

A Concurrent Physical and Digital Modeling Environment

Exploring Tactile and Parametric Interactions in Design Modeling

by

Lu Han

A Thesis Documentation

Submitted to the Faculty of Carnegie Mellon University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Computational Design

in the

College of Fine Arts, School of Architecture



© Lu Han

Carnegie Mellon University

Pittsburgh, PA

May 2017

Acknowledgments

Firstly, I would like to express my sincerest gratitude to my primary advisor, Assistant Professor Daniel Cardoso Llach, for his patience, motivation, and guidance. His breadth and depth of knowledge were of great assistance to me throughout the completion of this project – from its initial conception to its final presentation; I could not have done it without his help.

I also want to thank my thesis committee member, Assistant Teaching Professor Eddy Man Kim, for his insightful perspectives, suggestions, and encouragements. I am extremely grateful for his continued support throughout my time at Carnegie Mellon University.

Last but not least, I want to thank my cohort: Amit Nambiar, George (Qiaozhi) Wang, and Darcy (Hetian) Cao for all the help and camaraderie; and Ardavan Bidgoli, Chaoya Li, Hexing Ren, Robert Zacharias, Hugh Dyar, and the rest of my CodeLab-mates for lending mental, physical, and material support.

Thank you all for making the last two years an invaluable time of my life.

Table of Contents

Abstract	- 3 -
Chapter 1. Introduction	- 4 -
Chapter 2. Building Background	- 11 -
2.1 The Architecture Machine	- 11 -
2.1.1 Case Study: SEEK/Blocksworld	- 12 -
2.2 Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms	- 13 -
2.2.1 Case Study: MetaDESK (Tangible Geospace)	- 14 -
2.2.2 Case Study: URP	- 16 -
2.3 Case Study: A Dynamic Physical Model Based on A 3D Digital Model	- 17 -
2.4 Observation and Reflection	- 20 -
Chapter 3. Hypothesis	- 22 -
Chapter 4. Research Question	- 23 -
4.1 My Vision	- 23 -
Chapter 5. Method	- 25 -
5.1 IMU Sensing: Arduino MPU 9250	- 25 -
5.1.1 My Prototype	- 27 -
5.1.2 Advantages and Limitation	- 28 -
5.2 Fiducial/AR Markers	- 30 -
5.2.1 My Prototype	- 32 -
5.2.2 Advantages and Limitation	- 33 -
5.3 Final Prototype - Kinect Depth Tracking	- 35 -
5.3.1 Kinect and Voxel	- 35 -
5.3.2 The Set-up	- 38 -
5.3.3 Exploration of Interaction	- 41 -
5.4 Discussion	- 49 -
5.4.1 Benefits of the set-up	- 49 -
5.4.2 Observation of User Interaction	- 50 -
5.4.3 Contribution	- 53 -
5.4.4 Limitation	- 53 -
5.4.5 Next Steps and future potentials	- 55 -
5.5 Conclusion	- 57 -

Table of Figures

Figure 1: the Lego Mindstorms Kit	- 6 -
Figure 2: Spatial Reasoning Experiment with Physical and Digital Models.....	- 8 -
Figure 3: “Blocks” in SEEK	- 12 -
Figure 4: SEEK in Exhibition	- 13 -
Figure 5: Tangible Geospace on metaDESK	- 15 -
Figure 6: Urp - combining physical architectural elements with digital analysis	- 16 -
Figure 7: Case Study - Prototype System Physical Appearance.....	- 18 -
Figure 8: Case Study – 3D digital model representation	- 19 -
Figure 9: Case Study – Physical representation based on digital model	- 19 -
Figure 10: Orientation of Axes of Sensitivity and Polarity for Accelerometer and Gyroscope, Orientation of Axes of Sensitivity for Compass	- 25 -
Figure 11: MPU to Arduino hook-up guide.....	- 26 -
Figure 12: MPU to Arduino hook-up.....	- 27 -
Figure 13: Mapping MPU Data to Digital Representation	- 28 -
Figure 14: Sifteos, the “Toys That Can Talk to Each Other”	- 29 -
Figure 15: Fiducial Markers on Cubes.....	- 30 -
Figure 16: How AR Markers Work	- 31 -
Figure 17: AR Marker Prototype Set-up.....	- 32 -
Figure 18: Mapping Shape to Markers in AR.....	- 32 -
Figure 19: Digital Representations of Makers in Model Space	- 33 -
Figure 20: Microsoft Kinect.....	- 35 -
Figure 21: Microsoft Kinect V2 Specifications	- 36 -
Figure 22: Voxel Representations.....	- 37 -
Figure 23: Depth to Voxel - Mapping Process	- 38 -
Figure 24: Physical Set-up	- 39 -
Figure 25: Fundamental Algorithm for Prototype	- 40 -
Figure 26: The User Interface	- 41 -
Figure 27: User Stacks Cubes While Editing the Geometric Representation.....	- 42 -
Figure 28: Multiple Points of Views.....	- 43 -
Figure 29: 4 Primitive Representations (Cubes, Spheres, Cones, tubes).....	- 44 -
Figure 30: User working back and forth between physical and digital modeling	- 45 -
Figure 31: Slide Bar Allows for Users to Edit Parametric Shape That Expands Beyond Its Respective Voxel Constraint.....	- 46 -
Figure 32: Saved Configuration Pops Up as a New window for Easy Comparison.....	- 47 -
Figure 33: Newly Placed Block Has a Different Color than the Older Blocks Until After 2 Seconds	- 47 -
Figure 34: Unsuccessful Attempt – Physical Objects as Global Interactors	- 48 -
Figure 35: Project Set-up in Most Wanted Art Gallery, PA	- 50 -
Figure 36: Multiple People Playing with The Cubes at Once	- 51 -
Figure 37: Froebel Blocks.....	- 52 -

Figure 38: Users Experiment Did Not Want to Follow the Grids	- 54 -
Figure 39: User Pushing the Limits of the System with a More Complex Configuration.....	- 54 -

Abstract

This thesis explores the potential of a concurrent physical and digital modeling environment. Inspired by constructionist notions of embodied cognition in design, a novel interface for design modeling is presented where designers can take advantage of the affordances of both physical and digital modeling environments, and work back and forth between the two. Using Processing, along with the Kinect depth sensor, the system uses depth data read from a physical modeling space to produce an enhanced digital representation in real time. The result is a proof-of-concept concurrent physical and digital modeling environment where users can design by moving and stacking wooden blocks in a physical space, which is represented (and enhanced) digitally as a “voxel space.” Crucially, the system combines design affordances specific to each media: while the physical space offers tactile and embodied forms of design interaction, the digital space offers different views and parametric editing capabilities —as well as save configuration, and the capacity to perform basic analyses. Following a short review of experimental computational and tangible interaction design interfaces, the thesis discusses the system's implementation, its limitations, and next steps.

Chapter 1. Introduction

As children, we often think of creativity as being something hands-on and playful. However, as adults in a digital age, the design and creativity process became something more restricted behind the digital tools that dominate the profession and market place.

The advent of computer aided design and modeling software were breakthroughs that helped to transform and reshape the entire design and engineering industries. There are many strengths to digital modeling tools, including but not limited to...

- Drawing accuracy
- Encapsulation of data within different components
- Capability to create realistic renderings through texture mapping
- Performance and contextual simulation
- Navigation from different views on the same screen
- Saving and preserving data

With these computational advantages simplifying many processes while improving drawing accuracy, there is a shift in which we're moving away from physically prototyping things with our hands to directly drawing on the digital screen. However, what are some benefits to conventional physical modeling that many of us seem to be moving away from?

One major strength to physical models is that users can touch them directly. The affordance offered by the shape and materiality of graspable objects arouses our natural intuitions and desire to touch and play. For many, getting our hands moving can help fuel creativity and quickly generate ideas without focusing on unnecessary details.

This is supported by Constructionism's underlying idea of "learning-by-making," where the "learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe" (Papert, Harel, "Constructionism"). This concept was coined by Seymour Papert, who published his work titled "Constructionism" in 1991. He was largely inspired by Swiss psychologist Jean Piaget's theory on the nature of knowledge; Piaget believes that people actively construct "robust system of belief" through their experience with their world. His main argument was that as learners, we can understand things better after we construct them ourselves. Take the example of children, they might enjoy learning things like math more if they could apply that knowledge to activities such as building physical models.

According to Papert, "building and playing with castles of sand, families of dolls, houses of Lego," all have a quality of "learning-richness" (Papert, Harel, "Constructionism"). Papert tried to reproduce this quality of richness by designing many "construction materials" for children. One notable experience for him was seeing students freely practicing their imaginative fantasies with a soap sculpting project in an art class. He wondered why a traditional math class that uses "instructionist" method of education (learning through teacher's demonstration and doing problem sets), cannot be more like the art classes, where kids can construct something personal and meaningful. One example of Papert's legacy work is the Lego Mindstorms product line, which links software and hardware to allow kids to customize and build programmable robots. The first iteration of this work was called the LEGO TC LOGO, where children are engaged with 3 types of constructionist learning:

1. By building structures from LEGO pieces
2. By creating computer programs to control the LEGO pieces
3. By constructing knowledge in their minds from step 1 and 2.

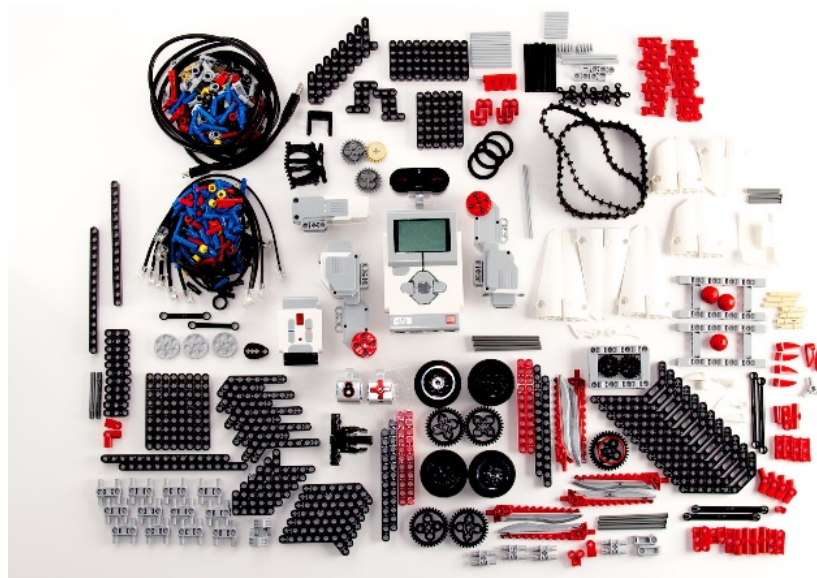


Figure 1: the Lego Mindstorms Kit¹

In addition, Papert's theory of the learning environment separates the "planners" from the "tinkerers." The "planners" are those who maximize workflow efficiency by planning everything out in advance and get to work. In contrast, the "tinkerers," like to "engage in a sort of 'dialogue' with their construction. This involves not necessarily having a solid plan all along, but rather, making incremental improvements by doing something first, stepping back to reflect, and then deciding on what to do next. Papert argues that the traditional schooling system tends to structure and formal, thus favoring the learning style of the "planners." I believe this is also evident even outside of school, where the typical work flow thrives on structure and schedule. However, creative work cannot necessarily be forced out of a structured environment, and as Papert asserted, forcing the "tinkerers" to work like "planners" is like forcing left-handed people to write with their righthand (Papert & Turkle, 1992).

¹ Lego Mindstorms. Digital image. <https://www.lego.com>. Web.

Another benefit of physical models is that many people can look and interact with one model at the same time from different angles. It is also inviting because it is a way of communicating with other people who may or may not be design experts; not only to colleagues, but also to clients. In the interview article “Why Do Architects Make Models?” conducted by Elisabetta Bono, Architect and educator Jan Schevers support this point with his account, “we have quite a lot of people coming into the office, and I always know that if I’m in front of my iMac, drawing something, everybody will sort of ‘leave me alone’. But if I’m working on a model, suddenly people will come up to me and ask questions about a project. I feel that I’m more open to say: ‘Where should this column be placed and why’, or: ‘That part is not working because . . .’” (20). In addition, Schevers also points out that he likes taking pictures of physical models because it easily allows us to study lighting. In rendering software, it often takes a certain amount of competency and expertise to generate realistic and accurate lighting renderings, whereas physical models are already situated in the natural environment (Schevers 21).

Connecting to the idea of the natural environment, could there also be a difference in how we interpret size and scale between physical and digital models? A group of students and researchers at Osaka University in Japan explored this topic in their research, “Spatial Understanding Between Physical and Virtual Models.” In the study, 24 participants were told to reason about spatial relationships using physical and virtual models. The models replicated an anonymous city and were built to scale in their respective realms. As defined in their paper, “spatial reasoning” refers to the “ability to understand the shape, size, location, and texture of an object or space.” The participants were asked to judge and compare size, height and scale of various elements in the models. The findings indicated that physical models allowed participants to make better judgements in both a quicker and more accurate manner (Sun, Lei, et al 34).



Figure 4 The experiment images (left: physical model and right: virtual model).

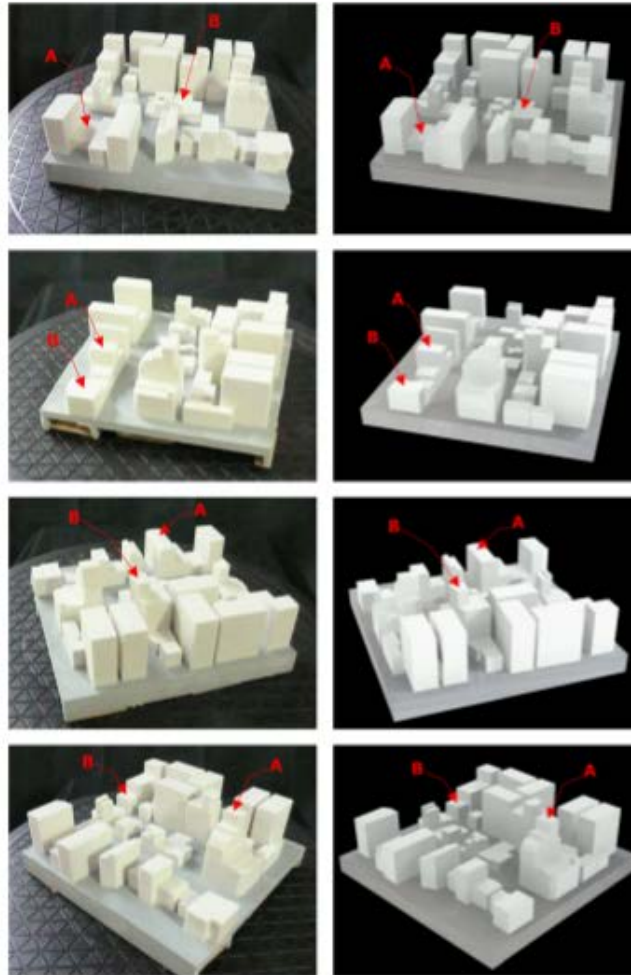


Figure 2: Spatial Reasoning Experiment with Physical and Digital Models²

² Sun, Lei, et al; the experiment concludes that we can reason about spatial relationships quicker and more accurately through physical models versus digital models.

In a quick recap, here are just a few relative advantages of physical models:

- Affordance
- Can help to quickly generate concepts without fussing over details
- Inviting for conversation
- Many people can interact with it at the same time
- Situated in the natural environment
- Allow for users to quickly understand overall formation and see the big picture

Often, the benefits of both physical and digital modeling can help to complement the other's shortcomings. It is ideal for architectural designers to use both mediums during their design process. However, the workflow between them is segmented. For example, in architectural studios, we often follow a sequential "waterfall" design process where we first develop conceptual low-fidelity physical model to generate ideas, and then move on to work with digital modeling to refine the details, and then, along with final digital renderings, building a finished high fidelity physical model to present the final design idea. In the field of software development and user experience design, the waterfall model is often criticized for being too rigid to account for the ever-changing design and development needs, especially after conducting user-testing. Instead, the current trend is to adopt an agile based model to develop a system through fast incremental cycles to improve design and development iteratively, making backtracking possible and inexpensive.

Could this agile, iterative approach to applied to the architectural design process, at least in the conceptual stage, by merging physical and digital modeling? One trending approach to integrate aspects of the real world to the digital worlds is to create immersive experience through virtual reality with haptic feedback; but I was more interested in examining a different approach

on the other side of the spectrum, where many have explored the idea of taking advantages of the affordance properties of tangible objects to simultaneously operate on the digital.

Chapter 2. Building Background

This section catalogues a few particularly inspiring academic literatures and case studies that explore various ideas on conjointly working with the physical objects and digital information.

2.1 Literature Review: The Architecture Machine

Nicholas Negroponte's 1970 book, "the Architecture Machine," marked one of the first visions of computer aided and participatory design. Negroponte envisioned a common language between the designer and the machine so that one does not need to be an expert to use the computer aided design system (9). Traditionally, design process started with 2D and then extrapolated to 3D. But at the time of the book, Negroponte already observed a trend (that's still evident today) where 2D information is being extracted from the 3D computer drawing. Negroponte critiqued that the formation of 3D model at an early stage "unconsciously implies the form of final solution." As such, this design process is limited by predefined elements, it is computerized, not a dialogue (13).

Alternatively, Negroponte brought up the idea of a "machine intelligence," where machines can become design partners with the humans. As mentioned in the book "Computational Design Thinking," Negroponte imagined as process where the "collaboration of human and machine is a circular event," and this could allow a "designer-machine unity" that "provokes a dialogue and the dialogue promotes a stronger designer-machine unity" (Achim & Ahlquist 78).

2.1.1 Case Study: SEEK/Blocksworld

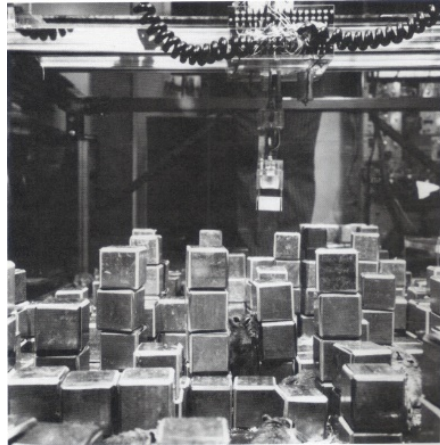


Figure 3: “Blocks” in SEEK³

The SEEK (Blocksworld) project signified one of Negroponte’s (and the Architecture Machine Group that was under his direction) attempts to explore machine intelligence. As the name suggests, it is a metal-block based “city” inhabited by gerbils. The machine gerbils are placed inside of a 5’ x 8’ glass box full of cubes as their environment. There is a computer extended by mechanical arm that makes sure the blocks remain in their original positions despite the constant disruptions caused by the moving gerbils. As the gerbils moves around knocking and pushing blocks, the machine is in a continuous loop to react to these unpredictable changes; this includes being able to recognize which blocks have shifted positions, and correct it in two ways: in a block has shifted from a grid cell, straighten it; if a block is on the floor and touching another block, stack it on top of another block.

³ “1969-70 - SEEK - Nicholas Negroponte (American).” Cyberneticzoo.com. N.p., 30 Dec. 2011. Web. 15 May 2017.

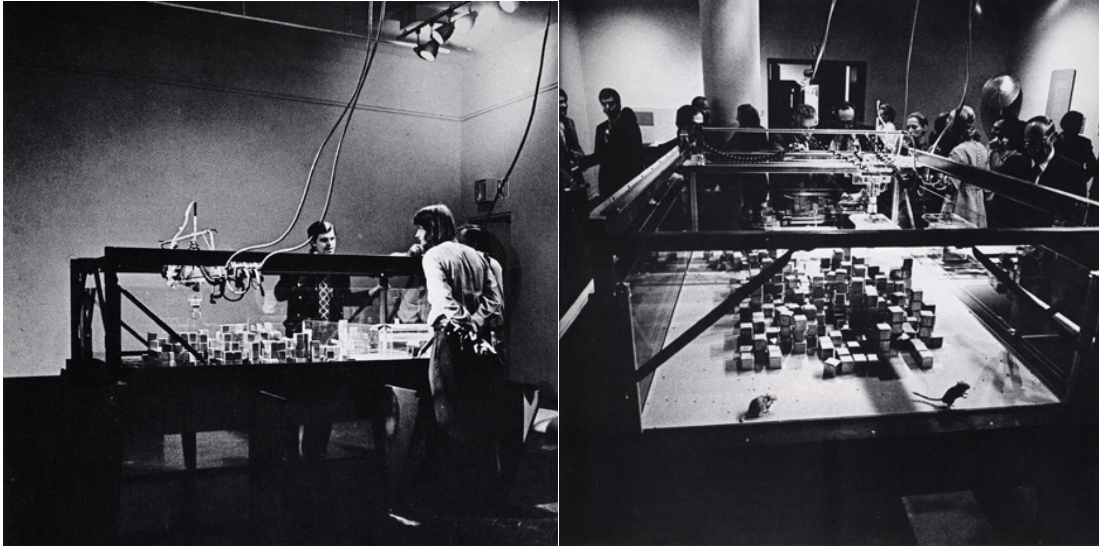


Figure 4: SEEK in Exhibition⁴

SEEK attempts to predict the randomness, and unpredictable nature of the world. In such ways, the computer can eventually become intelligent problem solvers, and thus worth of being noted as “collaborative partners” with the designer.

2.2 Literature Review: Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms

In 1997, Hiroshi Ishii, a professor at MIT’s Media Lab, along with his PhD student Brygg Ullmer, co-authored the paper “Tangible Bits: Towards Seamless Interfaces between People,

⁴ “1969-70 - SEEK - Nicholas Negroponte (American).” Cyberneticzoo.com. N.p., 30 Dec. 2011. Web. 15 May 2017.

Bits and Atoms.” The paper is often referenced as one of the pioneered publications on the field of Tangible User Interfaces.

The concept of Tangible User Interface (TUIs) calls for interaction designers to go beyond the dominant, but restrictive GUI model bonded by the computer monitor displays, keyboard and mouse. Instead, we can aim to embrace the duality between the realms of the physical world and the cyberspace (or “bits and atoms”) by allowing users to process and manipulate digital information through direct haptic interaction with graspable physical objects (to control digital information that pertains to them). The “physical embodiment of GUI,” as well as the coupling of physical and digital spaces, seek to take advantage of “human senses and skills people have developed through a lifetime of interaction with the physical world” (Ishii, Brygg 8).

2.2.1 Case Study: MetaDESK (Tangible Geospace)

For Ishii and Ullmer, the metaDESK signifies one of their first attempts at fulfilling the vision of “bridging the gaps between both cyberspace and the physical environment” by bringing tangibility to digital information (Ishii, Brygg 4). The metaDESK enables physical instruments to serve as tangible interfaces to many types and forms of graphical information.



Figure 5: Tangible Geospace on metaDESK⁵

One application of the metaDESK as demonstrated is the Tangible Geospace. This project uses models of MIT's Great Dome and the Media Lab as physical handles that users can hold on to control and manipulate graphical map of the MIT campus. An additional component is the "active LENS," where users can move around to see a simultaneous 3D representation of a particular building space.

The metaDESK is one of the first exploration of using graspable objects to sense and respond to physical stimuli, thereby becoming physical embodiments of digital information. This project becomes a prototype to "break free" from "the image of the computer as a monitor, keyboard, and pointer-endowed terminal.

⁵ Ullmer, Brygg, and Hiroshi Ishii. "The MetaDESK." *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology - UIST '97*, 1997, doi:10.1145/263407.263551.

2.2.2 Case Study: URP

In another example, Hiroshii Ishii and John Underkoffler's 1999 tool URP daps in the field of urban planning through the projection architectural elements. The URP application is based on an I/O bulb infrastructure, and serve as an example of "luminous-tangible interaction," where physical objects are coupled with visual information projected into them in real time reflecting the manipulation from each user interaction (Ishii & Underkoffler). In this project, physical architectural forms and models are placed onto a "workbench," and based on these forms, useful information such as shadows, proximities, reflections, wind and visual space are calculated and projected onto the "workbench."

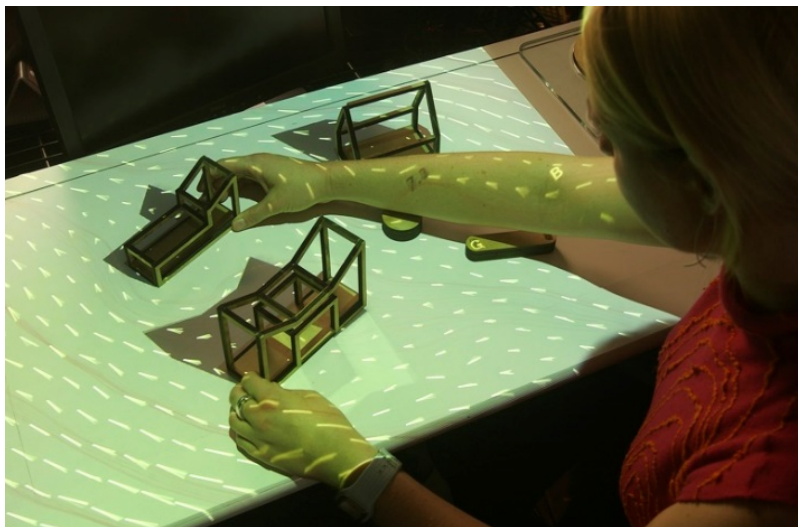


Figure 6: Urp - combining physical architectural elements with digital analysis ⁶

This project is significant because while MetaDESK allowed for physical objects to act as controllers, this tool allows digital and physical interfaces to be coupled to visually present

⁶ "I/O Bulb and Luminous Room." Tangible Media Group. N.p., n.d. Web. 15 May 2017.

information and analysis. The intend is for urban planners to gain computational and analytical powers of a computer through a natural interface.

2.3 Case Study: A Dynamic Physical Model Based on A 3D Digital Model

In a more recent example, “A Dynamic Physical Model Based on a 3D Digital Model for Architectural Rapid Prototyping” was a 2015 publication by Tomohiro Fukuda, Toshiki Tokuhara, and Nobuyoshi Yabuki of Osaka University. In its core, the study recognized that “both a physical model and a 3D digital model are three-dimensional visualization media to study space and shapes in a design and presentation process in the architectural and urban planning fields; however, they both have distinct strengths and limitations” (Fukuda, Tomohiro, et al 9). The best way to rectify this difference is by creating and using both physical and digital models conjointly. One way is to update both models back and forth. However, this task is not only time-consuming and costly, but also hard to respond and adapt quickly to frequent design changes. As such, the researchers have developed a “dynamic physical model system for volume simulation of buildings or a city based on 3D digital models” (9).

While the researchers mentioned many previous works involving previous studies on “architectural and urban presentation systems that combined physical models and 3D digital models” through technologies such as laser pointer, VR, LCD projections, there were still a few major challenges that the researchers hope to resolve in their research:

- 1) A system that synchronizes a physical model with a 3D digital model.

- 2) A system that represents a physical model at an urban scale, and individual buildings of various sizes and heights rapidly.

The proposed system is a take on a dynamic modeling system that synchronizes physical model with 3D model for volume simulation of buildings or city. a hardware based module consists of 105 (15x7) rods in a grid. A software extracts information from 3D data and can then control and lift the rods to different height based on its calculation, using a stepper motor.

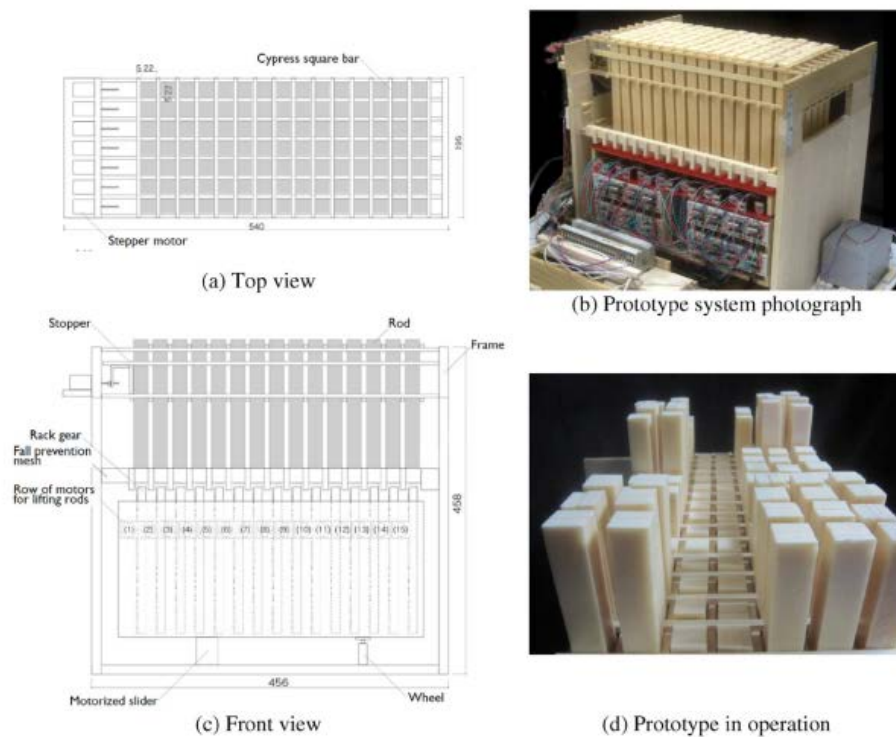


Figure 7: Case Study - Prototype System Physical Appearance⁷

^{7,8,9} Fukuda, Tomohiro, et al. "A Dynamic Physical Model Based on a 3D Digital Model for Architectural Rapid Prototyping." *Automation in Construction*, vol. 72, 2016, pp. 9–17., doi:10.1016/j.autcon.2016.07.002.

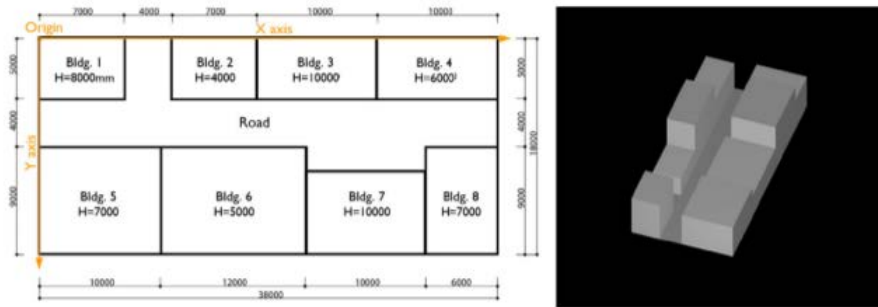


Figure 8: Case Study – 3D digital model representation⁸

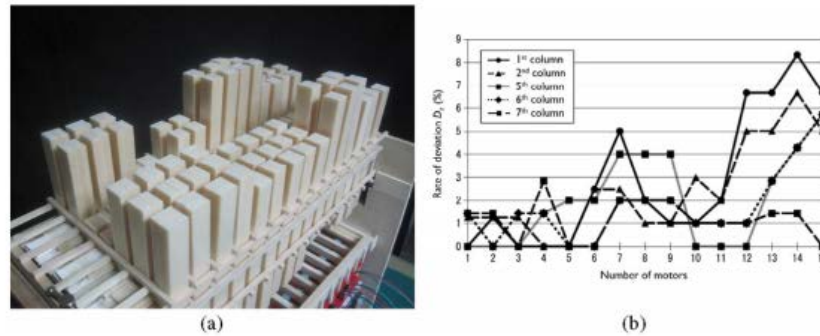


Figure 9: Case Study – Physical representation based on digital model⁹

To use the system, users will need to input a 3D BIM model software application and select the desired scale. Based on this information, the software will then calculate the height of each rod to be raised to represent and reflect a volume.

2.4 Observation and Reflection

Reflecting on these theoretical framework and case studies, it's easy to recognize the desire for many to connect the physical and digital worlds through synchronized interactions.

Negroponte's SEEK sought ways to present the machine as an interactive design partner by helping to manage a "chaotic" micro-world of cubes. The gerbils inhabiting the cube world represented unpredictable interactors (like humans in the real world), and in this process, we can see potential for programs to "understand" spatial configurations and manifest that knowledge in helping to construct (and reconstruct) a physical world. Going back to his vision, the machine can transform from a "yes-man" to a decision-making partner. While this study was focused on having the machine follow a pre-determined template to act in an uncontrolled environment, the cubes units mostly served as triggers whose displacement caused the computer to act. There could be potential to explore how the machine can be a helpful partner in a more intentioned design environment, where the cubes go beyond to serve as design interactors to help us make decisions to create something completely new.

Ishii's projects inspired many others to create tangible user interfaces that not only as physical controllers, but also be coupled with digital interfaces to present useful information. These work echoes particularly aspects of Mark Weiser's 1991 vision of "ubiquitous computing," where computing can appear anywhere in our lives, and our interaction with computers can be both intuitive and unnoticeable; but different from Weiser's vision, the use of big computational pads/boards transform physical objects into "GUI-style metaphors". As summarized in *Tangible Bits*, these projects were more about "awakening richly-afforded physical objects, instruments, surfaces, and spaces to computational mediation" (Ishii & Ullmer).

In the Dynamic Physical Model for Architectural Prototyping, the concept of synchronized prototyping modeling environment was presented. In their case, the synchronized modeling process went one direction from digital to physical because a software model must be first be finished, imported, and calculated for the physical model to be constructed. The physical model then merely becomes a realization of a digital representation, not something with its own identity that we can interact with.

Could there be another way to create this synchronized rapid prototyping environment by applying the concept of human-computer dialogue, as well as having a tangible user interface to couple digital information to physical objects? For example, instead of having to import a software model to construct a physical representation, can we synchronize the creation of a digital model from a physical configuration? If so, could this help us to have a more iterative design interaction where we can go back and forth between the two media? Would this allow us to simultaneously benefit from the advantages of physical and digital modeling?

Chapter 3. Hypothesis

If physical objects can serve as interfaces to their own digital representations, we would be able to benefit from the complementary values of physical and digital modeling by working concurrently and iteratively between both media. This would bring tangibility to digital data while allowing us to take advantages of both physical and digital affordances.

Chapter 4. Research Question

What kind of design processes can be enabled by a low-fidelity hybrid modeling environment that's distributed across physical objects and digital representations?

4.1 My Vision

To help test my hypothesis and assist my research, there were two early goals that underlined my project:

1. To prototype a modeling system that allows us to have a “conversation” between the physical and digital media by working back and forth amongst them
2. To experiment with tangible modular cubes in a structured gridded system that can be stacked and moved around, and at the same time, the digital representation of shapes and positions of the cubes would be reflected on a virtual modeling environment

In such ways, the tangible cubes can serve as interfaces to their own digital representations, we would be able to benefit from the complementary values of physical and digital modeling by working concurrently and iteratively to physically configure the cubes while editing and performing basic analysis in the digital environment.

Chapter 5. Method

This chapter documents my prototyping journey marked by a selection of prototypes and various technologies used before coming to my final prototype. I experimented with series of sensing technologies including the Arduino MPU, fiducial markers, and the Microsoft Kinect's depth sensor. I chose Processing as my software programming platform due to its ease of use and arrays of readily available libraries.

5.1 IMU Sensing: Arduino MPU 9250

The first experiments conducted was using an IMU sensor called MPU 9250. IMUs are inertia measurement unit sensors used commonly in electronic gadgets (i.e. smartphones, wearables, game controllers).

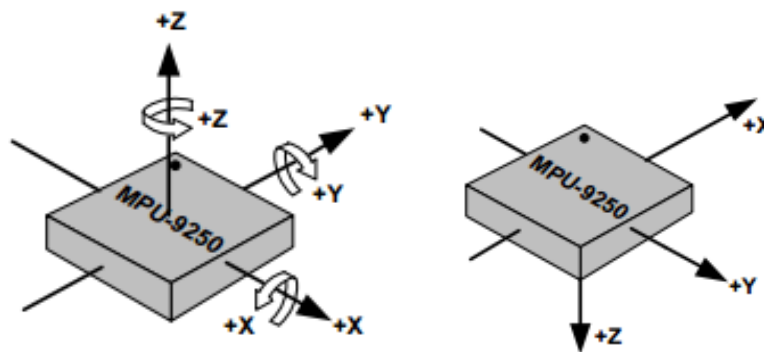


Figure 10: Orientation of Axes of Sensitivity and Polarity for Accelerometer and Gyroscope, Orientation of Axes of Sensitivity for Compass ¹⁰

^{10,11} "MPU-9250 Product Specification" InvenSense. Web. 16 January 2017.

The MPU 9250 is a type of IMU and an 9-axis (gyroscope, accelerometer, magnetometer) sensor from InvenSense.¹¹ The gyroscope measures rotation with respect to an axis, the accelerometer measures change in velocity, and the magnetometer measures magnetic fields. Working together, these sensors can help to report forces, angular rates, and the magnetic fields, allowing us to track positions based on rotational movements.

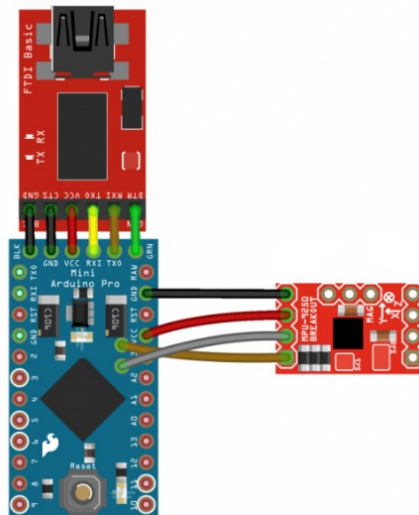


Figure 11: MPU to Arduino hook-up guide¹²

¹² “SparkFun IMU Breakout - MPU-9250.” Learn at SparkFun Electronics, learn.sparkfun.com/tutorials/mpu-9250-hookup-guide. Accessed 14 February 2017.

5.1.1 My Prototype

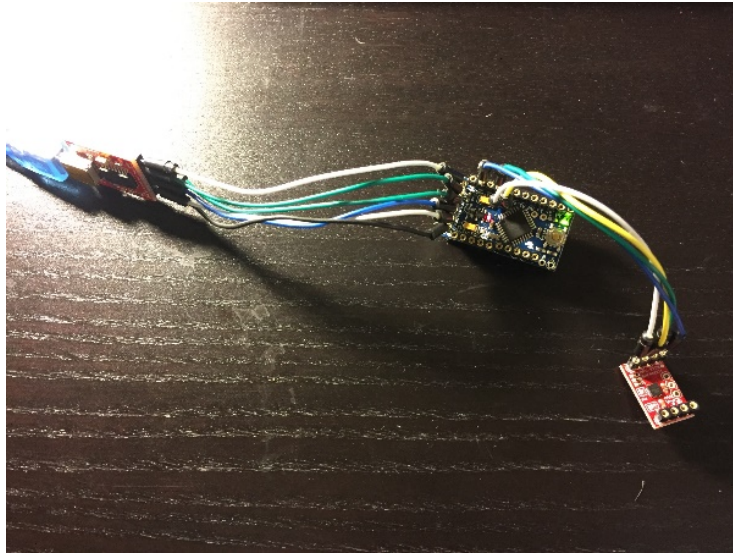


Figure 12: MPU to Arduino hook-up

Following the hook up guide and with the help of Arduino's MPU9250 library, I was able to read the values from the sensors, and map these values to yaw, pitch and roll degrees on a pre-drawn box on the screen. The idea was then to embed the sensor into a physical cube, and then its rotation will be seemingly mirrored by its digital representation (the pre-drawn box).

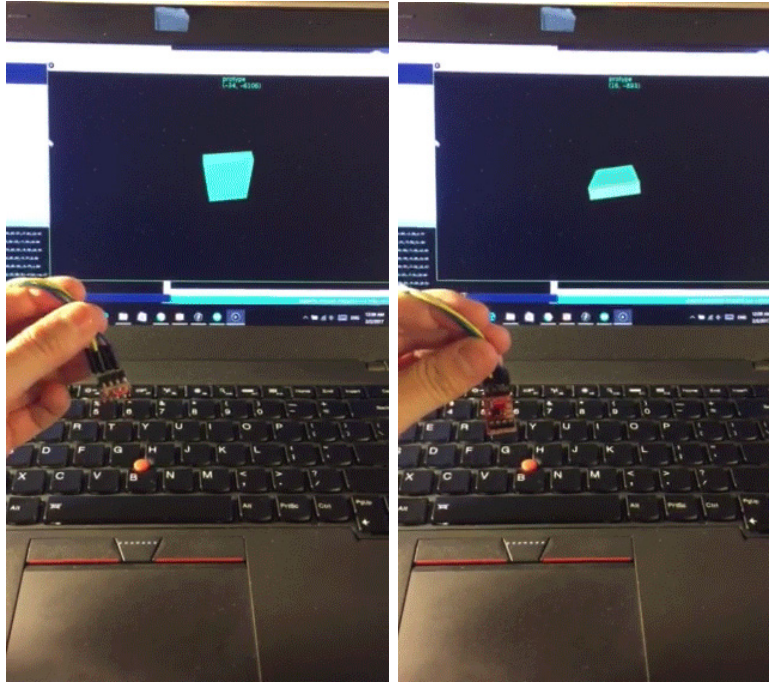


Figure 13: Mapping MPU Data to Digital Representation

5.1.2 Advantages and Limitation

In order to track the movement and rotation of each physical “cube”, each geometric unit would require separate sets of MPU sensor and wireless transmitters embedded within. The advantage of this is that we can receive data for each unit of “cube.” We don’t have to limit these data to merely positional, but also add more sensors to make each cube unit interactive. This idea is similar to the concept of “cookie scale computing,” demonstrated in David Merrill’s Sifteo cubes, where we can interact with groups of small pieces of computational devices (that can also communicate with one another) at once¹³.

^{13,14} Sifteo. "Sifteo Cubes." YouTube. YouTube, 22 Nov. 2012. Web. 16 May 2017.



Figure 14: Sifteos, the “Toys That Can Talk to Each Other”¹⁴

However, there are also a few disadvantages to this type of technology. One example is IMU sensor’ accumulation error. The sensor works by integrating acceleration to calculate velocity and position, therefore, any small measurement errors will become accumulated and thus lead to increasing inaccuracy in the calculation of positioning. As such, additional tracking assistances, such as a GPS, must be used to accurately track positional change (Goodrich, “LiveScience”).

Although there are a lot of potential for further exploration of this idea as an example of cookie scale computing, I wanted to focus on investigating the interaction potentials of a fusion modeling environment and developing a prototype quickly and at a relatively cheaper cost. Thus, I began to consider other technologies that may help to deploy this system in a quicker and cheaper manner.

5.2 Fiducial/AR Markers

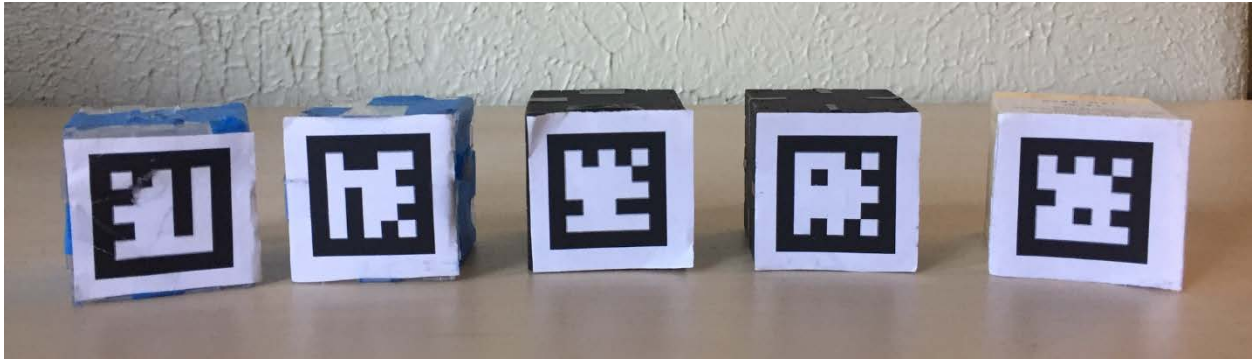


Figure 15: Fiducial Markers on Cubes

The second prototype was experimented by using Augmented Reality marker tracking. The library I used was the NyARToolkit library for Processing.

Augmented Reality combines the physical and virtual worlds through computer vision by fusing data together to overlaying digital information on top of the real world. One way to do this is through marker based recognition. A “marker” image was pre-defined, and later recognized by the camera, the computer then can calculate its position and orientation, allowing us to overlay with desired graphics or information.

How do AR markers work?

AR markers are usually square, marked with uniquely defined black patterns in the center, and surrounded by contrasting light (usually white-colored) padding. The software, such as a library, searches through each frame of a given captured camera view for square markers. If a square marker is found, then the software tries to match and identify the square’s embedded pattern to something known or pre-defined pattern in memory, if successful, the software will then calculate the position and orientation of the markers relative to camera. With this

information, we can create a computer graphics model with position and orientation as an offset based on the position and orientation of the camera, all in real time.

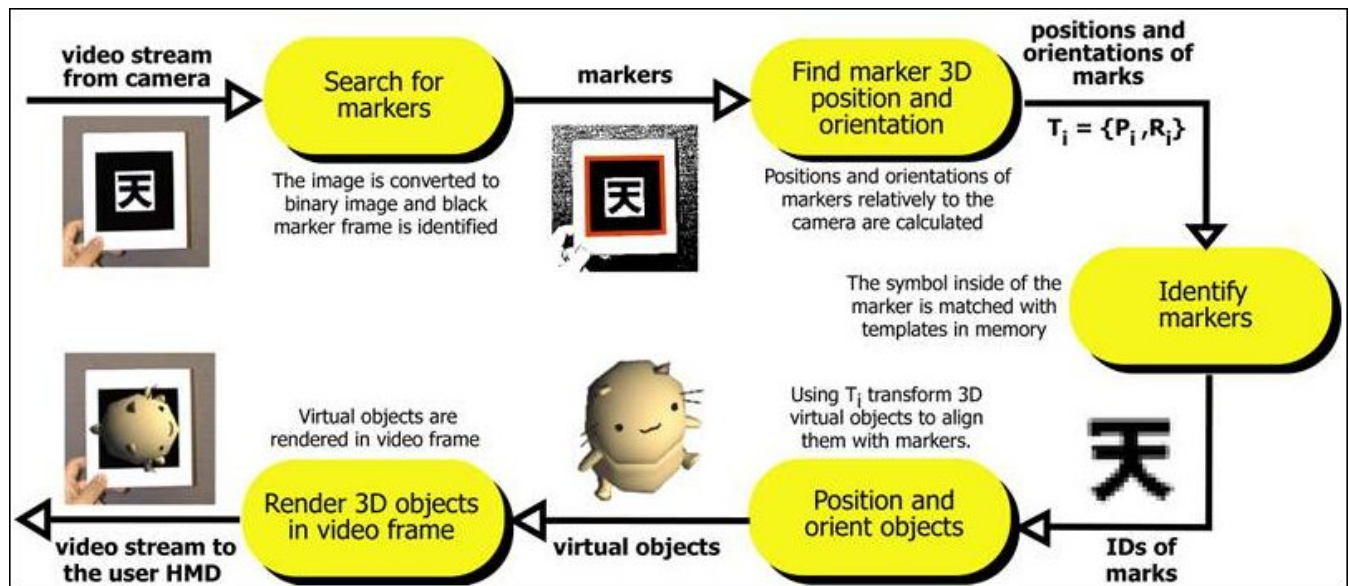


Figure 16: Flowchart - How AR Markers Works¹⁵

¹⁵ "About the Traditional Template Square Marker." ARToolKit. N.p., n.d. Web. 6 March 2017. <https://artoolkit.org/documentation/lib/exe/fetch.php?cache=&media=diagram.jpg>

5.2.1 My Prototype

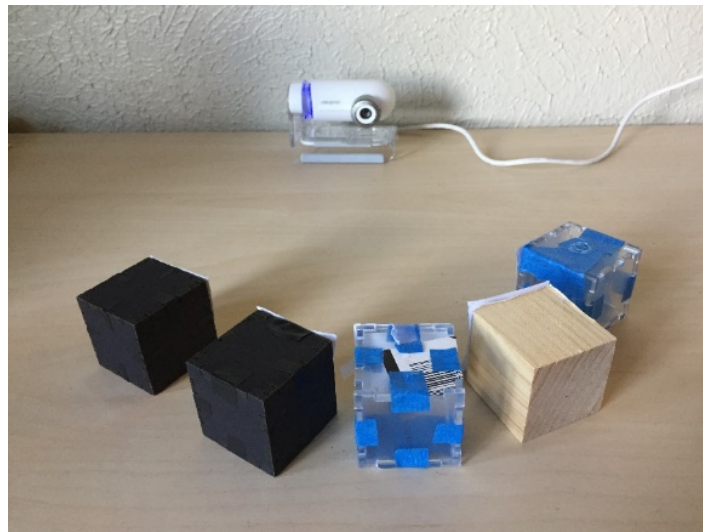


Figure 17: AR Marker Prototype Set-up

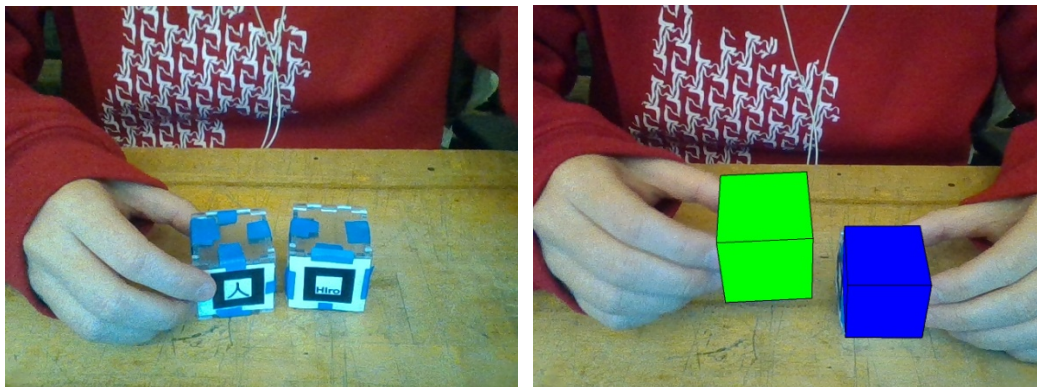


Figure 18: Mapping Shape to Markers in AR

For this prototype, I cut out 5 pre-defined markers recognizable by the NyARToolkit library and taped them to 5 corresponding 1.5in cubes. In Processing, the library can help us to recognize the marked cubes on the screen, and we can even draw different primitive geometries onto the place of the marker.

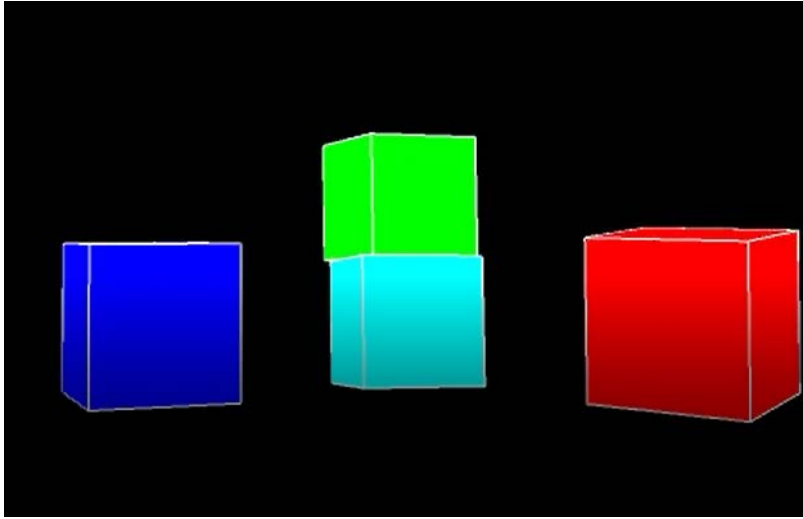


Figure 19: Digital Representations of Makers in Model Space

5.2.2 Advantages and Limitation

Like the MPU sensor, the marker based system provided positional tracking of desired objects in real time, including rotational changes. Although using computer vision, the markers enables it to eliminate noise data such as hands and other background objects on screen, allowing us to get a clear and desirable view of the model space.

There were three main limitations of the Marker based system. First, the marker data can be easily interrupted if the computer's perception of the markers is disturbed (i.e. covered by hand or another object). Secondly, tracking results are affected by lighting conditions; glare spots and shadows can easily disrupt tracking of the markers. Finally, the markers were not very good at recognizing depth. For example, moving a marker-tracked cube away from the camera caused the cube to become smaller only because the marker appeared to be smaller to the camera, not

because a change in depth was recognized. In other words, the markers were being tracked on a relatively shallow x-y centric basis, without knowledge of relative and absolute depth information.

5.3 Final Prototype - Kinect Depth Tracking

To better track object positions in a 3D space, I decided to consider depth sensing technologies.

The final prototype uses the Microsoft Kinect's depth sensor to track cubes on a gridded physical plane, and maps that information to voxel representations in the digital space.

5.3.1 Kinect and Voxel



Figure 20: Microsoft Kinect¹⁶

Kinect is a depth camera that takes computer vision to go beyond just RGB pixels. It uses infrared light technology to record where things are in space.

¹⁶ Microsoft Kinect. Digital image. Microsoft. N.p., n.d. Web.

Kinect 2 - Specs

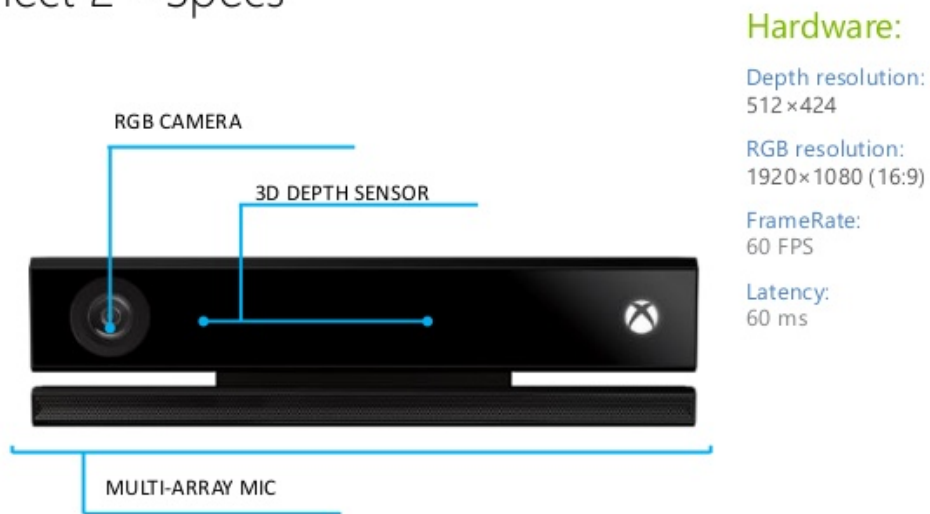


Figure 21: Microsoft Kinect V2 Specifications¹⁷

The Kinect has a regular RGB camera as well as a 3D Depth Sensor and a microphone. In this project, I mainly used its 3D depth sensing feature. The depth sensor emits infrared light to create a depth image that captures where things are in space. Unlike a traditional RGB camera, which presents how things look like by capturing light and displaying them through colored pixels, the depth camera can store information on where each pixel is located at in a 3D space (Borestein 2).

¹⁷ Valoriani, Matteo. "Programming with Kinect V2." 3 June, 2015 Microsoft PowerPoint file.

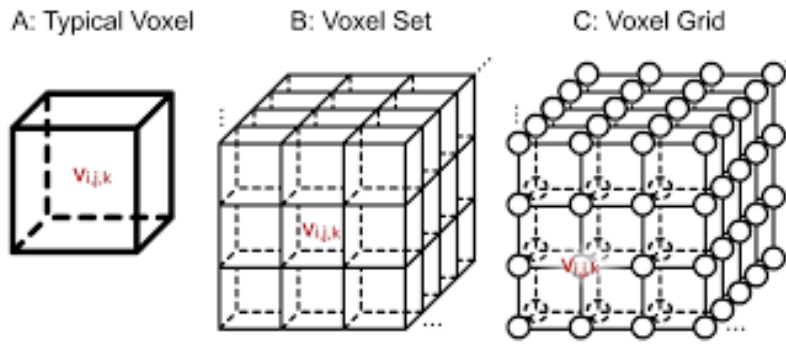


Figure 22: Voxel Representations¹⁸

Voxel, commonly described as a “3D pixel”, is a value on a regular spaced grid in a three-dimensional space. Like a pixel, which has x and y coordinates, a voxel has x, y and z coordinates. With depth information captured with the Kinect, we can then we can reconstruct a 3D model of what the Kinect “sees” by mapping data points to virtual voxel points, and even manipulate the model and reference different views.

¹⁸ "Scientific Visualization: Volume Surface Rendering." Scientific Visualization - Project 1 Report - Richie Zirbes. N.p., n.d. Web. 16 May 2017.

5.3.2 The Set-up

The underlying concept is to convert depth data from physical space to voxel representation in digital space. Reference the steps in the process figure below:

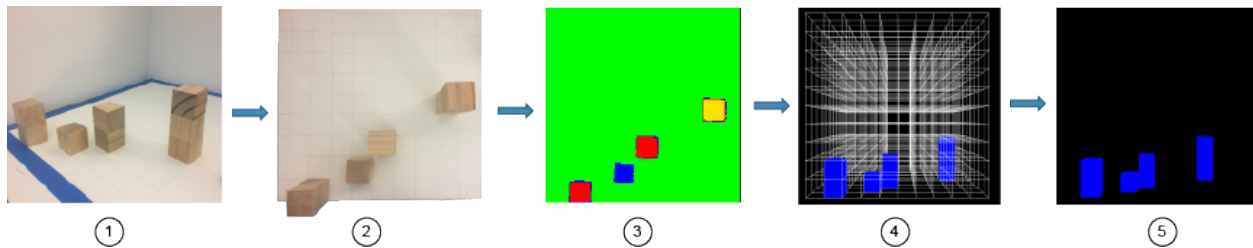


Figure 23: Depth to Voxel - Mapping Process

- 1) Cubes in physical space
- 2) Top view of cubes from Kinect's perspective
- 3) Kinect's depth data as represented by color
- 4) Boxes are drawn inferred from depth data in voxel space
- 5) Digital representations of cubes in physical space

The physical set-up

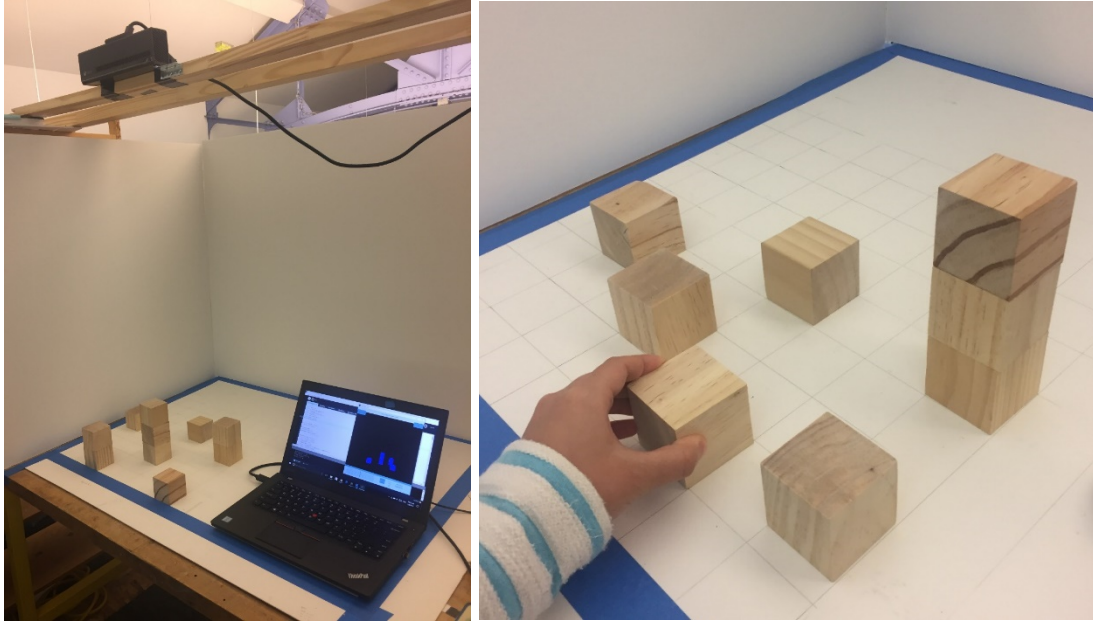


Figure 24: Physical Set-up

The Kinect is hung 3 feet from the table and looks down, its viewport cropped to an evenly gridded modeling space that fits exactly ten 2 inch cubes horizontally and vertically. The Kinect then loops through the cropped viewport and surveys the center points of each voxel in digital space. If it senses a height taller than the distance from the table, then at least one cube must have been placed on the point surveyed; we can infer the height (or how many cubes are stacked) based on this depth data, and as a result, the drawing space turns on the corresponding voxels, and we can define the geometry that we would like placed onto that voxel.

The Basic Algorithm:

```
1 void kinect() {
2     // first, get depth data in array
3     int [] depth = kinect.getRawDepthData();
4
5     // each voxel takes up about 20 pixels, we'll survey depth data from every 10th voxel
6     // this will allow us to survey only the center point
7     int skip = 10;
8
9     // cropping kinect's viewport to fit confined physical area
10    int crop_xStart = 165;
11    int crop_yStart = 20;
12    int crop_xEnd = KinectPV2.WIDTHDepth-130;
13    int crop_yEnd = KinectPV2.HEIGHTDepth-187;
14
15    // looping through pixels in the cropped viewport, survey every 10th pixel
16    for (int x = crop_xStart + skip; x < crop_xEnd; x += skip) {
17        for (int y = crop_yStart + skip; y < crop_yEnd; y += skip) {
18
19            //calculate and pinpoint depth array index to read from
20            int offset = x + y * KinectPV2.WIDTHDepth;
21            int depth_data = depth[offset];
22
23            // map pixel from kinect to voxel in digital space
24            int voxel_xPos = int(map(x, crop_xStart, crop_xEnd, 0, dimension));
25            int voxel_yPos = int(map(y, crop_yStart, crop_yEnd, 0, dimension));
26
27            // use this information to figure out how tall the cube in voxel space should be
28            // turning on and turning off columnes of voxels appropriately
29            map_depth_to_cube_height(voxel_xPos, voxel_yPos, depth_data);
30        }
31    }
32 }
```

Figure 25: Fundamental Algorithm for Prototype

5.3.3 Exploration of Interaction

With the set-up ready, I began building various features to explore and enhance user interaction and experience this concurrent modeling environment

The User Interface

The user interface was created with the G4P library in Processing.



Figure 26: The User Interface

Concurrent modeling environment

First and foremost, the prototype allowed users to move and stack the cubes orthogonally in the x, y and z directions and these changes were reflected in the digital space.

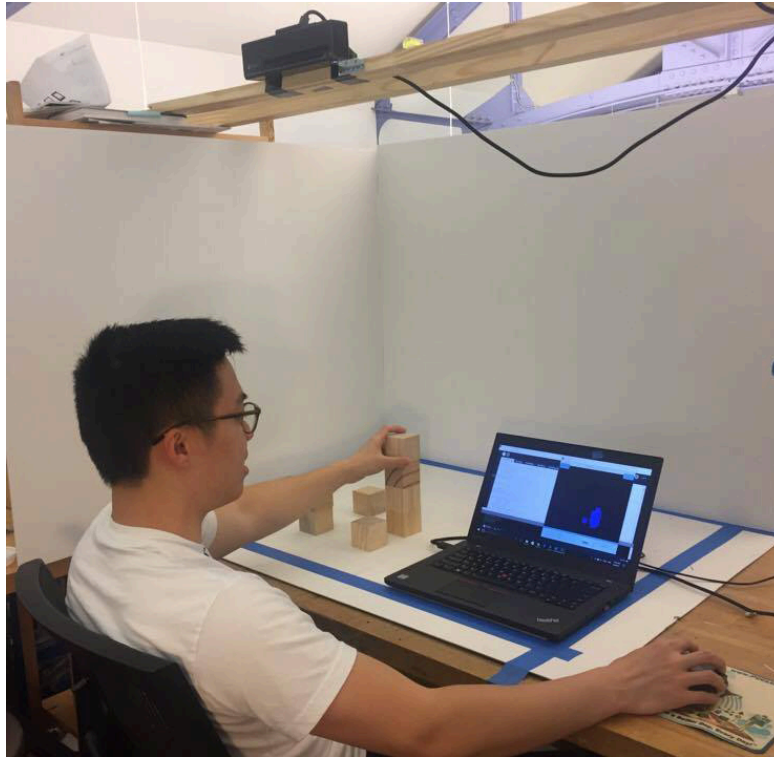


Figure 27: User Stacks Cubes While Editing the Geometric Representation

multiple points of view

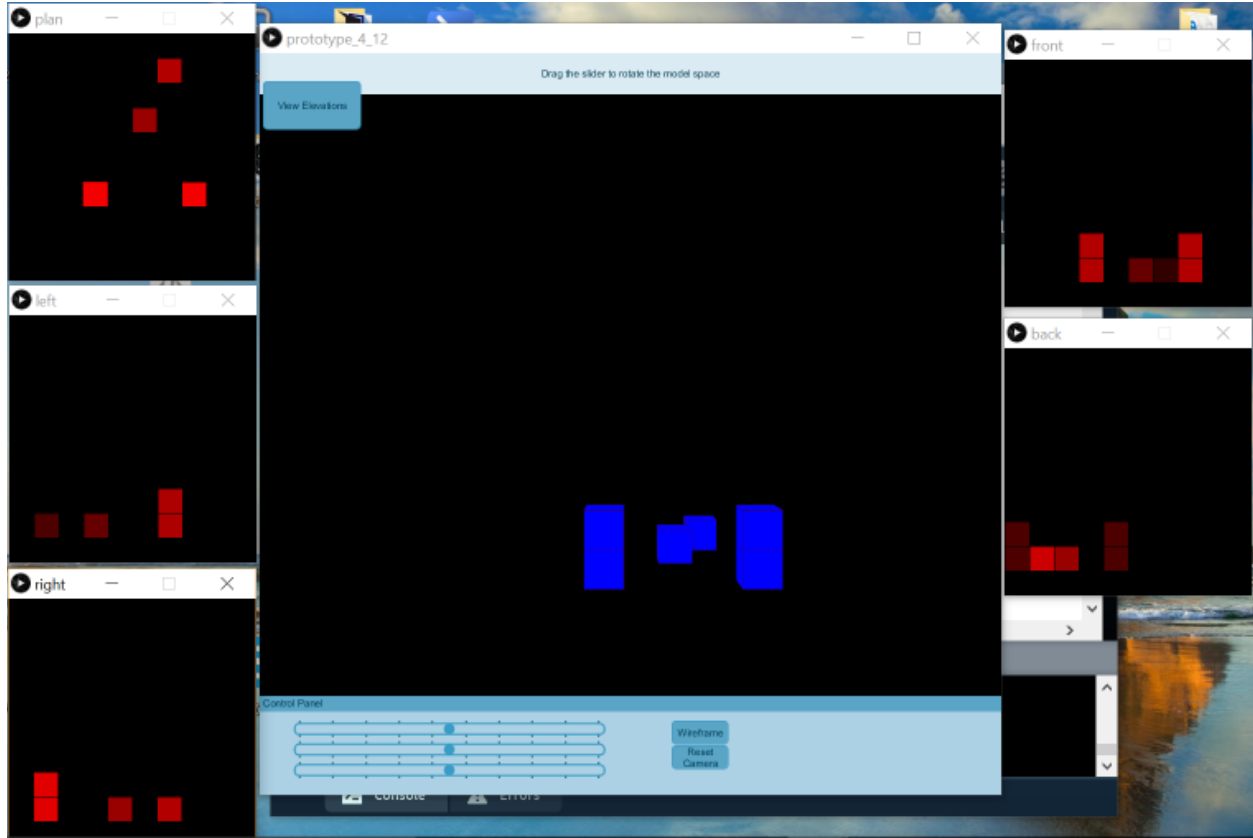


Figure 28: Multiple Points of Views

Like most 3D modeling software, we can offer perspective views that can rotate about the x, y, and z axis, as well as plan view, left, right, front and back elevations. The brightness of the color in the plan and elevations indicate how close or far away it is from the view of the “camera” – the brighter the color, the closer the object from that particular viewpoint. These different viewpoints get updated in real time with user interactions.

Abstraction of Geometric Representations

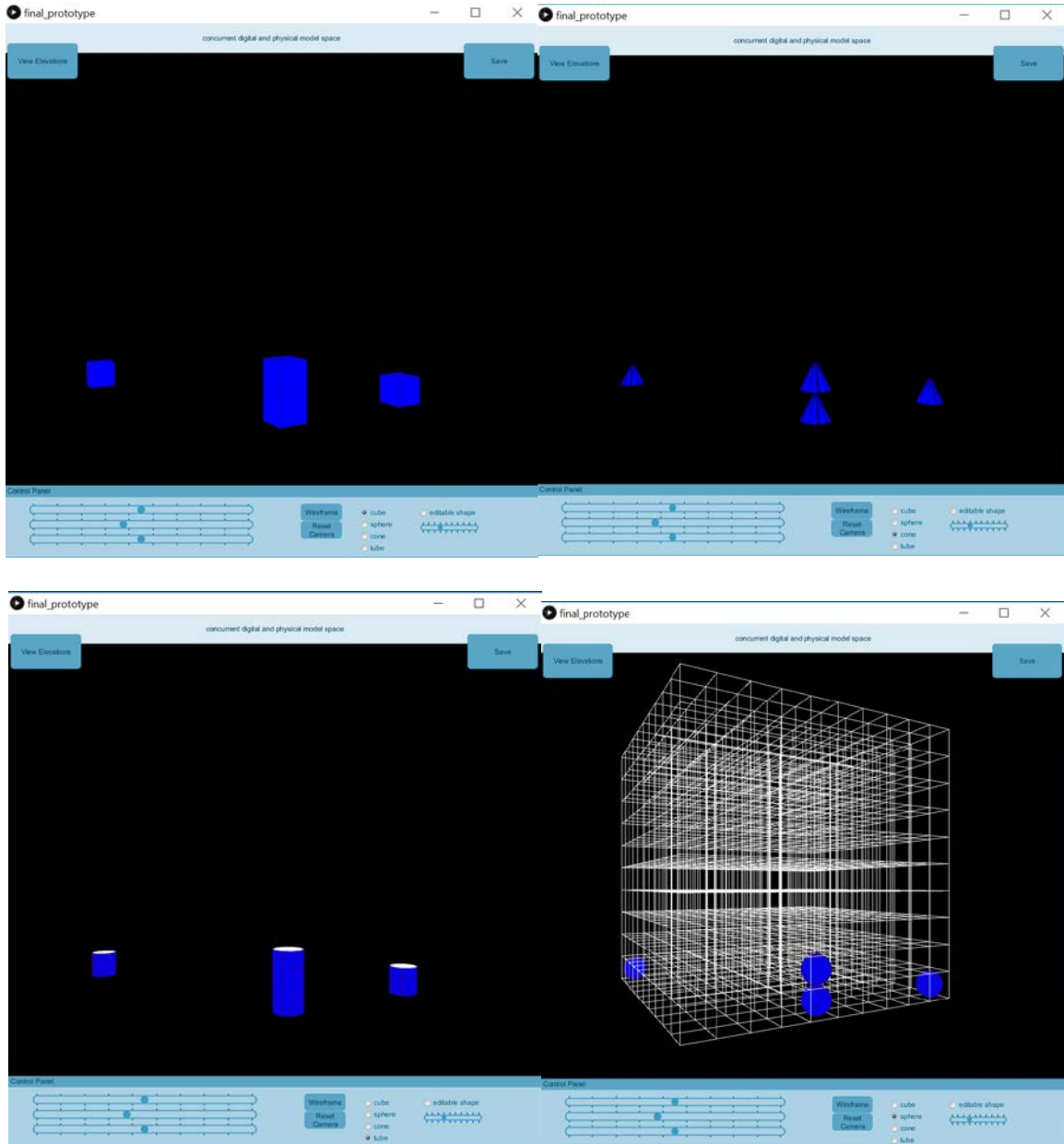


Figure 29: 4 Primitive Representations (Cubes, Spheres, Cones, tubes)

The interface offers 4 basic primitives to quickly allow for the change of geometric representation on the digital screen. The physical object becomes merely an interactor, encouraging the experimentation of formations with different shapes

Editable Geometric Representation

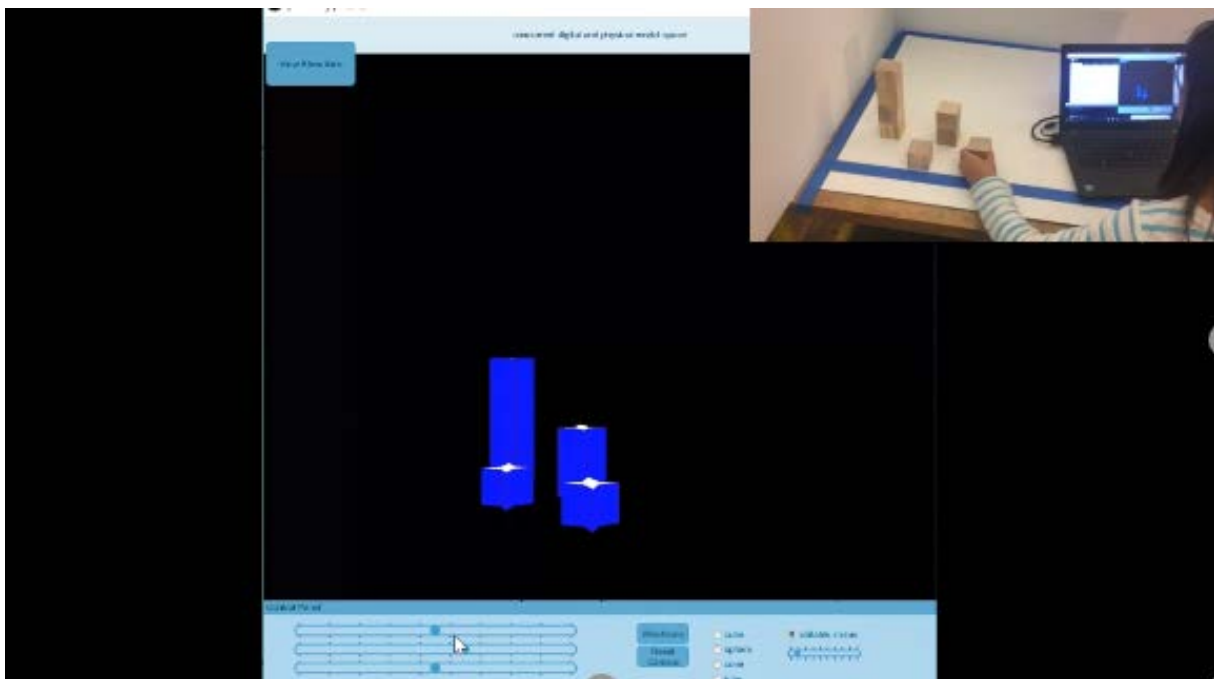


Figure 30: User working back and forth between physical and digital modeling

In addition to primitive 3D shapes, the system has a simple prototype for shape editing ability, where users can slide a bar to explore basic parametric manipulations. This was built on the idea that users can work back and forth between the two media, playing around with the configuration of the cubes in physical space, while editing shapes in the digital space

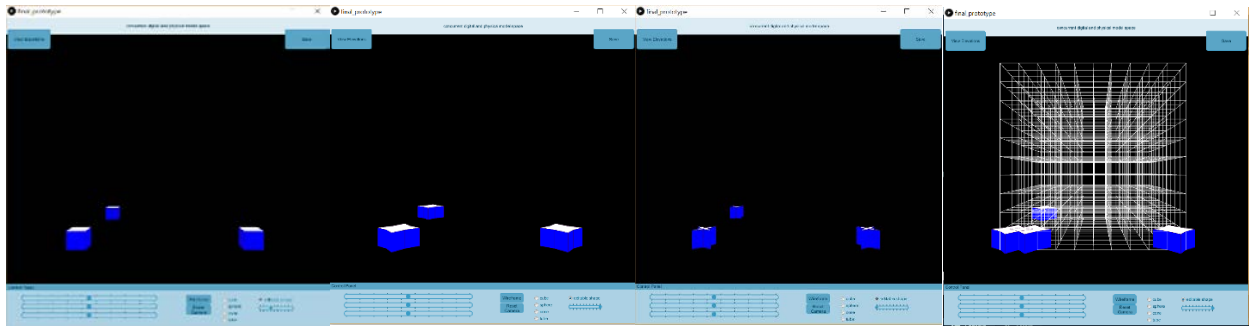


Figure 31: Slide Bar Allows for Users to Edit Parametric Shape That Expands Beyond Its Respective Voxel Constraint

Saving Configurations

In the physical context, quickly generating many prototypes means we either have to generate many models with different objects, or reuse material by losing a certain configuration to recreate another. In software modeling, we can save configurations and even preserve and revert to previous states through version control tools. With the save feature in my prototype, we can save configurations before rearranging physical objects and editing geometric forms. The saved configuration immediately pops up as a new window, allowing for quick comparisons.

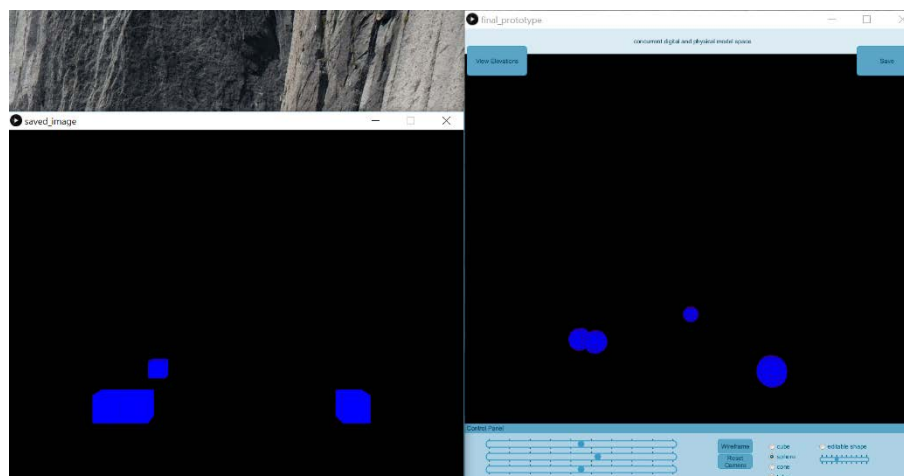


Figure 32: Saved Configuration Pops Up as a New window for Easy Comparison

Representing the abstraction of time through color

Another possibility is to track how long we have placed down certain objects, and represent this information by changing the color of the geometric forms in the digital space. The potential of this feature is that we can distinguish newly placed objects from older ones, to differentiate the temporary movements and noise data from the permanent and intentional. We can also take this concept further to map brightness of the colors to reflect the time for which each unit of geometry was placed.

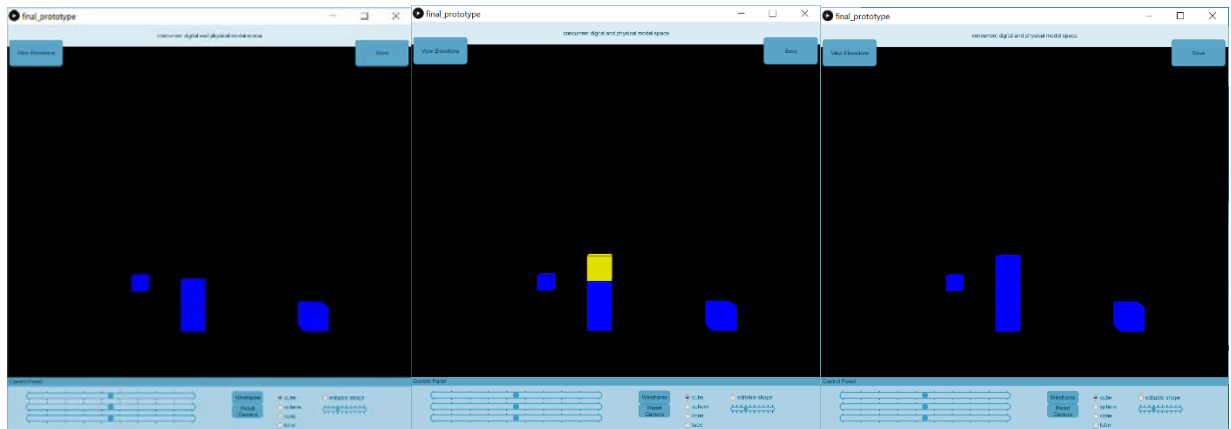


Figure 33: Newly Placed Block Has a Different Color than the Older Blocks Until After 2 Seconds

Failed experiment – physical objects as global manipulators

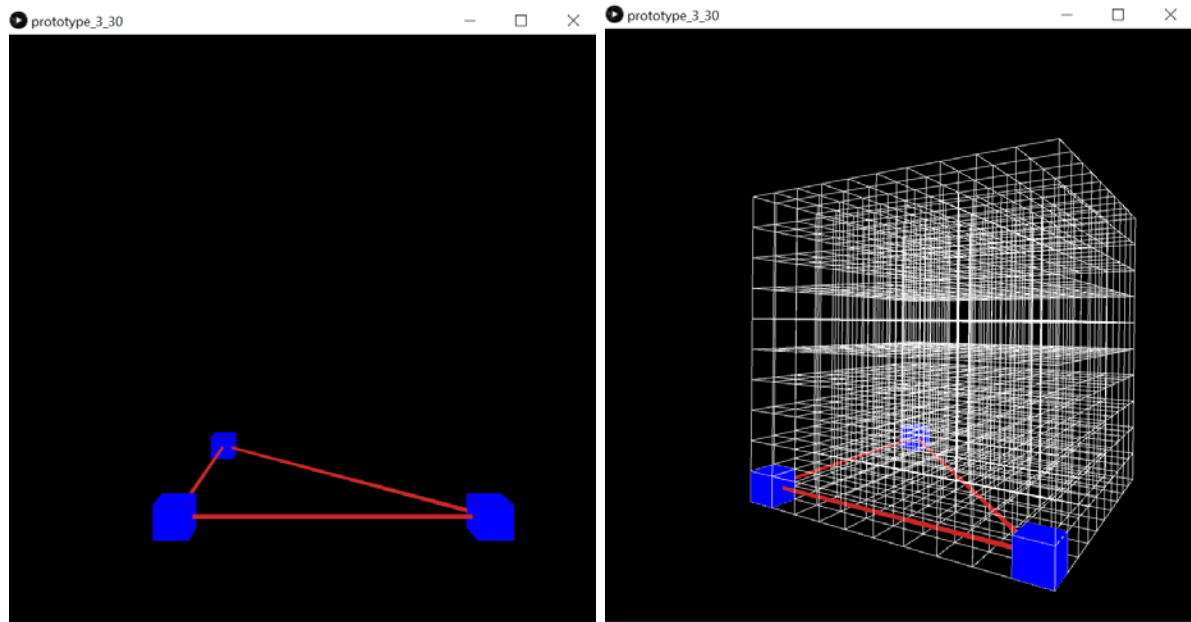


Figure 34: Unsuccessful Attempt – Physical Objects as Global Interactors

In addition, I also spent some time experimenting using the cubes as global interactors. The concept and (the naïve algorithm employed) was to simply draw a line between each cube. The length of the line can be calculated to represent distance between each pair of cubes. The enclosed space between the lines can then calculate the area formed by the cubes. The vision was to be able to update these lines and corresponding enclosures in real time along with movement of each cube. Although the naïve algorithm worked well for the first 3 stationary cubes shown in Figure 34, as soon as the number of cubes increased, there are cubes being stacked, or noise data from user's hand came into the space, the system becomes overwhelmed and crash. Perhaps future exploration of the project can dive into building a more robust algorithm to account for these factors.

5.4 Discussion

Our Code Lab cohort had the opportunity to display and showcase our thesis work in Most Wanted Art Gallery in Pittsburgh, and we became a part of the Pittsburgh Cultural Trust's monthly Gallery Crawl event. From there, we had the perfect chance to demo our projects to a range of gallery crawlers consisted of local Pittsburghers, students and even travelers, and thereby both test and observe their interaction with our projects. I will base my discussion on the observation of around 30 user interactions that I've witnessed. Most of them did not know anything about my work.

5.4.1 Benefits of the set-up

Due to Kinect's depth capturing capabilities, it is not sensitive to lighting conditions; as a result, the lighting conditions had insignificant effect on the prototype. Even though the final display was set up in the basement of an art gallery without abundant lighting, this did not affect the performance of the system.



Figure 35: Project Set-up in Most Wanted Art Gallery, PA

5.4.2 Observation of User Interaction

When users approached my system, everyone's first action was to touch the cubes. Relating to Hiroshi Ishii's arguments, people seemed more comfortable using their natural senses through playing with physical affordances offered by the cubes; despite the computer and mouse being placed in juxtaposition. There was also an observable difference between children and adults...while the children immediately went to play with the cubes, the adults appeared more cautious and carefully surveyed the set-up before attempting to move the cubes around. The children merely enjoyed playing with the physical cubes and seeing the mirroring digital

representations, but the adults tried to understand how the system worked, and tested its limits by moving things around to find the “boundaries” and making various configurations to what works and what doesn’t. It was through playing with the physical objects that they learned about how the system worked, what it was, and what the limitations were.

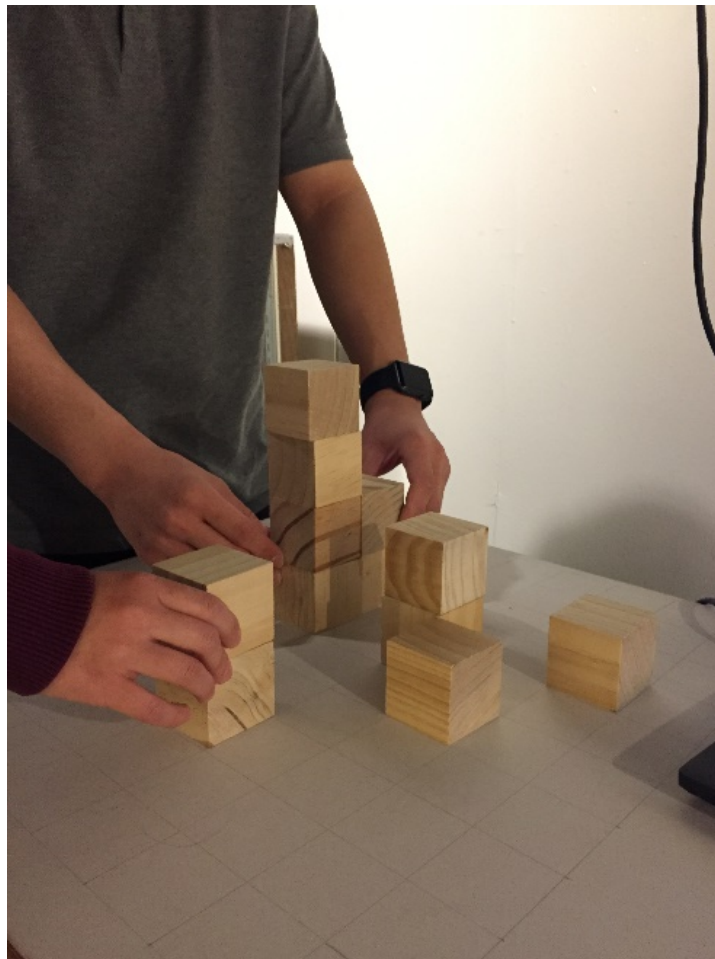


Figure 36: Multiple People Playing with The Cubes at Once

Echoing the idea of constructionism, it turns out that similar ideas involving playfulness and learning were also presented in 1800s by famous German educationalist name Freidrich Froebel.

Froebel was a believer of innate productivity and creativity of humans, and he appealed to developing an educational environment that involved the direct use of materials because it allows children to be creative and become aware of the environment at the same time. As such, he developed the Froebel gifts, play material to be given to children.



Figure 37: Froebel Blocks¹⁹

This set of toys influenced world renowned architect Franklin Lloyd Wright, who attributed to learning of the geometry of architecture through childhood play with these blocks (Brosterman 13).

¹⁹ "Friedrich Froebel (Fröbel)." Infed.org, 7 Jan. 2013, infed.org/mobi/friedrich-froebel-frobel/. Accessed 14 May 2017.

5.4.3 Contribution

Relating back, the main contribution of my project is a modeling environment that enables users to manipulate a design representation concurrently via tangible and digital representations, all while taking advantages of each representation with their own affordances. The physical context supports aggregation, stacking, removal, combination. The digital context supports basic geometric representations and editing, semantic modeling, camera space rotation and viewpoints, and visual analysis (by implementing a timer that allowed virtual geometries to change color based on how long ago they were placed).

5.4.4 Limitation

Relating to the playfulness and testing-the-limits traits displayed in many users, it's no surprise that the gridded system at the physical context had many limitations. Besides those who were already familiar with my project, first-time users mostly did not want to follow the grid that was drawn, they wanted to break the rules. In addition, many experimented with stacking up very creative configurations (such as rotating the cubes, stacking with overhangs) to test my program and push its limits, the program is currently too rigid and not robust enough to support these types of configurations.



Figure 38: Users Did Not Want to Follow the Grids

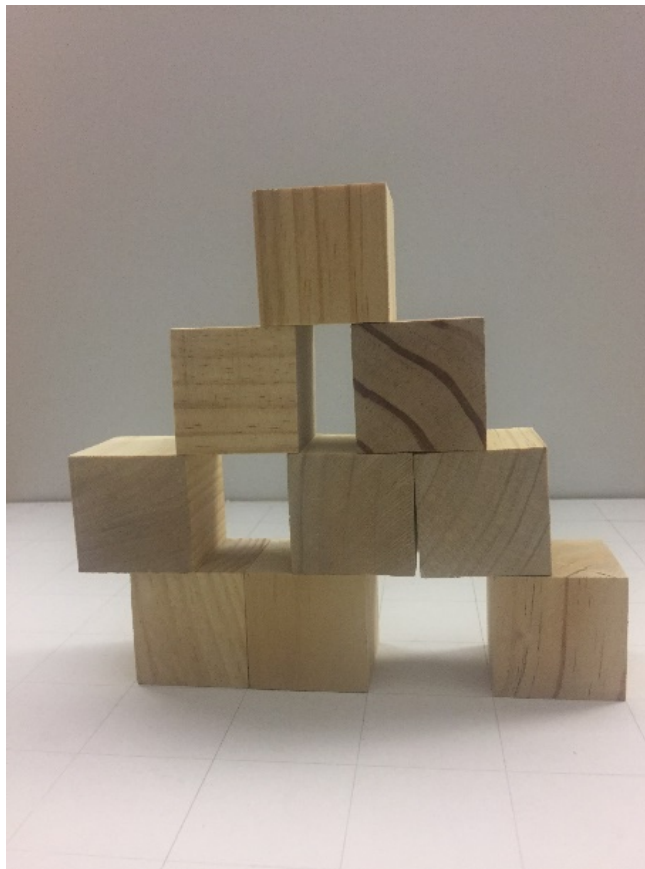


Figure 39: User Pushing the Limits of the System with a More Complex Configuration

In addition, the physical cubes currently only serve as inputs to the digital model space to communicate orthogonal location in space, but the digital model cannot communicate with the cubes; thus, the interaction in this front is limited.

Lastly, the system cannot distinguish between temporary movement (from humans) versus movement of physical cubes. The noise data from movement of human hands were fascinating to some, but also distracted many others. I observed one man who flinched when he first tried to touch the cubes after seeing the noise data appearing in digital model (he thought he had broken the system). Although there were recommendations to remedy this by using colors and timer, the shortcoming was still that our hands must be away from the physical model space in order for us to see the accurate digital representations.

5.4.5 Next Steps and future potentials

This project can benefit from more formal task oriented testing, especially to explore producing work in both digital and physical realms. The observations I made at the Art Gallery were interesting, but as an art installation, people were significantly more attracted to the element of “play” embodied in the presentation and set-up itself. Although I told many about the editing abilities in the digital space, most people were more interested in playing and stacking the cubes while seeing the digital representation on the screen, than figuring out what the buttons and slider bars were doing in the model space.

In addition, there are several improvements that can be made. For example, we can use the learnings from early prototypes that used fiducial markers and physical computing, and combine these technologies would allow for us to have a more robust program.

Employing markers and physical sensors to be added to the cubes can help with both tracking accuracy as well as tracking additional positional movements such as rotation and flipping of the cubes while eliminating noise data from user's movements. Another benefit is that we can then be able to identify each cube as individual elements, and thereby take advantages of object oriented programming, and open up possibilities of a physical realization of the MVC framework to allow elements of physical and digital model space to engage in a two-way conversation. For example, what if clicking on a particular cube in digital space cause the corresponding physical cube to light up or vibrate?

Finally, the ideas and concepts taken from this project can be applied to malleable physical objects in future studies. We can take user manipulation to go beyond merely spatial movement to also include the recognition of object deformation.

5.5 Conclusion

In conclusion, my thesis prototyped a low-fidelity modeling system that distributes across physical and digital media. It enables real-time interaction of tangible objects and digital primitives. It offers direct physical affordance to cyberspace geometrics, allows for geometric manipulations for users to work directly and back and forth between the two realms, multiple digital views and perspectives, a “save” feature to quickly capture and preserve various configurations, as well as a simple visual analysis of time using color. Based on a limited number of informal user interactions observed, the system appears to offer the complementary benefits of both physical and digital modeling. However, more formal user-testing can be conducted to gain data-driven and quantifiable data.

As a low-fidelity prototype, there exist some shortcomings with robustness and freedom of user interaction, which can be further improved in future works by adding other technologies such as fiducial markers and physical computing. Lastly, I hope that the ideas and methods presented in this thesis can serve as an inspiration and base springboard to other studies to allow for concurrent works in physical and digital spaces in higher fidelity prototypes.

References

- “1969-70 - SEEK - Nicholas Negroponte (American).” *Cyberneticzoo.com*, 30 Dec. 2011, cyberneticzoo.com/robots-in-art/1969-70-seek-nicholas-negroponte-american/. Accessed 27 April 2017.
- "About the Traditional Template Square Marker." ARToolKit. N.p., n.d. Web. 6 March 2017.
- Alfred Adler, Friedrich Froebel, & Frank Lloyd Wright. N.p., n.d. Web. 16 May 2017.
- Borenstein, Greg. *Making Things See: 3D Vision with Kinect, Processing, Arduino and MakerBot*. Beijing, O'Reilly, 2012.
- Brosterman, Norman, and Togashi. *Inventing Kindergarten*. New York, NY: Kaleidograph Design, 2014. Print.
- “Friedrich Froebel (Fröbel).” *Infed.org*, 7 Jan. 2013, infed.org/mobi/fredrich-froebel-frobel/. Accessed 14 May 2017.
- "Froebel Gifts." Wikipedia. Wikimedia Foundation, 12 Apr. 2017. Web. 16 May 2017.
- Fukuda, Tomohiro, et al. “A Dynamic Physical Model Based on a 3D Digital Model for Architectural Rapid Prototyping.” *Automation in Construction*, vol. 72, 2016, pp. 9–17., doi:10.1016/j.autcon.2016.07.002.
- Goodrich, Ryan. “Accelerometer vs. Gyroscope: What's the Difference?” *LiveScience*, Purch, 1 Oct. 2013, www.livescience.com/40103-accelerometer-vs-gyroscope.html. Accessed 14 May 2017.
- Menges, Achim, and Sean Ahlquist. *Computational Design Thinking*. Chichester: J. Wiley & Sons, 2011. Print.

“MetaDESK.” *Tangible Media Group*, tangible.media.mit.edu/project/metadesk/. Accessed 14 February 2017.

Microsoft Kinect. Digital image. Microsoft. N.p., n.d. Web.

Negroponte, Nicholas. *The Architecture Machine: toward a More Human Environment*. Cambridge, Mass, MIT Press 1970. 72.

Papert, Seymour. *Constructionism: a New Opportunity for Elementary Science Education*.

Massachusetts, Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group, 1986.

Schevers, Jan. “Why Do Architects Make Models?” www.anarchi.cc/.

"Scientific Visualization: Volume Surface Rendering." Scientific Visualization - Project 1 Report - Richie Zirbes. N.p., n.d. Web. 16 May 2017.

“Seek” *Negroponte's Seek—Gerbil's BLOCKWORLD*, cyberneticians.com/slideshow/seek2.html. Accessed 14 May 2017.

Sifteo. "Sifteo Cubes." YouTube. YouTube, 22 Nov. 2012. Web. 16 May 2017.

“SparkFun IMU Breakout - MPU-9250.” *Learn at SparkFun Electronics*, learn.sparkfun.com/tutorials/mpu-9250-hookup-guide. Accessed 14 May 2017.

Sun, Lei, et al. “Differences in Spatial Understanding between Physical and Virtual Models.” *Frontiers of Architectural Research*, vol. 3, no. 1, 2014, pp. 28–35., doi:10.1016/j.foar.2013.11.005.

Turkle, Sherry, and Seymour Papert. "Epistemological Pluralism: Styles and Voices Within the Computer Culture." *Humanistic Mathematics Network Journal* 1.7 (1992): 46-68. Web.

Ullmer, Brygg, and Hiroshi Ishii. "The MetaDESK." *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology - UIST '97*, 1997,
doi:10.1145/263407.263551.

Underkoffler, John, and Hiroshi Ishii. "Urp." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems the CHI Is the Limit - CHI '99*, 1999,
doi:10.1145/302979.303114.

Valoriani, Matteo. "Programming with Kinect V2." 3 June, 2015 Microsoft PowerPoint file.

Vardouli, Theodora. "Nicholas Negroponte: an Interview." *Open/Architectures*, 26 Oct. 2011,
openarchitectures.com/2011/10/27/an-interview-with-nicholas-negroponte/. Accessed 14
May 2017.