SiteInSight: Assigning physical computing a role in architectural ideation

Antony M. Wood

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Architecture

University of Washington
2015

Committee:
Brian R. Johnson
Robert J. Corser

Program Authorized to Offer Degree:
Architecture (Design Computing)

University of Washington

Abstract

SiteInSight: Assigning physical computing a role in architectural ideation

Antony M. Wood

Chair of the Supervisory Committee:
Associate Professor Brian R. Johnson
Architecture

The benefits of a robust architectural site-analysis are well understood through countless examples of built architecture. The tools necessary to achieve such an analysis, much like the site itself, continue to evolve. It is common to find physical computing systems – systems designed to engage the physical and environmental qualities of our world -- being employed to expose, capture, and store an architectural site's peculiarities in the form of data. However, what is less common is employing this data in real-time, on-site and within a common framework of ideation. This thesis demonstrates a relationship between physical computing and a designer that benefits from real-time data employment. It is a hardware and software platform that assigns physical computing a collaborative role during an initial architectural site visit to not only perform the common task of data collection, but to present it to the designer in a familiar way; a way that might allow for richer data interrogation resulting in a more informed architectural proposition. Ultimately, this thesis exists to demonstrate a way physical computing can be a meaningful participant in architectural ideation.

Table of Contents

INTRODUCTION

It is well understood that design changes become increasingly expensive, both in time and money, as an architectural project moves through the design process. It is also understood that design decisions made early in the design process act as drivers for subsequent decisions. If an architectural project is to be realized as a physical artifact, then arguably, the architectural site, along with client needs and practice goals, provide the necessary constraints for early design decisions (Zimmerman, 2000). In light of these realities, a system capable of revealing otherwise unseen site peculiarities should be viewed as a meaningful participant in the design process. This thesis documents the development of such a system.

According to Floyd Zimmerman, "A well-executed site analysis forms the essential foundation for a cost-effective, environmentally sensitive, and rational approach to project development." (2000) Untimely, the outcome of a "good" site-analysis allows an architectural proposition to "exploit the full … potential of a site." (Zimmerman, 2000) To better understand the site, several types of data are collected for better understanding. Information involving a site's climate, topography, soil conditions, utilities and immediate surroundings (edge conditions) are often investigated, recorded and compiled to form the bulk of any site-analysis. The tools, however, employed by architects and engineers to develop a "good" site-analysis continue to evolve. This can be attributed to an ever-increasing understanding of our physical surroundings and our changing relationship with it. One author states such a view:

> "Our understanding … is improving every day due to our ability to scan and sample the environment around us at increasing resolution using better technologies and sensors. Thus the space is revealed as a ever changing data field, and our capability to … design the space itself can be enhanced by the use of tools able to inform better the design processes with more accurate, specific and variable, in space and time, data streams." (Anonymous, 2013)

A tool that not only samples the aforementioned environmental "data-field," but also employs such data in the development of early design ideas informs the primary goal of this work. This work, culminating in the development and limited deployment of the *SiteInSite* system, is designed to work

the hardest during the initial site visit of an architectural project; a time when ideas are less-defined and more capable of change.

The *SiteInSight* system is positioned within two research spaces. The first concerns itself with the process of architectural ideation and, more specifically, the use of sketching in the early stages of ideation. In this space, current research is employed more as a blueprint for the *SiteInSight's* interface strategy rather than expanding the field's findings. The second is the growing field of research broadly defined as physical computing. Here, current knowledge is exploited and transformed to develop appropriate strategies of locating, capturing, visualizing and storing data. The goals are few, but broad: meaningful representation of site data in as-close-to-real-time as possible to the designer through an easily accessible interface that also facilitates ideation; the robust recording of usable site data necessary to maintain its value for future use in the design process. Together, these research spaces and project goals inform the development of the *SiteInSight* system; a system hoping to expand what a designer can "see" on-site while, at the same time, using what is seen in the development of ideas.

PHYSICAL COMPUTING

In the first chapter of *Experiencing Architecture*, Steen Eiler Rasmussen (1959) shares his story of casually watching five boys playing a game similar to football on the steps of the church of S. Maria Maggiore in Rome. In this game, the boys used

> "… a curved wall, which they played against with great virtuosity. When the ball was out, it was most decidedly out, bouncing down all the steps and rolling several hundred feet further on … among motor cars and Vespas down near the great obelisk." (Rasmussen, p. 18)

What makes this story remarkable is not the newly discovered game, but rather an improved way of understanding a physical space in time. Rasmussen, after watching the boys play, remarks, "[a]s I sat in the shade watching them, I sensed the whole three-dimensional composition as never before." (Rasmussen, p. 19) Rasmussen, understanding how the ball improved his understanding of the

physical space, broadens the discussion:

> "… [A] child familiarizes himself with all sorts of playthings which increase his opportunities to experience his surroundings. If he licks his finger and sticks it in the air, he discovers what the wind is like in the low strata of air in which he moves about. But with a kite he has an aerial feeler out high in the atmosphere. ..." (Rasmussen, p. 19)

Although written in 1959, this passage serves as an exceptional analogy to physical computing systems being developed and employed today for the very same reasons.

Physical computing, as it is understood in its current form, is a field of study focused on the relationship between humans, objects, environmental spaces and technology. (Corlett, 2010) By definition, it is an exploration of human interactions between our world, its contents and each other. From walls that change color when people approach to thermostats that learn occupant habits to better provide comfort, the merits of such interactions in the design world are just now being uncovered. Physical computing systems commonly involve a hardware/software platform employing sensor technology capable of quantifying natural phenomena through transduction. Until very recently, however, those educated and trained in design had few opportunities to practice in a field that included computerized systems.

The field of physical computing, until recently, has been dominated by engineers working in research centers capable, both in knowledge and funding, of designing meaningful interactive computing systems. According to University of Philadelphia professor Tod Corlett, "The technology was expensive, highly-trained people were needed to use it and sharing information was awkward. These factors combined to limit [its] development." (p. 2) This began to change in 2001 when MIT media lab members, Ben Fry and Casey Reas, developed the graphic-oriented programming language *Processing*. What was revolutionary, according to Corlett, was not the language, nor its free and open-source distribution model, but rather *Processing's* ability to be altered and improved by its users. "It became possible, and even easy, to learn programming by example and by modifying and combining parts, rather than by creating new code from first principles." (p. 3) Although this became an inroad

to learning software development, it lacked the necessary "physical" apparatus necessary to interact with the real world. (Corlett, 2000)

In 2005, the Arduino Integrated Development Environment (IDE) and hardware platform was born from four instructors at the Interaction Design Institute in Ivrea, Italy. Stemming from Humberto Barragan's adaptation of *Processing*, called *Wiring*, the *Arduino* platform was "[a] philosophical break from all the physical-prototyping hardware that preceded it." (Corlett, p.6) For the first time, a platform had been developed aimed at designers rather than engineers, "with maximum simplicity and convenience at minimum cost." (Corlett, p.6)

Similar to the development of *Processing*, *Arduino*-based systems today tend to be developed based on previous users' work posted to forums populated with interested and involved users. The *SiteInSight* system, as will be detailed later in this writing, employs the common method of development accurately described by Corlett:

> "the practical, modern way to work on a physical computing problem is to surf around, grab pieces of code and circuit specifications, recombine and modify them, iterate and then post your own results so the process can continue." (p. 6)

The development of the *SiteInSight* system exemplifies this process and offers evidence for its viability. However, writing code that captures data through an *Arduino*-based platform provides only part of the answer. The goal of the *SiteInSite* system, again, is how to make the system a meaningful participant in architectural ideation. Simply stated: how does a physical computing system assist a designer in exposing and developing ideas?

IDEAS AND IDEATION

An idea can be understood as a basic element of thought. (Jonson, 2006) However, the process of developing these basic elements into a complementary system of ideas can be understood as ideation. It is well documented that design professionals often employ "unstructured forms of pictorial representations" – known as the conceptual sketch – as a tool of ideation. (Purcell, Gero, 2006)

Although researchers lament the lack of empirical evidence supporting conceptual sketching as a means of creative discovery, most agree the problem lies with developing adequate research methodologies to capture such data. (Lawson, 2004) In 1998, A.T. Purcell and J.S. Gero cataloged in "Drawings and the Design Process" current research at that time surrounding the use of the conceptual sketch in early stages of design. Consistent in all research reviewed was sketching's affordance of re-interpretation. Simply stated: "… sketching allow[ed] for [the] emergence of new ways of seeing the perceptual (drawn) representation of a potential design." (Purcell, Gero, p. 396) Their summary findings relating to the conceptual sketch remain valid today; key findings are as follows:

1. Either explicitly or implicitly, sketches are image based.
2. Sketches are dense, unstructured and ambiguous, more easily allowing for reinterpretation
3. Either during or following sketching, new knowledge becomes part of the problem solving process.
4. "[T]he examining of the new knowledge produced by reinterpretation [sketching] result in further reinterpretations and access to new knowledge; that is, there is a cyclic, dialectic process involved." (Purcell, Gero, p. 399)

Gero and Purcell's last summary finding is better understood through Gabriela Goldschmidt's diagram from 1994. Her diagram [Figure 1.1] clearly illustrates the process of emergence and re-interpretation embedded in the process of sketching. It illustrates the demand put on any tool of ideation to be interactive. In addition to their summary findings, one important realization noted by Gero and Purcell is the void in research on "what to draw." Although lacking, some researchers argue the question of "what to draw" is unique to each designer and should not exist as a prescription (Lawson, 2004)). Numerous anecdotal examples exist through archived sketches of successful designers to support this claim.

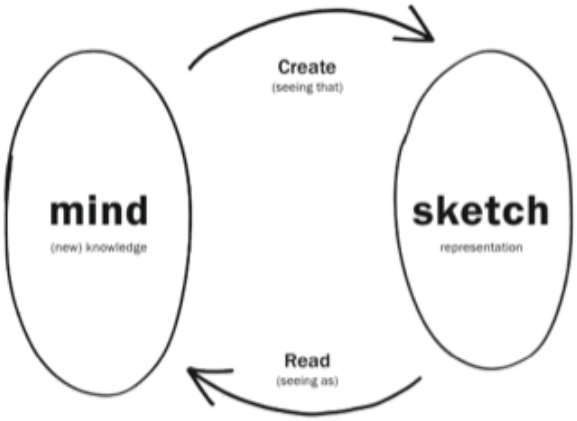


Figure 1.1

Goldschmidt's sketch, in addition to the summary findings by Gero and Purcell, raise questions regarding Rasmussen's discovery at the Basilica. However enlightened Rasmussen became after watching this game on the steps, it would be of value to learn how Rasmussen recorded this newly discovered understanding of S. Maria Maggiore. How, perhaps, would he have used this information to inform an architectural proposition situated in the church plaza if given the opportunity? Questions of this nature illustrate the difference between acquiring information and employing information in the development of an idea. An Arduino-based physical computing system capable of capturing and representing environmental data graphically to a user is commonplace. What is less common is the affordance of real-time interrogation of environmental data in the spirit of Gabriella Goldschmidt's sketching diagram.

Enter the *SiteInSight* system.

SITEINSIGHT DEVELOPMENT GOALS

SiteInSight, fundamentally, exists to demonstrate a physical computing system's ability to make a visible mark capable of altering the trajectory of an idea. Early efforts in locating such a mark proved difficult as attempts quickly turned into predictable patterning -- marks incapable of facilitating emergence and reinterpretation. A more robust data type was needed to be useful in respect to ideation.

A simple question restarted the system's stalled development: What is the view from the sixth floor? With this question, image data was selected as the type to be captured, geo-located, displayed, interrogated, stored and retrieved with the use of an on-site physical computing system. The human-computer interface would be a client-server system viewed through a web browser, allowing for a 'common' and well-understood interface. And, the system needed to be capable of viewing the site in three-dimensions. Simply stated: the system needed to fly. The participating designer would ultimately determine the success or failure of the *SiteInSite* system and its ability to meaningfully participate in process of ideation. Although exact moments of design "change" based on information provided through the *SiteInSight* system may not be recorded, an overall design experience that resembles a designer-directed collaboration would be the ultimate test of success. Failure would be a system that provided visualizations that effected no change and served only to complicate the designer's ability to ideate. However, if the *SiteInSight* system allowed a designer to understand and react meaningfully to site conditions previously unknown or less understood, the system should be viewed as a success.

DEVELOPMENT CHALLENGES

For one with limited education and experience with computerized systems involving both electronic hardware and software, imagining what is possible with a physical computing system greatly outweighs the realities of what can actually be built. Even with a platform as barrier-free as the

Arduino, solutions closely resembling imagined ideas remained the greatest challenge in developing the *SiteInSite* system. An endless process of trial-and-error, alongside the searching and reading of countless forum posts containing perhaps a sliver of needed information paved the way to the eventual deployment of the system. Again, it is by no means the realization of the ideal physical computing system imagined at the start, but rather a collection of code and strategy borrowed from countless others, peppered with what might be understood as original work. The following outlines in more detail the hardware/software employed, followed by challenges encountered.

SYSTEM HARDWARE COMPONENTS

Platform

As mentioned earlier, the *Arduino* platform was chosen for the development of the *SiteInSight* system. More specifically, the microcontroller serving as the electronic backbone was the *Arduino YUN*. The choice of the *Yun* was two-fold. First, the YUN had built-board WiFi capability necessary for the streaming of image and location data. Second, the YUN contained a Linux Stack for on-board computing. This allowed for a local web-server to be built complete with an installation of MySQL Server for data recording.

Image Data Capture

In addition to the Arduino Yun, a Logitech USB web cam was chosen to stream and capture image data. Although the picture quality is not as good as some other cameras, it was necessary to use a compatible USB webcam to interface with the application installed on the Linux side of the Arduino YUN. Popular cameras like the Hero, or other digital cameras with better image quality proved incompatible with the installed Linux software.

Location Data

Two strategies were explored to capture an image's associated location data. Ultimately, one was chosen for deployment test. The two sensors were the Adafruit Ultimate GPS and the Adafruit

10DOF board, the latter containing an accelerometer, gyroscope and altimeter. Both sensors were capable of providing much desired elevation data; however, the 10DOF sensor was chosen over the GPS for two reasons. First, it was impossible to test the GPS module indoors due to the lack "fixity" with orbiting satellites. Second, the 10DOF offered the ability to capture the camera's view-vector (compass orientation), a necessary data point if the image were to be of value beyond the architectural site.

Communications Network

The system, as it was imagined, included a browser-based interface capable of piloting the system and displaying sampled data. To achieve this goal, a WiFi capable iPad Mini was chosen to communicate with the 'flying' system components. Because of its availability onboard the Arduino YUN, a Wi-Fi network was employed. An Airport Extreme was chosen (non-flying) to provide the necessary Wi-Fi signal.

Power

Three *Radio Shack Portable Power Banks* (Output: 5V, 1A). Two banks were used to power each Arduino YUN, while the remaining bank powered the on-board servos mentioned below.

Camera Control

Two servos were used to control a custom pan-tilt mechanism allowing the user to control the direction of the camera. This servo-control would be accomplished through a browser-based interface communicating with the Arduino sketch.

Flight Vehicle

A 5.5' diameter *CloudBuster* weather balloon using 80cubic feet of helium and a net lift of 4.2lbs was chosen to move the camera and YUN's 'flying' components through various elevations. This allows the user to see things at desired elevations. The balloon was purchased as part of a system from PublicLab.org, an organization that employs kites, masts and balloons for the purpose of producing aerial maps. The organization has thousands of hours of documented balloon time and freely shares

the results of their work. This proved invaluable as the flight portion of the system came late in development.

Fuselage

A custom housing was designed to hold the hardware components, with the exception of the Airport Extreme. In determining the appropriate enclosure, the combined weight of the flying components remained a great concern. The entire system weighed 2.98 lbs upon completion. The largest contributors to the overall weight were the three batteries. The fuselage was constructed of foam-core and laser cut for accuracy. Quarter-inch threaded rods and a variety of nylon nuts, and washers were chosen for their lightweight characteristics. All hardware was purchased at local hardware stores and electronic shops. See Appendix A for a detailed parts list.

Browser-based Interface

As mentioned earlier, the interface uses HTML, JavaScript and associated client/sever based languages to visualize in real-time all image and location data. An iPad Mini running Google's Chrome browser is used to display the interface. See Appendix B for development diagram.

CHALLENGES: Hardware

The *Arduino Yun*, at the time of development, was the newest microcontroller board to be released by Arduino. The bricolage approach to development mentioned earlier was not as effective as it might have been had the *SiteInSite* system employed an older board, such as the *Arduino UNO*, long in use by the design community. And although the *Yun* was based on the established *Arduino Leonardo* board, the *Yun* community lacked examples demonstrating the Yun's published capabilities. Of the many challenges specific to the board, communicating with the on-board Linux stack proved difficult when more than one "line" of communication was necessary.

A second hardware challenge existed in the limitations of *MicroSD* memory cards. Current knowledge regarding this type of memory suggests the cards has an average limit of 3000 writes

(http://superuser.com/questions/17350/whats-the-life-expectancy-of-an-sd-card). Simply stated: the card will eventually become corrupted if data is continuously written to it. As the only means of data storage available on this platform, this presented a fundamental system challenge with regards to data collection and storage. The solution: only write data to the *MicroSD* card that was deemed valuable by the designer during the site visit. All other means of transmitting data to the interface would have to bypass the *MicroSD* card completely.

A third challenge surrounded the use of image capture technology in addition to geo-location data within an *Arduino* sketch. As of this writing, no code solution allowing both location data and image data to be captured using a single *Arduino Yun* sketch was discovered. The solution, therefore, was to employ a separate *Arduino Yun* for each data stream, adding both cost and weight to a system not-yet proven.

CHALLENGES: Software

The major software challenge was, of course, authoring the software. Software used included an installation of MySQL on the Linux stack using PHP and HTML as the conduit. Standard browser-based code was employed to create interactions between the designer, the Arduino YUN, and captured data via a browser-based interface. JavaScript and AJAX (Asynchronous JavaScript And XML) did the major lifting with respect to interaction. CSS was used to style the control interface; HTML5 and the *canvas* tag were used to develop the sketch interface. Again, using a bricolage approach allowed software not-yet-fully-understood to be employed with some degree of success. As the project advanced, so did the understanding of the software. Eventually, the line between "original" and "borrowed" code slowly faded, resulting in more intentional software and a greater understanding of the relationships between each of them.

In addition to the development of 'working' code, it was imperative the interface remained viable for control and data interaction. Designing an interface that essentially 'gets out of the way',

while at the same time providing the necessary tools proved challenging. The *SiteInSight* system's interface needed to be more than a visualized data-dump, it required a thoughtful design that would allow the data to become a meaningful mark.

CHALLENGES: Flight Vehicle

The primary challenge with the 'flying' of the *SiteInSight* system was the development of a system that required limited piloting, allowing the designer, instead, to focus on the data being presented and the development of ideas while on-site. Also, in the spirit of Rasmussen, the system needed to afford the opportunity for the designer to linger on-site without adding additional distractions (noise). This required the system to be 'on' and 'in the background' for extended periods of time. Kites, masts and tethered helium-filled balloons were considered as potential aerial vehicles; the balloon was eventually chosen for deployment. The reasons are described below.

First, the balloon can remain airborne as long as the helium remains viable. Second, The balloon requires little to any directed attention once properly positioned on-site, and finally, the balloon can achieve greater controlled altitudes than a mast or kite. The one drawback to a balloon, however, is the cost of helium. The 5.5 diameter *CloudBuster* balloon chosen requires between $50-$80 per flight depending on payload (the difference in cost is dependent on how 'full' the balloon is with helium).

Additional challenges facing the flying system were weight and battery-life. These two challenges were intertwined as flying more 'power' increased the overall weight of the system. In the end, three battery packs were included, providing over 3hrs of system operation. This amount of flight time, it can be argued, affords the designer ample opportunity to linger, which, arguably, increases the chance for a better understanding of the architectural site.

SiteInSight DEPLOYMENT

The *SiteInSight* system's maiden flight took place on June 2, 2015 at 5:22pm in the atrium of Gould Hall, University of Washington. The atrium is roughly 4.5 stories tall and protected from the environment (winds) allowing for more idealized flight conditions during the initial flight. The goals of this first flight were (in no particular order):

1. Verify a satisfactory Wi-Fi connected network between the ground station, the *Arduino YUN* microcontrollers, and the iPad Mini at increasing altitudes.
2. Verify that the *CloudBuster* balloon was capable of lifting the *SiteInSight* system to differing altitudes.
3. Verify that the servos could move the camera as desired by the designer at different altitudes
4. Verify that the system was capable of streaming video, capturing desired images and allowing the designer to mark the images as necessary to develop their ideas.
5. Identify tethering tactics for the *CloudBuster* Balloon. (At the time of the first flight, the tethering system was still in the design phase.)

RESULTS

The maiden test flight of the *SiteInSight* system demonstrated, by all accounts, that a physical computing system capable of streaming, capturing, geo-locating and interrogating image data at any desired location on-site is possible. The *CloudBuster* balloon easily lifted the *SiteInSight* system to varying elevations. (See Appendix A & B) The Wi-Fi connection remained viable for the duration of the flight and at the maximum achieved elevation of 56' (according to the 10DOF sensor.) The iPad interface performed as designed; data was viewed, captured, geo-located, and stored for future use. The sketch interface performed as designed as well. One area of concern, however, remains in the design of the tethering system. In our 'idealized' setting, the tether performed adequately, however, if exposed to less-than-ideal site conditions, the balloon might easily drift off-

site into neighboring areas reducing the systems ability to deliver data in areas desired by the designer. One potential solution, although untested, is a three-way tether with ground-based (system controlled) motorized 'winches' located neat the edges of the site. This configuration might minimize drift and improve position control over the site.

It is worth noting that the servo-controlled camera system presented an unintended visual experience. The servo system was designed to move the camera in order to 'look' in a desired direction; however, that is not exactly how it behaved. As imaginable, a flying camera presents stability problems while in flight. The fuselage's attachment to the balloon was designed (Appendix B) to limit such movement without the help of expensive and heavy gimbals. Interestingly, the live video feed expressed this instability as wavering, and resulted in an aesthetic resembling a casual walk-thru. The servo movement, instead of directing the view, forced the camera into differing degrees of waver. This type of control was unexpected but welcome.

CONCLUSION

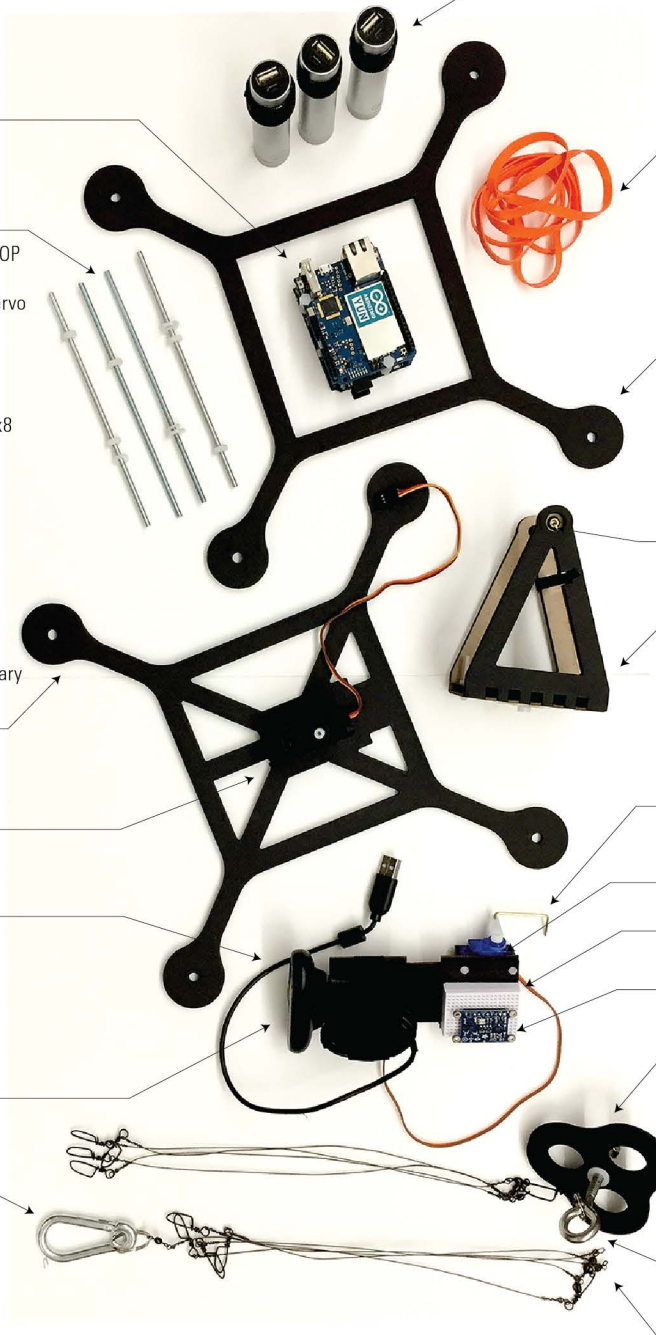
The successful development of the *SiteInSight* system provides an intellectual framework for developing a physical computing system with the potential to meaningfully participate in architectural ideation. It does so by demonstrating a method of extending a physical computing system beyond mere data collection and representation to include an interface allowing real-time data interaction. Although the desired goal of demonstrating the system's ability to alter the trajectory of an idea during an architectural site visit remains untested, valuable outcomes were still achieved:

First, this work adds to the Arduino YUN knowledge base, providing other designers examples of working code and integration with other systems. Second, the overall development process provided ample intellectual resistance necessary for individual growth. It introduced examples of logic; it demanded explicit understanding of problems; it allowed for creativity through debugging

strategies. Fundamentally, the *SiteInSight* system succeeds by offering its "working" system as evidence of the DIY/hacker process employed.

The *SiteInSight* system, imagined in the spirit of Rasmussen's experience in Rome and grounded in a solid understanding of sketching's role in architectural ideation, offers those looking to engage the data-sphere one example of how it can be done. This work also provides a strategy to allow other data types to enter the ideation process; data whose employment leads to a more informed architectural proposition.

Appendix A: Hardware Components



Radio Shack® Portable Power Banks x3 (output 5V, 1A)

Arduino® Yun© Microcontroller x2

3" Dia. Rubber Bands, x5. Provides extra support for the Pan/Tilt mechanism while turning.

6", 1/4-20 Threaded Rods Provides the mechanism to space the TOP and BOTTOM fuselage assemblies as necessary for system protection and servo movement.

1/4-20 Nylon Nuts x20 +
1/4 Nylon Washers x20 +
1" 1/4 Nylon Washers (not pictured) x8

1/4" Foam Core, black. Provides a lightweight material necessary to form the TOP section of the fuselage structure.

Brass eyelets, x2 + Thin Brass Tube

1/4" Foam Core, Black adhered to 1/8" Chipboard. Both materials are necessary to support the weight of the Camera / Sensor Platform

1/4" Foam Core, black. Provides a lightweight material necessary to form the BOTTOM section of the fuselage structure.

Radio Shack® Standard Servo. Cat. No. 2730766

Thin Brass Tube to attach to Pan/Tilt Housing.

TowerPro® MicroServo 9g/s690

Camera/Sensor Platform. 1/4" Foam Core adhered to 1/8" Chipboard for stability.

Solder-less Bread Board (small)

AdaFruit® 10DOF IMU Breakout Board

1/4" Foam Core, black, Portable Power Bank support housing.

LogiTech® USB Web Cam

Stainless Steel Carabiner

1-1/2" 1/4-20 Stainless Steel Threaded Eyelet. Serves as the connection point between the fuselage and power.

12" Stainless Steel leaders (fishing) x8

Not Pictured:

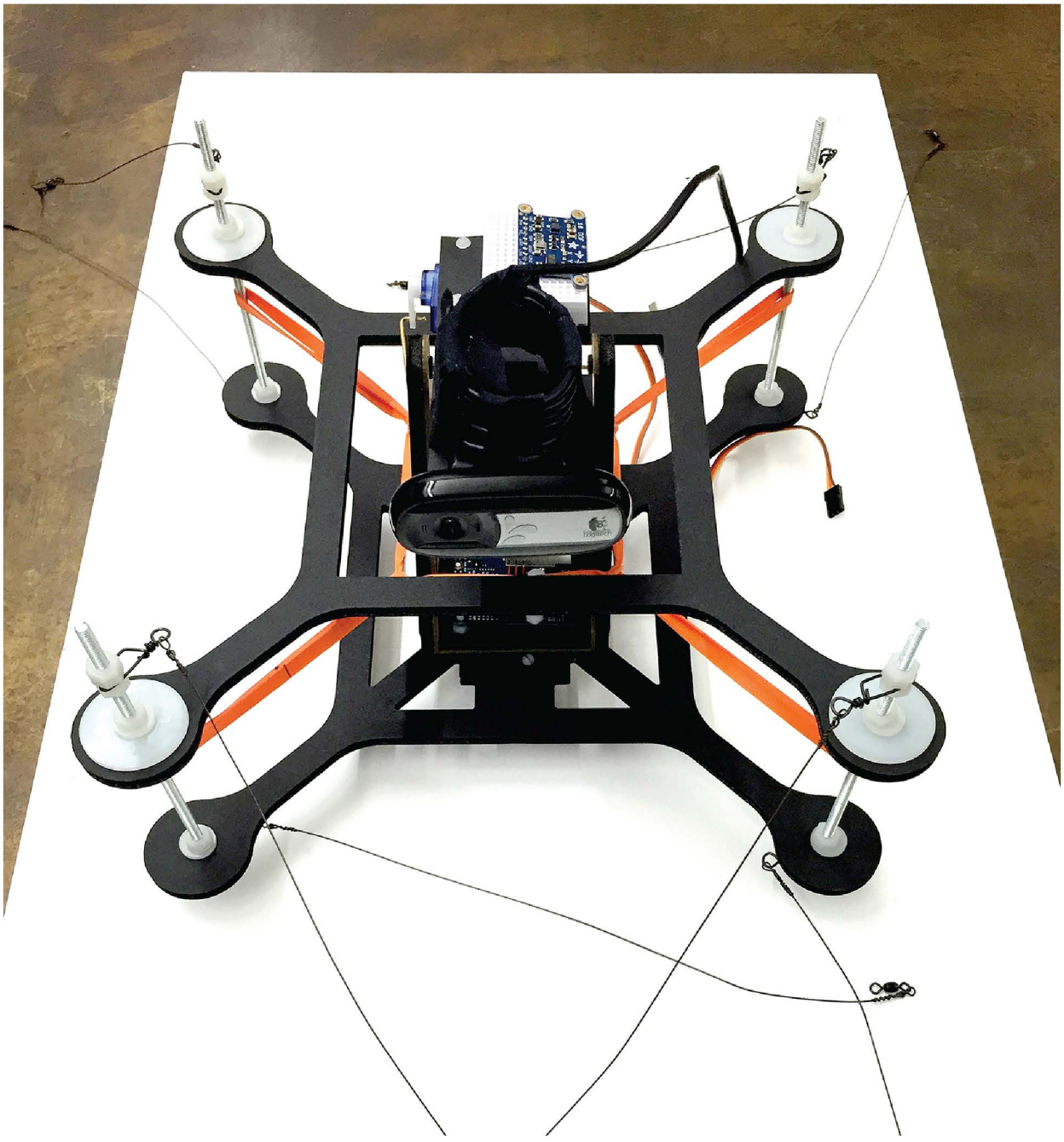
22Ga "Jumper" Wire (Cut to Length) necessary for connecting Electronic Components to Arduino Micrcontroller

5.5' Dia. CloundBuster® Weather Balloon

Apple® Airport Extreme© Wireless Router

MicroB USB cables x3. Used to connect Portable PowerBanks where needed

Appendix A: Hardware Components



Hardware Components Assembled

Appendix A: Hardware Components

Hardware Components in Operation

Appendix B: SiteInSight Interface

IMAGE VIEW AND CAPTURE CONTROL INTERFACE

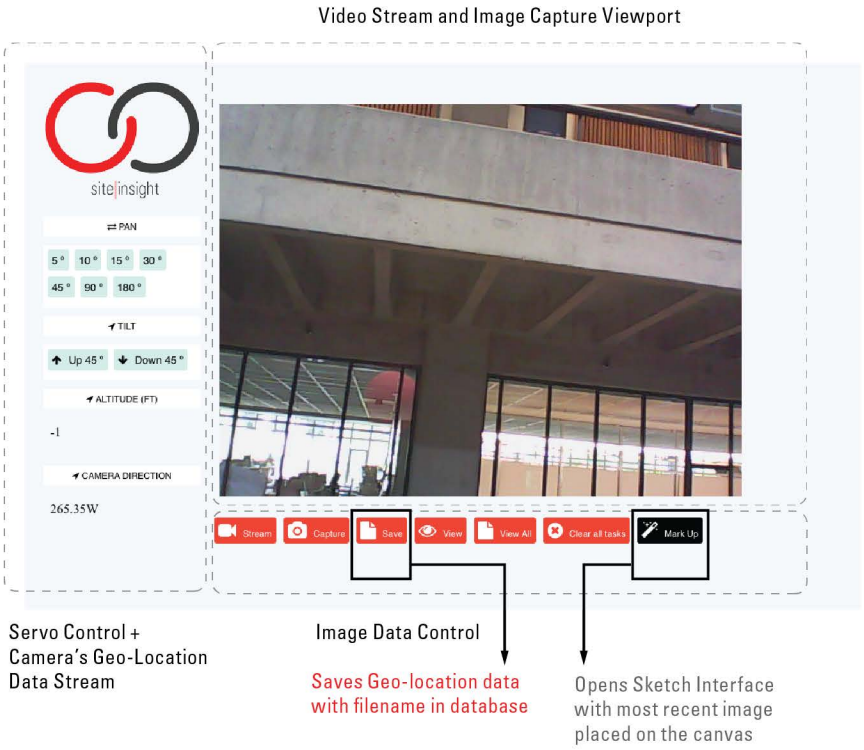
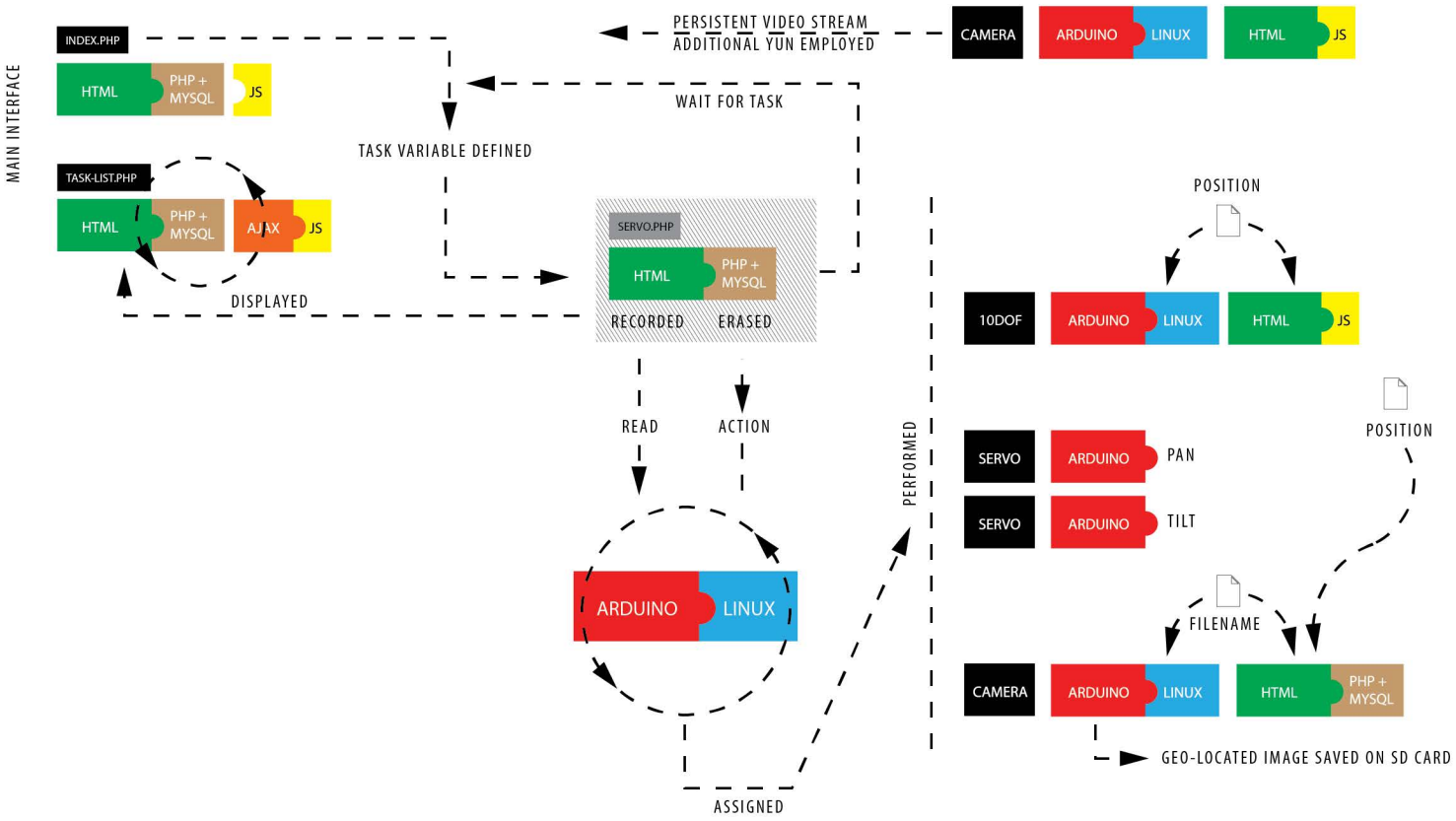
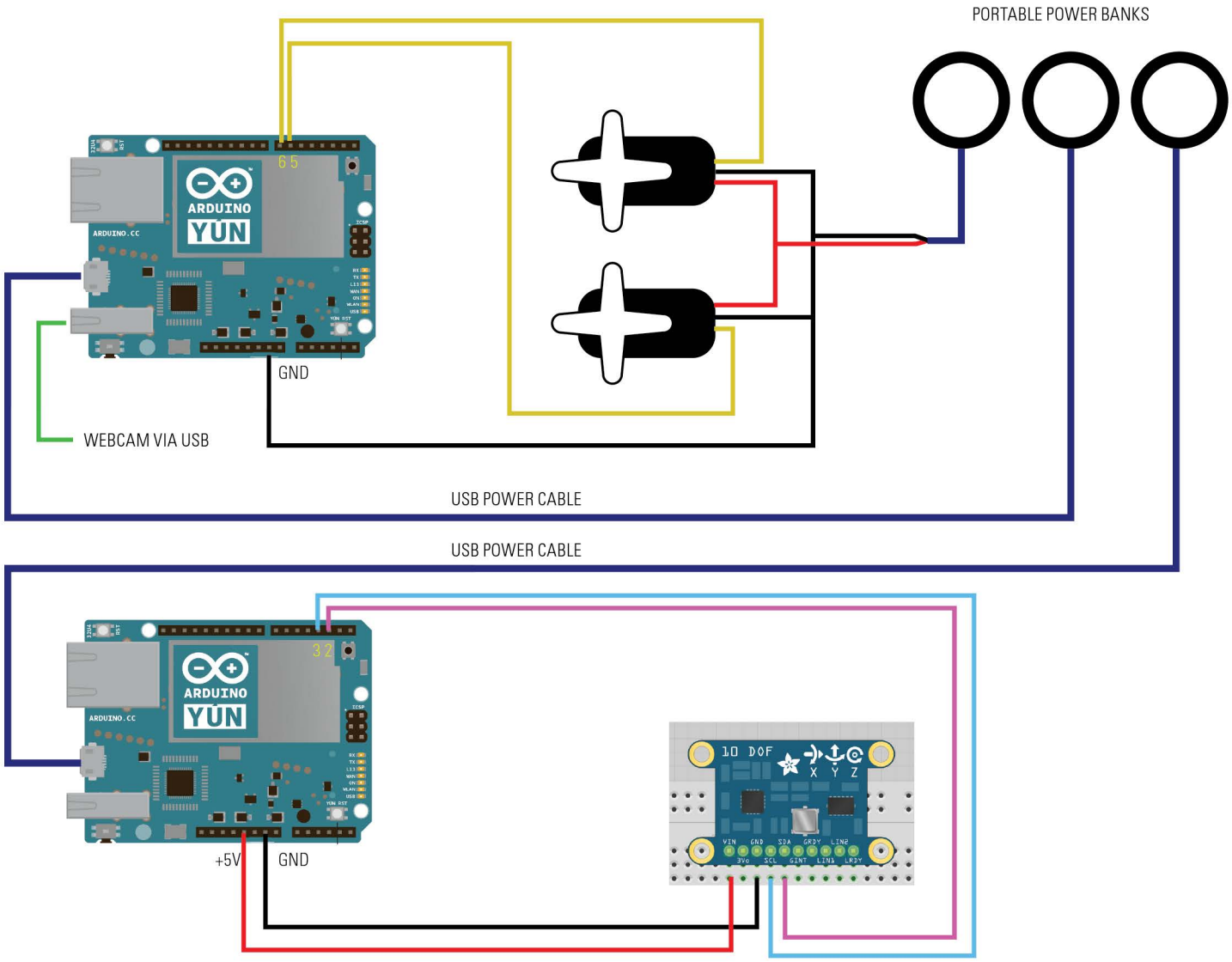


IMAGE DATA INTERROGATION INTERFACE (SKETCH)

Appendix C: SiteInSight Software Schematic



Appendix D: SiteInSight Arduino Wiring Diagram



References:

Corlett, T. (2010). Physical Computing: Joining the DIY Revolution.*IDSA*. Retrieved from http://www-old.idsa.org.

Goldschmidt, G. (April 1994) On Visual Design Thinking: The Vis Kids of Architecture. *Design Studies*, 15(2), 158-174.

Jonson, B. (2006) Design ideation: the conceptual sketch in the digital age. *Design Studies*, 26(6), 613-624

Lawson, B. (2004) Schemata, Gambits and Precedent: Some Factors in Design Expertise. *Design Studies* 25.5, 443-57.

Menezes, A., Lawson, B. (2006) How Designers Perceive Sketches. *Design Studies* 27.5, 571-85.

PérezGómez. A. (2005) Questions of representation: the poetic origin of architecture. *Architectural Research Quarterly, 9*, 217-225

Purcell, A. T., Gero, J. S. (2006) Drawings and the design process. Department of Architectural and Design Science, University of Sydney, Sydney, Australia

Rasmussen, S.E. (1964) Experiencing Architecture. The MIT Press, 2nd ed.

Suwa, M., Gero, J., Purcell, T. (1998) The Roles of Sketches in Early Conceptual Design Processes. *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, Morton Ann Gernsbacher, Sharon J. Derry (Eds.)

Tversky, B. (2002). What do sketches say about thinking? In T. Stahovic, J. Landay, and R. Davis (Eds.), *Proceedings of AAAI spring symposium on sketch understanding*. Pp. Menlo Park, CA: AAAI Press.

Zimmerman, F. (2000) Site Analysis. *The Architect's Handbook of Professional Practice*, 13th ed.