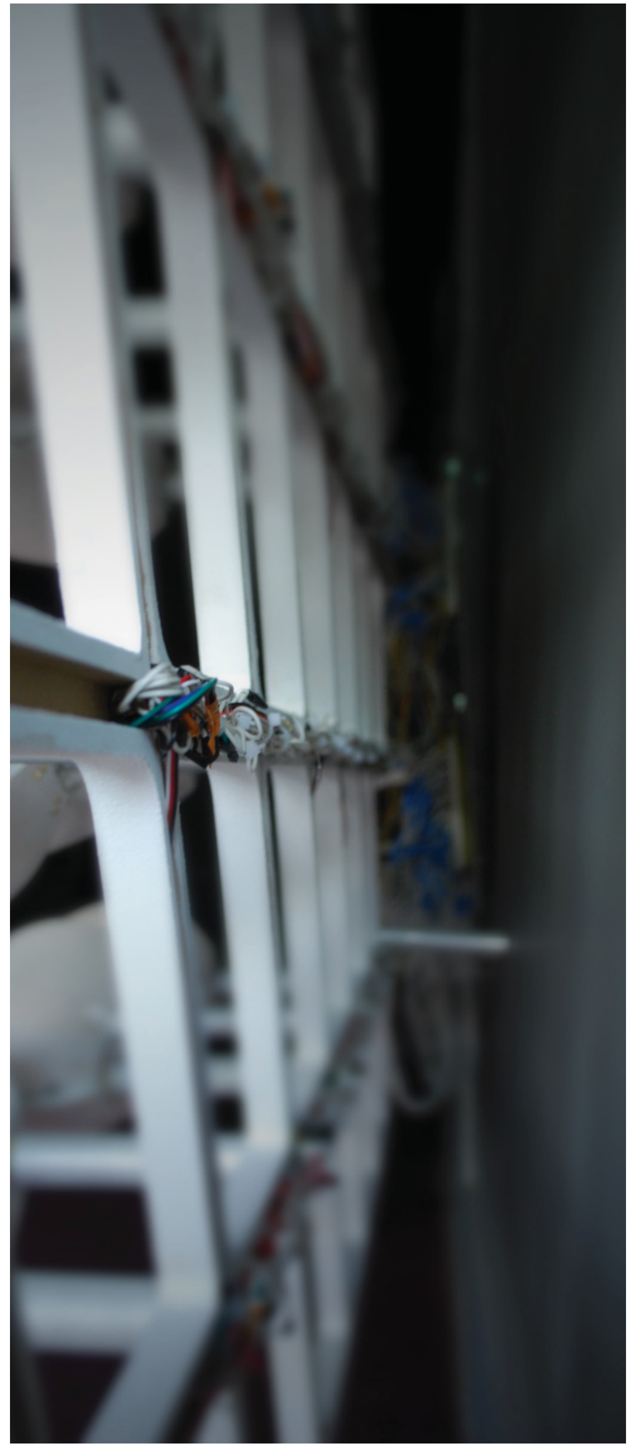
Figure 83 left-top, detailed image of head assembly of a2o. The assembly houses an infrared range finder, photocell and LED. Image also highlights the use of a live-hinge made of HDPE.

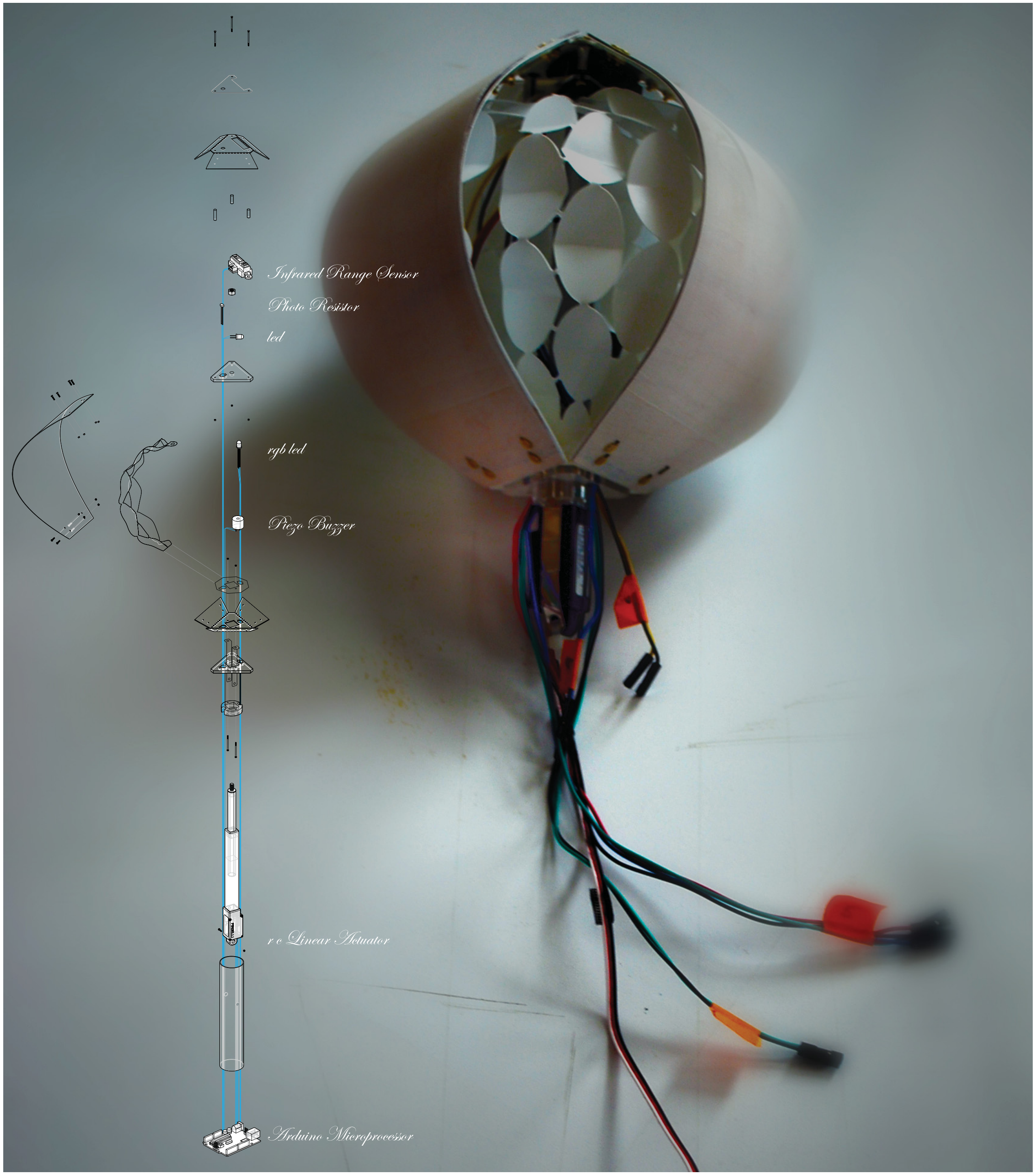
Figure 84 left-center, reverse detailed image of head assembly of a2o.

Figure 85 left-bottom, detail image of nodal assembly.

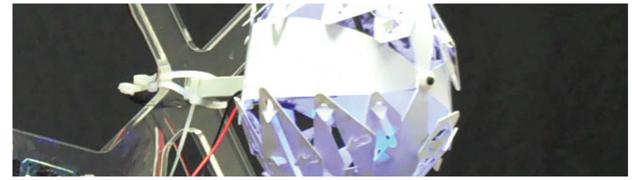
Figure 86 center, image of processing units. Assembly consist of 5 Arduino Mega Microprocessors and a USB serial connection. The PARTeE team decided to leave the hard logic exposed for purposes of visitors education and an expression of complexity.

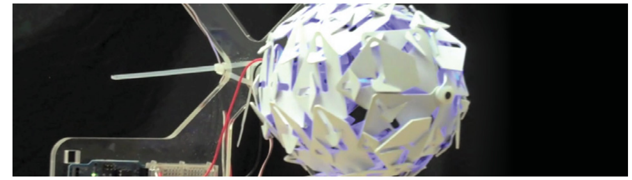
Figure 87 right, rear locking image of a2o frame. The frame was CNC routed from medium density fiberboard (MDF). CNC routing allowed for the production of wiring channel to each node of the assembly.

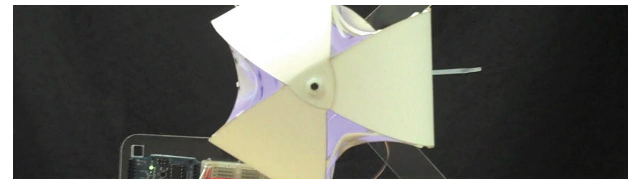






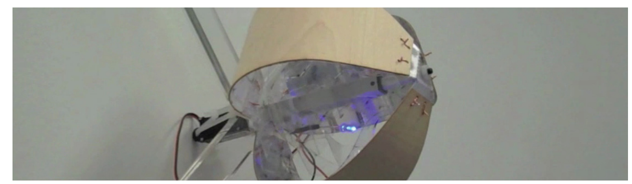


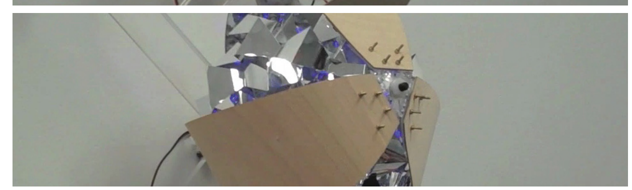
Infrared Range Sensor
Photo Resistor
led
rgb led
Piezo Buzzer
r c Linear Actuator
Arduino Microprocessor

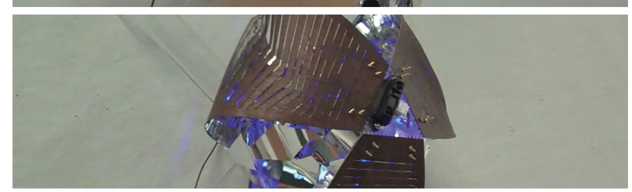


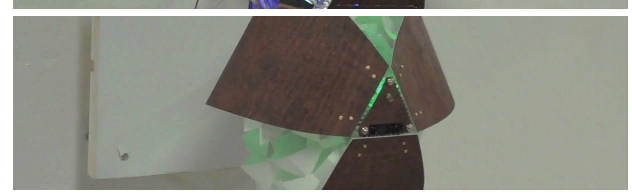












Light Mediator

a2o uses swarm logic to actively shade space. Through localized photocells, a2o measures the light levels falling on individual units. Throughout the day, the units blossom with increased light, increasing their profile and in turn reducing solar gains. The result is an ever-changing gradient, which is responsive to physical adjacencies and time.

Human Mediator

a2o allows users to become units within a parametric system, able to actively influence their environment. Through the use of localized infrared distance sensors, a2o uses gestural interface to allow for controlled mitigation of solar gains while also producing isolated views and privacy.

Time Mediator

a2o uses a model that is similar to a low-level motivated agent model that allows for time to fold and bend upon itself. a2o is capable of being self-aware. If it gets bored - if the system has not been actuated within a given time - a2o can react with a preprogrammed surface pattern. This pattern is accomplished through the use of simple time delays to describe physical motion. Each unit's time on and off relative to its adjacent units allows for fluid-like movement. In the future, a2o will be able to play back more exciting interactions recorded throughout the day.

Material

Kinetic transformation presents many formal opportunities. a2o sought to produce a phenomenological transformation. This transformation is an example appliance architecture's ability to express the uncanny, or *Das Unheimliche*, which is literally translated as 'un-home-ly.' The use of a plywood shell presents an original reading of a rigid, yet soft, furniture-like, application. When the unit compresses, the formal transformation presents the illumination of a fractal ornamentation that can be read as organic and amorphic. In this way, the aggregation of units takes on a formal and material vocabulary that is in constant flux throughout the day.

Subassemblies

As a gallery instillation, a2o employs a series of subassemblies to illicit its users to reflect amorphic and anthropomorphic emotions onto its units. Through a series of delays and pitch 'libraries,' the units of a2o produce a flocking 'noise.' As users interact with the units, they orchestrate a 16-bit wave of sounds that could be read as the unit's conversing or communicating with one another, and the result is a flocking and fleeing interpretation of their response.

Figure 88 left, montage of the subassemblies of single unit of a2o.

Figure 89 right, image series of a2o's design versioning.

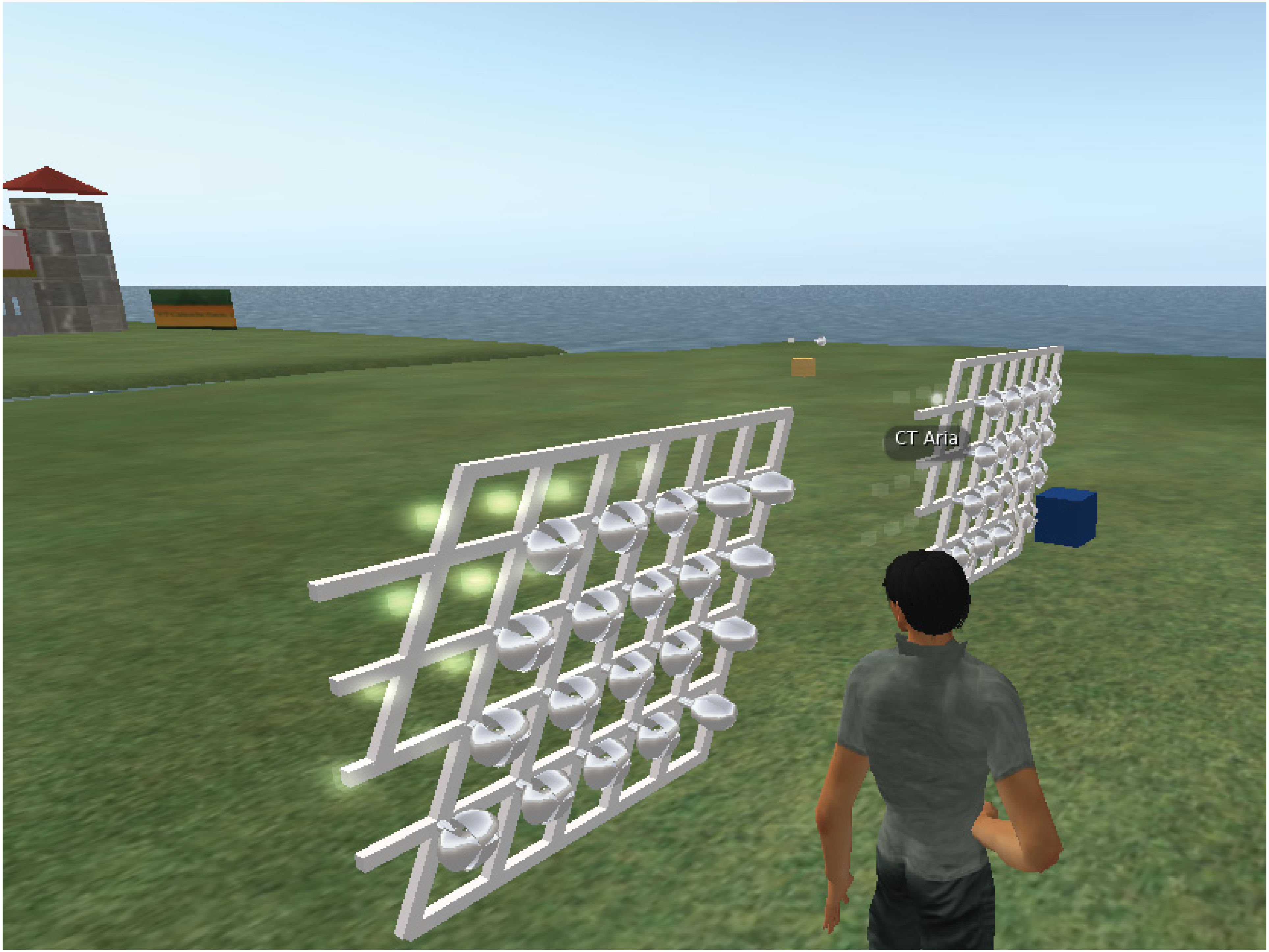

CT Aria

Figure 90 left, image of a2o in Second Life. Through the Second Life virtual environment users (Avatars) can experience a2o's real-world reactions in real-time. Image by Spencer Lee.

Virtual Interface

Through a serial connection, a2o uses Pachube, a real time internet data host, to connect to Second Life. Users can interact with a2o virtually with Avatars or see a2o's response within the Second Life environment to real world stimuli. a2o in Second Life asks how will we virtually interact with physical environments in the future?

Conclusion

As architecture seeks out a post-digital 'ism,' and realizes the tools that have been developed for architects have allowed its design process to become analogous to those of fashion and the media-centric world it resides in, a2o and the work of PARTeE does not seek to answer what appliance is, but rather ask, what architecture can it do? The ever-expanding toolkit of off-the-shelf robotics, open source computing and participatory communities have lowered the threshold for designers to explore this question. a2o's development as an advanced working prototype provides a construct in which the following questions can be asked: Can architecture actively and dynamically change physical environments in real time while becoming a social medium? Can architecture connect the virtual and the physical? Can architecture become an interface to connect ideas that were once thought to be disparate?

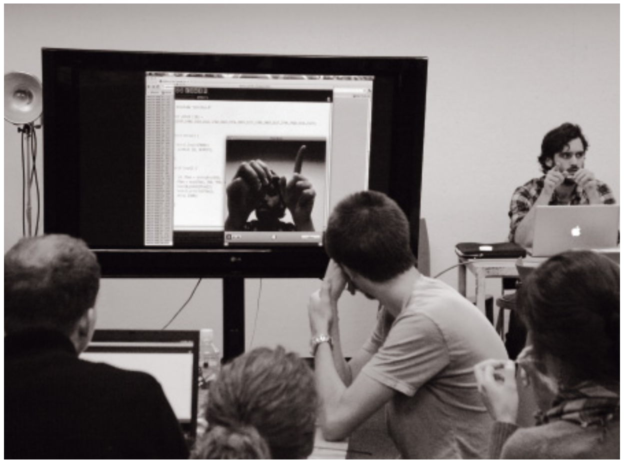

INTERACTIVE
ARCHITECTURE
WORKSHOP>no.1
7 DAYS OF QUESTIONING PHYSICAL COMPUTING, KINETIC SYSTEMS AND SIMULATION OUTPUT
Completed in ONE WEEK, the PHYSICAL COMPUTING WORKSHOP focused on the skill-sets necessary to understand the new re-emergence of INTERACTIVE ARCHITECTURE in the last decade. The goal of the workshop was for students to learn THREE basic elements necessary to PRODUCE dynamic responsive systems.

WORKSHOP › no.1

WORKSHOP > NO.1

In the fall of 2010, this author held a seven day physical computing workshop with twenty third year students in Virginia Tech's College of Architecture + Urban Studies (CAUS). The workshop represents the direct application of the pedagogical approach of the invisible college and represents the culmination of the research and teaching methodologies established by PARTeE in its first year. The goal of the workshop was to teach students introductory concepts related to the subassemblies of appliance architecture and to exhibit the workshop's research to the student body of CAUS. The exhibition sought to make the workshop's studies transparent, invite critique and begin a dialogue about appliance architecture within CAUS.

PARTeE provides a direct precedence for the teaching methodology used for the physical computing workshop. A bottom-up approach was taken for each element of the workshop and the five points of the invisible college severed as the framework for teaching. The workshop addressed these five points: Hacking—learning through borrowing, editing and splicing of preexistent media, off the shelf—providing a working model with direct heuristic learning and feedback as well as integrated coding libraries, simulation is realization—diagramming of behavior logic is realized as direct actuation and digital versioning—the ability for systems to employ the novel tectonics of digital fabrication and [inter]disciplinary blind spot—a cross-fertilization and new synergies between research fields.



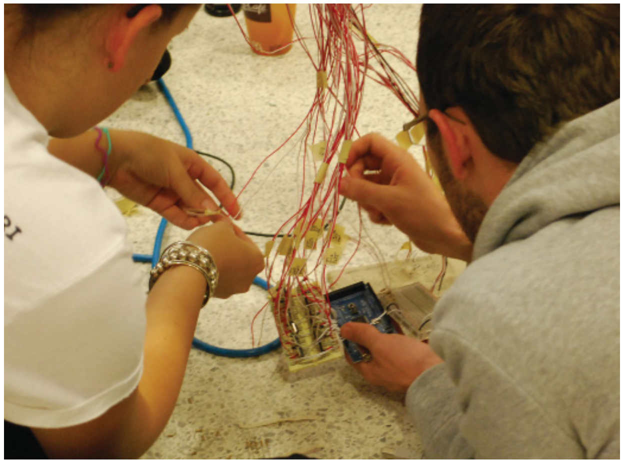


Figure 91 previous page, image of graphic narrative for exhibition at CAUS.

Figure 92 top, image of collective Pecha Kucha style research.

Figure 93 image of lecture style instruction. Image by Kelly Folts

Figure 94 image of student wiring transistors.

INTRODUCTION

[Note. Although the systems, materials, and subassemblies of appliance architecture are in their youth, the spectrum of its capabilities is vast. The study of appliance architecture requires a finite scope. For this reason the workshop was limited to physical kinetic motion.]

As described earlier under the pedagogy of the invisible college, appliance architecture represents those architectural constructs that embody behavior logic described through computation processes; however, complexity within appliance architecture does not necessitate complex coding or algorithms. In order for students to understand the disparity of complexity and algorithmic logic, as well as the scope of intended research, a Pecha Kucha (20 slides for 20 seconds each) style research assignment and presentation was given prior to the first day of the workshop. This author supplied the workshop participants with twenty-four practitioners and academics in the field of appliance architecture to research. The focus of research ranged from established institution such as MIT's media lab and Chuck Hoberman, to emergent movers such as Pranav Mistri and to the more abstract ideas and platforms such as those seen in Michael Resnick's Scratch programming software. The research and subsequent presentation explored concepts related to software and mechanical behavior, human interface and interaction, as well as direct architectural application. The presentation served to provide inspiration, precedents and a framework for the students' studies.

Disparities in student skill levels related to digital media were also a concern. The application of digital design software within academia is not always geared towards computational logic; a student who excels with Nurbs-based 3D modeling environments may not necessarily understand the structure of computational logic. Kostis Terzids describes this disparity:

> Computation is a term that differs from, but is often confused with, computerization. While computation is the procedure of calculating, i.e., determining something by mathematical or logical methods, computerization is the act of entering, processing, or storing information in a computer or a computer system. Computerization is about automation, mechanization, digitization, and conversion. Generally, it involves the digitization of entities or processes that are preconceived, predetermined, and well defined. In contrast, computation is about the exploration of the indeterminate, vague, unclear, and often ill-defined processes; because of its exploratory nature, computation aims at emulating or extending the human intellect. It is about rationalization, reasoning, logic, algorithm, deduction, induction, extrapolation, exploration and estimation. In its manifold implications, it involves problem solving, mental structures, cognition, simulation, and rule-based intelligence, to name but a few. (Terzids, 2003)

This discrepancy led to the adoption of a team learning environment which establishes a peer-to-peer learning structure. Peer-to-peer learning also provided a means to identify and promote student skills.

Research in the field of appliance architecture requires the cross-fertilization or contamination of multiple fields of study. As a whole, its study requires experts from multiple fields including computer programming, computer science, mechanical engineering, electrical engineering, behavioral science, and material science. For this reason and because of variable skill-levels with digital design tools, the workshop sought a bottom-up structure for learning. Students were divided into teams of four. The use of team learning provides a peer-to-peer learning environment that encourages participation with varying skills and at varying levels. A student may excel in one area and take on the roll of 'expert' within the group, while allowing another student to excel in another area of learning (Jerkins, Purushotma, Clinton, Weigel, & Robison). The result is a feeling of competence within the students. Clay Shirky describes the importance of instilling competence by saying:

> Learning on the job may seem opposed to the desire to feel competent, but competence is a moving target. Taking on a job that is too large or complex can be demoralizing, but taking on a job that is too simple that it presents few challenges can be dull and demoralizing. The feeling of competence is often best engaged by working right at the edge of one's abilities' (Shirky, 2010)

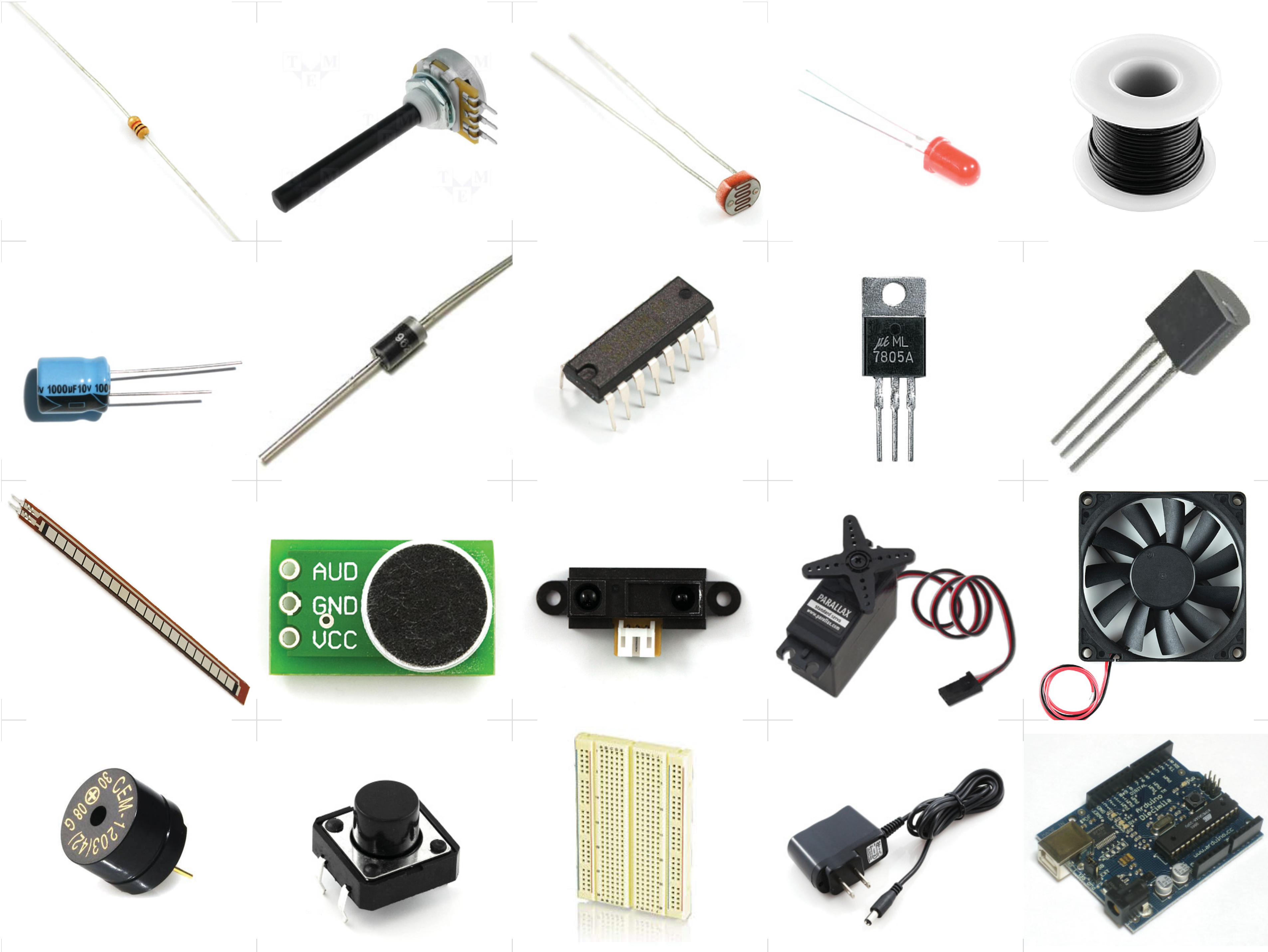
The teams were used for the first three days during which the workshop consisted of two daily seminars that introduced basic concepts related to three major topics: hardware, logic and simulation. For the final exhibition, the teams were reconfigured based on similar design typologies (as well as student competence), and resulted in a strengthening of the peer-to-peer learning structures. The three topics covered in the seminars are described below:

Hardware

Students were taught basic circuitry and principles in electrical engineering. Each student team was given a kit-of-parts including: Arduino microprocessor; sensors (distance, temperature, light, sound, flex); actuators (servomotor, fan, RGB LED, buzzer) and circuit components (hookup wire, breadboard, resistor, diode, potentiometer, transistor).

Basic principles related to circuitry needed to be established and the concepts include: direct / alternating currents, resistance (ohms law) and parallel vs. serial circuits. Once basic concepts were established, simple circuits were produced without the use of coding. These circuits were used to describe fundamental concepts in circuitry, such as resistance through the use of potentiometers, as well as to show that aspects of appliance architectures may not require coding. Later, more advanced circuits were developed concurrently with lessons in coding. These lessons provided a one-to-one understanding of how a circuit produces sensory data and in turn, how the computation logic processes that data.

Figure 95 image series showing the 'kit-of-parts' for the workshop > no. 1. Read from the top left: resistors, potentiometer, photocell, LED, hookup wire, capacitor, diode, 'H' bridge, variable transistor, thermal transistor, flex sensor, microphone, infrared range finder, servomotor, fan, buzzer, button, breadboard, power supply and Arduino Duemilanove. Images by sparkfun.com


ML
7805A
AUD
GND
VCC
PARALLAX

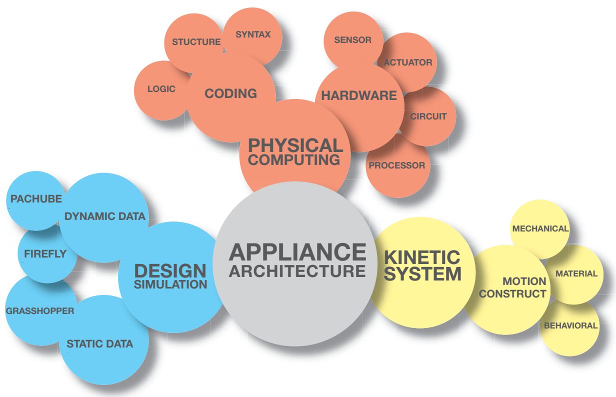


Figure 96 diagram of the subassemblies of appliance architecture introduced in workshop › no. 1.

Coding

The workshop focused on the use of open source input/output Arduino microprocessors, and the corresponding coding language, Arduino (Java based). In order to effectively teach introductory concepts related to algorithmic design, the workshop adopted a two-pronged approach to teaching basic coding structure and syntax. Seminars included lecture-style presentation that introduced broad concepts. At the same time, these seminars were injected with hands-on team learning and application of coding concepts through basic circuits. The Seminars included:

Syntax

Programming software is developed for specific applications, and these software use syntax that are specific to their programming environment, which is why there is no universal language. Although low-level processes such as arrays and loops are almost universal, the syntax used to describe these functions are not. Syntax is the way in which the programming environment understands code; it is the grammar of programming. In other words, syntax governs the use of characters that define the structure of code (imagine reading this work without any punctuation).

Variables

Variables are the way data is stored within a code (i.e thisPageNumber = 116)

Functions

Functions are coding modules built into the programming environment that allow for specific tasks to be operated. These functions 'call' codes that are built into the low-level operations of the software but are outside the coding environment. Functions are an example of how microprocessors communicate with their input/output peripherals (i.e. analogRead (pinA)). Functions are also used to perform complex algorithms through simple coding.

Structure

Structure defines the flow and organization of data within programming languages. It is described through operations (i.e. if x then b).

Simulation

Grasshopper

Grasshopper, a graphic algorithm editor designed for Rhino 4.0, provides a visual interface for simulating appliance architecture. Grasshopper simulations allow students to study and more accurately understand movement and the part-to-whole relationships of their designs. These simulations also provide an expedited media for testing ideas and concepts. Multiple design versions can be tested against a controlled logic system.

Firefly

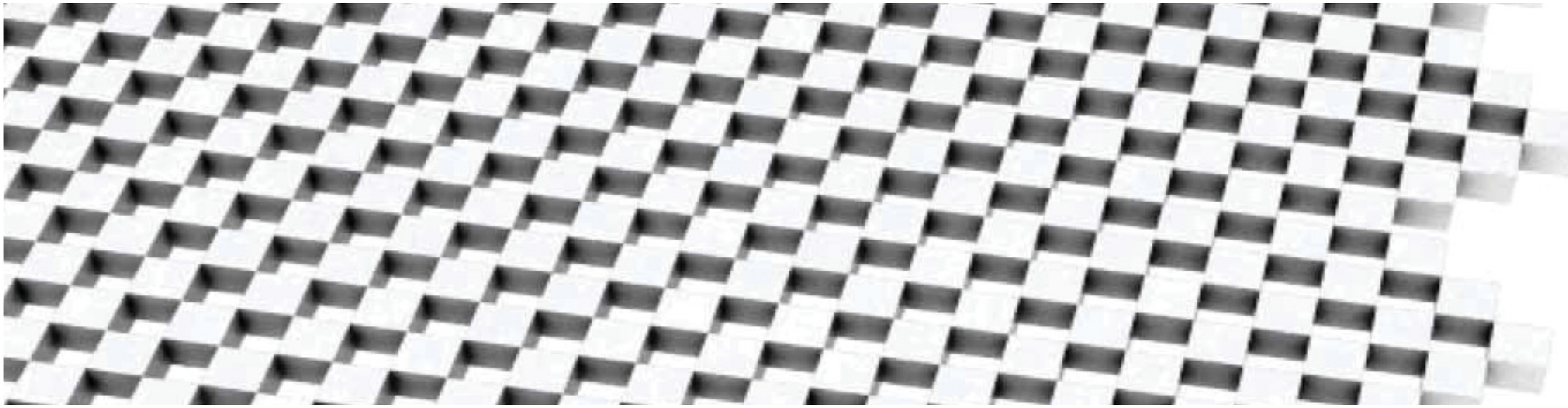
Firefly is a Grasshopper plug-in developed by LiftArchitects that allows Grasshopper to connect the computer's serial port to an Arduino microprocessor. Through the use of firefly, students can translate Grasshopper definitions, which are the behavior logic of a simulation, into working physical constructs. In other words, they can simulate, redefine and re-scale real-time sensory data into virtual landscapes.

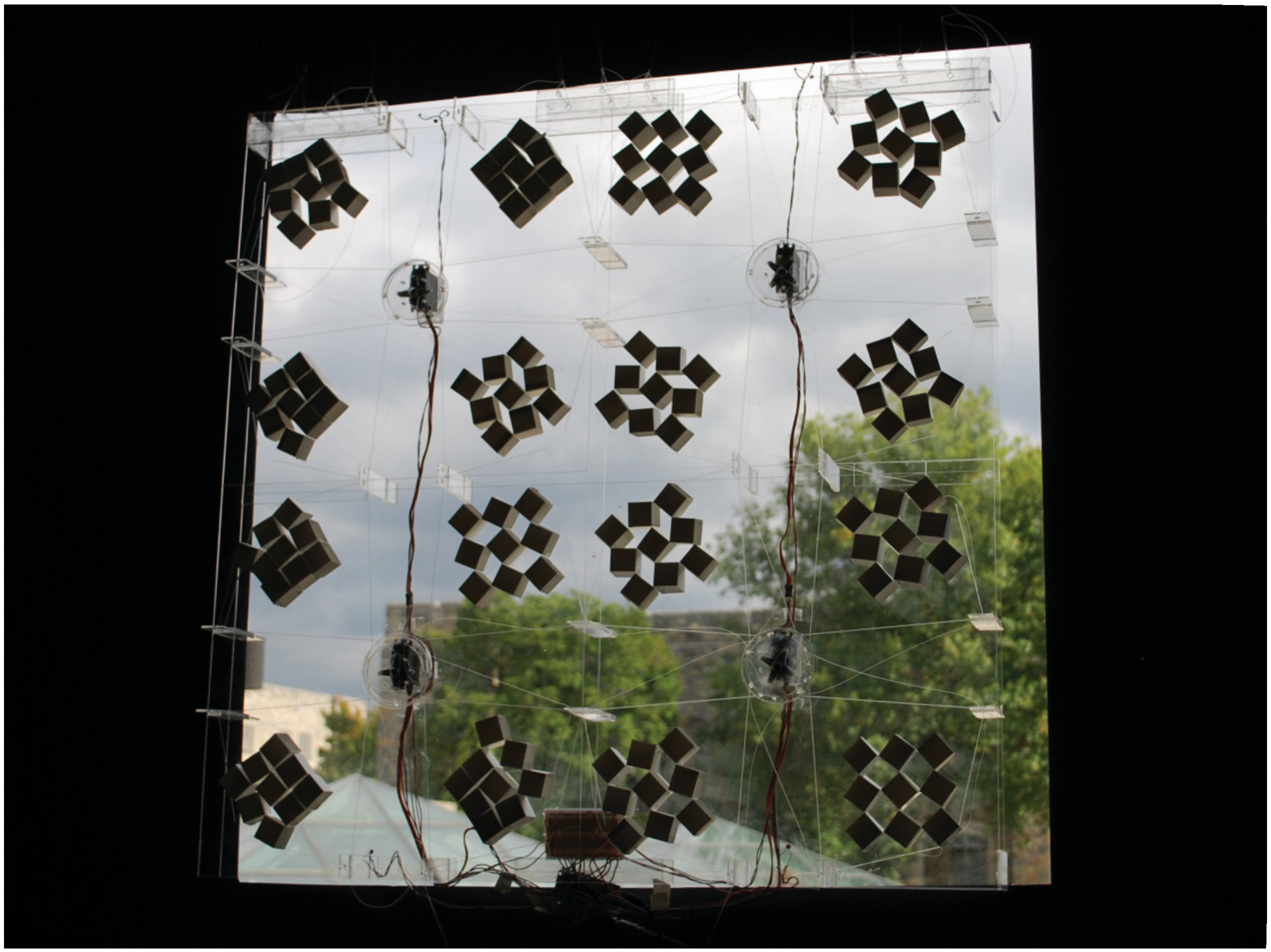
Design Prompt

Concurrent with seminar attendance, students were assigned Kostas Terzids' Passage design prompt, an exercise that was used to establish the bottom-up development of emergent typologies of PARTeE. Students were required to independently design four motion constructs related to the Passage design prompt. These constructs were intended to explore the tectonic relationships of motion, as well as to provide a system within which initial mechanical behaviors could be explored as the seminars evolved. Simply, the students were expected to create working models that could also house actuators, servomotors, fans, etc. The addition of actuators to the students' physical constructs provided hands-on learning and an opportunity to apply the circuits and codes developed during the lecture style seminars. The passages also provided a critical mass of experiments from which exhibition concepts could be evaluated.

The second half of the workshop focused on a full-scale lobby exhibition. The student teams were re-structured based on the emergence of similar typologies, competence and theoretical discourse. The selection process highlighted the need for instructors to guide and provide transparency. Although many of the concepts were novel and provided promising approaches to appliance architecture, it was necessary to select and shape explorations toward systems that could be realized in the limited time of the workshop.

The appliances that follow highlight the final student instillation:





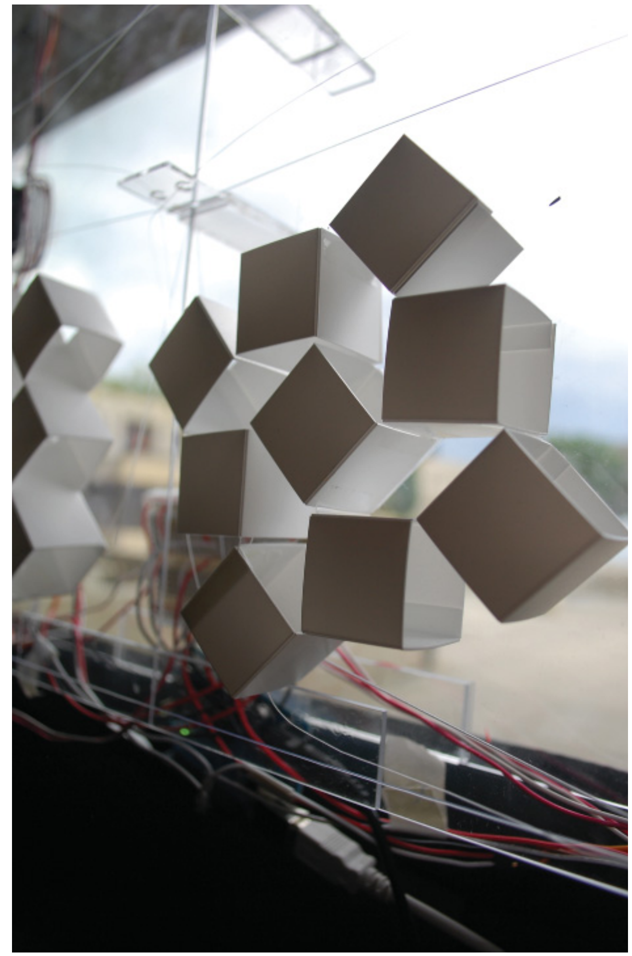


Figure 97 top, rendering of an aggregation of PHOTOtroph units depicting the instillations potential for expansion through a fractal geometric system.

Figure 98 bottom-left, image PHOTOtroph instillation showing the proportional articulation of the pixels (clusters of four units) to light levels falling on the storefront.

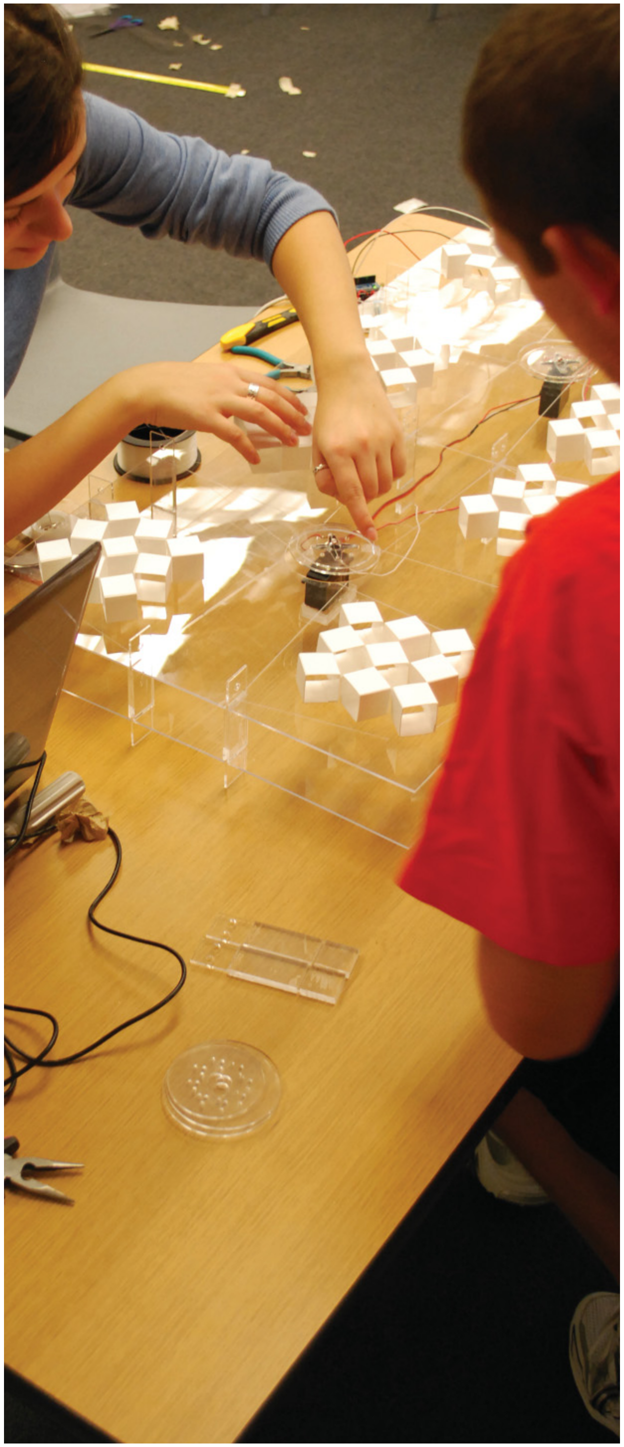
Figure 99 bottom-right, image single unit of PHOTOtroph.

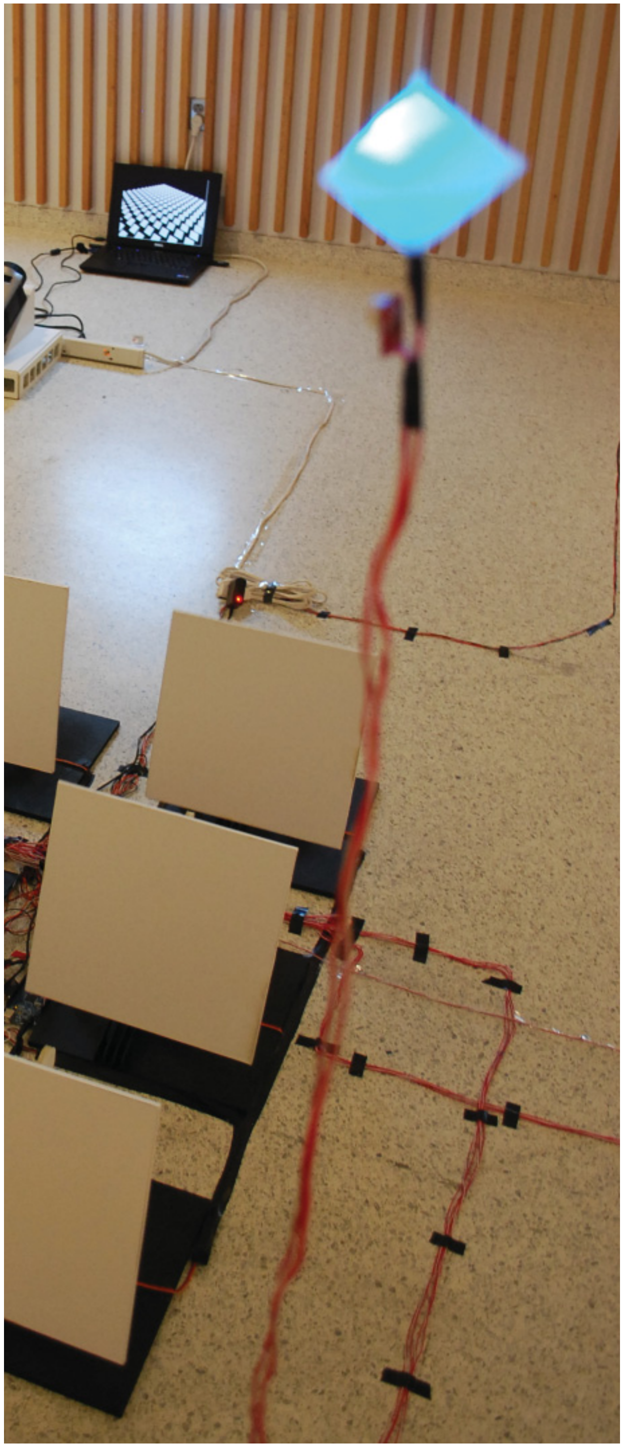
Figure 100 right, image of PHOTOtroph construction.

PHOTOtroph

Julian Akogyeram, Daniel Hoogenboom, Sandra Sierdzinska

PHOTOtroph is a fractal system focused on two basic systems. First, a motion construct consisting of nine one-inch-by-one-inch cubes arranged in a square pattern, which were able to rotate 180 degrees about their common vertices. The system is an example of a fractal unit and the beauty that can reside within physical constructs. The nine one-inch-by-one-inch cubes become a single unit within a larger system of nine three-inch-by-three-inch cubes and so on. The result is a complex ornamentation that is legible on multiple scales. Second, the module is an organism that changes its form depending on the input from the environment. Each of the four pixels consists of clusters of four three-by-three cubes that respond to a single photocell. The result is several different orientations reacting to the amount of light and shadow on the surface. Functionally, the system is a prototype of an adaptive façade—an examination of an active fritting— that reacts to changing environmental conditions relative to internal programming.





SPAZ

Lauren Bogaard, Sarah Durkin, Kelley Folts, Tess Kelly

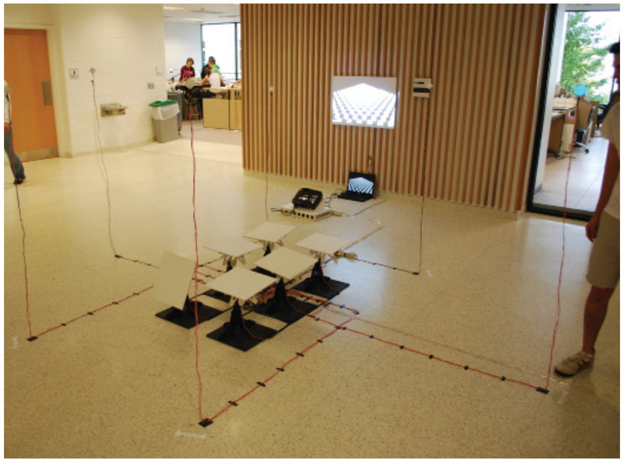
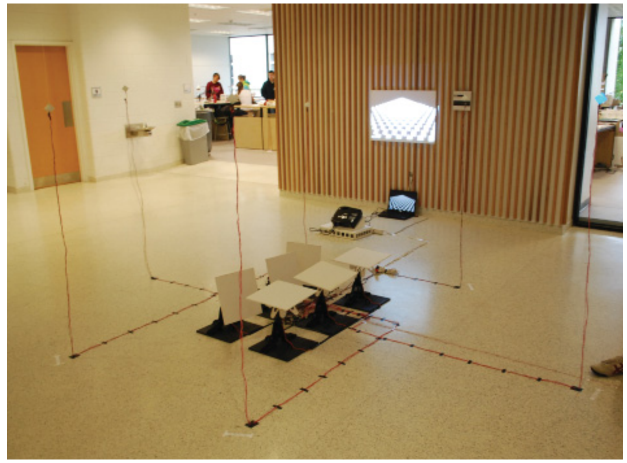
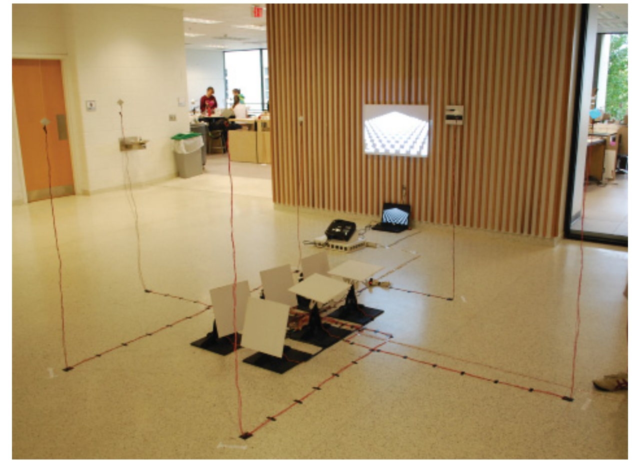
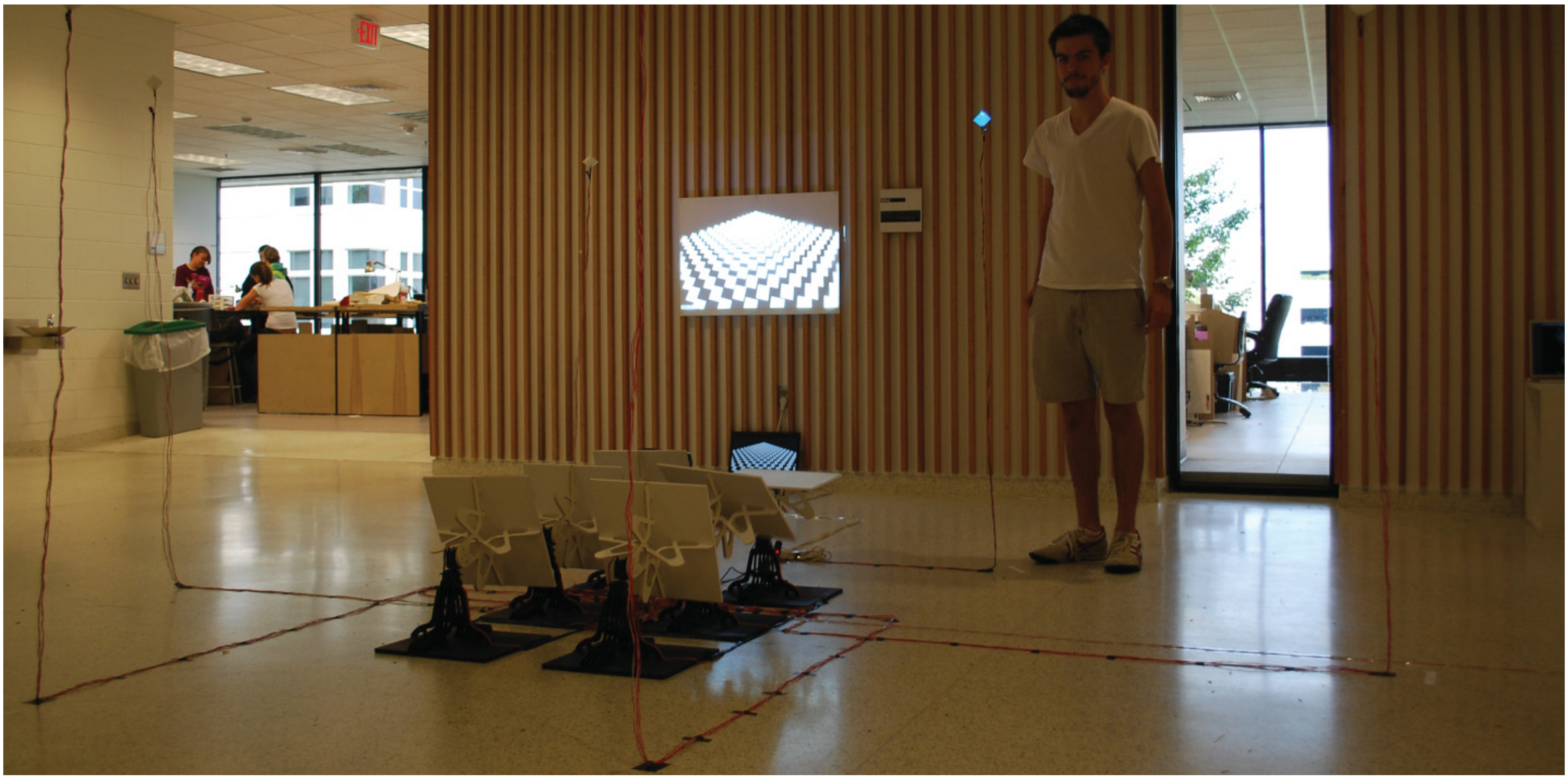
Torque can be described as the tendency of a force to rotate an object about an axis. The system is based on the ability to react and respond to sound as an occupancy sensor. When the units hear a loud noise, they each rotate about a three-dimensional node and look toward the source of the noise. The use of two servomotors in each unit provides a multi-axial nodal system able to 'look' at any point within a hemisphere.

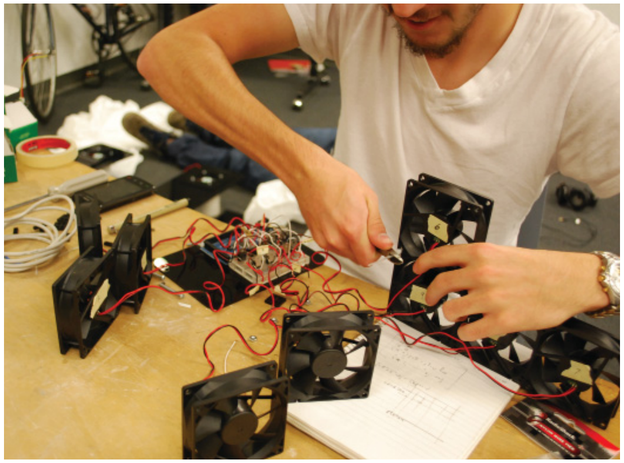
SPAZ's reaction results in an anthropomorphism through falsification. In a sense, SPAZ declares 'you are not one of us.' Instead of empathy, SPAZ evokes apologies, guilt and shame. The precise movement and ability to locate the noise in space creates a disturbance within the equilibrium of the horizontal surface produced by the units of SPAZ. Users sense that the units of SPAZ have been disturbed and are now looking back at them as if to say 'shh.'

Figure 101 left, image of microphone unit. When SPAZ recognizes a noise, the corresponding microphone unit illuminates, signifying sensing of data.

Figure 102 right-top, image series showing SPAZ's units actuation in response to the detection of noise.

Figure 103 right-bottom, image of SPAZ. Image depicts the six units of SPAZ capable of sensing noise through the use of 6 microphones distributed equally about the units bounding circumference.





EXIT



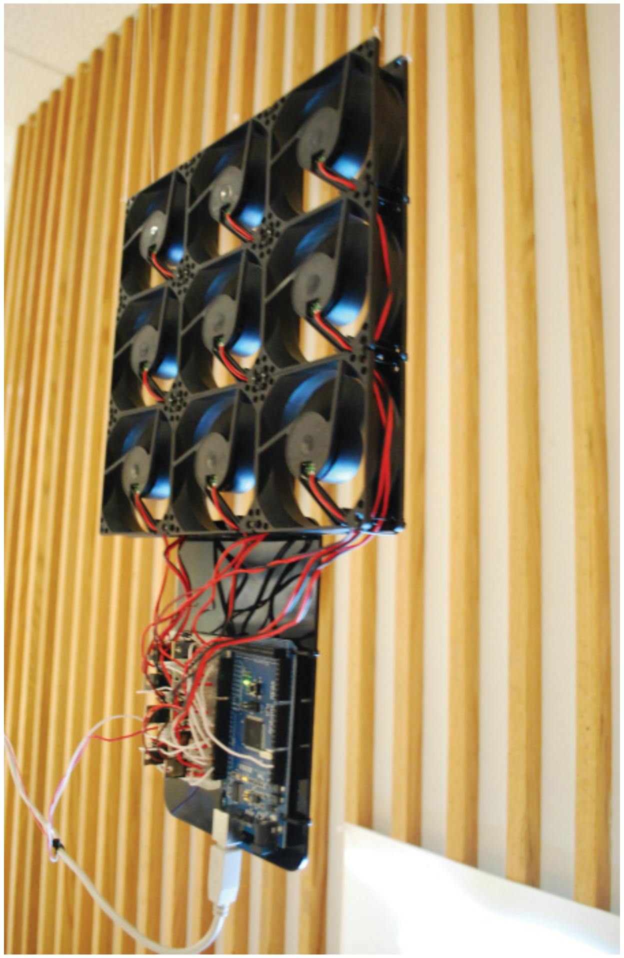


Figure 104 left, image of PHYxel. Image depicts the recognition of a body in space through a black and white 1000 x 1000 pixel digital image, which was coded in Processing, and the actuation of a sample 3x3 physical pixel.

Figure 105 left-top, image of PHYxel construction.

Figure 106 left-bottom, image of circuit and physical pixel composed of 9 cooling fans.

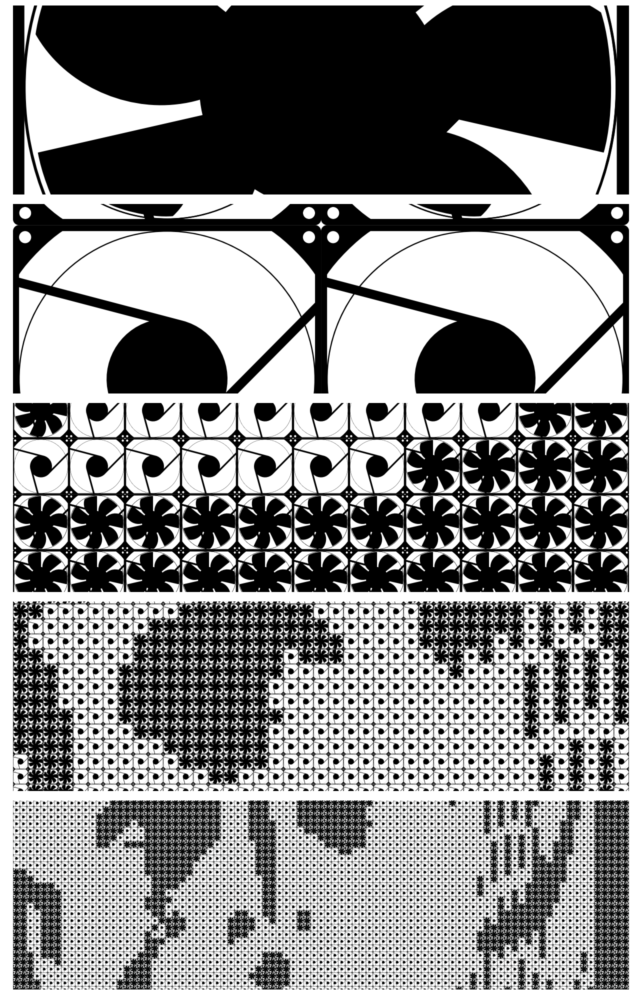
Figure 107 right, pixel resolution diagrams of PHYxel's potential expansion.

PHYxel

Thomas Keane

The three-by-three array 'off-the-shelf' computer cooling fans is a graphic kinetic interface. The fans are able to reconstruct real-time video imagery as physical pixels. Images are reduced to two basic pixel types: black (off), and white (on) through video capturing in Processing. Each fan within the 3 by 3 physical array represents a pixel within the larger 1000x1000 array. The fans present unique physical behavior able to mimic these values by adjusting its rotation speed, thus changing its optical density. The installation is a sample of what could be a vast array of pixels, creating a virtual (and physical) reflection of the user in real-time.

Similar to the optical button, PHYxel demonstrates the opportunity for learning through mimicry. Although the workshop focused on the Arduino programming environment, Processing's open source ancestry to Arduino allowed for easy translation of coding for the student. The student used source code similar to that of the optical button used in previous PARTeE studies. Through online communities and tutorials, the student was able to manipulate the code even with only limited programming experience. This manipulation also shows the benefit of learning through mimicry. Editing requires that the student posses a low-level understanding of code structure and syntax, but by actively editing and seeing one-to-one results, the student was able to develop a more advanced understanding of the source code. As a result the student learned how to use one and two-dimensional arrays, a subject not taught during the seminars. This example further supports the effectiveness of learning through mimicry, or 'hacking,' as a bottom-up emergent learning typology used to teach appliance architecture.



Hosiery

Hannah Catlow, Sarah Haase, Courtney Horst, Yi Yao

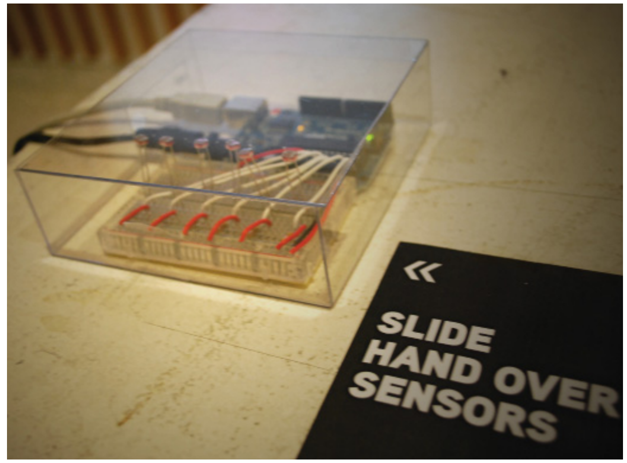
Hosiery, a mechanism for controlling threshold and volume through real-time Boolean operations, takes form as a system of suspended cylinders that creates a grid of volumes. These volumes are actuated through a gestural interface, which highlights the two-fold roll of the appliance—an interface device was imagined as both an instrument for physical actuation and virtual design. Hosiery's logic was completely designed within the Grasshopper environment and utilizes the Firefly plug-in for serial communication with the Arduino board. A linear array of photocells acts as an interface similar to the keys of piano. As a user's fingers move over the photocells, their proximity is mapped through the amount of shadow falling on the photocell. Each sensor represents a control point along a Nurbs surface that was modeled through Grasshopper in the Rhino 3D environment. Therefore, a one-to-one relationship can be established between the curves of the surface and the user's hand. If the user's hand makes an upside-down 'U' the resulting surface is an extruded upside-down 'U.' Once more, this surface represents a one-to-one relationship to the bottom surface of the suspended cylinders. As the user's fingers move and undulate, they actively shape the void space of the suspended volume.

Due to the scale of the intended instillation and the limited torque of the servomotor, Hosiery's physical construct could not be realized within the time frame of the workshop. However, the design 'instrument' shows the potential for subassemblies of appliance architecture to be re-embedded into the design tools of the parametric software developed through Schumacher's Parametricism.

Figure 108 left, image of Hosiery volumes.

Figure 109 right, top-left, image of Hosiery's gestural interface.

Figure 110 right, top-right, image of Hosiery's motion construct.



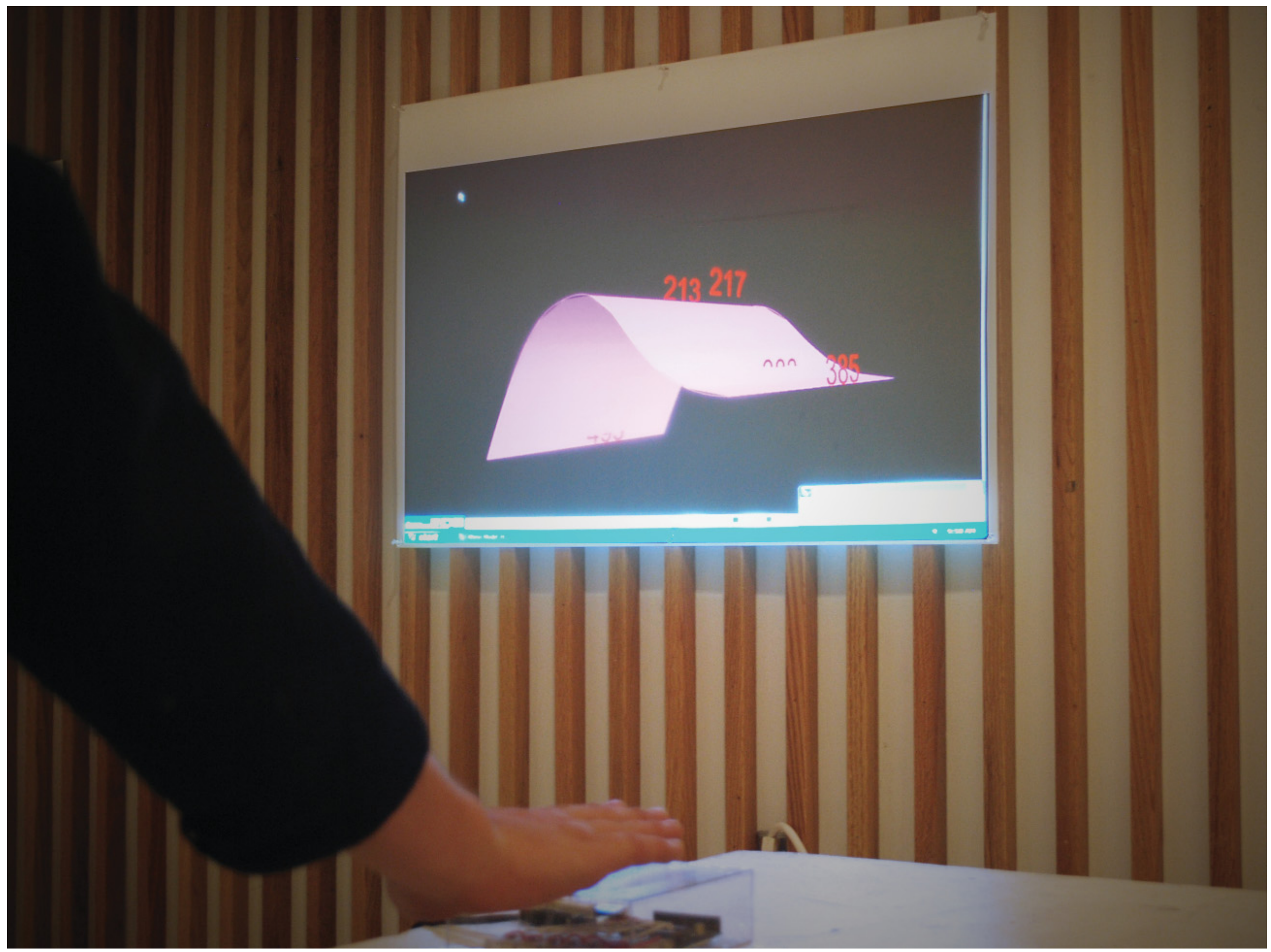


Figure 111 right, bottom, image of mapping of gestural interface in Rhino 3D environment

TAGGERS

Chase Daniel, Eric Rolaf, Robert Vance, Chris Warren

TAGGERS is an architecture in constant flux; it is an active, sensing, observing, thinking, lighting, and creating environment in communication with its human stimuli. Through a gestural engagement, the panels become a canvas for light graffiti. Inspired by Gjon Mili, TAGGERS is an environment consisting of gradiated parts, with each panel of the system containing RGB LEDs that cast patterns of light according to a person's gesture. The gesture is measured both by its place (left or right) and its magnitude (up or down) by three infrared range finders. Through this two-dimensional matrix the body can orchestrate a surface of light that is capable of capturing the intensity, chaos and emotion of the users.

The creation of light graffiti is recorded with time-lapse photography, which is then projected onto an adjacent wall. With the fluctuation of light, it amplifies the passage through space. In turn, each individual creates his or her own tag on the façade.

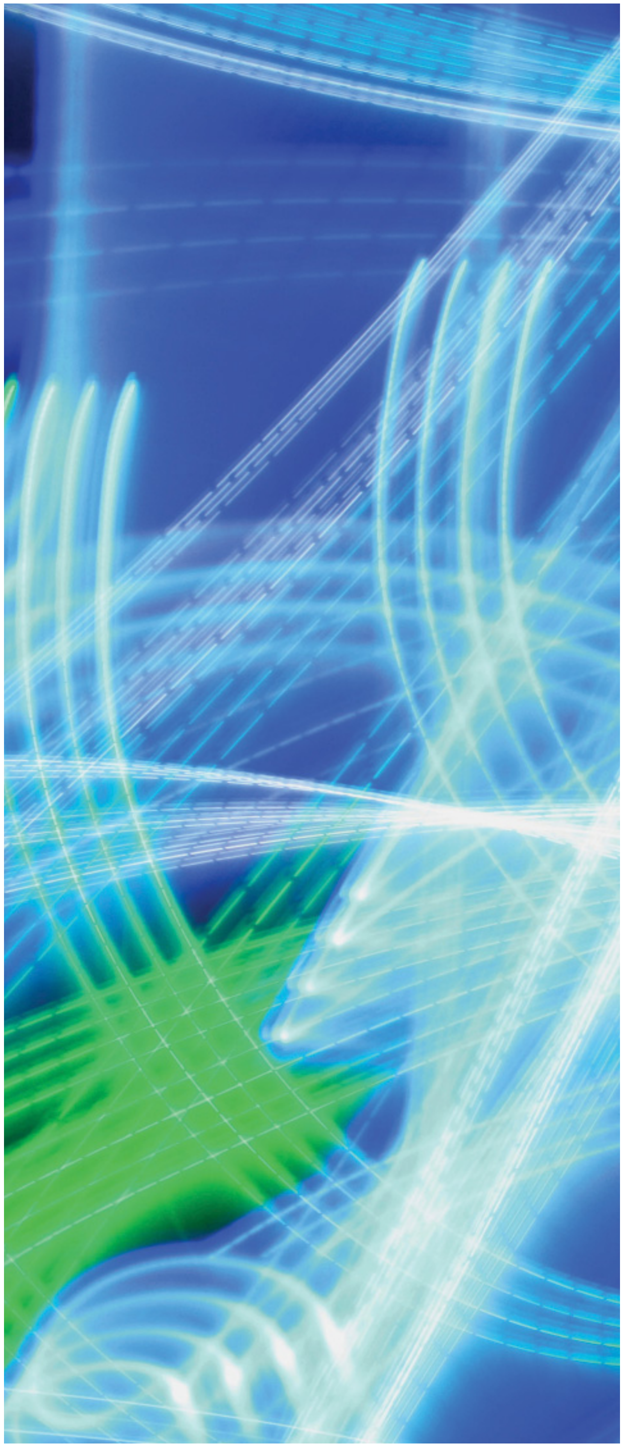
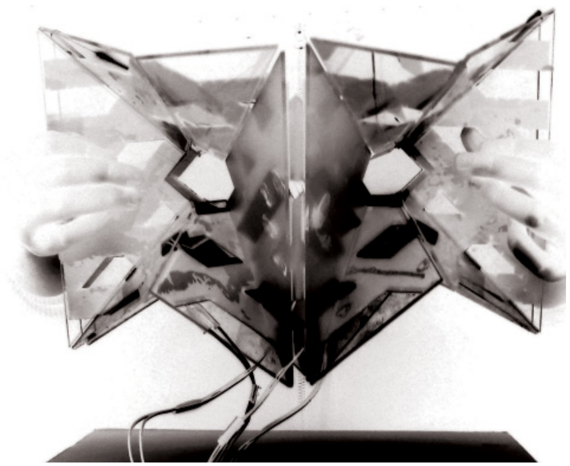


Figure 112 left, images of mapping of TAGGERs gestural orchestration.

Figure 113 right, TAGGERs time-lapse 'graffiti.'



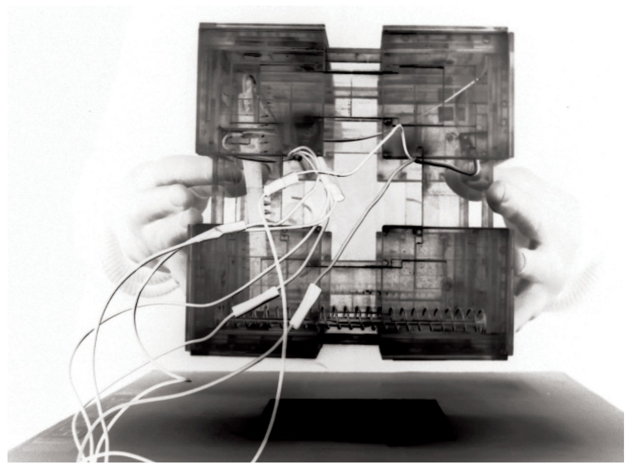


Figure 114 left-left, image of Diamond.

Figure 115 left-right, image of Cube.

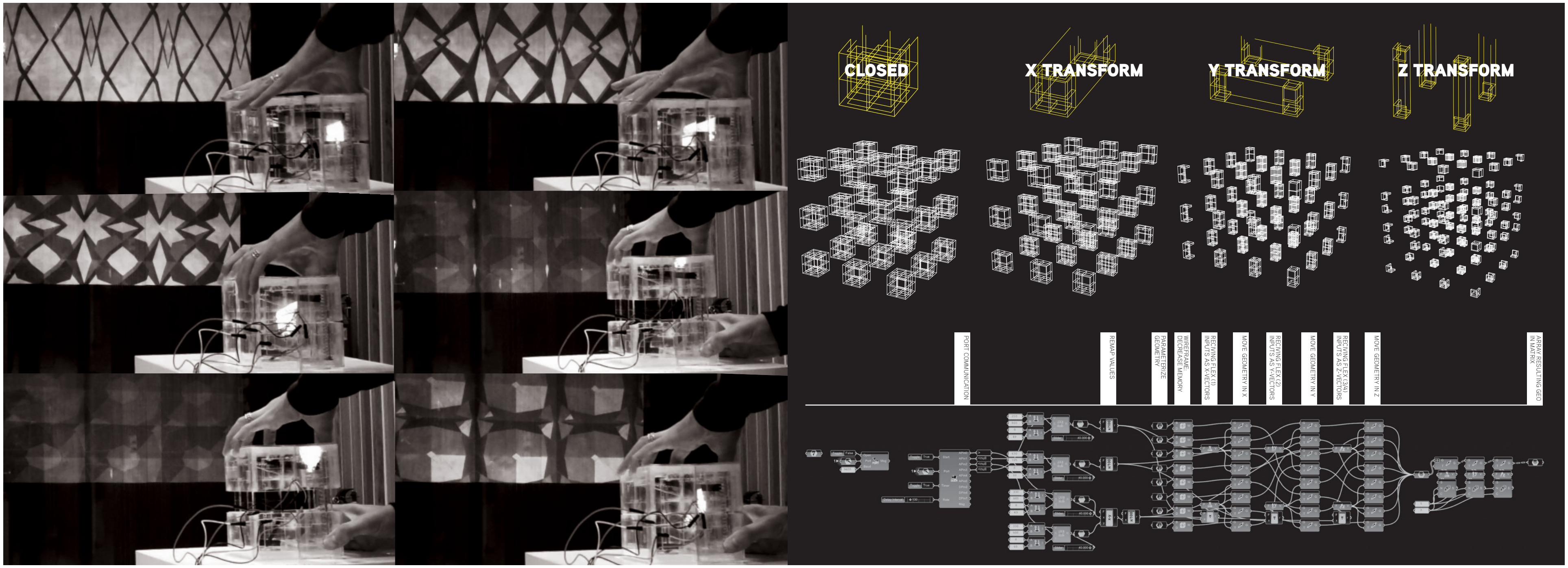
X_ref

Christopher Ames, Christopher Birr,
Chris Morgan, Grace Cabrera

As the paradigm of the new media culture seems to shift from the analog to the digital, X_ref is an effort to make an architectural process that is both more physical and more virtual. If the interface to digital media is more analogue, can this further the virtual (abstract) understanding of physical form and materiality? Two kinetic objects, different in form (physical) but similar in diagram (virtual), were designed as a kinetic interface to a graphic representation of spatial constructs. Each sensation is cross-referenced to its virtual counter-part; its movement is "mapped" so that the haptic-manipulation of one system translates to the digital manipulation of the other. This cross-association accesses abstract relationships between the two different objects, such as responses to action, conditions of openness and orders of movement. X_REF is about a process of mutually intensifying both the analogue and the digital in order to gain access to the haptic and tectonic questions of the physical, cross-referenced to the computational and iterative questions of the virtual; all of which are questions of appliance architecture.

Figure 116 right-top, montage of X_ref's Cube. Image shows the manipulation of the physical interface and the resulting digital augmentation in Grasshopper.

Figure 117 right-bottom, montage of X_ref's Diamond. Image depicts the manipulation of the physical interface and the resulting digital augmentation in Grasshopper.


CLOSED
X TRANSFORM
Y TRANSFORM
Z TRANSFORM
PORT COMMUNICATION
REMAP VALUES
PARAMETERIZE GEOMETRY
WIREFRAME: DECREASE MEMORY
RECIVING FLEX (1) INPUTS AS X-VECTORS
MOVE GEOMETRY IN X
RECIVING FLEX (2) INPUTS AS Y-VECTORS
MOVE GEOMETRY IN Y
RECIVING FLEX (3/4) INPUTS AS Z-VECTORS
MOVE GEOMETRY IN Z
ARRAY RESULTING GEO IN MATRIX

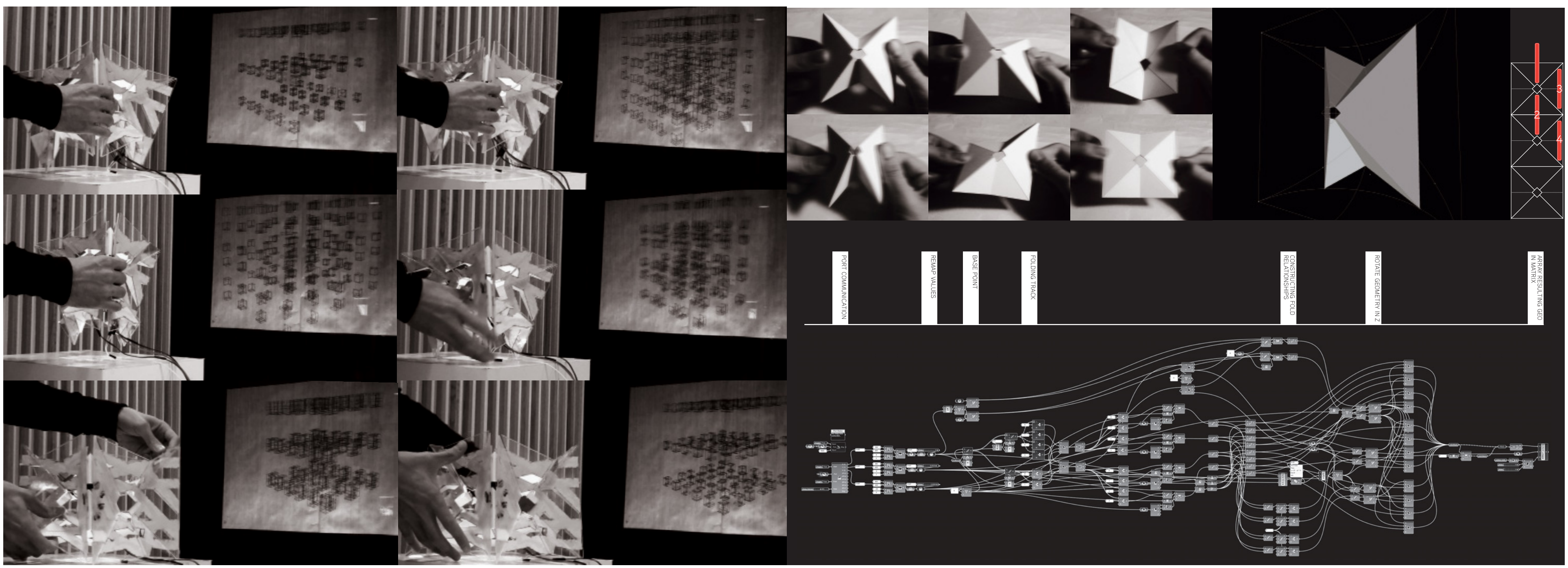

PORT COMMUNICATION
REMAP VALUES
BASE POINT
FOLDING TRACK
CONSTRUCTING FOLD RELATIONSHIPS
ROTATE GEOMETRY IN Z
ARRAY RESULTING GEO IN MATRIX

Figure 118 image series of interaction with Reasonable Attraction.

Reasonable Attraction

Ryan Boyland, Joshua Eager, Kirsten Halik

Reasonable Attraction's student designer describe the system's evocative logic:

> Passage may be understood as a mutual act or transaction that takes place between two people. This shared experience can take on many natures - coincidental, circumstantial, or intimate. Passage may also be interpreted as a process or transition from one state of being to another; in both cases, there is a transfer of energy. Whether the energy transferred is physical or mental, the transaction is violent. Electrons flicker constantly in and out of existence, ripping through the fabric of space-time with extreme force. This chaotic and beautiful choreography drives our physical world and the chemistry of our mental constructs at a vividly invisible scale. Through an evolution of transaction and transition, irrational passion grows into reasonable attraction. The incubation of these two passages creates an environment favorable for interaction.

Perhaps the most socially provocative, if not sexual, prototype, Reasonable Attraction explores the potential of emergent typologies along with 'hacking' to provide a medium through which spontaneous, accidental and improvisational design may arise.

Reasonable Attraction is a partition which houses six inflatable forms on both its interior and exterior surface. The forms are sewn plastic trash bags that are inflated and deflated by a pair of fans working in tandem to move air between the conjoined forms on each surface. As a body approaches already inflated forms, an infrared range sensor triggers the code to deflate the forms. This interaction provokes the user to imagine the forms are running away from them and fleeing to the other side. However, occupants on the other side of the wall could also understand the fleeing act as invasion of privacy.

The system also has the ability to produce a pattern across its surface. Through a loop of on/off signals and time delays, Reasonable Attraction produces a slowly shifting pattern.

Reasonable Attraction designers also added secondary actuators, LEDs and Piezo buzzers, to the system. The result is an amalgamation of parts that seamlessly work together to produce a provocative and uncanny appliance. However, the project was admittedly full of mistakes. As the students layered loop upon loop of actuation, the structure, in a way, became confused. This layering produced unanticipated patterns and sounds, and the result was exactly what the student creator had hoped, if not more. The smooth system of fleeing and invasion was paired with intense and unpredictable illumination as well as an animal-like 'breathing' noise.

This type of exploration through 'hacking' is a unique facet of the study of appliance architecture. The appliance's materiality is capable of embodying emergent behavior that could not be teased out of the mathematics rigidity of virtual environments. Improvisational systems are fragile and therefore, it was important that the students identify the mistaken structure in order that they could apply a proper, 'debugged,' logic. Debugging ensures that the emergent behavior will not be lost in subsequent edits.


YEAH


caus


How would you rate the instructors ability to convey concepts related to electricity and wiring circuits?
4.11
How effectively did assignment one, 'passage,' develop concepts related to electricity and wiring circuits?
3.17
How effectively did the final group installation develop concepts related to electricity and wiring circuits?
4.39
How would you rate the instructors ability to convey concepts related code syntax and structure in the Arduino programming environment?
3.78
How effectively did assignment one, 'passage,' develop concepts related code syntax and structure in the Arduino programming environment?
3.22
How effectively did the final group installation develop concepts code syntax and structure in the Arduino programming environment?
4.06
How would you rate the instructors ability to convey concepts related serial communication between Grasshopper, specifically 'Firefly,' and Arduino?
3.72
How effectively did assignment one, 'passage,' develop concepts related serial communication between Grasshopper, specifically 'Firefly,' and Arduino?
3.33
How effectively did the final group installation develop concepts related serial communication between Grasshopper, specifically 'Firefly,' and Arduino?
3.72
How would you rate the instructors ability to convey concepts related to the use of sensory data in physical computing and architecture?
4.06
How effectively did assignment one, 'passage,' develop concepts related to the use of sensory data in physical computing and architecture?
3.72
How effectively did the final group installation develop concepts introductory concepts related to the use of sensory data in physical computing and arch?
4.06
How would you rate the instructors ability to convey concepts related related algorithmic logic (i.e. if, then)
4.17
How effectively did assignment one, 'passage,' develop concepts related related algorithmic logic (i.e. if, then)
3.56
How effectively did the final group installation develop concepts related related algorithmic logic (i.e. if, then)
4.11
How would you rate the instructors ability to convey concepts of time in relationship to physical computing in architecture
4.11
How effectively did assignment one, 'passage,' develop concepts of time in relationship to physical computing in architecture
3.56
How effectively did the final group installation develop concepts of time in relationship to physical computing in architecture
3.94
How well did the Pecha Kucha style research presentations develop concepts related to responsive architecture and architectural robots?
4.17
How effectively did assignment one, 'passage,' develop concepts related to responsive architecture
4.00
How effectively did the final group installation develop concepts related to responsive architecture
4.11

Figure 119 previous page, image of exhibition installation.

Figure 120 histogram of student mean response to a series of questions related to the physical computing workshop. A response of 1 shows low correlation, a response of 5 shows high correlation.

RESULTS AND DISCUSSION

After the workshop, a voluntary and anonymous survey was conducted in order to establish pedagogical feed back. A sampling of eighteen student's were asked to evaluate eighteen statements related to the workshop and were given the opportunity to write in direct feedback to six free-response questions.

The first series of questions asked students to read and rate a statement between one and five; a response of one showed a low correlation and a response of five showed a high correlation. These questions were used to establish direct feed back on the three major teaching methods of lecture, Passage and exhibition. The statements assessed the effectiveness of these methods in developing concepts related to wiring and circuits, coding structure and syntax, Grasshopper simulations and serial communication, algorithmic logic, sensory data, time and the relationship to architecture.

As a whole, the results showed positive correlations between each of the three teaching methods and the development of the many themes discussed. By evaluating the mean response for the lecture, Passage and exhibition methods, we may extrapolate a series of correlations. The students were first asked to rate their knowledge of computer programming prior to the workshop, the students mean response was 2.5, or below moderate understanding. One third of the students responded with a one, or low, meaning they had little to no experience with software programming. These numbers can establish two methods of interpolation. First, they can establish a baseline for the effectiveness of the teaching methods. Or second, they can be used to establish that prior knowledge of computing is limited, and therefore it may be extrapolated that students may also have limited exposure to computer programming teaching environments in which they can rate the effectiveness of these methods. We will assume that students are rating the teaching methods based on their personal experience and feelings of competence.

When looking at the mean rating of the effectiveness of the lecture, the Passage and the group exhibit, we may evaluate the effectiveness of these methods through correlative methods. Lecture style teaching, the Passage and the group exhibit received a mean rating of 3.92, 3.4 and 4.0 respectively. The mean value of these ratings, 3.8, shows that, as a whole, the teaching environment's effectiveness was just under moderately high. When compared to the students' original understanding of computer programming, they also show a 52% improvement of understanding.

When comparing lecture style teaching, the Passage and the group exhibit side-be-side, we may establish another set of correlations. If the above statement is true, 'students may also have limited exposure to computer programming teaching environments to which they can rate the effectiveness of these methods,' then we can compare the Passage and exhibition against the lecture style teaching. In this case, the lecture style teaching becomes a baseline for the ratings, given that it is a controlled, top-down teaching method that differs from the bottom-up method of both Passage and the exhibit, and therefore could be understood as a precedent of knowledge. In this case Passage is 13% less effectively then the lecture as a teaching environment, and the exhibit was 4% more effective then the lecture as a teaching environment. More importantly, it indicates that the exhibit was 17% more effective then Passage.

This information shows positive correlations for a bottom-up teaching environment. Bottom-up systems are meant to be adaptive. The physical constructs and behavioral logics designed within Passage were edited, scaled and evolved for the lobby exhibition. Therefore, the 17% increase in effectiveness of the exhibition shows that the process of emergent typologies and bottom-up learning can produce increased effectiveness with time.

A second layer of statements is used to evaluate secondary teaching tools, the Pecha Kucha research and presentations, and a website designed by this author to host lessons and provide valuable links related to appliance architecture. The students were asked to rate how effectively the topics of the Pecha Kucha research related to physical computing and architectural robotics, as well as how effectively these topics prepared them for the workshop. The students rated the topics at 4.17, or moderately high, and rated their ability to prepare them for the workshop at 3.78, just under moderately high.

This information shows that although broad topics can be discussed and learned prior to the workshop, it is more difficult to prepare a student for the subassemblies and finite elements of physical computing.

The workshop served as a test of the effectiveness of teaching and exchanging information through a web-based platform. A website was designed specifically for the workshop and was used for three reasons: to supply media related to the workshop such as assignments and readings, to host valuable links to participatory communities related to physical computing, and to provide images and source code for the daily seminars. The students were asked how often they used the website and asked to assess the effectiveness of the website site to link to online communities as well as the appropriateness of these links. The students rated their use of the website at 3.17, or moderate, and the effectiveness of the website to direct them to links as 3.88, or just under moderately high. The effectiveness of the links was rated at 3.63, or moderate. These results showed that the effectiveness of teaching through a website and links to online communities was moderate and the subsequent workshops may need to assess the application of the website as a teaching tool.

The second section, which was comprised of free response questions, provided a more finite understanding of the students' understanding and opinion of the physical computing workshop. The section below provides the students' responses and has been edited for the purpose of removing repeat answers.

What aspect of the workshop did you find most beneficial?

'Learning about sensory objects in their relation to architectural components.'

'It taught us a new type of architectural thinking, asking what architecture can do rather than what architecture is.'

'The Pecha Kucha peaked my interest in the ideas of interactive architecture and just learning how to code and wire were fascinating.'

'Having to work through things mostly on our own helped to actually learn some things that would have been missed in only lecture-style learning.'

'Having the opportunity to explore innovative design mediums.'

'It truly felt like a "design laboratory" during this workshop. I felt free to experiment, and really watched projects iterate and evolve.'

'Beginning to learn about programming, and its relevance to architecture.'

'I found learning the circuitry most beneficial because it became a skill that we could all have for life that I couldn't previously do.'

What aspect of the workshop (other then more time) would you change?

'Perhaps having the introductions to wiring, coding, etc. earlier since it all didn't really have time to percolate and be processed.'

'I would've liked to understand the coding part more, so that we would have had to rely less on you for that.'

'A more in depth study to understand how grasshopper, Arduino, and coding in general works. Didn't fully understand all the computing.'

'I finally started to understand how each thing worked when I saw how it was applied to a project so possibly showing those applications earlier so that when were learning to program them I understand what it will do.'

'I don't think learning the actual coding was that beneficial for understanding of the technology? Perhaps using some sort of program that has a user-friendlier interface (similar to grasshopper) that would allow someone with little coding knowledge more freedom.'

'The coding process needs to be better explained, or more people need to help out (although I know you are just one person) and the timeframe needed to be longer.'

'It would have been beneficial to have more powerful or a greater number of motors, fans, and flex sensors.'

'Having others do wiring and coding so there would be more time for design and execution of that design.'

'It might benefit further workshops to position (or at least recognize) our efforts next to those of past student exhibitions in other schools of architecture.'

'Make it at least a week longer.'

What other topics would you be interested in pursuing as a result of the workshop?

'More extensive circuitry and wiring.'

'I think material technologies is very relative to this workshop and how we approach architecture in general, and I think a study of new material and fabrication technologies would be inspiring.'

'Applying this to an actual building design.'

'I have found interest in design research efforts related to technology and the body. Recognizing hacking as practice of researching. A consciousness of interfacing and processes as issues of designing and products of design generative architecture.'

Would it be beneficial to include students from other disciplines? If so, what departments would you include?

'Someone whose specialty is coding, some of us could work out the wiring because it's physical and logical, but coding is a whole other animal'

'Yes, I would like to include some computer programmers or maybe engineers for circuits.'

'Computer Science, Industrial design, Interior Design.'

'I'm eager to collaborate with good students from computer science, industrial design, mechanical engineering, and interior design.'

'Yes, I would include students from computer engineering who could help us with programming.'

Please include any additional comments here.

'It was an awesome experience and has moved my interest in architecture to buildings, which interact with you!'

'One of the best parts of the semester, I learned a lot.'

Conclusion

The students' free response answers provide valuable and constructive criticism. Their answers provide three major topics of discussion. First, the shifting paradigm of architecture; the students realized new ways of think about architectures that respond to users and have embedded motion, and the elements necessary to produce these actions. Second, the need for multidisciplinary synergies; students strongly advocated the need for more time, better explanation of coding and circuitry and a desire for a user-friendlier interface. These issues could be addressed in future workshops by producing a cross-fertilization of disciplines including students from the fields of computer science/ engineering, electrical/mechanical engineering and industrial design. This synergy would allow students to learn physical computing while fulfilling the roll of spatial consultant/designer. Third, a 'design laboratory' approach is an effective means of learning; students' responses showed the effectiveness of designing through doing. The direct application of concepts covered in lecture style teaching to one-on-one designs provided a means of learning where students 'worked through' problems and were rewarded with a sense of competency.

As a whole, the responses proved that the workshop was a positive experience for the students. When asked whether students would be interested in participating in another workshop related to physical computing, the mean rating was 4.11, or moderately high. It should also be noted that 50% of the students rated this statement as 5 or high. This potential shows a promising future for what might be the beginning of workshops > no.1.

APPENDICES

APPENDIX A

CONSULTED WORKS

Boigen, B., & Kwinter, S. (1991). Manual for 5 Appliances in the Alphabetical City: A Pedagogical Text. Assemblage (15), 30-41.

Corbusier, L. (1931). Towards a New Architecture. New York: Dover Publications, inc.

Dunay, R., Schubert, R., Wheeler, J., & Grinham, J. (2010). Eclipsis System - An innovative Sun Control and Insulation Fenestration. The Virginia Society of the American Institute of Architects.

Eisenman, P. (2010, 05 12). Lateness and the Crisis of Modernity. London.

Eisenman, P. (2008). Ten Canonical Buildings 1950-2000. New York: Rizzoli International Publications, Inc.

Fox, M., & Kemp, M. (2009). Interactive Architecture. New York: Princeton Architectural Press.

Frens, J., Djajadningrat, J., & Overbeeke, C. (2003). Form, Interaction and Function, An Exploration for Interactive Products. Proceedings of the Journal of the Asian International Design Conference , 1.

Gerber, D. (2009). SOM + SCI-Arc on CF: Responsive Kinetic Facade. Retrieved 2009 from Core Form: http://Core.form-ula.com

Goulthrope, M. (2006). HypoSurface at BIO: A Case Study of the Impact of New Technology on Even Marketing. Retrieved 2009 from HypoSurface: http://www.hyposurface.org

Hegel, G. W. (2010). Science of Logic. New York: Cambridge University Press.

Hookway, B., & C, P. (2006). Responsive Systems / Appliance Architectures. Architectural Design , 76 (5), 74-79.

Jerkins, H., Purushotma, R., Clinton, K., Weigel, M., & Robison, A. J. (2006). Confronting the Challenges of Participatory Culture: Media Education for the 21st Century. Retrieved 12 10, 2010, from New Media Literacies: http://www.newmedialiteracies.org/files/working/NMLWhitePaper.pdf

Kant, I. (1987). Critique of Judgment. Indianapolis : Hackett Publishing Company, Inc.

Kemp, M., & Fox, M. (2009). Interactive Architecture. New York, New York: Princeton Architectural Press.

Koolhaus, R. (1994). Delirious New York: a Retroactive Manifesto for Manhattan. New York: The Monacelli Press.

Leddy, T. (1991). Moore and Shusterman on Organic Wholes. 49, 63-73.

Ley, R., & Stein, J. (2009). Reef. From Reef Series: http://www.reefseries.com

Maher, M.-L., & Merrick, K. Agent Models for Dynamic 3D Virtual Worlds .

Maki, F. (2008). Nurturing Dreams: Collected Essays on Architecture and the City. Cambridge: MIT Press.

Marshall, D. (n.d.). New Media Culture. Retrieved from New Media Culture: http://www.newmediacultures.co.uk/

Negropnte, N. (1970). The Architecture Machine. Massachusetts: The MIT press.

Negroponte, N. (1969). Toward a Theory of Architecture Machines. Journal of Architectural Education , 23 (2), 9-12.

Petrunia, P. (2008, May 14). Eisenman's six point plan . Retrieved November 9, 2010, from Archinect.com: http://www.archinect.com/news/article.php?id=75217_0_24_0_C17

Raffle, H., Ishii, H., & Tichenor, J. (2004). Super Cilia Skin: A Textural Interface. Textile , 2 (3), 1-19.

Reas, C., & Fry, B. (2007). Processing: A Programing Handbook for Visual Designers and Artists. Cambridge, MA: MIT Press.

Reiser, J. (2006). Atlas of Novel Tectonics. New York, New York: Princeton Architectural Press.

Rothernberg, J., & Small, D. (2006). Design Research on Responsive Display Prototypes:

Integrating Sensing, Information and Computation Technologies. Architectural Design , 76 (5), 46-49.

Schumacher, P. (2008). Parametricism as Style - Parametricist Manifesto. Dark Side Club. Venice.

Schumacher, P. (2010 6-May). Patrik Schumacher on parametricism - ‘Let the style wars begin’. Retrieved 2010 6-May from The Architect’s Journal: the Home of British Architecture: http://www.architectsjournal.co.uk/the-critics/patrik-schumacher-on-parametricism-let-the-style-wars-begin/5217211.article

Shirky, C. (2008). Here Comes Everybody! The Power of Organizing without Organizations. New York: The Penguin group.

Shirky, C. (2010). Cognitive Surplus. New York: Penguin.

Simeonova, M. (2010 21-January). http://greenerbuildings.com. Retrieved 2010 from Greener Building.

Sterk, T. (2003). Building Upon Negroponte: A Hybridized Model of Control Suitable for Responsive Architecture. Digital Design. 21st eCAADe , (pp. 407-414). Graz (Austria).

Terzidis, K. (2006). Algorithmic Architecture. Burlington, MA: Elsevier Ltd.

Terzids, K. (2003). Expressive Form: A Conceptual Approach to Computational Design. London: Spoon Press.

APPENDIX B

SOURCE CODE: FLOWer

Author(s):
Jonathan Grinham, Spencer Lee

```
#include <Servo.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2

//Tone tone1;
Servo myservo[16];
int bend = 3;
int led[16] = {
23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,
53};
int pin[16] = {
24,22,28,26,32,30,36,34,40,38,44,42,48,46,52,
50};

int pitch[16] ={

1397,1480,1568,1661,1760,1865,1976,2093,2217,
2349,2489,2637,2794,2960,3136,3322};
int myArray [16];
int inByte;
int k = 0;
int photo = 400;
int flex = 600;
int memory = 800;
int heat = 200;
int cam = 1025;
int num = 40;
int degree[40];
int turn = 0;
int mag = 0;
boolean button = false;
int buttonVal = 0;
boolean done = false;
boolean lightcond = false;
boolean heatTemp = false;
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

void setup(){
  //tone1.begin(13);
  for (int i = 0; i < 16; i++)
  {
    pinMode(led[i],OUTPUT);
      myservo[i].attach(pin[i]);
  }
  pinMode(11,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(13,INPUT);
   for (int i = 0; i < num; i++)
  {
    degree[i] = 0;
  }
    sensors.begin();
  Serial.begin(9600);

}

void loop(){
  int val = analogRead(0);
  int photoVal= analogRead(1);
  int flexVal= analogRead(2);


  if (val >=0 && val < heat) {

     sensors.requestTemperatures();
    float temp =
sensors.getTempCByIndex(0)*9/5+32;

    if (temp > 0) heatTemp= true;
    else heatTemp = false;
    if (heatTemp){
```

```
    temp =
map(temp,70.00,75.00,0.00,160.00);
    Serial.print(temp);
    Serial.println();
    for (int i = 0; i < 16; i++)
   {
    myservo[i].write(temp);
    noTone(11);
   }

    delay (10);
   }

  }

 else if (val > heat && val < photo) {

   if (photoVal < 50) lightcond = true;
   else lightcond = false;

   photoVal = map(photoVal,0,450,0,10);
   int photoDegree = map(photoVal,0,10,0,140);

   for (int i = 0; i < 16; i++)
   {
    myservo[i].write(photoDegree);
   }

   if (lightcond){
    for (int i = 0; i < 16; i++)
    {
     digitalWrite (led[i],HIGH);
     noTone(11);
      ;
    }
   }
   else {
    for (int i = 0; i < 16; i++)
    {
     digitalWrite (led[i],LOW);
     noTone(11);

    }

    delay (10);
   }
 }

else if
 (val > photo && val < flex) {
 flexVal = map(flexVal,600,300,0,10);
 int flexDegree = map(flexVal,0,10,0,140);
 for (int i = 0; i < 16; i++)
 {
  digitalWrite (led[i],LOW);
  noTone(11);
 }
 for (int i = 0; i < 16; i++)
 {
  myservo[i].write(flexDegree);
  noTone(11);
  delay (10);
 }
}
else if
 (val > flex && val < memory)
 {
 buttonVal = digitalRead(13);
 //delay(300);

 if (buttonVal == 1)
 {
  button = true;
 }

 //buttonVal = 0;

 if (button)
 {
  buttonVal = 0;
  done = false;
  digitalWrite (12,HIGH);

  while (button)
  {
   if (done!=true)
   {
    for (int i = 0; i<num; i++)
    {
     degree[i] =
map(analogRead(2),600,300,0,140);

     delay (100);
    }

    done = true;
   }

   buttonVal = 0;
   Serial.println(buttonVal);
   buttonVal = digitalRead(13);
   Serial.println(buttonVal);

   if (buttonVal == 1 || button == true)
   {
    button = false;
    digitalWrite (12,LOW);
    buttonVal = 0;
    noTone(11);

    Serial.println("exit!");
```

```
      }
    }
  }
  else
  {
    mag = map(analogRead(3),0,1024,0,50);
    for (int i = 0; i<num; i++)
    {
      turn =  degree[i];

      for (int j = 0; j < 16; j++)
      {
        myservo[j].write(turn);
        tone(11,pitch[i]);
        Serial.print(turn);
        Serial.print("A,");
        delay(mag);
      }

      Serial.println();
    }
  }
}


else
if
  (Serial.available())
{
  inByte = Serial.read();
  Serial.print (inByte);
  myArray [k]=inByte;
  k++;
  if (k == 16){
    k=0;
  }
  for (int i = 0 ; i<16; i++){
      if (myArray [i]==0){
        myservo[i].write(140);
        digitalWrite (led[i],HIGH);
        tone(11,pitch[i]);
      }
      else if (myArray [i]==255){
        myservo[i].write(0);
        digitalWrite (led[i],LOW);
       noTone(11);

      }
    }
    delay(20);
  }
}
```

APPENDIX B

SOURCE CODE: FLOWer Optical Button

Author(s):
Jonathan Grinham
*Adapted from Brightness Thresholding by Golan Levin.

```
import processing.video.*;
import processing.serial.*;
Serial portCom;
color black = color(0);
color white = color(255);
int  numPixels;
Capture video;

void setup(){
  size(640, 480, P2D);
  strokeWeight(1);
  video = new Capture (this, width, height, 24);
  numPixels = video.width * video.height;
  portCom = new Serial(this,Serial.list()[1],
9600);

  smooth();
}
void draw(){

  if (video.available()){
    video.read();
    video.loadPixels();
    int threshold =127;
    float pixelBrightness;
    loadPixels();
    for(int i=0; i< numPixels; i++){
      pixelBrightness = brightness(video.pixels[i]);
      if (pixelBrightness > threshold){
        pixels[i] = white;
      }
      else {
        pixels[i] = black;
      }
    }
    updatePixels();
    for (int y=(video.height/5); y<(video.height);
y+=(video.height/5))
      for (int x=(video.width/5); x<(video.width);
x+=(video.width/5)){
        int testValues = get(x,y);
        float testBrightness =
brightness(testValues);
        if (testBrightness > threshold){
          fill (black);
        }
        else {
          fill (white);
        }
        ellipse(x,y,20,20);
        int arduino = int(testBrightness);
       println("p" + arduino + ",");
        portCom.write(arduino);

      }
  }

  delay (320);
}
```

APPENDIX B

SOURCE CODE: a2o

Author(s):
Jonathan Grinham, Spencer Lee

```
#include "pitches.h"
#include <Servo.h>
#include <Wire.h>

Servo myservo[5];
int servoPin [5] = {22,24,26,28,30};

int tonePin [5]   =  {23,25,27,29,31};
int ledPin [5]    =   {33,35,37,39,41};
int distPin [5]   =  {0,1,2,3,4};
int lightPin [5]  = {11,12,13,14,15};

int old[5]        = {0,0,0,0,0};
int old_light[5] = {0,0,0,0,0};

int lightArray [5]   = {0,0,0,0,0};
int range_reading[5] = {0,0,0,0,0};

int brightness[5] = {0,0,0,0,0};

int redPin[5]   = {13,12,11,10, 9};
int greenPin[5] = { 8, 7, 6, 5, 4};
int bluePin[5]  = { 3, 2,45,44,46};

int sensorMin = 200;
int sensorMax = 650;
int length = 0;
int color = 0; //RGB light

int lightMin = 100;
int lightMax = 650;

int first = 1;
int first_light = 1;

int d = 1;     //CHANGE SPEED HERE, KEEP
ABOVE '5'

int current;
int def_pos = 0;

unsigned long t;
unsigned long howlong = 180000;

int data[6];
int mega = 5;
unsigned long interval = 200; // milli second
unsigned long t1;

int top[5][4] = {
{NOTE_G1  , NOTE_GS1 ,NOTE_A1  ,
NOTE_AS1 },
{ NOTE_B2  , NOTE_C3  , NOTE_CS3 ,
NOTE_D3  },
{ NOTE_DS4 , NOTE_E4  , NOTE_F4  ,
NOTE_FS4 },
{NOTE_G5  , NOTE_GS5 , NOTE_A5  ,
NOTE_AS5 },
{NOTE_B6  , NOTE_C7  , NOTE_CS7,
NOTE_D7  }
};

int second[5][4] = {
{NOTE_B1  ,NOTE_C2  , NOTE_CS2 ,
NOTE_D2  },
{NOTE_DS3 , NOTE_E3 , NOTE_F3  ,
NOTE_FS3 },
{NOTE_G4  , NOTE_GS4 , NOTE_A4  ,
NOTE_AS4 },
{NOTE_B5  , NOTE_C6  , NOTE_CS6 ,
NOTE_D6  },
{ NOTE_DS7, NOTE_E7  , NOTE_F7  ,
NOTE_FS7}
};
```

```
int third[5][4] = {
{ NOTE_DS2 , NOTE_E2  , NOTE_F2  ,
NOTE_FS2 },
{ NOTE_G3  , NOTE_GS3 , NOTE_A3  ,
NOTE_AS3 },
{ NOTE_B4  , NOTE_C5  , NOTE_CS5 ,
NOTE_D5  },
{ NOTE_DS6 , NOTE_E6  , NOTE_F6  ,
NOTE_FS6 },
{NOTE_G7  , NOTE_GS7 , NOTE_A7  ,
NOTE_AS7}
};

int bottom[5][4]= {
{ NOTE_G2  , NOTE_GS2 , NOTE_A2  ,
NOTE_AS2 },
{NOTE_B3  , NOTE_C4  , NOTE_CS4 ,
NOTE_D4  },
{NOTE_DS5 , NOTE_E5  , NOTE_F5  ,
NOTE_FS5 },
{ NOTE_G6  , NOTE_GS6 , NOTE_A6  ,
NOTE_AS6 },
{ NOTE_B7  , NOTE_C8  , NOTE_CS8 ,
NOTE_D8  }
};


void setup()
{
  Wire.begin(mega);
  data[5] = mega;

  Serial.begin(9600);

  t  = millis();
  t1 = millis();

  for (int i = 0; i < 5; i++)
  {
    pinMode(redPin[i],OUTPUT);
    pinMode(greenPin[i],OUTPUT);
    pinMode(bluePin[i],OUTPUT);
    pinMode(tonePin[i],OUTPUT);
    pinMode(ledPin[i],OUTPUT);

    myservo[i].attach(servoPin[i]);
  }
}

void loop()
{
 if (millis() - t >= howlong)
 {
  for (int m = 0; m < 5; m++)
  {
     myservo[m].write(50);
     delay(2000);
     myservo[m].write(170);
  }

    t = millis();
 }

  for (int k = 0; k < 5; k++)
  {
    range_reading[k] = analogRead(k);
    int current_light = analogRead(lightPin [k]);

    color = map(range_reading[k], sensorMin,
sensorMax, 0, 3);
    length = map(range_reading[k], sensorMin,
sensorMax, 170,0);

     data[k] = length;

   if (range_reading[k] > sensorMin)
   {

     if(current_light > lightMax/2)
     {
       length = 170;
     }
     else
     {
       length = 50;
     }

     digitalWrite(ledPin[k], HIGH);
     servo_move(k, length);
     rgb_light(k, color);

     //////////////////////////////////////////
     tone(tonePin[k], top[k][color], 20);
   }
   else
   {
     digitalWrite(ledPin[k], LOW);
     light_sensor(k);
   }

     delay(21);
     noTone(tonePin[k]);
  }

  send_data();

} // end of loop()

void send_data()
{
  if (millis() - t1 > interval)
```

```
  {
    Wire.beginTransmission(1);
    Wire.send((uint8_t *)data, sizeof(data));
    Wire.endTransmission();
    t1 = millis();
  }
}


void servo_move(int k, int length)
{
      range_reading[k] = analogRead(k);

      if (first == 1)
      {
        old[k] = length;
        first = 0;
        myservo[k].write(length);
      }
      else
      {
         myservo[k].write(length);
         old[k] = length;

      }
}

void rgb_light(int k, int color)
{
    switch (color)
    {
        case 1:    //red
          analogWrite(redPin[k],0);
           analogWrite(bluePin[k], 127);
           analogWrite(greenPin[k], 255);
break;
        case 2:    //green
          analogWrite(redPin[k],255);
           analogWrite(bluePin[k], 255);
           analogWrite(greenPin[k], 0);
break;
        case 3:  //blue
          analogWrite(redPin[k], 255);
           analogWrite(bluePin[k], 0);
           analogWrite(greenPin[k], 255);
break;
    }
}

void light_sensor(int j)
{
  lightArray[j] = map(analogRead(lightPin [j]),
50, 350, 170, 0);

      if (first_light == 1)
      {
        old_light[j] = lightArray [j];
        first_light = 0;

        myservo[j].write(lightArray [j]);
        brightness [j] = map(lightArray [j], 170, 0,
255, 0);

        //noTone(tonePin [j] );
        analogWrite(redPin[j],brightness [j]);
        analogWrite(bluePin[j],brightness [j]);
        analogWrite(greenPin[j],brightness [j]);
        //digitalWrite(ledPin[j], HIGH);
      }
      else
      {
        if (lightArray [j] < old_light[j] + 7 & lightAr-
ray [j] > old_light[j] - 7)
        {
          //don't move
          //old_light[j] = lightArray [j];
          //digitalWrite(ledPin[j], LOW);
        }
        else
        {
          old_light[j] = lightArray [j];
          myservo[j].write(lightArray [j]);
          brightness [j] = map(lightArray [j], 170, 0,
255, 0);

          //noTone(tonePin [j] );
          analogWrite(redPin[j],  brightness [j]);
          analogWrite(bluePin[j], brightness [j]);
          analogWrite(greenPin[j],brightness [j]);
          //digitalWrite(ledPin[j], HIGH);
        }
      }
}

void servo_speed(Servo servo, int degree, int
msec)
{
  current = servo.read();
  int diff = degree-current;

  if (diff > 0)
  {
    for (int i = 0; i < diff; i++)
    {
      current++;
      servo.write(current);
      delay(msec);
    }
  }
  else if(diff < 0)
  {
```

```
    for (int i = 0; i > diff; i--)
    {
      current--;
      servo.write(current); // all the servo.write
commands should be here.
      delay(msec);
    }
  }
  else
  {
    //nothing
  }
}
```

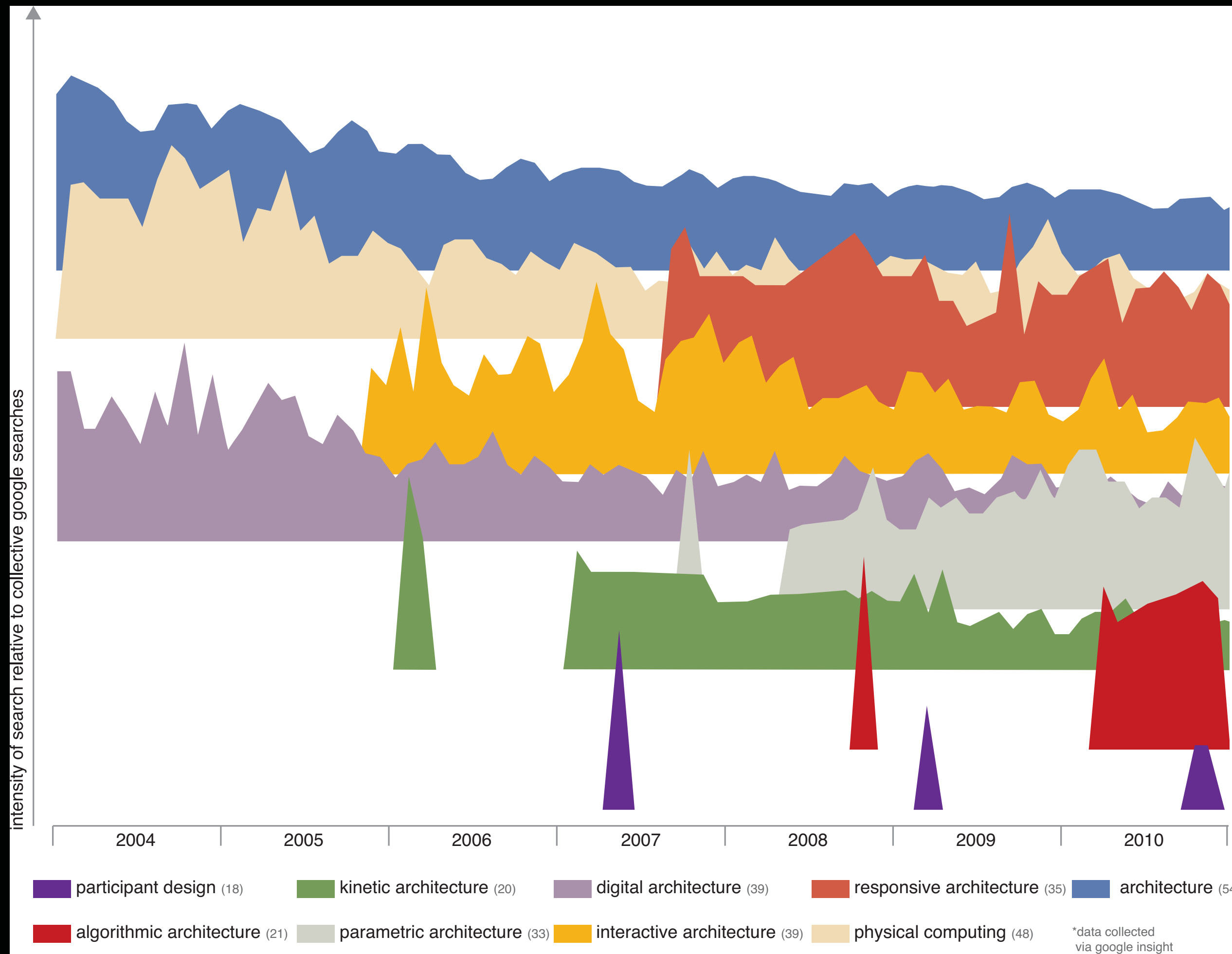


Figure 121 Diagram of the intensity of google search of themes related to appliance architecture throughout google existence.

...how has the new media culture affected architecture and style?