Aware Surfaces: Large-Scale, Surface-Based Sensing for New Modes of Data Collection, Analysis, and Human Interaction

by

Daniel Hayim Goodman

Bachelor of Science in Mechanical Engineering, 2013
Massachusetts Institute of Technology

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

Author Signature redacted

Daniel Goodman
Program in Media Arts and Sciences
May 18, 2015

Certified by Signature redacted

Kent Larson
Principal Research Scientist
Changing Places Group
Thesis Advisor

Accepted by Signature redacted

Pattie Maes
Academic Head
Program in Media Arts and Sciences

Aware Surfaces: Large-Scale, Surface-Based Sensing for New Modes of Data Collection, Analysis, and Human Interaction

by

Daniel Hayim Goodman

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on May 18, 2015, in partial fulfillment of the
requirements of the degree of
Master of Science in Media Arts and Sciences

Abstract:

This thesis describes the design and construction of pressure sensing matrices for capturing human location and activity data from large surfaces in a space such as the floors, walls, tabletops, countertops, and furniture. With the ability to operate either alone or connected to others in an assembly, each sensor module is 0.3m x 2m, contains 512 force sensitive resistors, and has a refresh rate of about 8Hz. Each module was made with conductive inkjet printing and PCB fabrication, creating a low-profile sensing surface with robust signal-collecting circuitry. Several experiments were conducted on an assemblage of three modules to assess parameters such as response time, sensitivity, measurement repeatability, spatial and pressure resolution, and accuracy in analyzing walking data as compared to a camera. Applications that could employ such a system are explored and two visualizations were prototyped that could ambiently provide data and trends to a user.

Thesis Supervisor: Kent Larson
Title: Principal Research Scientist, Changing Places Group, MIT Media Lab

Aware Surfaces: Large-Scale, Surface-Based Sensing for New Modes of Data Collection, Analysis, and Human Interaction

by

Daniel Hayim Goodman

The following individuals served as readers for this thesis:

Signature redacted

Advisor____

Kent Larson
Principal Research Scientist
Changing Places Group

Reader__________ Signature redacted ________

Chris Schmandt
Principal Research Scientist
Living Mobile Group

Reader_________ Signature redacted ____________

Joseph Paradiso
Associate Professor of Media Arts and Sciences
Responsive Environments Group

Acknowledgments:

To my advisor, Kent Larson. Thank you for your mentorship, input, and advice over the last two years. I feel that my time in the Changing Places Group has allowed me to grow as both a designer and engineer.

To my readers, Chris Schmandt and Joe Paradiso. Thank you for your time and suggestions during my project design and thesis writing process.

To my UROPs, Maria Fabre, Sasha Levy, Shruthi Narayanan, and Izzy Gomez. Thank you for all your hard work this semester. I'm extremely happy with the final outcome of the project and I have all of you to thank for making it happen.

To my parents, Marilyn and David Goodman**.** Thank you for everything. I love you both very much and I am grateful for the guidance, support and occasional kick in the ass you have given and continue to give me.

To my brothers, Aaron and Jonah. I am proud of both of you and look forward to seeing all the interesting things you will do in the coming years. I love you guys.

To my colleagues, professors, and friends at the Media Lab. Thank you for helping me along the way. Thank you for the impromptu design help, enjoyable and thought provoking conversations, and inspiring projects.

Table of Contents:

0. Overview 21

1. Introduction and Context 23

2. Connected Devices 25
- **2.1** Intro and Overview 25
- **2.2** CityHome 25
  - **2.2.1** Transformation
  - **2.2.2** Smart Elements
  - **2.2.3** UI/UX
  - **2.2.4** App Ecosystem and Store
- **2.3** CityFarm 32
  - **2.3.1** Aeroponics
  - **2.3.2** Hydroponics
  - **2.3.3** Sensing Modules
  - **2.3.4** Data Recipes
  - **2.3.5** Potential
- **2.4** Athena Breast Pump 36
  - **2.4.1** Prototype System
  - **2.4.2** Hardware and Sensing
  - **2.4.3** Affordances of a Human-Centered Pump Interface

3. Research Motivations 39
- **3.1** Overview 39
- **3.2** Aging In Place 39
- **3.3** Cognitive Decline 41
  - **3.3.1** Activities of Daily Living
  - **3.3.2** Reporting ADLs
  - **3.3.3** Using Surface-Based Sensing to Monitor ADLs
- **3.4** Fall Detection 42
  - **3.4.1** Using Surface-Based Sensing to Monitor Fall Likelihood
- **3.5** A Solution Involves Multiple Devices Working Together 43
- **3.6** Use of Cameras For Geriatric Monitoring 44
  - **3.6.1** Monitoring of Individuals With Dementia
  - **3.6.2** Identify Abuse or Neglect
  - **3.6.3** Use of Surface-Based Sensing in lieu of Cameras
- **3.7** Closing Remarks on Aging in Place and Age-Related Diseases 46
- **3.8** The Quantified Environment - Not "Wearables" but "Liveables" 47
- **3.9** Data-Driven Transformation of Space 48
- **3.10** Person-to-Person Communication and Connection 48
- **3.11** Games, Exercise, Education and Other Applications 49

4. Previous Work on Surface-Based Sensing 50
4.1 Overview 50
4.2 Physical Surface-Based Sensing in Academia 50
4.2.1 Magic Carpet
4.2.2 Z-Tiles
4.2.3 Conductive Inkjet Sensors
4.3 Physical Surface-Based Sensing in Industry 55
4.3.1 Tekscan
4.3.2 Elsi Technologies
4.3.3 Tactonic
4.4 Hobbyist Physical Surface-Based Sensing 58
4.5 Digital Surface-Based Sensing in Academia 60
4.5.1 IR Multitouch Table
4.5.2 GravitySpace
4.5.3 LuminAR

5. Development of Pressure Sensor Matrices 63
5.1 Overview 63
5.2 The Force Sensitive Resistor (FSR) 63
5.2.1 Calibration of FSRs
5.2.2 Circuitry to Drive FSRs
5.3 Final Project for Human 2.0 (Human Augmentation) Course 65
5.4 Designs Leading to The Member's Week 2014 Prototype 67
5.4.1 Overview
5.4.2 First Modular Design
5.4.3 Single Sided Sensor Matrix
5.4.4 Eliminating Ghosting
5.4.5 Single Sided Prototype With Diodes
5.4.6 Larger Sensors and Different Fabrication Processes
5.4.7 Spray Painting Sensors
5.5 Member's Week 2014 Prototype (MW2014) 71
5.5.1 Overview
5.5.2 Circuitry of MW2014
5.5.3 Hardware - Sensor Sheets
5.5.4 FSR Construction
5.5.5 Reading the Sensor Values
5.5.6 Visualizations
5.6 Member's Week 2014 Demo 78
5.6.1 Modes
5.6.2 Closing Remarks

6. Thesis Prototype 84
6.1 Overview 84
6.2 Technical Overview of the Conductive Inkjet Printing Process 84
6.3 Conductive Inkjet Printing Experiments 85
6.3.1 First Set of Inkjet Prints
6.3.2 Characterizing the Resistance
6.3.3 Assembling the Square Grid Sensor
6.3.4 Assembling the Rectangular Grid Sensor
6.4 Final Prototype Hardware and Software 92
6.4.1 Triangular Inkjet Printed Sensors
6.4.2 Slave Module PCB
6.4.3 Arduino-to-Arduino Communication With I^2C
6.4.4 Communication with an Assembly of Slave Modules
6.4.5 Sensor Sheet Assembly
6.4.6 Masking Experiments

7. Visualizations of Sensor Data 101
7.1 Overview 101
7.2 Previous Work 101
7.2.1 Ambient Orb
7.2.2 Digital Family Portraits
7.3 "Digital" Clock 103
7.3.1 Family Clock Interface
7.3.2 Personal Clock Interface
7.4 Potted Plant 106

8. Experiments and Verification 108
8.1 Overview 108
8.2 Sensor Characterization 108
8.2.1 Single Sensor
8.2.2 One Cell of Two Sensors
8.2.3 Two Cells Containing Four Sensors
8.2.4 Discussion
8.3 Shape Characterization 116
8.3.1 Shapes
8.3.2 Force Distribution
8.3.3 Experimental Setup
8.3.4 Experimental Procedure
8.3.5 Triangular and Square Pixel Output Visualizations
8.3.6 Miscellaneous Activities
8.3.7 Results and Analysis
8.4 Walking Tests 124
8.4.1 Gait

8.4.2 Experimental Preparation and Setup
8.4.3 Experimental Procedure for General Walking Tests
8.4.4 Experimental Procedure for Gait Abnormality Tests
8.4.5 Analysis and Discussion of General Walking Tests
8.4.6 Analysis and Discussion of Gait Abnormality Tests
8.4.7 Closing Remarks

9. Future Work ____ 144
9.1 Wireless Connectivity ____ 144
9.1.1 XBee
9.1.2 ZigBee
9.1.3 Using XBee With the Thesis Prototype
9.2 Alternative Sensor Configurations ____ 146
9.2.1 Different Sensor Densities in the Same Assembly
9.3 User Testing with Multiple Surfaces ____ 148
9.4 Everything Else ____ 148

10. Conclusion ____ 150

References ____ 152

Appendix A: Arduino Code ____ 159
Appendix B: Processing Code ____ 165
Appendix C: MATLAB Code ____ 172
Appendix D: Lawton Instrumental Activities of Daily Living Scale ____ 181
Appendix E: Tekscan Walkway Gait Analysis System ____ 183
Appendix F: Masking Tests ____ 185
Appendix G: Shape Characterization Test Images ____ 187
Appendix H: Slave PCB Bill of Materials ____ 193
Appendix I: Slave PCB Schematic Detail ____ 194

List of Figures:

1. Introduction and Context 23
Figure 1a: FitBit Surge fitness watch
Figure 1b: Nest learning thermostat
2. Connected Devices 25
Figure 2: CityHome prototype
Figure 3a: Sleep mode
Figure 3b: Work mode
Figure 3c: Dining mode
Figure 3d: Lounging mode
Figure 3e: Bathroom mode
Figure 3f: Kitchen mode
Figure 4: FSR pad to actuate the CityHome wall module
Figure 5a: Using voice control to trigger party mode
Figure 5b: Party mode
Figure 6a: Pointing to engage the lights
Figure 6b: Commanding the desk to move
Figure 6c: Adjusting the hue of the lights
Figure 6d: Deploying the bed
Figure 6e: Adjusting the hue of the lights
Figure 6f: Locking in a specific light color
Figure 6g: Newspaper projected onto the desk
Figure 6h: Media remote projected on the wall
Figure 7: Sample downloadable app
Figure 8: CityFarm hydroponic/aeroponic research lab
Figure 9: Air sensing modules
Figure 10: Water sensing module
Figure 11: Athena Breast Pump prototype
Figure 12: Athena Breast Pump electronics
Figure 13: Personalized pumping interface for a tablet
4. Previous Work on Surface-Based Sensing 50
Figure 14a: Magic Carpet set up in the Media Lab
Figure 14b: Schematic drawing of the Magic Carpet system
Figure 15a: Assemblage of Z-Tiles
Figure 15b: Individual Z-Tile
Figure 16: Z-Tile data transfer schematic
Figure 17: Bounding areas of high pressure with ellipses
Figure 18a: Surface composed of multiple tiles
Figure 18b: Close up of PCB used to read the tiles
Figure 18c: Different components of a single tile
Figure 19: Sheet folded to accommodate different surface geometries

Figure 20: User with a shoe tag stepping on the surface
Figure 21a: Tekscan Walkway system in use
Figure 21b: Output from Walkway system sensor readings
Figure 22a: Elsi system installed under laminate flooring
Figure 22b: System sensing an elderly individual walking on the floor
Figure 23a: Output from Tactonic floor tiles
Figure 23b: Capturing a user's foot pressure profile
Figure 24: Pressure sensing pads underneath the exhibit flooring
Figure 25: PCBs for the pressure sensors
Figure 26: IR multitouch surface
Figure 27: Sample outputs from different IR table configurations
Figure 28: GravitySpace prototype
Figure 29a: Schematic of GravitySpace
Figure 29b: GravitySpace output image
Figure 30a: LuminAR
Figure 30b: LuminAR internal view
Figure 31: Sample LuminAR interfaces
Figure 32: Shunt mode and through mode FSR configurations
5. Development of Pressure Sensor Matrices______ 63
Figure 33: Force/Resistance and Force/Capacitance curves
Figure 34: Voltage divider circuit for a single FSR
Figure 35: Row and column configuration of the pressure mat
Figure 36: Tekscan pressure sensing matrix schematic
Figure 37: Smaller pressure mat prototype
Figure 38a: Larger pressure mat prototype
Figure 38b: Control circuitry for larger pressure mat prototype
Figure 39a: Bottom surface of a 3 x 3 pressure matrix
Figure 39b: Traces masked with electrical tape
Figure 39c: Upper and lower surface unfolded
Figure 40a: Shunt mode FSR traces
Figure 40b: Squares of Velostat placed over the sensor pads
Figure 41a: One tile with diodes for ghosting mitigation
Figure 41b: Sensor output visualized in Processing
Figure 42a: Vinyl-cut traces
Figure 42b: Diodes connected to every sensor trace
Figure 43a: Laser cut masking pattern
Figure 43b: Spray painted sensor traces
Figure 43c: Folded sheet to create rows and columns
Figure 44: MW 2014 deployed in the CityHome
Figure 45a: Completed connector board with SR and MUX breakout boards
Figure 45b: MUX, connector board, and SR
Figure 45c: Front side of connector board
Figure 45d: Back side (routed) of connector board

Figure 46a: Close up of the circuit connection adaptor and sheet traces
Figure 46b: View of one sensing strip
Figure 47: One sensing strip with pads at each FSR location
Figure 48a: View of the six-strip prototype assembly
Figure 48b: View of the electronics connected to the strips
Figure 49: View of the carpeted prototype in the CityHome living room
Figure 50: Raw data and threshold visualization
Figure 51: Path plot visualization
Figure 52: Heat map visualization
Figure 53: Projector overlay on the carpeted prototype
Figure 54: Member's Week 2014 demo space
Figure 55: Icons representing each mode the system could run
Figure 56: Raw data mode
Figure 57: Light mode
Figure 58: Music mode
Figure 59: Heat map mode
Figure 60: Orbit mode
Figure 61: Path plot mode
6. Thesis Prototype____84
Figure 62: Conductive inkjet printed circuitry
Figure 63a: Different print patterns and settings
Figure 63b: Close up of printed pattern with interlocking fingers
Figure 63c: JST FFC connector with 0.1" pitch pins
Figure 63d: Off-the-shelf Interlink square FSR
Figure 64: Close up of the traces and sensor pads for the first prototype
Figure 65a: Entire inkjet printed sheet with Square and Rectangular Grid
Figure 65b: Close up of the Square and Rectangular Grid patterns
Figure 65c: Square Grid
Figure 65d: Rectangular Grid
Figure 66a: Square and Rectangular Grid sheets cut from the main sheet
Figure 66b: Tabs cut in Square Grid sheet and FFC connectors placed
Figure 66c: Tabs cut in Rectangular Grid sheet and FFC connectors placed
Figure 67: Square Grid sheet plugged into MW2014 electronics
Figure 68: Square Grid sheet with paper masking
Figure 69a: Square grid sensor connected to the computer
Figure 69b: Sensor activated and the output visualized on the computer
Figure 69c: Testing two hands on the sensor
Figure 70a: Rectangular Grid sheet
Figure 70b: Large Velostat sheet
Figure 70c: Half of the mask applied to the sheet
Figure 70d: Complete mask applied
Figure 70e: Velostat attached along the edge of the sheet
Figure 71: Laminated inkjet printed sensor and Velostat

Figure 72: One cell of two triangular sensors
Figure 73: Sample section of triangular sensors being activated by a foot
Figure 74a: Close up of triangular sensor configuration
Figure 74b: Alternate view of the inkjet printed triangular sensors
Figure 75: Overlapping sensor sheets
Figure 76: Long and short ribbon cable section of the printed sensor traces
Figure 77: Schematic for a single slave module PCB
Figure 78a: A rendering of the final three-layer board
Figure 78b: Fabricated and assembled board
Figure 79: Arduino ISP board
Figure 80a: View of inkjet printed sheet upon arrival
Figure 80b: Close up of guide border ribbon for the cable section
Figure 81: Cut and flipped ribbon cable section for the lower sensing sheet
Figure 82a: Bottom inkjet sheet plugged into the PCB
Figure 82b: Top and bottom inkjet sheets plugged into the PCB
Figure 82c: Masking layer applied to the top and bottom inkjet sheets
Figure 82d: Assembled sensor sheet architecture
Figure 83a: View of three slave module assemblies connected together
Figure 83b: View of PCBs and master Arduino
Figure 83c: Additional view of the PCB/sensor sheet interface
Figure 84a: Primed and spray painted sensor sheet
Figure 84b: Sensor sheet run through an office inkjet printer
7. Visualizations of Sensor Data____101
Figure 85a: Ambient orb
Figure 85b: Orb changing color in response to price of electricity
Figure 86a: Digital family portrait of a Grandmother
Figure 86b: Digital family portrait of a Granddaughter
Figure 87a: Physical clock display
Figure 87b: Side view of clock
Figure 88a: Default clock face
Figure 88b: Default clock face on the physical display
Figure 88c: Inspiration for the default clock face design
Figure 89a: Data visualization showing one active room
Figure 89b: Visualization of the active room and previously active rooms
Figure 89c: Data visualization on the physical display
Figure 90a: Data visualization for the individual
Figure 90b: Personal visualization on the physical display
Figure 91a: Grass short indicating low activity
Figure 91b: Grass tall indicating high activity
Figure 91c: Random colors to demonstrate highlighting the grass with color
Figure 92: Seven planter boxes representing the seven days of the week
Figure 93a: Exploded view of the Potted Plant system
Figure 93b: Potted Plant system with artificial grass placed

Figure 93c: Top view of the internals of the system
8. Experiments and Verification ________ 108
Figure 94: Diagram of two cells of sensors
Figure 95a: Location of force application
Figure 95b: LEGO® force applicator
Figure 95c: Water container and applicator loading a single sensor
Figure 96: Static loading tests for a single sensor 250g
Figure 97: Applied loading tests for a single sensor 250g
Figure 98: Unloading tests for a single sensor 250g
Figure 99a: LEGO® force applicator for one cell
Figure 99b: 7.47kg block of steel
Figure 100: Static loading tests for a single cell with 7.47kg
Figure 101: Unloading tests for a single cell with 7.47kg
Figure 102a: LEGO® force applicator for two cells
Figure 102b: Steel block applying weight to the cells
Figure 102c: Other angle of the steel block applying weight
Figure 103: Static loading tests for two cells with 7.47kg
Figure 104: Force/Resistance and Force/Capacitance curves
Figure 105: Sensor actuated by a weight
Figure 106: Shapes used in the characterization experiments
Figure 107a: Arched foot profile
Figure 107b: Flat foot profile
Figure 107c: Circle profile
Figure 107d: Triangle profile
Figure 107e: Square profile
Figure 108a: Sensor mat system placed on the table
Figure 108b: Foam sheet covering the system
Figure 109a: One water jug weighing about 42lbs
Figure 109b: Two water jugs weighing about 84lbs
Figure 110a: Toe out or "duck-footed" orientation
Figure 110b: Toe-in or "pidgeon-toed" orientation
Figure 110c: Parallel feet orientation
Figure 111: Square orientation for 0° and 45°
Figure 112: Triangle orientations for 0°, 45°, and 90°
Figure 113a: Circle object with triangular pixel visualization
Figure 113b: Same object with the square pixel visualization
Figure 114a: Person lying down
Figure 114b: Person sitting upright on their hands
Figure 114c: Person sitting crosslegged
Figure 114d: Person performing a push up
Figure 114e: Office chair
Figure 114f: Person sitting in office chair
Figure 115: Outputs of different shape tests

Figure 116: Man and woman standing side by side
Figure 117: Two representations of the human gait cycle
Figure 118: Pressure output and gait cycle analysis by Tekscan Walkway
Figure 119a: Slave modules taped in place with wood plank protecting PCB
Figure 119b: Close up of the tape
Figure 119c: Close up of the gaffer's tape/foam layer
Figure 119d: View of the walking setup
Figure 120a: Walking experiment setup
Figure 120b: The two camera angles that are recorded
Figure 121: Reference directions and dimensions for the camera experiment
Figure 122: Diagram of clustering scheme
Figure 123: Calculating theta from the new centroid and bounding box
Figure 124: MATLAB output showing location of each footstep
Figure 125a: Subject wearing a constrained knee brace
Figure 125b: Weight taped to the subject's right foot
Figure 125c: Subject carrying a five-gallon water jug
Figure 126: Finding the time of each step
Figure 127: Different step conditions encountered in the video analysis
Figure 128: Diagram of a set of sensors on the system
Figure 129: Location of the centroid of a right triangle
Figure 130: Centroid approximation for each cell of two sensors
Figure 131: Gap between slave modules in the mat assembly
Figure 132: Histograms for the distance and time differences for each step
Figure 133: Three separate walking trials for the same person
Figure 134: Diagram indicating how step width, length and time is calculated
9. Future Work____144
Figure 135: XBee module with wire antenna
Figure 136a: XBee connected to the master Arduino
Figure 136b: XBee connected to the computer via USB
Figure 137a: Row and column configuration for reading nine sensors
Figure 137b: One-sensor-per-pin configuration
Figure 138a: Floor plan of an apartment using multiple sensor assemblies
Figure 138b: Sparse sensor module with large sensors
Figure 138c: Dense sensor module with small sensors
Figure 138d: Example of multiple sensor sizes on a single sensor module
Figure 139: Lawton Instrumental Activities of Daily Living scale
Figure 140: Tekscan Walkway gait analysis system
Figure 141: Points for testing resistance before and after masking

List of Tables:

Table 1: Resistance vs. Length For 1mm Traces ____ 87
Table 2: Resistance vs. Length For 0.5mm Traces ____ 87
Table 3: Static loading values (1-254): One sensor with 250g, one + two cells with 7.47 kg ____ 114
Table 4: Dimensions for each of the shapes used in the characterization experiments ____ 116
Table 5: Format of MATLAB and Video data outputs ____ 135
Table 6: Mean and standard deviation for the distance and time differences for each step ____ 138
Table 7: Mean and Stdev of step length/width for the normal and abnormal gait tests ____ 141
Table 8: Mean and Stdev of step times for the normal and knee walking tests ____ 141
Table 9: Mean and Stdev of step times for the ankle and weight carrying tests ____ 142
Table 10: Average number of steps for the normal and abnormal gait tests ____ 142
Table 11: Results of resistance, slide, crinkle, rub, and scratch tests ____ 186
Table 12: Bill of Materials for Slave PCB ____ 193

0. Overview:

The Internet of Things is a powerful paradigm in current technology for its ability to connect a multitude of distinct devices and provide high-volume collection of data for optimization, quantification, and enhancement. Single devices and appliances have already found a comfortable place within this construct but the scope of "things" can be expanded even further. In particular, the largest surfaces in any given space such as the floors, walls, tabletops, countertops, and furniture have the least amount of sensing associated with them. These vast surfaces are significant capture areas that are underutilized as a location of data collection. This work investigates the use of coordinate pressure sensing to passively capture human location and activity data from these surfaces. Through several prototype iterations, a modular system of sensing "sheets" was developed to provide sensing capability to these locations. The theory behind pressure sensing as well as the design and fabrication process for each sensor prototype is provided.

In addition to the design and construction of the sensor sheets, several applications of the technology are explored to demonstrate the potential power of the system as both a standalone device and when paired with other sensing and smart devices in a connected environment. The primary application of focus is aging in place. The use of such sensors to provide valuable information about walking, cognitive status, and activity level of an elderly user is explored. Algorithms written for the sensor can capture footsteps and calculate parameters of an individual's gait such as step width, height, walking rate, and number of steps taken. In theory, this information can be used to diagnose onset of mobility disorders, anticipate likelihood of falling, and on a macro scale understand the patterns of activities of daily living and how they vary on a daily, weekly, or monthly basis. Additional features investigated are the use of the system as an input device for different applications and as a controller for different objects around a home.

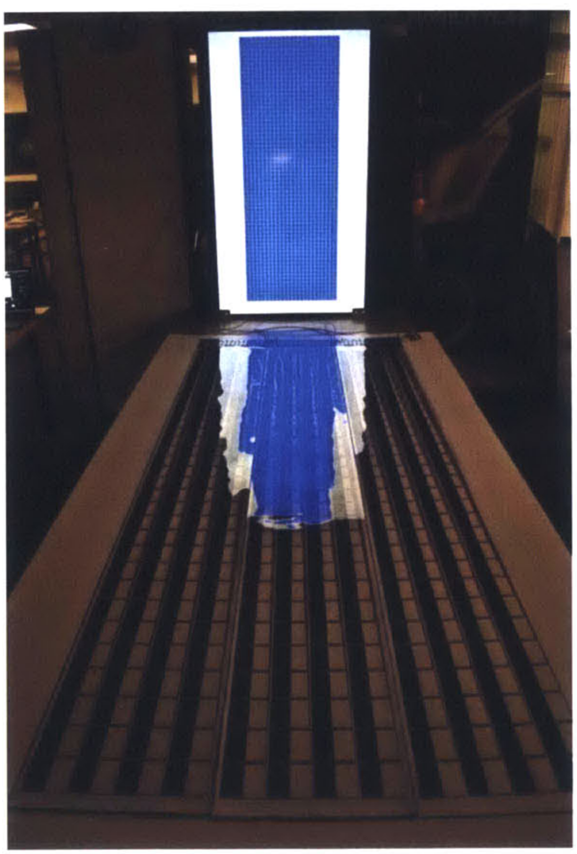
Several experiments were conducted on the system to assess both its spatial and pressure resolution as well as understand the accuracy of collecting walking data as compared to a camera:

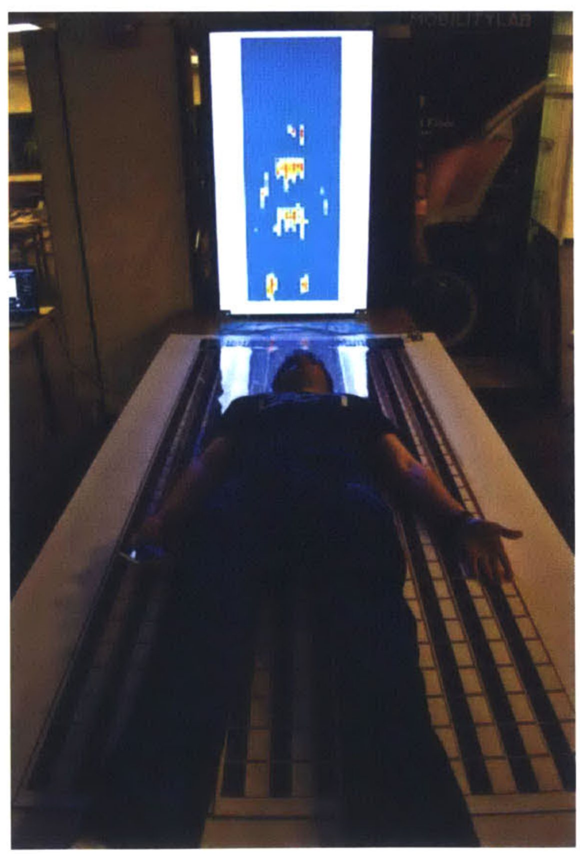
The first experiment tested the response time, sensitivity, and repeatability of individual and small groups of sensors to known loading conditions. Single sensors or groups of sensors were both statically and dynamically loaded and unloaded with known weights and the resultant outputs were plotted.

The second experiment involved placing differently shaped objects on the system to assess shape sensitivity. Each object was loaded with either forty or eighty pounds of weight to understand how the recorded output changed. The sensor data was visualized to more easily identify how differing pressures affected the recorded data.

For the final experiment, an assemblage of three sheet modules was set up in The Changing Places Group space and walking tests were then conducted on the system. Visual markers on the floor paired with webcam video recording allowed one to identify walking parameters by using frames of the video and the visual markers. In parallel, sensor data was also captured and algorithms generated foot positioning and time information. The algorithmically generated values were compared to the camera values to evaluate their agreement.

Anticipating a time when long-term, complex sensor data will need to be conveyed in a detailed, yet easy to understand way, two physical visualizations were created to ambiently provide data and trends to the user as well as their family or care providers (for the case of aging in place). The first visualization takes the form of a potted plant with actuated grass and LED lighting that can be used to give daily indications of activity level and variation from the norm for specific physiological variables. The second visualization takes the form of a digital clock, which normally shows the time, but when viewed up close, the display switches to a real-time visualization of location and activity level for the individual being monitored. Both devices require little mental computation to understand the outputs, and both are meant to fit easily into any living or working environments. Secondary applications and corresponding visualizations discussed but not developed are the use of these sensors to control elements of a connected environment, including entertainment and educational systems, as well as robotic furniture and connected appliances.





1. Introduction and Context:

Humanity is currently experiencing a period of unprecedented connectivity. With the advent of the Internet, people all around the world had the ability to communicate, interact, and work with each other almost instantaneously. Now there is a new paradigm in connectivity and it belongs to physical devices; The Internet of Things or the "concept of a world full of connected devices controlled through a consumer-friendly hub" [38]. With inexpensive manufacturing and a myriad of intelligent chips and components, it is easier than ever to connect a device to the Internet or any number of mobile phones, tablets, or computers and control it remotely. Objects ranging from lights to thermostats to gardens are becoming sensing and feedback enabled, allowing previously unavailable control over their function and use.

Having been present at the 2015 Consumer Electronics Show, the comment by the New York Times that CES was "energized by a wave of new exhibitors chasing a top trend in consumer electronics: the Internet of Things" [38] was certainly true. Most people are probably more familiar with devices such as the Nest Learning Thermostat and the Phillips Hue Light bulb for revolutionizing two very old and virtually unchanged staples of the home. But at CES, devices ranging from pet trackers, automated door locks, smart clothing, and Wi-Fi connected ceiling fans were just some of the new products that were featured. The current trend is to provide wireless connectivity to everything. Startups are not the only groups innovating in the space; large companies such as Apple and Samsung are also betting on a future filled with connected devices.

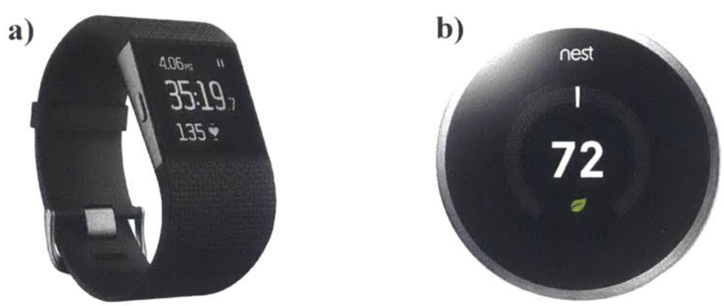


Figure 1: a) FitBit Surge fitness watch. b) Nest Learning Thermostat.
Image Sources: a) FitBit, b) Nest

Another class of pervasive devices was also featured prominently at CES; wearables and activity trackers. Spawning from the quantified self-movement, wireless heart rate monitors, breathing monitors, sleep monitors, and step counters have provided tech savvy individuals with a means to collect and visualize their personal data [39]. Companies like FitBit and Jawbone have designed elegant fitness trackers that capture multiple points of

sensing from activity, to heart rate, to step count. This data is sent to an app and beautifully visualized for the user.
Based on personal research as well as research of the Changing Places Group, it can be seen that the eventual culmination of Internet of Things and wearables will be seamlessly blending into the environment. Devices that quantify will no longer be tethered to the individual but rather embedded in the environments the individual interacts with the most; from the home to the workplace, to even the city itself.

Healthcare and diagnosis of disease will benefit from this future because every person will have a rich set of behavioral and physiological data associated with them. Those with chronic diseases that require vigilant monitoring, elderly individuals who want to live longer in their homes unassisted, people with disabilities who need new ways of controlling devices and interacting with their home will all stand to benefit from pervasive sensing and control.

For this thesis, a system was developed that could provide coordinate and pressure awareness to many of the commonly used surfaces within a home and be able to adapt to the different conditions of the space and requirements of the user. This hardware platform could not only serve the healthcare sector but any other industry who's goal is to understand and capitalize on human behavior and data. From entertainment to security to fitness, different developers could use the same hardware to create different interactions and applications for a connected environment. Several years from now, environments will be app-based. Devices in a space will communicate with each other like components in a computer. All the sensors, screens, and peripherals will operate under the same communications umbrella. It is at this point that developers can take all of the smart objects in a space and put them to work. Make them work together to create new and meaningful applications that are more personal and more specifically cater to the individual because they are informed by the very data of that individual. As argued in this thesis, sensing capabilities on the largest surfaces in any given space, such as the floors, walls, tabletops, can be used to passively capture human location and activity data as well as actively control elements within a connected environment. For this paper, a focus is given to surfaces and surface-based sensing, but the expectation is that such a system will be just one of the many components of the future connected home or environment.

2. Connected Devices:

2.1 Intro and Overview:

When sensing and networking capabilities are given to historically unconnected devices or products, an entire world of new possibilities is opened. Optimization, feedback and control become standard features, and with those abilities, the ways users operate or interact with these devices can be enhanced. Three projects developed in the Media Lab that demonstrate the power of giving intelligence to historically unintelligent objects are the CityHome, CityFarm, and Athena Breast Pump. For each project the existing mode was redesigned and prototyped to incorporate sensing, feedback, and interactive capabilities. The CityHome was a prototype of a small-scale intelligent home. The CityFarm is a sensing and connected hydroponic and aeroponic farming laboratory. And the Athena Breast Pump was a software-defined breast pump with a digital interface and data feedback capabilities.

2.2 CityHome:

Figure 2: CityHome prototype.

The CityHome project was a prototype solution to the problem of high cost of living in the major metropolitan centers of the world such as San Francisco, Tokyo or New York. As more people move into these cities, the limited real estate available becomes prohibitively expensive and forces many to relocate. Because only certain demographics are able to afford the high prices, gentrification begins to occur and much of the original cultural diversity of that area disappears. Our group's hypothesis is that innovation flourishes if more people are able to live and work in the city centers rather than commuting to work and leaving at the end of the day.

New real estate development is an obvious solution to the space issue, but it is certainly not a cost effective and sustainable approach. Enough space to accommodate the living requirements is already there; it just isn't being utilized efficiently. One popular approach is the "micro apartment" where small, few hundred square foot apartments are being offered in lieu of the larger more costly flats. While more space efficient, tenants are now faced with the problem of trying to fit a normal-sized apartment's worth of furniture and furnishings into a space a third or quarter of the size. Multi-use or "transformable" furniture pieces such as the Murphy bed are one way to optimize space usage in a micro-apartment but almost all are purely mechanical and require manual operation to move between modes.

The CityHome project began with a single question "how can you live on a 200 ft^2 footprint?" The solution has two parts, robotic transformation and digital interaction. Rather than having mechanical furniture pieces that must be manually actuated, what happens when those same pieces become robotic and able to move themselves? For the CityHome project, we designed a single wall module, which contained powered furniture including a bed and a desk that could deploy or collapse on demand. The wall itself was also able to roll across the space. Robotic furniture meant that the space was able to reconfigure itself without any human intervention (though that option was still designed into the system). Using the same floor area, one could summon a living room, office, bedroom, dining room, or bathroom. Rather than having a room for each, the available area in the apartment becomes multimodal. On top of the robotic transformation, a suite of satellite embedded devices was incorporated including lights, a projector, and blinds, which were addressable with voice or gesture commands. A Kinect sensor running off of a central Windows PC received the voice and gesture commands. These sensors and commands anticipated the likely future of the "app based home" where features and functionality can be downloaded into your home like apps on a cell phone.

2.2.1 Transformation:

The lower portion of the wall module contains the bed and a desk as well as the robotic systems needed to deploy and collapse them. The upper portion of the wall contains modular shelving which allows customization of the design to fit different apartment spaces. The home is able to transform into six primary modes. Sleeping, working, dining, lounging, bathroom, and kitchen.

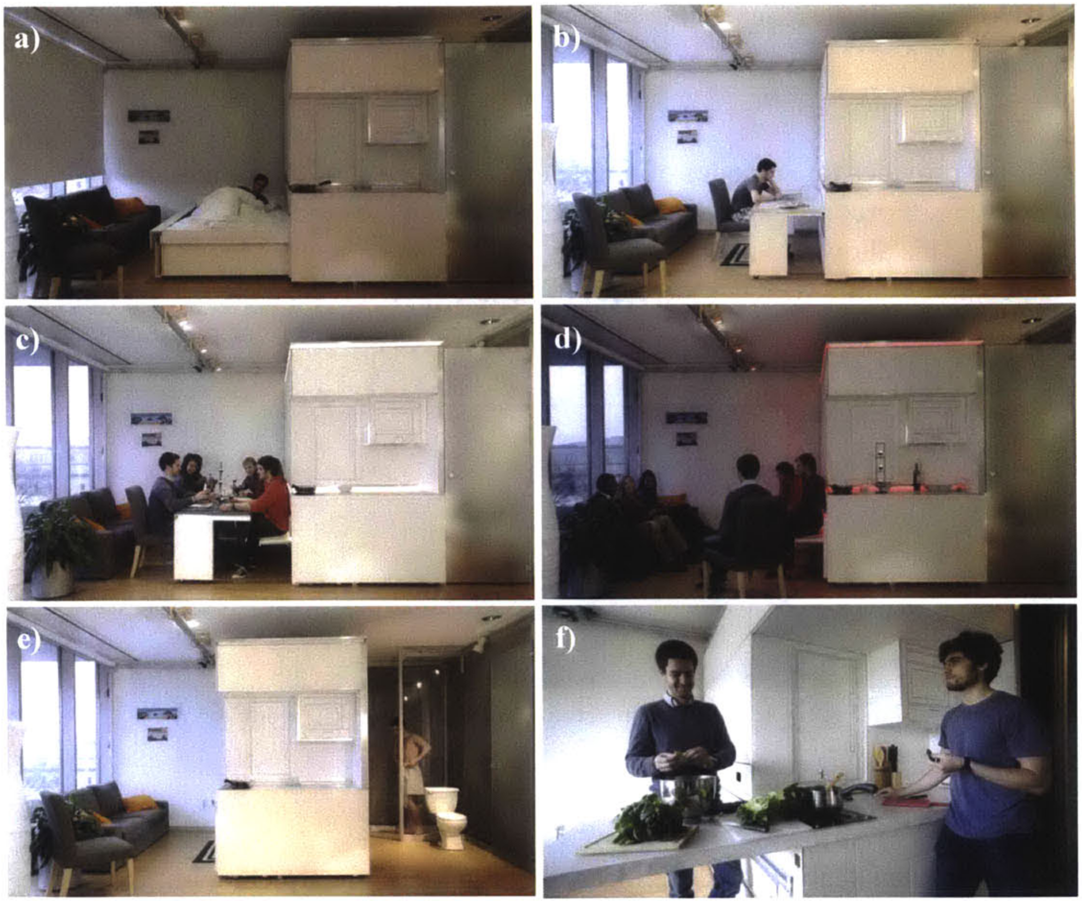


Figure 3: a) Sleep mode. b) Work mode. c) Dining mode. d) Lounging mode. e) Bathroom mode. f) Kitchen mode.

2.2.2 Smart Elements:

The intelligent elements of the CityHome, representing possible Internet connected devices were all powered by Arduino microcontrollers [61]. From controlling motors, to positioning servos, to switching LEDs the Arduino microcontroller provided an easy-to-use yet powerful tool for connecting these disparate devices under a single operational protocol. To further unite the devices, all were connected to a single PC running Windows 7. This PC served as the brain of the CityHome and anticipated a future where homes are equipped with their own computation. Having a Windows PC enabled the easy use of a Kinect sensor to handle gesture and voice commands. A C# script combining Kinect functions and calls to the Arduino allowed the team to program features and modes incorporating multiple devices.

2.2.3 UI/UX:

Touch:

The CityHome provides three modes of interaction and control of the system, touch, voice and gesture. Force sensitive resistors embedded into the wall, desk and bed enabled touch. This allowed a user to manually "move" the system. Motors provided drive force commensurate with the force of the push. Forcefully pushing the wall or pulling out the bed made the action occur faster, while a delicate push or pull made the system slowly deploy.

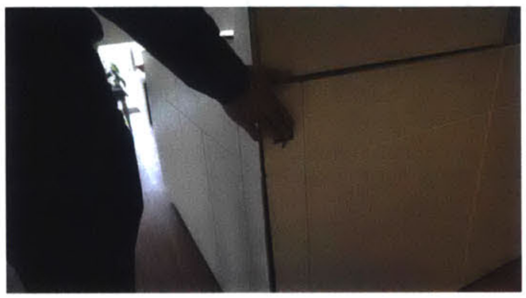


Figure 4: FSR pad to actuate the CityHome wall module.

Voice:

Voice commands were enabled by the Kinect's built in speech interpreting library. Though we had to create a dictionary of predetermined phrases, we were able to demonstrate a breadth of possible commands incorporating different aspects of life. To prevent the system from listening to every conversation and accidentally moving elements, an inclusion of "CityHome" at the beginning of the command ensures the system pays attention. Saying "CityHome, sleep mode" caused the blinds to come down, the lights to dim, and the bed to deploy from the wall. Likewise, saying "CityHome, wake up" after getting out of bed caused the blinds to rise, the lights to turn on, and the bed to collapse back into the wall, freeing the space in the living room. Many other modes were programmed into the system including ones that displayed the weather on the wall, projected a newspaper onto the desk, and even a party mode, which flashed the lights and played a party soundtrack using the Spotify API.

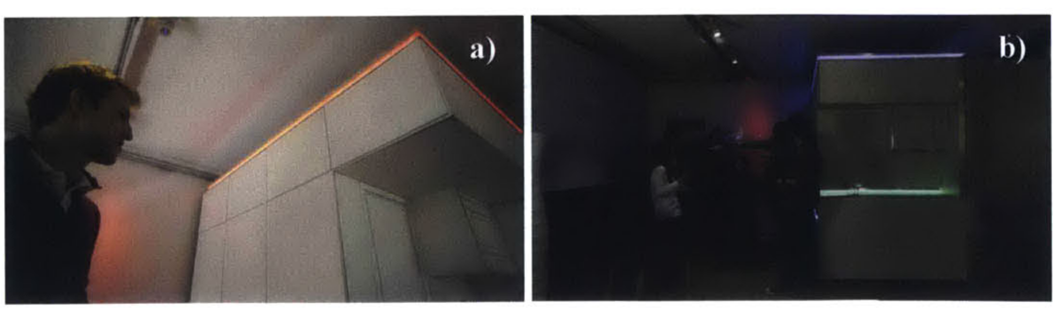


Figure 5: a) Using voice control to trigger party mode. b) Party mode.

Gesture:

Like voice, the gesture commands were enabled by the Kinect's built-in skeleton tracking capabilities. To enter gesture mode, a user had to stand in the living room facing the wall and raise his/her right hand and hold it in place. This was done to ensure gestures would not be interpreted from general activity within the space. Once the hand is raised for two seconds, the built in wall lights turn red and flash twice to indicate that gesture mode has been entered. The lights then switch to white and pulse from dim to bright continuously to indicate that the system is waiting for a command. If the user points at a controllable element, it then begins to respond to the gesture commands given.

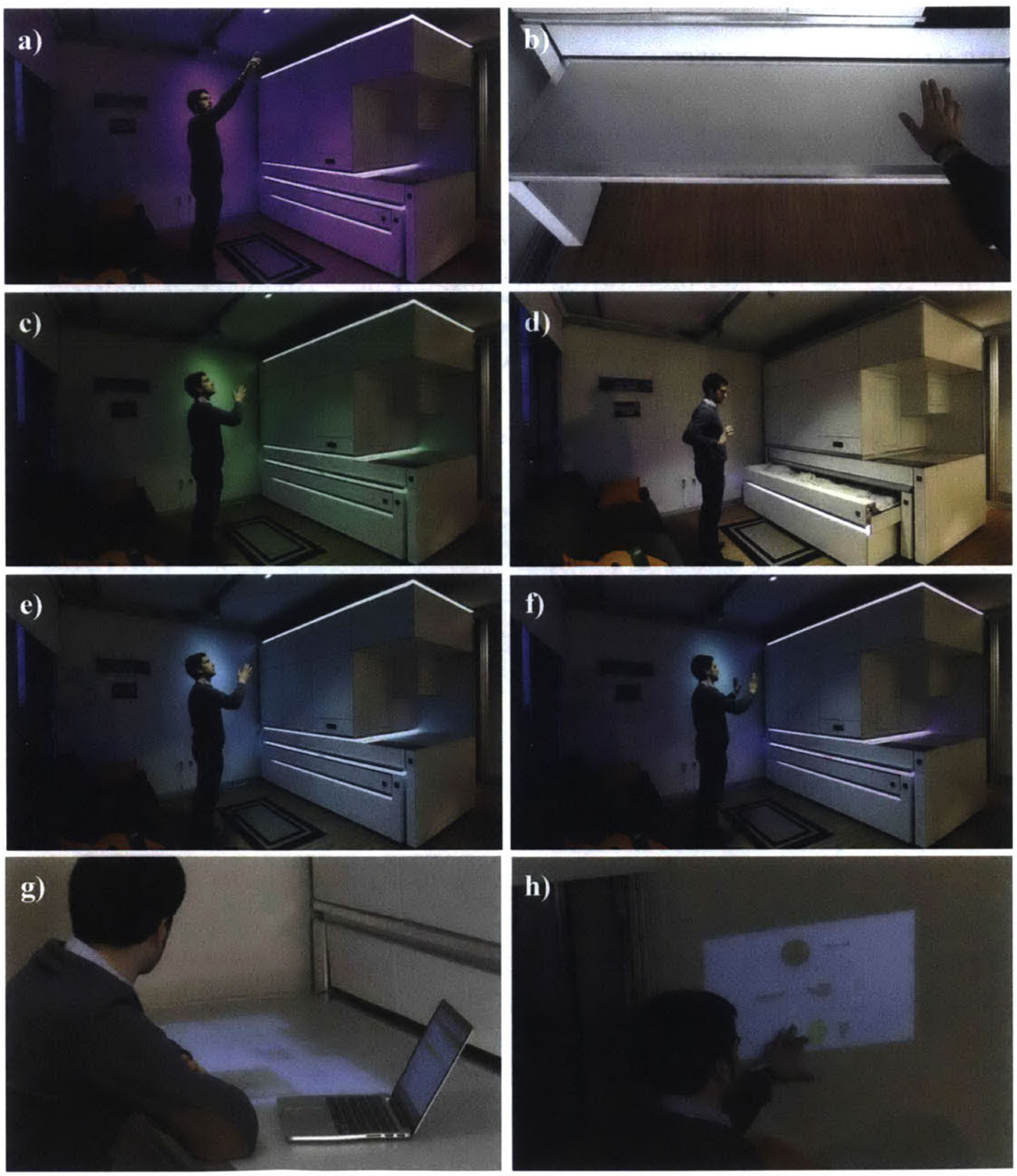


Figure 6: a) Pointing to engage the lights. b) Commanding the desk to move. c) Adjusting the hue of the lights. d) Deploying the bed. e) Adjusting hue of the lights. f) Locking in a specific light color. g) Newspaper projected onto the desk. h) Media remote projected on the wall.

Pointing at either the bed or table and then holding their hand either close or away from their chest will allow a user to deploy or collapse one of the furniture elements from the wall. Pointing at the lights and moving their hand side to side or up and down will allow the user to scroll through different colors and brightness levels respectively. Pointing at the blinds and either raising or lowering their extended arm allows the user to raise or lower the height of the blinds. Finally, pointing at the pico-projector and then pointing at the desired surface (apartment wall, wall module, desk, or near the couch) will cause the projector to orient itself at the surface and project a custom screen on the area. Near the couch, it is a TV remote. On the apartment wall, the weather is displayed. On the desk, the system scrolls through a New York Times article, and on the main wall module, the App Store is displayed.

2.2.4 App Ecosystem and Store:

From descriptions of the various CityHome features, one can begin to see the potential for meaningful and useful applications that can be downloaded directly into the home and make use of the various smart elements incorporated into the architecture. Currently there are many Internet connected devices each operating alone and communicating over different protocols. Each device probably has its own iPhone or Android app, which allows a user to control that device specifically. In designing and building the CityHome we wanted to understand what the interactions may be like if all devices were communicating with each other and all run under the same operational umbrella. In doing so, the devices are no longer standalone, but rather pieces of the greater connected ecosystem.

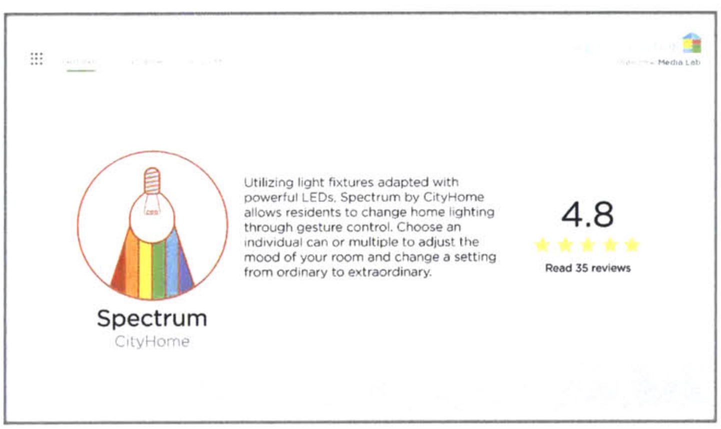


Figure 7: Sample downloadable app that uses the existing hardware of the home.

A developer can then use one or many of these devices to design new and meaningful applications that can be downloaded directly into the system. For example, rather than buying an alarm clock, a developer can design an alarm clock app, which incorporates the robotic bed, lights, and music built into the CityHome. When the user needs to wake up, the bed gently shakes back and forth, the lights gradually get brighter, and a selected song begins playing and gradually gets louder.

As more devices are added to the system, the opportunity for new applications is expanded. Companies can focus on designing devices and apps rather than designing an ecosystem to accompany the former. That is the power of the single protocol device ecosystem.

This aspect of the CityHome inspired the bulk of the investigation described in this thesis. Of particular interest was understanding what the possibilities were if one were able to use the connected devices of the home as the basis for applications. How can data collected about the user, from their decisions, patterns of activity, and preferences be then fed back into the system for meaningful and interesting applications?

If all the devices and sensors are working together, sensor fusion can be used to paint a more complete picture of the space. Different sensors monitoring different aspects of the environment i.e. thermostat monitoring temperature, smoke detectors monitoring gasses, and automatic lighting monitoring ambient light levels can all contribute data to give a comprehensive environmental snapshot. Multiple sensors working together can also provide missing or complimentary information to inform a system decision as well as corroborate observations and conclusions drawn by the system.

2.3 CityFarm:

The CityFarm is a set of projects which aim to address the problem of sustainable food production and food security. The current system of food production and food distribution is not a long term solution; vast areas of land are devoted to growing single types of crops, and forests are cut down to make more room to grow. Water, pesticides, and fertilizer are used in large quantities and are often oversupplied to the plants to account for losses into the soil. Once the food is grown it needs to be picked, packaged, and shipped by land air and sea to their final destination, often outside the country of their origin. Fuel is used for transportation and refrigeration and still crops are lost in transit. As more people move into the cities and the population increases in general, this method of growing will no longer be able to provide the food necessary. Even now, many of the crops lack nutritional value and clearing more land for more farming wont solve the problem of providing nutritious food to a large population, all while conserving vital resources such as fuel and water. The CityFarm group believes that the future of growing is not in the soil but rather soilless, with aeroponic and hydroponic agricultural methods. These techniques use significantly less resources, create faster grow cycles, and generate plants with better flavor and higher nutritional content. These farming technologies can then be deployed right in the urban centers to provide food, jobs, and education for the people of those communities.

Figure 8: CityFarm hydroponic/aeroponic research lab. Image Source: Wired UK

Even though these technologies are able to provide a sustainable alternative to conventional agriculture, we feel that the US is not quite ready to accept it. Most people associate these growing techniques with the production of Marijuana, which itself is a water-hungry plant. Countries like the Netherlands and Japan already use these techniques for food and crop production and as a culture they understand the benefit of using these techniques. To encourage more acceptance we aim to develop open source growing, control, and containment technologies for the personal level, research laboratory level, and industrial production level. Designing standard kits of parts, sensing systems, and control software will enable more people to build their own systems and make improvements, upgrades or changes. This information can then be shared as hardware components or software updates.

2.3.1 Aeroponics:

Originally developed for use in space, aeroponic growing involves spraying a fine mist of water mixed with fertilizer at the roots of growing plants. These types of growing systems can reduce water usage by 98%, fertilizer usage by 60%, and pesticide usage by 100% [37]. Crops grown this way (as verified in our own experiments) grow faster, taste better, and have a higher nutritional content.

2.3.2 Hydroponics:

Like aeroponic growing methods, hydroponic growing removes the need for soil and instead submerges the roots of the plants in a water and fertilizer mixture. Faster plant growth is associated with this method as well as reduced water and fertilizer consumption [37].

If fluorescent or LED lighting provides the light source, these types of growing systems can be placed anywhere there is room. There is no need for sun visibility. The only requirements then become access to water, electricity, fertilizer, and HVAC.

The only final factor to consider is the environment around the growing plants. If the air and water qualities of the grow environment can also be controlled, plants can be grown anywhere around the world, regardless of local environmental conditions. We have designed sensor modules and containment environments that isolate the growing plants to ensure that everything about their growth is controlled. The sensors provide nearly constant readings of the local and global air and water parameters and the data can be accessed via a web interface. With this connectivity, any variation in the data will trigger an alert so that the user could then manually fix the problem or have the system self-correct.

2.3.3 Sensing Modules:

Rather than having sensors monitor the general grow environment, we wanted to create modules that provide sensing on a plant scale. If a set of plants is grown in a certain container, that container should have its own air and water sensing, so that the current health and status of every plant can be monitored and controlled. To achieve this goal, two modules of sensing were created, air and water. The air sensor modules sat amongst the plants on top of the growing container and the water sensors were placed in waterproof boxes with probes sitting inside the container's reservoir.

Air Sensors:

Figure 9: Air sensing modules. Same sensors but different form factor.

Using mainly Seeed Studio sensors, Grove connectors and a Grove Arduino shield, we monitor temperature, humidity, light intensity, carbon dioxide, oxygen, nitrous oxide, carbon monoxide, and dust [77]. The data is passed from the Arduino to a Raspberry Pi via USB serial and then to the server via Ethernet for visualization on the website. The enclosure was designed to be low profile and to allow air circulation while deterring possible splashing or moisture contact with the sensors. The design was laser cut and held together with nuts and bolts rather than adhesive in case repairs, sensor swapping or additions or other modifications were needed.

Water Sensors:

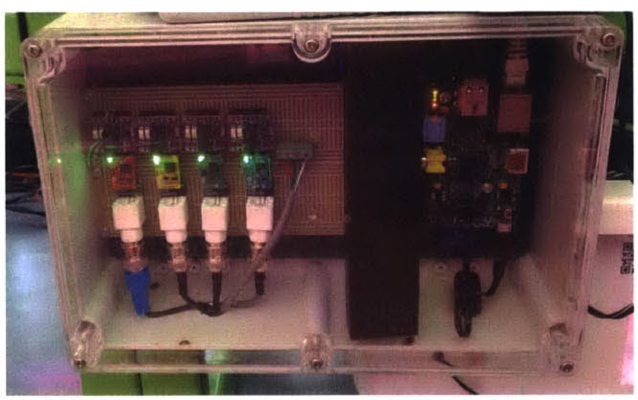


Figure 10: Water sensing module. The sensors are lit up and the Raspberry Pi receiving the data is to the right.

The water sensing was done with Atlas Scientific's water sensing boards and probes [82]. We measure pH, Oxidation Reduction Potential (ORP), Electrical Conductivity (EC), and Dissolved Oxygen (DO). These sensors report to either an Arduino or Raspberry Pi and then get visualized on a web interface.

2.3.4 Data Recipes:

Using a combination of sensing and environmental control such as heating and air venting, one is able to set specific values and let the system adjust itself to those levels. Currently this feedback control only works with air temperature, but we soon hope to have humidity, and fertilizer levels controllable as well. With this control, one person is able to mimic the growing conditions of another. For example, a user has figured out how to grow a specific, exotic plant. They have set the temperatures, light levels and schedule, fertilizer amounts etc. and it has yielded a successful plant. The system can store all of those settings and generate a "data recipe" that someone else with a growing system can then take and download. Their machine then sets the environmental and water parameters to those specified in the recipe. Then with only seeds to begin, the other user can replicate the first user's success at growing that crop without having to go through all the trouble of finding out what works.

2.3.5 Potential:

Using sensing, feedback, and an easy to understand and access UI, we hope non-farmers are able to take these technologies and be able to grow crops at a personal as well as large scale level. The data collected from each grow cycle can inform future growth and get contributed to a centralized repository which will determine the best parameters for each crop people try to grow. Recipe sharing is possible because of the ability to control the grow environment. Suddenly an entire group of people who had no knowledge of farming techniques or agriculture can grow their own food, just through data sharing and sensing.

2.4 Athena Breast Pump:

The Athena Breast Pump is a software driven, responsive breast pump, which can intelligently modify pumping patterns and is able to both optimize lactation as well as reduce user stress via a novel interface to pumping.

Figure 11: Completed breast pump showing the white cylindrical enclosure and collection flange and bottle from an off-the-shelf pump.

The current gold-standard breast pumps virtually all use a process-centric interface, which, through knobs or buttons, allows the user to control the pumping strength and frequency. Most nursing mothers we have talked to dislike the interface and often have to fidget with the settings until something suitable is reached. The stress of discovering these settings, on top of the stress associated with pumping in general creates a negative experience. We explored possible mitigation of these issues by developing a novel pump software-control mechanism that enables the use of a complimentary user-centered (rather than process-centered) interface.

2.4.1 Prototype System:

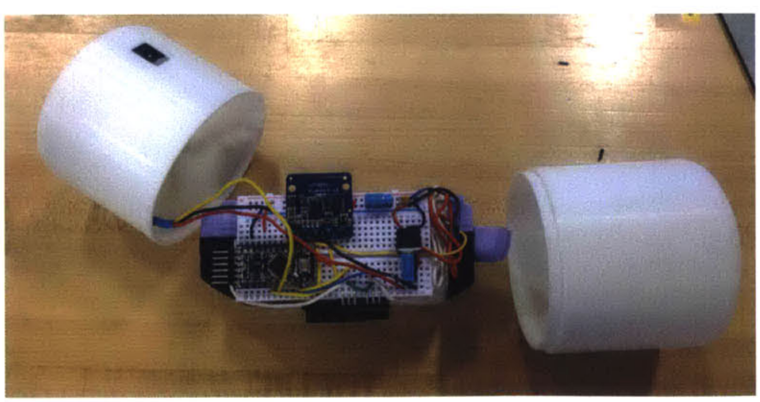


Figure 12: Electronics of the pump prototype including an Arduino Nano, bluetooth module, and other circuitry to drive the motors and collect sensor data.

The goal of the current prototype is a proof of concept for three principles. The first is to create a pump that can drive a variety of collection mechanisms (from different companies) through flexible software control. The second is to create a responsive platform that enables the use of a user centered interface design. The third is to create a platform that lowers the barrier for further research and enables big data collection of pumping parameters and experience, which in turn allows for optimization and improvement in the field.

2.4.2 Hardware and Sensing:

Most existing breast pumps are hardware-defined, with their pumping mechanism hardcoded on to the electronics of the device, and without driving these mechanisms with sensor data, these pumps are unaware and therefore unresponsive to changing pump or user states. By being software-driven and sensor-equipped, our pump prototype can drive a variety of mechanisms such as pumping cycles based on user and pump data:

Sensing at the pump: Using a pressure sensor at the internal pump, we are able to monitor generated vacuum. This in turn provides feedback information on the frequency and amplitude of the pumping generated by a given actuation pattern of the pump.

Sensing at the breast: A pressure sensor incorporated at the flange (between the last vacuum separation membrane, if any, and the breast) provides information about the actual pressure profile being generated at the breast.

Pump-as-platform: This is a data-driven pump + API. What this also allows is further research to be conducted and collected more quickly and more easily than is done today.

2.4.3 Affordances of a Human-Centered Pump Interface:

A software-driven, sensor-equipped pump affords a highly responsive and human-centered user interface. Our pumps API allow this interface to be instantiated on a mobile device, computer, or custom hardware.

Personalized Pumping: We demonstrate a person-centered interface for adjusting pumping speed and pressure. Rather than expose controls for increasing and decreasing these settings, like the mechanism-centric interfaces of existing pumps, we frame the adjustments in humanistic terms: comfort, discomfort, and time constraints.

Timely, Data-driven Insights: Sensor data from the pump can be used to queue a collection of meaningful insights integrated onto the pumping platform. These insights can also take the form of data-driven support and encouragement, interventions and device warnings, celebration or success, or learning.

Accessible, connected support and learning resources: Pumping data and learning/ help resources can be tightly coupled. Users can quickly access advice that addresses pumping challenges, and these topics can be organized based on user and pump data.

Implicitly social support: Pumping can be an isolating experience, in part because social attitudes toward pumping and breastfeeding mean many women must pump in private. A software-defined pump is uniquely capable of integrating web-enabled social technology into the pumping interface.

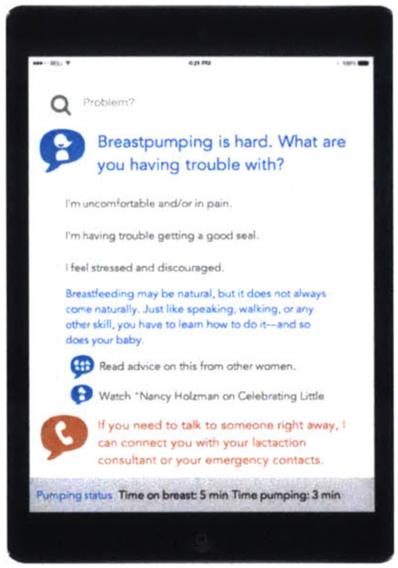


Figure 13: Personalized and more informative pumping interface running on a tablet. Options are provided for assistance, information and comfort.

We demonstrated through our prototype that small improvements to a historically unintelligent device not only has the potential to improve the experience for those using the device, but it can also provide informed guidance and information for the user and others who are facing similar issues. Sensing on the pump paired with an intelligent controller can fine tune the settings and remove that burden from the user. Data collected can inform the user and medical professionals where problems may be occurring so that help is more direct and faster. Finally by designing a novel UI built around the experience rather than the action, we aimed to lower the stress of pumping and provide personalized, and informed encouragement and assistance to those who need it.

3. Research Motivations:

3.1 Overview:

Pervasive, surface-based sensing could have a variety of meaningful applications across multiple areas from architecture to healthcare. Discussed below are the motivations for researching and developing this type of sensing. The primary area of focus is aging in place, and using surface-based sensing to assist the elderly population by monitoring for mobility and cognitive decline as well as unexpected events such as falls or other medical emergencies. Secondary areas of focus are new interactions enabled through pervasive sensing. How could such a system be used to enhance the way somebody interacts with their home or living/working environment?

3.2 Aging In Place:

Compared to only 100 years ago, human life expectancy has increased by almost 30 years and by 2030, the number of Americans age 65 and older will be about 71 million [6]. Today, because people are living longer, there are two generations that make up the aging population with the fastest growing segment consisting of individuals 85 years and older. 80% of seniors have one chronic disease, with half of that same group suffering from two chronic conditions. Even among US adults, 45% have at least one chronic condition and by the time seniors reach 85, some may have three or four chronic conditions [7]. These chronic ailments, the cost of which is about $2 trillion yearly, range from diabetes, to visual or mobility impairments, to more serious conditions such as Alzheimer's or dementia.

As the population ages, the number of caregivers is expected to decline while the wait times for nursing homes has been increasing yearly. For example, in Ontario, since 2005 the wait times for nursing homes have tripled [5]. The lack access to long-term care paired with healthcare cost cuts has shifted responsibility from hospital/long-term care to the home. Now it is more common for care to "be delivered by family members" [5, 6].

Many aging or senior individuals do not want to go to homes and more would rather stay independent and live in their homes, i.e. age in place. Thus the CDC defines aging in place as "The ability to live in one's own home and community safely, independently, and comfortably, regardless of age, income, or ability level" [14].

With many families living far apart, monitoring and sensing technologies are being explored as a powerful solution to address some of the problems associated with aging in place. Because of the promise of extended autonomy and peace of mind, more seniors are agreeing to being tracked by their caregivers. The use of wearable and internet connected devices will soon know "all of your vital signs, know your nutritional habits, know exactly where you are and how you are moving, note your weight, blood sugar, etc.

and deliver them to caretakers" [4]. The movement towards platforms such as Apple HealthKit and Samsung Healthcare promise even more opportunities for diagnosis and monitoring when data from different devices can be exchanged and combined.

With the current quantified self-movement, wearables are tracking anything and everything, but Wired argues that what's missing is "common sense or ambition" in the data [7]. Fitness trackers provide superficial information for tech-savvy millennials who probably don't benefit as much from this constant monitoring. More than half of the US consumers who have owned an activity tracker no longer use it, with a third stopping within the first six months. The director of Qualcomm Life urged developers to "stop screwing around in a saturated market for activity trackers and embrace the red-tape and regulatory friction of the healthcare industry" [7]. Simply put, more developers are designing non-clinical or pseudo-diagnostic health monitoring tools for those who really don't need them. Meanwhile, of the adults with chronic conditions in the US, 40% with one chronic condition track their health, and 62% of adults with two chronic conditions do so. There are fewer devices that are intended for that market. The aging and elderly population is no different. If wearable technologies and data-based monitoring and diagnostics were applied, more individuals would be able to live at home longer, and not only that, but more conditions can be caught before they become serious problems.

Professor Elizabeth Mynatt of Georgia Tech has researched aging in place and the role technology and sensing will take to assess some of these problems. Three key problem areas Mynatt and her group identify are crisis recognition, cognitive support, and awareness of daily life and long-term trends.

Crisis recognition is responding to an immediate crisis such as a fall, or an impending crisis such as a broken heater in the winter and the "house must recognize potential problems and notify residents and outside support" [9]. Cognitive support includes systems to "recognize forgotten or interrupted tasks and assist the occupant in resuming them" as senior adults are prone to lapses in memory [9]. Awareness of daily life means maintaining daily contact between family members. This contact provides a sense of security for the senior adult and peace of mind for the rest of the family.

By receiving sensor information that indicates health, fitness, environment, relationships, social interaction, events, and activity, a family is able to see quantitatively and qualitatively how their loved-one is doing. Cognitive and physical abilities change slowly over time, and an individual living alone and away from their family may not have someone to notice these changes as they are occurring. By using real-time and historical sensor data, information and trends about the loved-one is provided constantly to the family.

3.3 Cognitive Decline:

Of the 5.3 million Americans with Alzheimer's, an estimated 5.1 million are 65 and older and the number of Americans with Alzheimer's disease and other dementias will continue increase every year. By 2025, the number of people is estimated to reach 7.1 million [12]. Dementia and similar diseases is associated with compromised cognition and reduction of independent functioning and inability to perform activities of daily living is related to distress and reduced quality of life for the patient and caregiver.

3.3.1 Activities of Daily Living:

Impaired ability to conduct activities of daily living is a "diagnostic criterion for dementia" and an indicator of impending healthcare and caregiver burden [11]. The assessment of "everyday functioning" for older adults focuses on an individuals ability to carry out these ADLs. There are two general types of ADLs, basic ADLs (BADLs) and instrumental ADLs (IADLs) [11]. Basic ADLs include tasks like grooming, feeding, and using the bathroom, things that require less coordination but are necessary in the daily routine. Instrumental ADLs involve more complex behaviors including managing finances, handling medications, housekeeping, or cooking. BADLs are "highly correlated with motor functioning and coordination" while IADLs have been linked to cognitive function and is affected early in the course of dementia. Declines in IADLs may also be present in pre-clinical dementia states such as "mild cognitive impairment" [11]. For a specific ADL test, see Appendix D.

3.3.2 Reporting ADLs:

Assessment in large-scale studies is usually conducted through "informant report" with the advantage being asking family members or caregivers who are familiar with the individual's performance in real-world environments. These people are asked to rate the individual's level of functioning and it is a more "time and cost-efficient" method [11]. Performance based ADL assessment allows a trained rater to directly observe behavior in well-defined functional tasks. The only problem with the observation method is that tasks performed for another may not accurately reflect the true abilities. Because most performance-based experiments are time-consuming, they are "less practical for large-scale research studies" [11].

3.3.3 Using Surface-Based Sensing to Monitor ADLs:

The best way to monitor ADLs would be right in the individual's home. Large area sensing could capture this activity information and when paired with other sensor data, such as tags in the furniture and appliances, a comprehensive picture of the individual's daily life could be created. Especially with individuals suffering from or at risk for dementia or Alzheimer's, making sure the person wears an accelerometer, GPS slippers or other "wearable" tracking is a difficult task. Surface sensing requires no direct interface with the individual. Over time, trends in the performance in specific activities including average time of day performed, duration, and order could indicate onset of dementia. The benefit of continuous monitoring and comparison is that all events are scrutinized. Subtle changes in time it takes to find the keys wouldn't normally stand out as a problematic but a computer could recognize a gradual increase in time it takes. These facts could be reported to family members and doctors when necessary. At the same time, pervasive monitoring could keep people in their houses longer by confirming to loved ones that the individual is still capable of living on their own. As Mynatt points out, "Parents are often forced to move out of their homes for emotional reasons rather than logical ones" [2]. Even when parents are doing well, she found that many children who were unable to check on them daily became so worried that this anxiety drove them to relocate their parents to assisted living. Comprehensive monitoring and analysis could serve to convince children or loved ones that the individual is still thriving at home and according to the data, the individual has no indications of dementia or other worrisome conditions.

3.4 Fall detection:

As the population ages and people are continuing to live longer, "the number of older Americans who fall and suffer serious, even fatal injuries is soaring" [25]. In 2013, the CDC estimated that one in three adults over the age of 65 falls [26]. The number of people that age who died after a fall reached nearly 24,000 in 2012 [25]. And in 2014, more than 24 million people over 65 were treated in emergency departments for injuries from falls. It's a serious problem. Many families have personal experiences with grandparents or elderly parents falling and severely injuring themselves. The cost to treat these injuries exceeded $30 billion in 2010 [26]. Most people are ambulatory before the fall, but the injury and subsequent healing often lead to long term negative impacts on quality of life and even risk of death. For those who do recover, many lose confidence in their stability while walking, and as a result avoid ambulation.

Retirement communities and nursing homes are encountering a problem in trying to "balance the resident's safety and their desire to live how they want". Most facilities are paying more attention to preventing falls. Even with safety upgrades such as path lighting and softer bathroom flooring, many residents "do not, or refuse to, recognize their own gradual deterioration, leaving them vulnerable despite efforts to protect them" [25]. The elderly population is their own worst enemy because they do not want to recognize that they are getting old. People resist transitioning to a cane or a walker because they feel stigmatized. Falls go unreported because residents worry that they will be taken somewhere else, even though one fall indicates risk for another. Electronic pendants for summoning aid in a timely manner are very useful but they require the person to be wearing it at the time of a fall. Countless stories entail a resident not wearing their pendant because of pride, taking a fall, and lying on the ground for hours or days until a nurse or housekeeping staff comes upon them. Some high risk residents have alarms that trigger an alert if they try to get out of bed but they are prone to false alerts and aren't taken as seriously. Many officials argue that the focus should be more on preventing falls.

3.4.1 Using Surface-Based Sensing to Monitor Fall Likelihood:

A large aspect of that prevention is anticipation. A "sluggish foot, eyesight that fails to catch a step down, slowed reflexes – these creep up slowly, often imperceptibly. This helps to explain why people tend to pay scant attention to their risk of fall until it happens" [25]. For both physical and cognitive decline, several clinical warning signs have been identified that are manifested as alterations to normal walking parameters (gait) or daily activity patterns. A pressure and location sensitive surface, especially in the home continuously monitoring the occupant's gait and activities, could continuously and unobtrusively capture a rich set of activity and behavioral data. These data can then be analyzed in real time and warning markers could be identified and notifications or even alarms in the case of a fall or medical event can be sent to the required people (children, caregivers, doctor etc.). In addition to floor based sensing for mobility information, identifying activities of daily living (grooming, working, eating, leisure etc.) via sensing in the tables, counters, and even walls could give details about the mental health of the individual by checking for shifts in behavior over time.

3.5 A Solution Involves Multiple Devices Working Together:

A variety of solutions exist that are intended for the aging in place market. They range from fall alert pendants, to doorway sensors, to pill taking reminders. Most deal with the symptoms of aging but very few attempt or do a good job of understanding the progression of aging and identifying problematic behaviors or events before they result in something catastrophic. Most systems are structured as a central hub with corresponding satellite pendants and sensors. For example, these sensors indicate when the individual has opened their front door, left their bed, or are wandering about their apartment. Alerts are sent to family for updates or medical responders in the event of a fall. Several companies including Phillips have systems that incorporate both the sensing and communication. But to achieve more innovative solutions, combinations of devices and sensors will have to work together to provide better and more comprehensive information. Companies such as OnKöl are working to incorporate multiple devices into the same ecosystem for the purpose of aging in place. In addition to the standard set of sensors and central hub, they also allow for multi-protocol connection to their system. Medical devices ranging from blood pressure cuffs, glucose meters, to heart rate monitors are able to send their data to the OnKöl system to be sent to the right people as well as stored for later analysis [13]. A more open ecosystem is the best way to address such a complex problem as aging. Many more types of sensing than a single manufacturer can provide are needed, and rather than owning multiple systems just to fill in the gaps, why not have all the varied devices sending their information to a single location? This is where surface-based sensing can find its place; as an intelligent component in a multi-platform, multimodal sensing environment.

3.6 Use of Cameras For Geriatric Monitoring:

Cameras are excellent sensors; for general object tracking and recognition, they are the best at doing so. All this data comes at a cost though; analyzing camera data is a computationally heavy task. But the low cost of current cameras and the increase in computational power of even low-end computers have lead to cameras being the go-to for tracking and visual identification applications. Monitoring people complicates the use of cameras as a tracking tool though. Privacy and respecting the individual's rights is a high priority when considering installing camera based monitoring systems for the elderly.

Cameras currently have two main purposes when applied to assisted care and aging in place for the elderly; 1) monitoring individuals with dementia or other cognitive impairments and reporting if any problems arise and 2) identifying and catching abuse or neglect of residents by nursing home staff.

3.6.1 Monitoring of Individuals With Dementia:

People who are no longer able to take care of themselves or are at risk of reaching this point due to various stages of dementia are prime candidates for camera based monitoring. Certain companies such as ResCare provide "telecaregiver" services where a home is instrumented with video cameras and a person remotely keeps an eye on the individual. ResCare states that "primarily the people using this at this time are in the beginning stages of Alzheimer's or dementia" so at the point where memory and judgment issues may put the individual at risk if left on their own [16]. Balance issues and potential for falling compound this risk.

Relatives and spouses seem to be supportive of camera intervention for their loved ones. Camera monitoring provides peace of mind for children and relatives who are not nearby. Some feel that the "burden's on [them] if something happened" and appreciate having the caregivers remotely watching over the loved one [16]. Able bodied/minded spouses also benefit from having camera monitoring. As is often the case, when a spouse is suffering from cognitive impairment, the roll of full-time caregiver falls on their husband/wife. With cameras watching out, the husband/wife is free to get out of the house and do their own activities.

Ultimately, the decision to install cameras falls on the spouse/children because individuals with dementia or other cognitive impairments "often lack the ability to consent to being recorded" [15]. Support for camera monitoring ranges from "workers see your rear end anyway…so if your family sees your naked behind, what's the big deal" to "they're [the telecaregiver] diligent…they're on the ball. And I like it" [15, 16].

A bigger concern with installing cameras, particularly in assisted living facilities, is inadvertently recording roommates, visitors or other people who pass through the room. Some states have laws that require letting people know if they are being recorded, or requiring that cameras be fixed only in positions "aimed at only the intended resident" [15].

3.6.2 Identify Abuse or Neglect:

Unfortunately, elder abuse occurs in some assisted living facilities, and families have taken to surreptitiously hiding "granny cams" throughout a loved one's apartment to catch offenders. This raises additional legal concerns, as some people want to know if they are being recorded. Some facilities that allow in-room monitoring usually require families to put a notice on the resident's door to "encourage transparency and reaffirm shared expectations for quality care" [15]. Some feel that even if recording is not occurring, the prospect will curb potential abuse.

3.6.3 Use of Surface-Based Sensing in Lieu of Cameras:

Cameras have definitely found a comfortable position keeping a watch on those individuals who are incapable of looking after themselves. They are an elegant and comforting solution for family members who want peace of mind as well as freedom to go about their day.

Where opinions become split is for able-minded but still at-risk individuals who may develop either cognitive impairment or are suffering from a chronic disease that requires special monitoring. For those who do not like the idea of being observed, they have strong feelings against a camera, "I figure what I do in this building is nobody's business but my own" [5]. Others who are suffering from a disease such as diabetes, and who know that they have to watch what they eat seem more open to a system that could encourage them eat healthier i.e. "if it knew how to make me healthy, I would probably like it" [4].

Based on the current uses and feelings towards cameras, a surface-based sensing system can provide an important source of complimentary information. For those with dementia and who are already being monitored via video camera, the data collected about their walking patterns and gait could be compared to the camera data to both confirm observations, as well as color in hypotheses about the individual's health. For example a sensing floor could identify that an individual has been dragging his or her left foot more. Video feed can confirm that the person is unable to pick up a plate or is not moving their left arm well. A stroke could be inferred from those two pieces of data, and even if that was not the case, help could be called sooner and assistance administered faster.

For individuals who are able minded, but possibly physically impaired, a floor is a non-camera solution to understanding the individual's physical state. It does not invade his/her privacy, it cannot ID anyone else who comes into the space, and it provides the qualitative as well as quantitative information a camera could give. If paired with other conventional sensors such as motion sensors in the cabinetry, sound sensors in the hallways, and tags on the various items within the apartment, a set of sensing surfaces could enable understanding without violating the privacy and bruising the ego of the individual. Surface sensors also provide additional information; for example pressure values are currently undetectable with conventional monitoring camera systems. Additionally, camera sensing systems still have "limited robustness to changing lighting conditions or clutter" while the floor only sees what happens on top of it; presence of shadows, a lit vs. dark environment, and furniture do not prevent footsteps, counter activity etc. from being recorded, assuming as the sensor gets actuated either directly or indirectly (such as through flooring or carpet) [42].

3.7 Closing Remarks on Aging in Place and Age-Related Diseases:

Continuous monitoring of activities of daily living and walking parameters via large-scale surface-based sensing could be a boon for healthcare professionals, care providers, family members and the individuals themselves. Any subtle early warning signs of a problem could be identified sooner and treatments or appropriate action could begin earlier. To collect the required spatial and load data, force and coordinate sensitive surface-based sensors will need to be developed that can cover a large area without being severely expensive. There are several companies that design pressure sensing mats for orthopedic applications, and though the resolution is more than adequate for gait analysis, the systems are often closed-source, requiring proprietary hardware and software to use [43,44]. Large pressure mats, on the order of a floor carpet, are also prohibitively expensive and preclude use outside of medical facilities and university labs. The project described in this thesis is an effort to create a practical, low-cost, and modular sensing platform for the home to fill this need for relatively fine-grained quantitative, as well as qualitative activity and gait information.

3.8 The Quantified Environment - Not "Wearables" but "Liveables":

The quantified-self movement has already demonstrated people's willingness to use devices that record and display their health and activity information. From heart rate watches, to breathing monitors, to sleep trackers, people are demanding more information about themselves. For the most part though, this data is merely graphed or presented in such a way that it is still a number. A number of steps walked that day, an average heart rate, or an indication of how restful your sleep was. What is more exciting is the next step, where multiple devices are working together to actually diagnose or predict disease.

Current quantified self-devices provide one piece of the picture; discretized variables like heart rate, breathing rate, steps taken, hours asleep etc. What about the complementary data set such as information that can be gathered from an individual's home or workplace? Anywhere somebody is spending long periods of time could be a source of physiological as well as behavioral information. The home in particular is a perfect environment to collect information. Your guard is down; the social constraints that could affect behavior in public are absent. A person is at their most natural state in their home. It is for that reason that the home is a perfect location to complete the quantification picture. Understanding how a person is moving around their apartment, whether they are watching a lot of TV or not eating enough could give greater understanding to why an individual is feeling a certain way.

As more internet connected devices are incorporated into spaces and talking to each other, this creates more interesting ways they can be used to understand a person within the environment. Rather than wearables, these devices would be **livables,** meaning the quantification is done around you. Ubiquitous computing applied to quantified self. Data from smart watches and monitoring apps would mix and co-mingle with behavior, activity, and sensing data coming from the floors, television, couch or bed. And rather than merely sending a text message or notification with pretty graphs, the data would actually be put to use understanding the individual's health and wellbeing. Powerful and sophisticated algorithms will take this information daily to create a profile of "you." And the new information can be continuously compared against previous data to assess if there is any change for the better or worse.

As discussed earlier, aging in place is a meaningful application that could benefit from more comprehensive data collection and analysis for an individual. Age related diseases and ailments usually come slowly and with some warning signs that may be overlooked. If a system is able to continuously check multiple data points against itself, more of those warning signs may stand out. Cognitive decline, motion disorders, and even immediate events such as a stroke, heart attack, or fall can be immediately recognized and responded to accordingly.

A system can also serve more purposes on behalf of a doctor. If after suffering an injury physical therapy is required, the system can encourage the individual to perform the required exercises. The data collected from each exercise performed can be sent to the doctor or physical therapist to keep them informed. As the data collection tools of the space become more sophisticated, all information obtained from a doctor's visit could even be collected from the comfort of the home. The number of in-office visits can be reduced, lessening the burden on the healthcare system and allowing those who actually need the in-person visits to be seen faster.

3.9 Data Driven Transformation of Space:

If spaces of the future are equipped with robotic furniture and connected devices, the ability to detect human habits and trends would allow these systems to work even more naturally around their users. For example, a system monitoring when somebody gets up in the morning, what music they put on, whether they go to the bathroom or make breakfast first etc. could allow for informed home automation. For a future connected apartment, getting out of bed in the morning could trigger the system to transform into bathroom mode, because it knows from floor and doorway sensor data that 95% of the time the user walks to the bathroom after getting up. From observing weeks of light setting data, the home can change the light colors to the preferred values on a minute by minute basis. Similarly, a user arriving home from work could trigger the space to transform into living room mode with their favorite TV channel already selected.

In addition to informed home automation, the system can adapt to disabilities or specific limitations it senses the person is experiencing. If a user breaks their leg, making it more difficult to move about the space, the space in turn can configure itself to meet these needs. Extraneous furniture is automatically driven out of the way to reduce risk of tripping. Commands shift to primarily voice control to eliminate a need to manually push elements. And as the individual heals, the system can monitor their progress and give encouragement. Progress data can be sent to the physician and family. Over time, the system could slowly revert to normal to accommodate the state of healing. Understanding the user's data allows a system to work with the individual more effectively.

3.10 Person-to-Person Communication and Connection:

Being able to monitor and sense individuals can allow greater connection between those who live remotely. In Section 7, two visualizations are described that are meant to give family members an indication of the health of their elderly loved ones. Research from Brigham Young University and UNC found that "social connection leads to a 50% increased chance of longevity" and a study by University College London found that social isolation "was associated with increased mortality" [1]. Clearly for elderly individuals, social contact and maintaining a connection with their family or friends is essential. But technology to enable communication/connection does not have to be for loved ones at risk because of age or disease. Why can't children who are away at college get their own presence in the home so their parents know how they are doing? Why not aunts, uncles, friends etc.? Through lights, transformation, music or any other feedback mechanism built into the space, a digital presence for an individual can be incorporated into the connected home. This can provide a range of meaningful applications from peace of mind for recently empty-nesters to comfort and a sense of community for families who live on opposite ends of the globe. The interactions can be made as simple or as complex as a developer desires.

For those relatives who have passed, their data can be incorporated into their family's homes permanently. A whole new meaning can be given to the phrase "gone but not forgotten." If a dataset of behavior and activity information has been collected about someone, that data can be downloaded into all the relatives' homes to merge with the native information and provide a slight, yet noticeable digital presence.

3.11 Games, Exercise, Education and Other Applications:

If the surfaces in a home were aware, many applications can be programmed to incorporate these surfaces into familiar activities. Video games are an obvious extension. Why sit on a couch pressing a button to move, when roaming around the living room could cause the character to move as well? And a pivoting projection screen can move the screens to different walls as the player turns, crouches and pivots. Virtual reality devices could receive input from the sensors in the home, and the map in the video game could become a digital version of the player's home. Exercise can be made more fun and involved. If a user is sitting on the ground watching TV, the lights can shine on the floor in a specific pattern indicating that it is time to stretch. The floor sensors can verify stretch quality and form and the home can respond accordingly to good form and give suggestions to improve poorer form. Watching TV for too long could cause the system to suggest taking a break and going for a walk. New types of TV media can be designed to incorporate devices of the home. Imagine if the shows a user enjoys also have corresponding devices that enhance the experience. Children's education can feel less forced by involving all of the familiar spaces of the home. Projected interfaces can place math problems on the floor that a child has to jump on to answer. Geometry can be taught by lighting up angles and points on the floor, and only by walking to connect the dots does the shape and area traced out reveal itself. Activities that require different settings such as sitting down to write or going to the gym to exercise are more prone to incompletion, because it becomes just an additional activity. But with sensing and intelligent feedback, activities can be seamlessly blended with an individual's normal daily activities.

4. Previous Work on Surface-Based Sensing:

4.1 Overview:

There have been multiple projects to design surface-based sensing system in Academia and in Industry with approaches ranging in both scale and resolution. Described below are some physical and digital approaches to capturing position and motion. For physical sensors, the two common approaches taken are either a modular "tile" approach, or an "all in one" design where the dimensions are predetermined. For digital approaches, most require the use of a camera to either directly identify position, or through intermediate hardware retrieve points of interest.

4.2 Physical Surface-Based Sensing in Academia:

4.2.1 The Magic Carpet - Responsive Environments Group, MIT Media Lab, 1997:

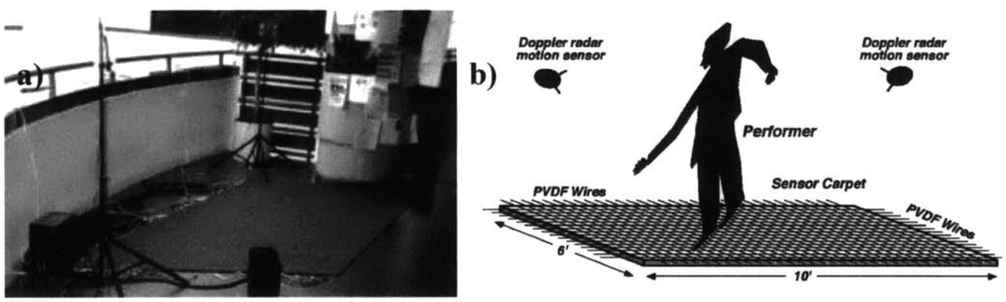


Figure 14: a) Magic Carpet set up in the Media Lab. b) Schematic drawing of the Magic Carpet system. Image Source: Responsive Environments.

The magic carpet project consists of a pair of Doppler radars to measure upper-body kinematics such as velocity, direction, and amount of motion, and a grid of piezoelectric wires to monitor dynamic foot position and pressure. The use of the system is for sound and music creation, where a user can wander around the area and their position and movements trigger sequences and musical sounds.

The grid of piezoelectric wires is 16 x 24 and covers an area of 6' x 10'. The wires are spaced at 4" intervals and allow the system to sense foot pressure and position. The wires also contain a special layer of piezoelectric copolymer, which causes the cable to produce a voltage when it is flexed or pressed. Previous sensor floors have measured foot pressure with force sensitive resistor sheet, but the wire is more rugged, readily available, and exhibits a higher dynamic range in pressure response [42]. The wires are able to distinguish between soft foot motion and harder impacts, and the wire shielding eliminates stray signals, pickup or crosstalk. An operational amplifier circuit buffers the signal from each wire, a diode/capacitor envelope detector produces a pressure profile, and a 68HC11 microprocessor scans the signals with a multiplexer. When a peak in pressure is detected, the processor plays a corresponding MIDI note corresponding to the position of the wire, and the note velocity corresponding to the pressure value [42]. The Doppler radars were needed to track the movement of the arms and upper body, something the floor sensor was unable to do. A signal is generated with a 2.4 gigaHertz continuous waveform (CW) oscillator, coupled to a four-element micro patch antenna [42]. The Doppler-shifted reflections from the performer return to the antenna, where they are mixed with the transmitted signal. This produces beat frequencies, which are analyzed to determine the dynamic state of the performer. Frequency of the beats is a function of velocity, and beat amplitude is a combined function of the size and distance of the reflecting object [42].

4.2.2 Z-Tiles - Responsive Environments Group, MIT Media Lab, 2004 :

Z-Tiles are modular, pressure sensing nodes which can be connected together to create a pressure sensing area of varying size and shape. Each node contains an array of force-sensitive resistors, and the information is output through a self-organized network of nodes. Previous projects, including the magic carpet take a different approach by creating pixelated surfaces using larger numbers of sensors (all in one) [41]. The systems worked well for the specific applications but lacked flexibility to be used in a variety of situations. The primary design consideration behind Z-Tiles was that it should be modular, consisting of individual nodes that can connect together during use but also be separated for transportation or for reassembly into a new shape.

The application for the Z-Tiles system was an interactive dance floor. The system has 4cm resolution and a response time of a few milliseconds.

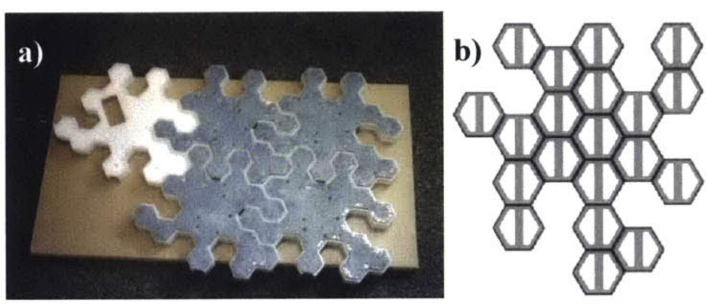


Figure 15: a) Assemblage of Z-Tiles. b) Individual Z-Tile showing location of pressure sensors. Image Source: Responsive Environments.

To achieve the required resolution, each floor node has twenty force sensitive resistors on its surface. The unique shape of each node allows them to lock together in a regular pattern and be self holding. The surface of each tile is protected by a 2mm layer of plastic.

Each node houses its own circuitry including five Cygnal C8051F2xx microcontrollers. These read the data from each node and communicate with other tiles in the floorspace. The communication is done via physical UART. When the nodes interlock, spring loaded connectors engage and provide electrical connectivity with the other titles. There are transmit, receive, power, and ground.

Sensor Network:

It is not feasible to have a data wire on each node, so the interlocking tiles must form a self-organizing network to pass the data to the single tile with a data connection. The physical data line reduces protocol overhead and allows for high speed and directed communication, "facilitating real-time data extraction" [41]. The code on each node is written such that data is always routed to reach the sink node in the fewest number of steps from one node to the another.

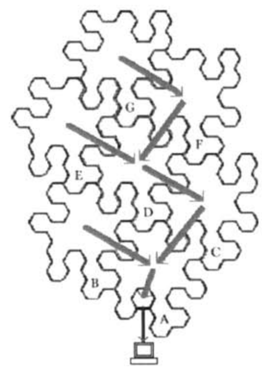


Figure 16: Schematic for how data is passed from each Z-Tile to the sink node for collection. Image Source: Responsive Environments.

Possible Applications:

The first application area is for music and dance control. Z-Tiles built in a floor can be used to generate music from the movements of dancers. Pressure sensing can also add another dimension of control for musicians using existing instruments if sensors are placed under a seat or " integrated into the seat itself" [41].

A second application is as an input device or the control of computer games. The system was tested within a VR game to control movements of a vehicle.

A surface of Z-Tiles can also be used in areas of medicine or sports science, where a therapist or doctor could obtain a dynamic view or recording of the footfalls of people walking, running, or jumping. Final work has been focused on matching areas of pressure on the floorspace to geometric figures such as circles or ellipses.

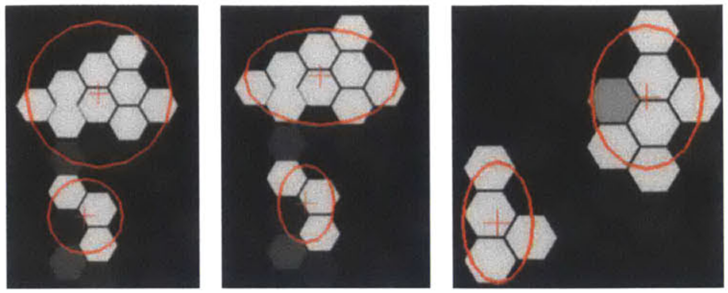


Figure 17: Using circles and ellipses to identify areas of high pressure. Image Source: Responsive Environments.

4.2.3 Conductive Inkjet Sensors - Responsive Environments Group, MIT Media Lab, 2011 :

This project designed and implemented a sensate surface based on conductive inkjet technology, which allows capacitive sensor electrodes and RF antennas to be cheaply printed onto a roll of flexible substrate. By placing the surface under a floor, it is possible to "detect the presence and whereabouts of users through both passive and active capacitive coupling schemes" [40]. Also incorporated into the prototype was GSM and NFC sensing and piezoelectric pressure/vibration detection.

Background:

Traditional electronic fabrication techniques are based on the PCB, which are rigid and have a fixed size once manufactured. Larger PCBs are difficult to manufacture, transport and deploy. New research into flexible and stretchable electronics is promising but is "far away from full-scale mass production" [40].

Conductive inkjet printing, now a commercial process, can allow for the design of low cost, large area sensors to detect users in an indoor environment. Unlike computer vision based tracking, inkjet printing and floor based sensing requires minimal computing power, is low cost, and can provide good resolution depending on the density of the sensors. A flexible sensor floor can be rolled out quickly and easily connected anywhere without constraints on lighting or issues with camera visibility.

For this project, a sensate surface based on a matrix of sensing tiles was crated. The tiles are printed in copper ink onto a thin flexible plastic substrate. Each tile is 0.3m x 3m and contains four electrodes for capacitive sensing, two RF antennas and another NFC pickup.

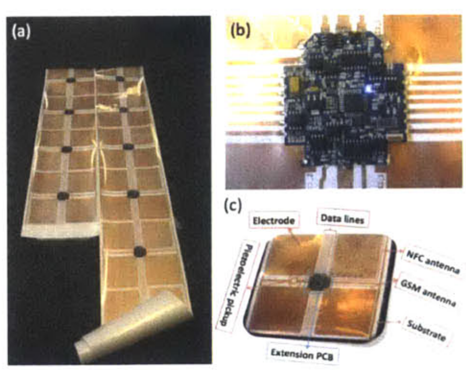


Figure 18: a) Surface composed of multiple tiles. b) Close up of PCB used to read the tiles. c) Different components of a single tile. Image Source: Responsive Environments.

Signal conditioning and processing circuitry was created on a conventional PCB because surface mounting components on the inkjet printed surface, though possible, is more difficult and limiting because it is single sided. Each circuit board contains the signal processing circuitry as well as an Atmega368 microcontroller, which can take the detected signals and send them to a PC over an I^2C bus. The prototype sheet is ~0.3m in diameter due to manufacturing limitations from the supplier. Folding has the potential to expand the coverage of the sensor to both flat and non-flat geometries without the need to cut or reconnect different pieces of the substrate.

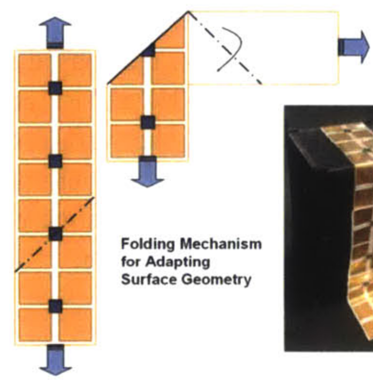


Figure 19: Sheet folded to accommodate different surface geometries. Image Source: Responsive Environments

Additional applications and features such as person recognition were also explored. The system would have to be paired with other sensing technologies to achieve the additional information, but the apparatus to incorporate different sensing types can be integrated into the sensor when it is printed.

Figure 20: User with a shoe tag stepping on the surface. Image Source: Responsive Environments.

4.3 Physical Surface-Based Sensing in Industry:

Several companies manufacture pressure sensing surfaces for medical applications as well as industrial testing and measurement. Three companies who's system design influenced the ultimate thesis project design are described below. They are Tekscan, Elsi, and Tactonic. Tekscan is one of the industry standards for medical monitoring and general pressure sensing. Elsi is a Finnish company who designs large capacitive sensing systems for assisted living facilities. And finally Tactonic is a modular, high resolution system with a variety of applications including gait analysis.

4.3.1 Tekscan:

Tekscan provides single force sensors, position sensors, and matrix pressure sensors for multiple fields including dental, medical, and test and measurement.

a) b)

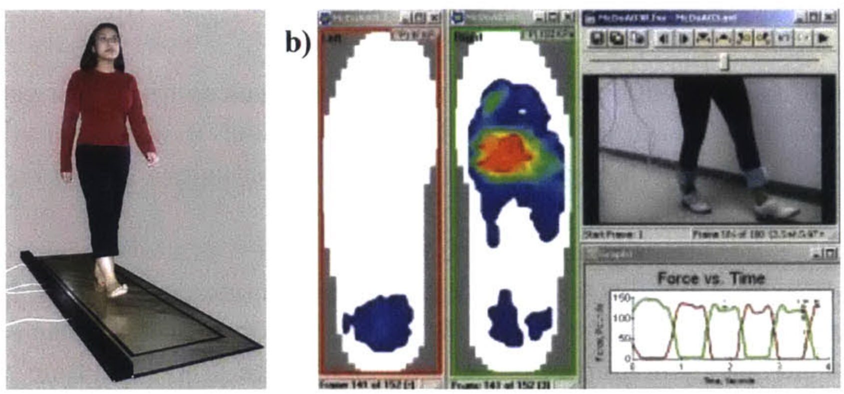


Figure 21: a) Tekscan Walkway system in use. b) Output from Walkway system sensor readings. Image Source: Tekscan.

Of particular relevance to this thesis is Tekscan's position as an industry leader in the area of gait tracking and analysis through high resolution, pressure sensing walk systems. Their Walkway System provides a "low profile, gait analysis walkway that captures accurate kinetic, timing and physical measurements" [43]. The system includes software to automatically calculate an array of gate parameters including step and gait time, distance, velocity and cadence. It can capture multiple footsteps for analysis and labels them as either right or left. Foot segmentation helps in the calculation of toe-in or toe-out angle. A variety of other features are offered with the system. The design is modular in that the customer can choose the system sensing area and sensor resolution. The company then fabricates the custom sensor to fit to those specifications.

The base system offers 1.4 points of sensing per cm^2 at 100Hz scan speed. The high speed system has a 440Hz scan speed, and the high resolution pediatric system offers four sensors per cm^2 at 185Hz refresh speed [43].

4.3.2 Elsi Technologies:

The Elsi system consists of strips of capacitive sensing foil connected together with signal processing circuitry along the wall. The resolution of the system is about 6" x 6" and can work under carpet and laminate flooring.

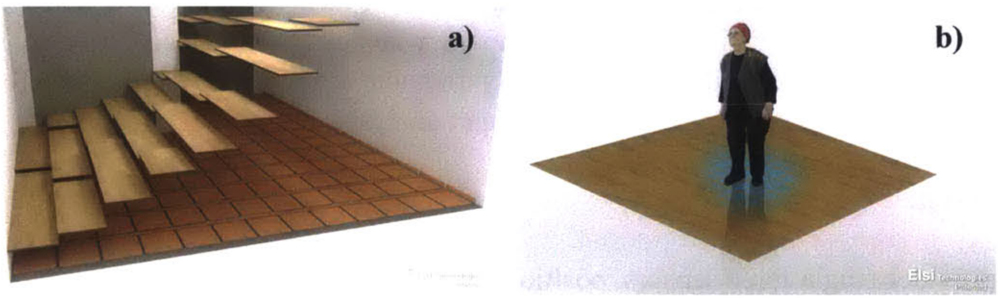


Figure 22: a) Elsi system installed under laminate flooring. b) System sensing an elderly individual walking on the floor. Image Source: Elsi Technologies.

Various alarm notifications can be activated in the Elsi UI that can trigger for various events. Nurses can receive alarms directly to their smart phones indicating anything from a resident getting out of bed to taking a fall. This enables assistance to occur faster [49].

Other intelligent features are programmed into the system such as a toilet visit counter, door entrance/exit alarms to indicate wandering or uninvited intruders, and lights that trigger when someone gets out of bed in the night. The system provides peace of mind for the residents and allows the nurses to "be at the right place at the right time" [49]. Routine checks are eliminated which saves resources and preserves privacy and dignity.

4.3.3 Tactonic:

Tactonic's goal is "extending the boundaries of human-machine interaction" with multi-touch and pressure imaging sensors [44]. Tactonic sensors allow for pressure sensing multi-touch and pressure imaging. They can replace conventional computer touch pads, as well as larger tablets and interactive whiteboards.

Specifically, Tactonic designs floor tiles that allow one to capture foot pressure over any size area. The 2' x 2' tiles can be connected to create regions that can be used for medical/gait analysis, interaction, people tracking/counting, or factory ergonomics. The tectonic tiles have a sample rate of between 110-500Hz and a resolution of 110 DPI [44].

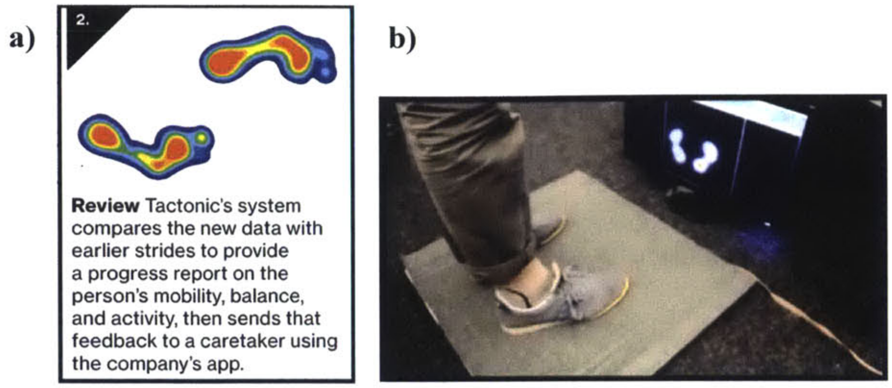


Figure 23: a) Output from Tactonic floor tiles. b) Demonstration of floor tile capturing a user's foot pressure profile. Image Source: Bloomberg.

The system compares new data with earlier walking data to provide progress on the person's mobility, balance and activity. If there is any deviation or alert, the system sends feedback to a caretaker with an app. Conditions like Arthritis, joint weakness, and Parkinson's disease can be monitored using the pressure sensing system [50]. In addition to home use, the Floor Tiles can be very useful to hospitals and physical therapists.

4.4 Hobbyist Physical Surface-Based Sensing:

Overview:

Attempts to create pressure sensing surfaces and especially floors have been attempted by the hobbyist community in a variety of implementations. One in particular that stood out for its large scale and semi-modularity was Sean Voisen and his team at Adobe.

Sean and his team were tasked with designing "something new" for the Children's Creativity Museum in San Francisco [51]. They decided to design a digital/physical environment for kids, which featured a 14' x 8' touch-enabled LED wall and a 14' x 12' pressure-sensitive floor. Sean's specific task was the design and construction of the pressure sensitive floor. It is 168 square feet with one pressure sensor per square foot [51].

The goal was to create a new type of digital input device "that could sense and track multiple children running or jumping in the exhibit, thereby allowing games running on the accompanying LED wall to respond appropriately" [51]. Pressure sensing was chosen over other tracking technologies because of its robustness and simplicity. Vision tracking is computationally complex and prone to failure when faced with multiple objects or people to track. 3D cameras also wouldn't have provided accurate information related to jumping force or weight.

Hardware, Layout, and Modularity:

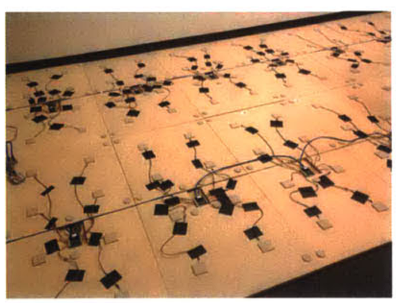


Figure 24: Pressure sensing pads underneath the exhibit flooring. Image Source: Sean Voisen

The floor consists of 21, 2' x 4' modular tiles in three rows of seven. Each tile contains eight pressure sensitive resistors and accompanying electronics. The tiles are slave devices on a long I^2C bus and controlled by a single controller, a BeagleBone.

The tiles are designed as a sandwich of materials. A bottom layer of MDF houses the FSRs, wiring, and electronics. Heavy duty felt pads protect the sensors and are sandwiched between the MDF and 1/2" piece of chip-board to disperse the weight. The final two layers are plastic sheeting and 5/8" close-celled foam [51].

Electronics:

Figure 25: PCBs used to connect separate pressure sensing modules.
Image Source: Sean Voisen.

Tekscan A401 FlexiForce FSRs were used as the pressure sensors. Circuitry to linearize the FSR output and improve the dynamic force range was designed with an LM324 op-amp. ATtiny84 microcontrollers with 74HC4051 multiplexers were used to read the signals into a single ADC pin. A PCB was designed to incorporate the ATtiny and RJ-45 ethernet connectors (for the board-to-board connections). The Ethernet connectors provided power and ground lines as well as the SDA and SCL lines of for the I^2C bus [51].

4.5 Digital Sensing in Academia:

4.5.1 IR Multitouch Table - Multiple Instances of Use, Summary by Seth Sandler:

Using IR LEDs and an IR sensitive camera, it is possible to design surfaces that can be used as multitouch surfaces. These types of surfaces have been used for a variety of academic projects and demonstrations.

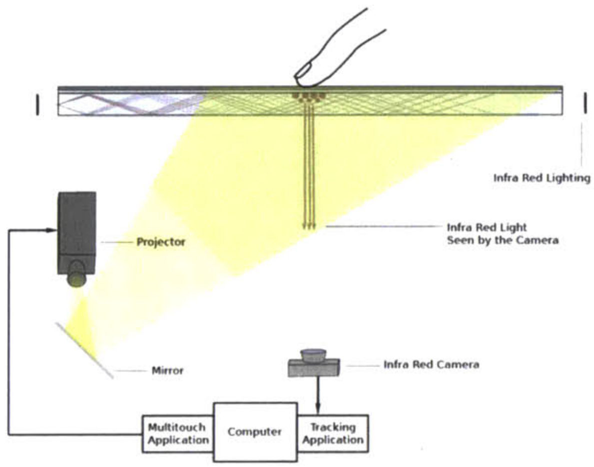


Figure 26: Diagram of an IR multitouch surface with projector overlay. Image Source: Seth Sandler

Most tables work by first shining IR light through a surface (usually acrylic). Due to total internal reflection, the light shines through the pane without scattering. When an object or finger is placed on the surface, the light scatters at that location and a camera with the IR filter removed can see a bright spot in the shape of the object. At the same time, projection from below can create interfaces that do not interfere with the touch sensitivity, because the camera does not see the regular light of the projector [47].

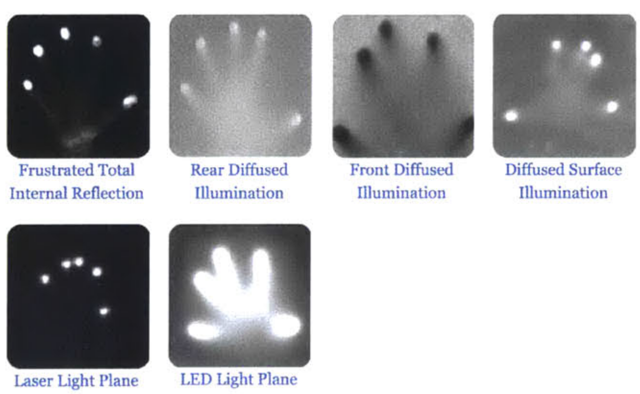


Figure 27: Sample outputs from different IR table configurations. Image Source: Seth Sandler.

Different variations on this same set of components (IR light, surface projection and cameras) are able to produce the results in Figure 27 above. Computer algorithms can then track the points as they move about the surface.

4.5.2 GravitySpace - Hasso-Plattner-Institute, 2013:

Most applications of the IR light table have been on the scale of a table or desk, but one group applied it to an entire floor area.

Figure 28: Students sitting on the GravitySpace prototype demonstrating the systems ability to recognize objects and project a digital mirror-image in those positions. Image Source: Hasso-Plattner-Institute

IR LEDs illuminate an entire floor area, which is raised above the camera and projector. Interfaces and games can be projected onto the floor and the camera records pressure information regarding how people are moving on the space. Though the system isn't truly weight sensitive, it can infer that a more defined shoe print has more force behind it as compared to one lightly showing through. Poses, individual users, and specific pieces of furniture can be identified by the system [46].

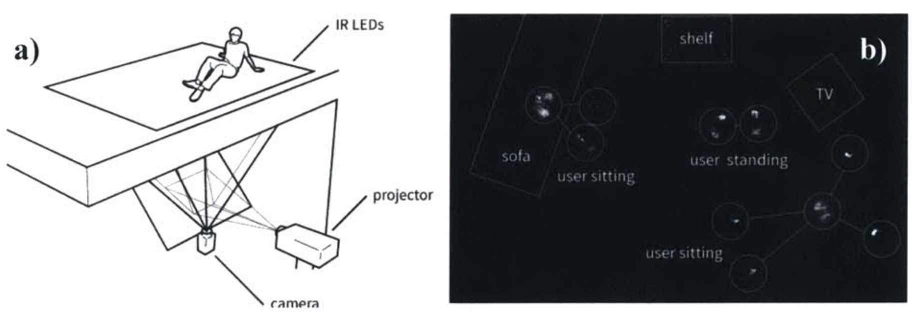


Figure 29: a) Schematic of GravitySpace elements and positioning. b) What the GravitySpace camera sees from the floor. Image Source: Hasso-Plattner-Institut

4.5.3 LuminAR - Fluid Interfaces Group, MIT Media Lab, 2013:

LuminAR is a "projected augmented reality system" that uses projection to create interfaces and camera vision to analyze motion, position, and activity on those interfaces [45]. It consists of a camera, depth sensor, projector, and onboard computer. The system can project interfaces onto any surface and recognizes and responds to objects in its field of view. Multitouch and gestural interfaces are possible with the system. The LuminAR also provides augmented experiences (augmented reality) that combine digital and physical information. The lens supports multi-touch interaction with any number of fingers, and also allows tracking physical objects in three-dimensional space. The lens can take pictures with the camera and track the presence and shape of arbitrary objects within the "bounding box of interest" [45].

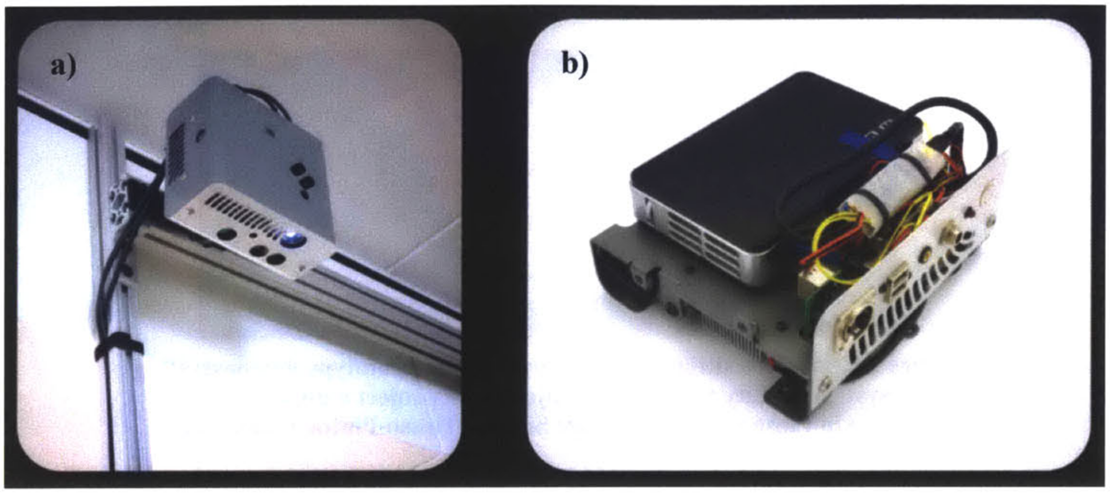


Figure 30: a) LuminAR mounted above a desk. b) Inside of LuminAR system. Image Source: Fluid Interfaces

The LuminAR provides a way to project interfaces on any surface and capture the use of those interfaces. Similar systems such as a Kinect have the ability to record activity and movement but do not have the interface aspect.

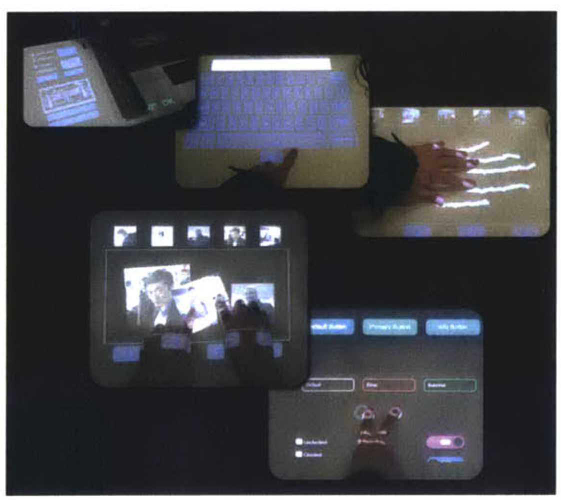


Figure 31: Sample interfaces the LuminAR can create and recognize activity from. Image Source: Fluid Interfaces.

5. Development of Pressure Sensor Matrices:

5.1 Overview:

After surveying the different modes of large-scale, surface-based sensing, force sensitive resistors or FSRs were chosen because of their simplicity, low-profile construction, and ability to scale to different shapes and conform to different sizes. What follows is an exploration of designing different configurations of pressure sensing matrices.

5.2 The Force Sensitive Resistor (FSR):

FSRs are polymer thick film devices that exhibit a decrease in resistance as force is increased on the active surface. FSRs have a variety of applications from musical instruments, to industrial robots, to joysticks and controllers for video games [52].

FSRs contain two primary elements that allow them to function. The first are the **electrical traces**. These are what connect to the computer or driving circuitry. These traces are printed onto a substrate such as mylar and contact the FSR material during use. To increase the contact area, on the FSR material, the traces have interlocking fingers at the active end of the sensor. For analog FSR sensors, there are two traces, one for power, and one to read the signal. The second element is the **FSR material**, which is a resistive polymer impregnated with carbon powder or similar conductive granules. The FSR material used in this project is Velostat by 3M [64]. Compressing the FSR material causes more granules to come into contact, increasing the amount of contact surface area, creating more paths for electrical current to flow, and lowering the overall resistance through the material [67]. Some FSRs also include a spacer of a non-conductive or dielectric material between the traces and FSR material to ensure that there is no contact until force is applied to the sensor.

As categorized by Sensitronics™, FSRs can come in two main form factors, Shunt Mode and Through Mode [52]. In shunt mode, both the electrical traces are printed on the same side of a substrate. A spacer layer is then placed on top of the circuit layer. The FSR material is the final layer. Interlink sensors are Shunt Mode FSRs. Through mode sensors have the electrical traces printed on separate layers and sandwich the FSR material and spacer layers. FlexiForce sensors by Tekscan are Through Mode FSRs [53, 54].

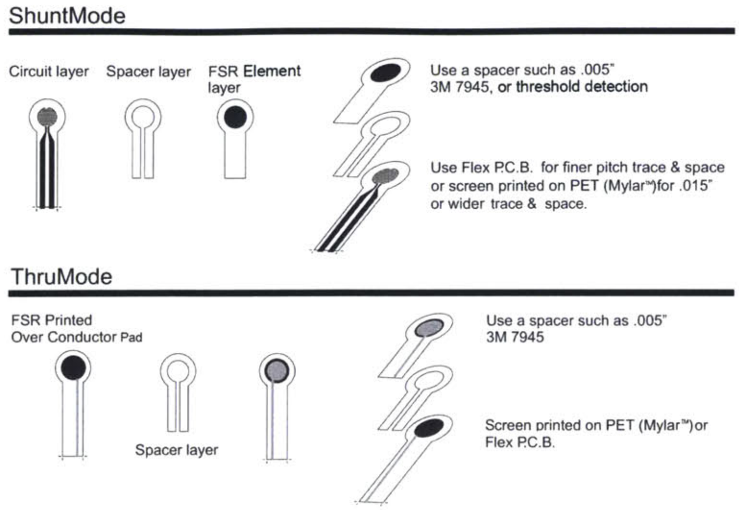


Figure 32: Shunt Mode and Through Mode FSR configurations. Image Source: Sensitronics

5.2.1 Calibration of FSRs:

Most FSRs have similar resistance/loading characteristics that is driven mainly by the FSR material used. A power-law characteristic (1/R) can be used to describe the resistance of the sensor as force is applied. With enough force, the changes in resistance become minimal and additional circuitry is needed to perceive the changes. The saturation pressure for a typical FSR is on the order of 100-200psi, so a 0.5" diameter sensor has a saturation force of about 25lbs. Larger forces can be measured by spreading the same force over multiple sensors, thereby increasing the effective area [53].

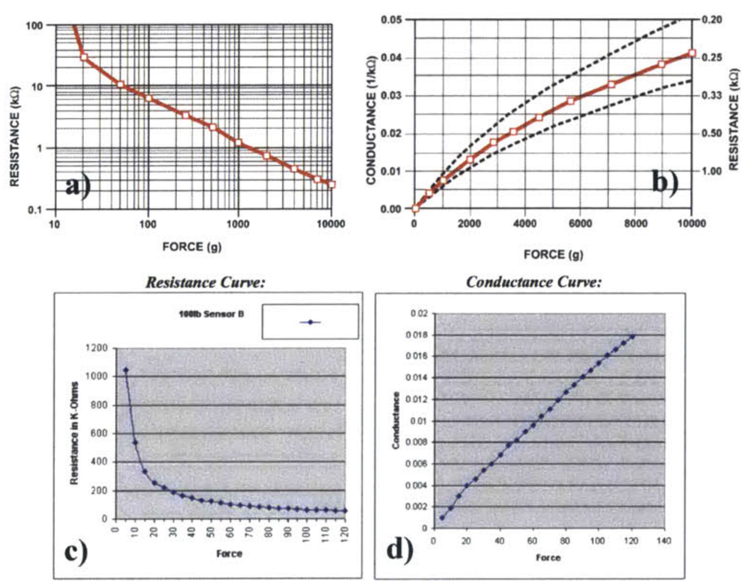


Figure 33: Force/Resistance and Force/Capacitance curves for Interlink (a, b)and FlexiForce (c, d) sensors. Image Source: Interlink and Tekscan.

5.2.2 Circuitry to Drive FSRs:

In common hobby usage, FSRs are powered with 5V and read with the analog pin (A0) on a microcontroller such as an Arduino. A voltage divider circuit is used to condition the FSR output; a pull down resistor of 10kΩ brings the FSR value into a range readable by the microcontroller.

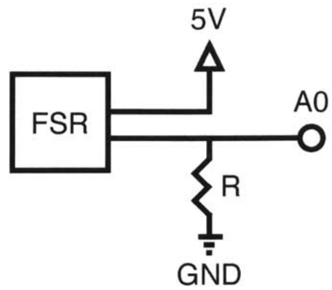


Figure 34: Voltage divider circuit for a single FSR.

5.3 Final Project for Human 2.0 (Human Augmentation) Course:

The first two sensor matrix prototypes were designed for the Human 2.0 course taught by Professor Hugh Herr at the MIT Media Lab. In order to collect data regarding the location of the force, a different configuration than the single sensor was needed. There were still two conductors sandwiching the polymer, but they were arranged as rows and columns on either side of the sheet of FSR Material. Each row is connected to a digital write pin on the Arduino, and each column is connected to an analog read pin. By writing one row high (5V) and reading each of the columns (A0-A3), a scanning read of the entire mat can be achieved. This method gives (row*column) unique values [48].

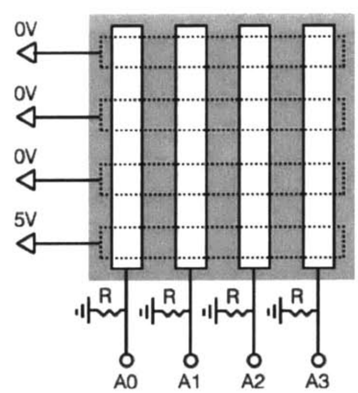


Figure 35: Schematic for row and column configuration of the pressure mat

Tekscan uses this collection method in their pressure sensor matrices. A key difference is that they use an inverting amplifier before reading the output signal. They did this to eliminate the ghosting effect (Section 5.4.4). For the prototype described in this paper, pull down resistors are used [43].

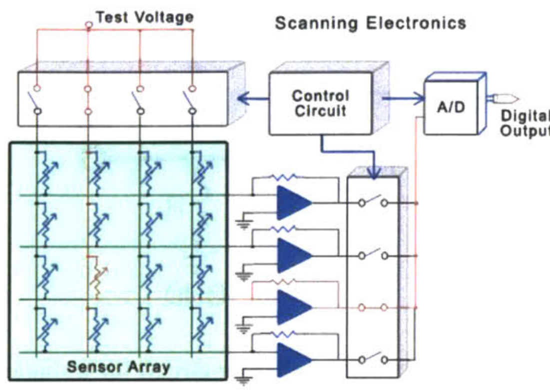


Figure 36: Schematic for Tekscan pressure sensing matrix. Image Source: Tekscan

Two iterations of the pressure mat were constructed, one that was ~ 12" x 12" and another that was 20" x 32". An Arduino Uno controlled the smaller version, and an Arduino Mega controlled the larger version. For the smaller iteration, a 1/4" thick square of acrylic served as the base while a 1/16" square of acrylic served as the top surface. Inspired by the work of Jie Qi and her circuit stickers, strips of copper foil tape were used for the conductor rows and columns [66]. A square of Velostat FSR material was placed in between the intersecting conductors, and the termini of the strips were connected to their respective pins on the Arduino. For this configuration, the Arduino would read 25 values (5 x 5 sensors) between 0 and 1023 (analog read range). The resolution was about ~ 2.5" x 2.5".

Figure 37: Smaller pressure mat prototype with 25 sensors (white dots indicate location).

The black material is the Velostat. The rows and columns can be seen above and below the Velostat.

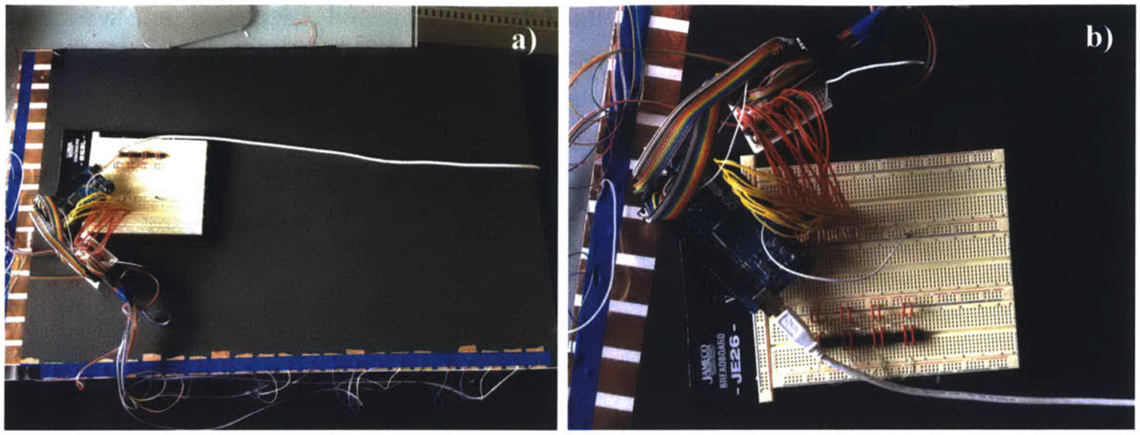


Figure 38: a) Larger pressure mat prototype with 345 sensors. b) Control circuitry for larger pressure mat prototype

The second iteration was closer in area to a small rug and was sized so that at least two steps could be captured. Instead of acrylic, the two main surfaces were thick paper. One of the lessons from the smaller design was that due to the more rigid nature of acrylic, any applied force was distributed over a larger area, even if the force wasn't being applied at the surrounding points. Paper is more flexible, so the applied forces don't bleed into neighboring nodes as much. The new pad had fifteen columns by twenty-three rows. The number of analog read pins limited the number of columns. Since the Arduino Mega has forty-eight digital pins, there were more than enough pins available for the rows [61]. The Arduino could now read 345 values. The resolution of the new mat was about 1.25" x 1.25".

5.4 Designs Leading to The Member's Week 2014 Prototype:

5.4.1 Overview:

Using the Human 2.0 Project as a jumping-off point, attention was focused to developing coordinate pressure sensing on a larger scale. Rather than having an area rug, the aim was to make entire floor able to sense what was happening on top of it. As a colleague said, "the floor is the monitoring camera". The only trouble with having floor-sized sensors was the cost and scalability. An affordable solution had to be on the order of ~$1000 for a decently sized floor area (20ft x 20ft). The resolution had to be small enough such that individual footsteps could be seen. Most of the Summer 2014 was spent designing and prototyping different pressure sensor configurations. The goal was to reduce cost and have a solution that was modular and able to provide relatively high-resolution data.

5.4.2 First Modular Design:

Because the final prototype needed to be modular, a version for modular tiles was designed such that multiple tiles could be connected together to create a larger surface. Electrical traces 1/2" thick were not needed so larger pads and smaller traces were tested. Electrical tape was used to mask the traces, and a small piece of Velostat was used to cover just the sensor trace pads. The two sheets are then brought together in a folding pattern; this FSR is a through mode sensor.

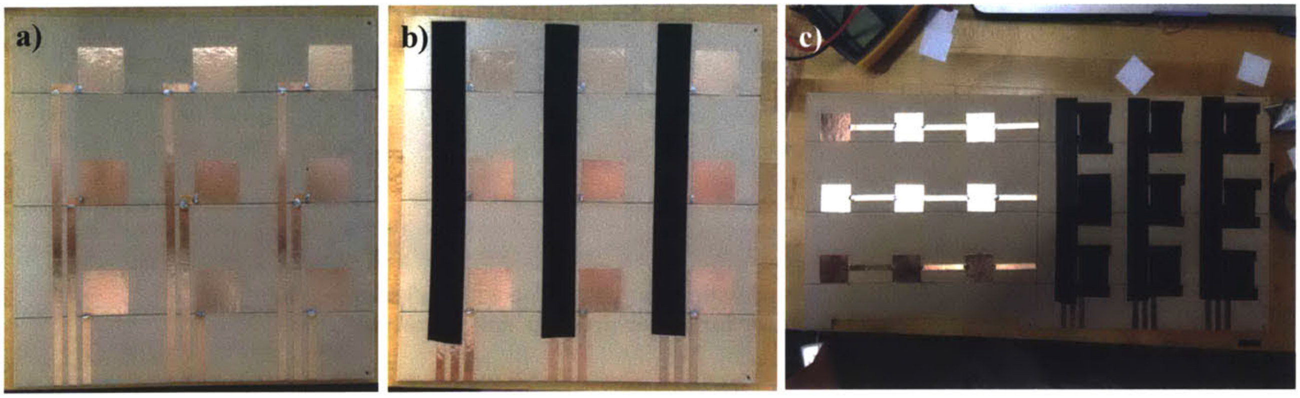


Figure 39: a) Bottom surface of a 3 x 3 pressure matrix. b) Traces masked with electrical tape. c) Upper and lower surface unfolded.

5.4.3 Single Sided Sensor Matrix:

Two separate sides of a single sensor matrix added complexity and made it more difficult to fabricate, so creating a set of rows and columns on the same side of a surface was investigated. This was achieved by placing the electrical pads side by side to create a shunt mode FSR. Electrical tape was again used to mask overlapping traces.

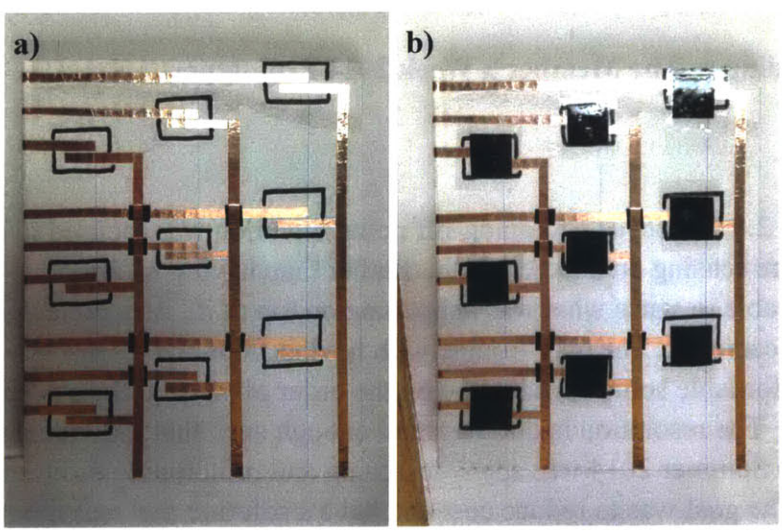


Figure 40: a) Shunt mode FSR traces. b) Squares of Velostat placed over the sensor pads.

5.4.4 Eliminating Ghosting:

When multiple sensors are pressed in close proximity, electricity is able to bypass an open connection and register as a pressed sensor to the computer [55]. This phenomenon was experienced in the first prototypes of the sensor mat. In order to solve this issue, diodes needed to be placed at every sensor location to ensure that there are no errant signals from pressing multiple sensors. Ghosting only occurs in the row and column configuration. If every sensor has a corresponding pin on a microcontroller, then it is a non-issue.

5.4.5 Single Sided Prototype With Diodes:

Three copies of this sensor prototype were created, and using a single Arduino on each, I^2C was successfully used to read all three at once. I^2C is discussed more in Section 6.4.3.

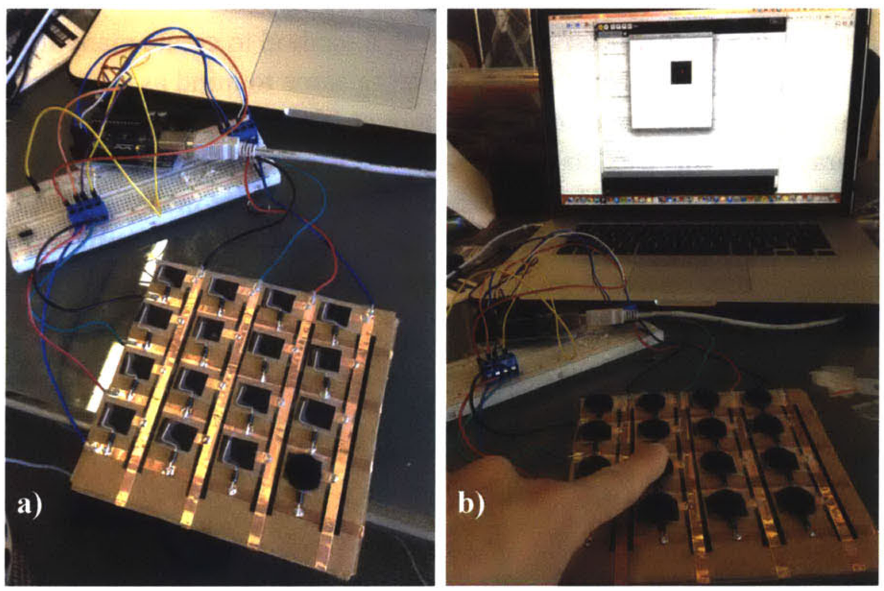


Figure 41: a) One tile with diodes for ghosting mitigation. b) Sensor output visualized in Processing.

5.4.6 Larger Sensors and Different Fabrication Processes:

Fabricating sensor matrices by hand was not a long term solution, so different manufacturing and fabrication methods were tested in creating the following prototype. Sensor traces were drawn as a Bitmap file and a Roland vinyl cutter was used cut them out of copper tape. Each trace was then connected to diodes to eliminate ghosting.

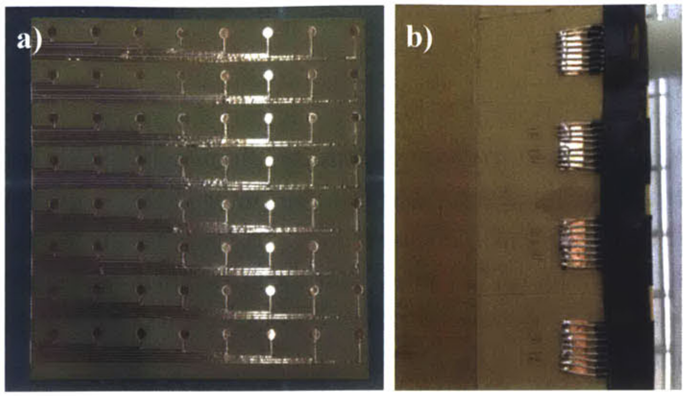


Figure 42: a) Vinyl-cut traces. b) Diodes connected to every sensor trace to prevent ghosting.

5.4.7 Spray Painting Sensors:

Placing copper tape was a time-intensive process, so a method to create many traces quickly and accurately was desired. Stenciling with spray-painting came to mind as a way to replicate a specific pattern and cover large areas in a short amount of time. A masking pattern was laser cut and grounding spray paint was used to create conductive traces. The final result was messy and had poor electrical characteristics. The resistance of the paint was incredibly high so it was difficult for electricity to flow. If the paint layer were thicker and more uniform, it might be better. Spray painting circuitry could be a viable method of creating sensors, but the variability in the process needs to be worked out.

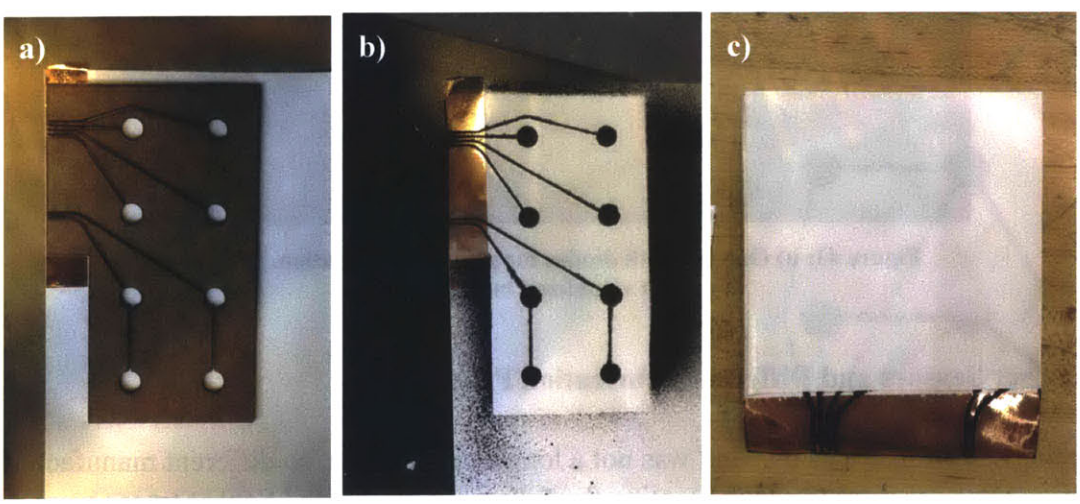


Figure 43: a) Laser-cut masking pattern. b) Spray painted sensor traces. c) Folded sheet to create rows and columns.

5.5 Member's Week 2014 Prototype (MW2014):

5.5.1 Overview:

During the design process we were made aware of a company called Elsi Technologies in Finland (see Section 4.3.2); they had developed a sensing floor to monitor the elderly in nursing homes. They used capacitive sensing rather than pressure, but what was really appealing was their sensor configuration. They made strips instead of using tiles; rather than having a sensor with 25 nodes arranged 5 x 5, there is just one long strip of 1 x 25. This configuration was used as a model for the Member's Week prototype.

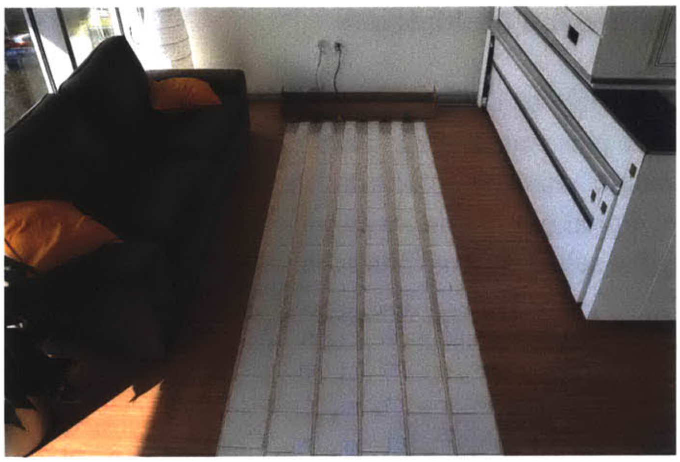


Figure 44: MW2014 deployed in the CityHome (Section 2.2).

MW2014 is 9' x 4' and contains 96 sensing points. The resolution is about 6" x 6" so the system is able to capture unique footsteps. Cost of parts including the Arduino was $200, meaning that for about $3500 the entire CityHome floor (200ft^2) could be made into a sensing surface. While that amount may seem high, the cost reduces with bulk. Components are also cheaper than the breakout boards commonly sold i.e. an ATmega chip costs far less than the Arduino UNO.

5.5.2 Circuitry of MW2014:

Shift Register:

A shift register (SR) is an integrated circuit that allows additional inputs or outputs to be added to a microcontroller. The shift register does this by converting data between serial and parallel formats. The microprocessor communicates in serial, and the SR takes that data and outputs information in parallel [56]. There are two main types of shift registers, named for how they handle data. The SIPO, Serial-In-Parallel-Out shift register takes in information as serial and outputs data in parallel. This is the type of shift register my circuitry uses. The second type is the opposite, PISO, where data comes in parallel and is output in serial. SIPO SRs are good for controlling a large number of outputs, such as turning on or off many LEDs [56]. Another added benefit of the SR is that they can be daisy-chained to multiply the number of additional pins available to a microcontroller.

Multiplexer:

A multiplexer (MUX) is an integrated circuit that acts like a rotary switch for signals. It selects one or several analog or digital input signals and forwards the selected input into a single serial line. A MUX with 2^n inputs requires n select lines which are used to select which channel to read. For example, a 16 channel MUX requires 4 select pins which cover 0000 to 1111, totaling 16 individual select channels [59].

Combining A Shift Register and Multiplexer:

Because multiple signals would be collected by a single Arduino, a method to capture all the data without using up all of the pins was needed. To achieve this goal, the SR and MUX's abilities were combined into a single board. The shift registers would allow for a high number of output pins requiring only five digital pins on the Arduino to control them. The shift registers would be responsible for sending the select channel bits to the MUX to indicate which signal to read.

Breakout Board Connector:

Before designing an integrated PCB, the SR/MUX scheme was tested on its ability to collect the required data. SparkFun offers two convenient breakout boards for the 74HC595 Shift Register and the CD74HC4067 Multiplexer. These breakout boards connect the pads on the IC chip with header pin positions (0.1" pitch) that can fit breadboards and other prototyping environments. A PCB cut pattern was designed in SolidWorks that when routed would provide anchor locations for each board and would connect the required pins between the SR and MUX. The board would also provide header pins for connecting sequential shift registers, routing power, ground and signal, and one pin for every input channel on the MUX. The board was cut on a Roland Modela CNC machine and the Sparkfun breakout boards were held in place with female headers soldered to the CNC machined board. Male header pins were then soldered on to complete the assembly. Six boards were created, which would be used to control six strips of sixteen sensors each.

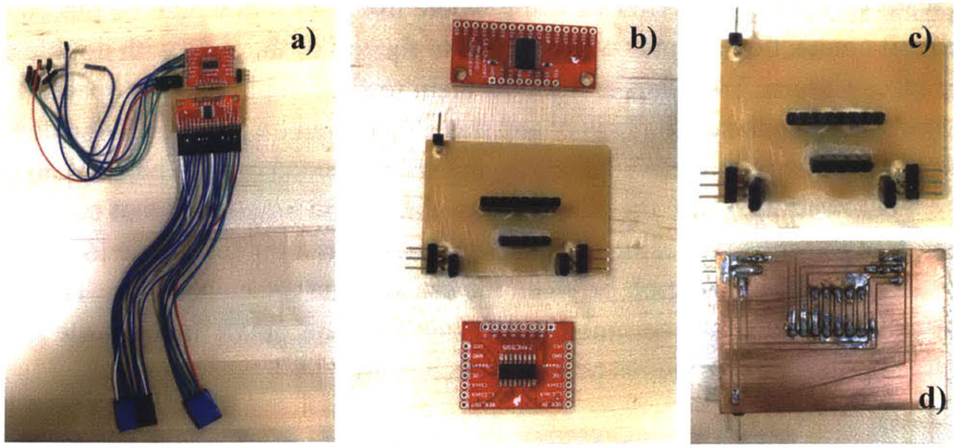


Figure 45: a) Completed Connector Board with SR and MUX breakout boards. b) MUX, connector board, and SR. c) Front side of connector board. d) Back side (routed) of connector board.

5.5.3 Hardware - Sensor Sheets:

As mentioned, the inspiration for this newest prototype was the Elsi Technologies sensing sheets (see Section 4.3.2). The sheets only contained the electronic traces for the capacitive pads, and all circuitry and electronics were at one end of the sheet. In this way, the profile of the sensor was as flat as possible, making it easy to fit under carpeting, flooring etc. Similarly, the pressure sensing sheets were designed such that the FSRs would be on the sheet and the circuitry would be placed at the far end.

Substrate:

Thick art paper 0.02" thick served as the base, as it was thicker than conventional poster paper. Since it was paper, it was easy to trim to size, tape and mark up for prototyping purposes. The dimensions of each strip were determined by the desired sensor-to-sensor dimension. Since there were sixteen sensors (one per channel on each MUX) spanning about ten feet, the distance between each sensor was about 7". In order for the sensors to form a square grid, the thickness of each strip had to be about 7" as well. Since each piece of art paper was about 32" long and 24" wide, each had to be cut into three separate strips, placed end to end and taped along the horizontal seams until the desired length was reached. Locations for sensors were marked with pencil.

Traces:

Because copper tape had worked so well and it was easy to place and cut, it was also used for this prototype. Rather than 1/2", 1/8" thick tape was used instead to conserve space and because thicker tape provided no additional electrical performance benefits. The size of the sensor square was chosen based on the size of the largest hobbyist FSR sold which was a square 1.75" x 1.75". To increase the amount of force the sensor could tolerate before saturating, the sensor dimensions were set to 3" x 3". After marking where these sensors would be placed along the length the circuit connection adaptor was adhered to the top of the strip with double sided tape. This adaptor takes sixteen, 1/8" pads and routes them to sixteen pin headers spaced 0.1" apart to fit into conventional breadboards. This board was also routed on a Roland Modela CNC router. Since the copper tape has conductive adhesive, the strips needed only to be stuck to the copper pads to make a connection. Once the adaptor was placed, the long lengths of copper were placed down the length of the substrate using the adaptor spacing as a guide. These long lengths of copper tape served as the connection between the sensors and the electronics. There are sixteen traces on each sheet and a single common power line that runs down the opposite side of the sheet. This trace provides continuous power to each sensor.

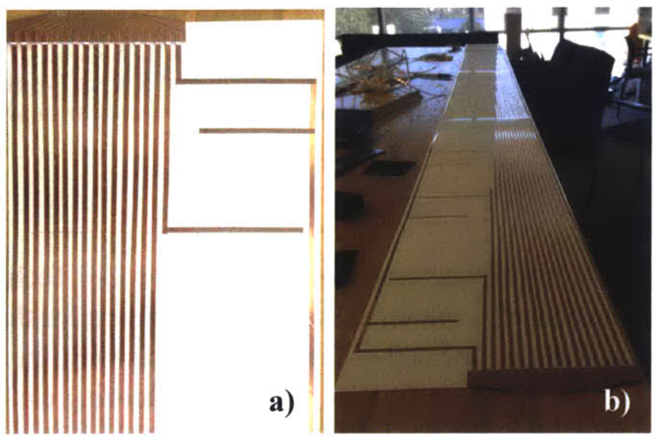


Figure 46: a) Close-up of the circuit connection adaptor and sheet traces. b) View of one sensing strip.

Pads:

Once the vertical traces were placed, the next step was to place the breakouts to each sensor and terminate the breakouts with large pads to increase the surface area on the FSR material. Larger solid pads were chosen over an the interlocking fingers configuration for the sake of time. It would have taken longer to place all of the small fingers, and since density of the sensors was already so low, the added sensitivity would not have contributed much to the overall system.

Figure 47: One sensing strip with pads at each FSR location.

5.5.4 FSR Construction:

With the traces placed, the next step was to create the FSR on top of the pads. This was done by adhering a 3" square of Velostat on top of the pads with scotch tape. Next, a square of the same art paper was cut and placed on top of the Velostat with double-sided tape to provide more uniform compression to the Velostat when pressed.

Figure 48: a) View of the six-strip prototype assembly. Note half the FSRs do not have a square of paper yet. b) View of the electronics connected to the strips.

Using female-to-female jumper wires, each strip was connected to its corresponding SR/MUX board. Each of the SR/MUX boards were also connected together with jumper cables, and finally an Arduino was connected to the end of the chain. A voltage divider with a 10kΩ resistor was placed between the signal line and ground to make the sensor values readable. The Arduino then plugged into the computer via USB.

5.5.5 Reading the Sensor Values:

As described in previous sections, the sensing system consists of an array of FSRs, whose output gets read as a ten bit analog signal (0-1023). At the end of all of the operations described below, the microcontroller simply does an analog read to receive the signal. The more complex part is selecting which signal to read. Since the signals all travel along a common signal line, it is necessary to ensure that other sources of signals are eliminated while one sensor is being read. For this description, the Arduino receiving the signal and connected to the computer is called the Master, while each SR/MUX assembly a slave.

Sensor Read Scheme:

For every slave, in this case six, the master sends a signal to tell each SR on the slaves to turn their respective MUXs off. This ensures that no stray signal is collected when one begins to read. Next, for every channel on the MUXs, in this case sixteen, the Arduino tells the shift register to write the required 4-bit code, from 0000 to 1111 (0 to16) to the MUX to select the specific channel. Next the Arduino reads the analog value and writes it to the Serial Port. This process repeats six times, one for every slave module.

At the end of the operations, there is an array of 16*6 = 96 analog values in the Serial Monitor representing all of the values in the assembly of 6 strips. These values are then ready to be read by either MATLAB or Processing for analysis or visualization respectively.

Sensor Data Output:

Once the hardware, electronics, and software were functioning properly, the prototype mat was set up in the CityHome, to get a feel for how a smart floor might function within a smart apartment space. Using 2' x 2' carpet squares, the mat was covered to not only protect the traces and sensors but also provide additional force distribution.

Figure 49: View of the carpeted prototype in the CityHome living room.

5.5.6 Visualizations:

To understand the data collection abilities of the system, three visualizations were programmed using Processing to help clarify what the mat was seeing and to explore the potential for different programmable applications. Processing is able to connect to the Arduino via the Serial Port and read in values that the Arduino writes.

Threshold:

For this visualization, processing first reads in the 96 sensor values as a single comma-separated array. It then transforms this array into a matrix representing the coordinate position of each sensor that corresponds to the value in the array. It then maps the analog value to a grayscale value of 0-255 and colors in the square that value. If the sensor value is past a certain threshold (in this case 50), the color would turn red, indicating the presence of a person or heavy weight.

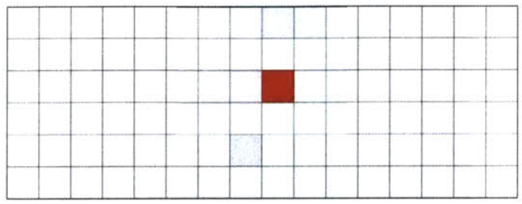


Figure 50: Raw data and threshold visualization.

Path Plot:

Once position of the person was determined, the next step was to visualize their walking history. So additional code was added to place a yellow point and connect it to a line for every location the person occupies. After a few steps a path begins to take shape showing where the person began, where they walked, and where they are currently standing.

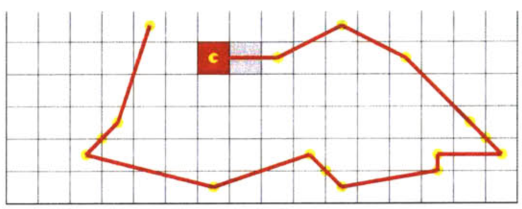


Figure 51: Path plot visualization

Heat Map:

The final visualization was a heat map to show which tiles were most frequently stepped on. Every time a tile was activated, its color incremented forward on a scale from blue to red, with all shades in between. A higher concentration of green/orange/red values indicates more walking has occurred in that area.

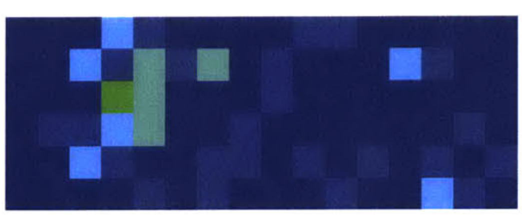


Figure 52: Heat map visualization. Green areas indicate a higher frequency of walking in that location.

Projection Overlay:

This final experiment involved adding enhanced interaction to the floor. Using a digital projector, interfaces and outputs were projected to scale directly onto the floor. As people walked around, the projected output on the floor matched the actual sensor activity of the system. For the experiment, the threshold visualization was projected onto the mat.

Figure 53: Projector overlay on the carpeted prototype. The red square shows where a sensor is currently being activated.

5.6 Member's Week 2014 Demo:

For Member's Week the system was set up in the Changing Places Group area. A large sheet of PVC served as the base and both the sensor sheets and carpet squares were adhered to the base material with carpet tacks. This way the sensor sheets did not slide around as people walked. The system was oriented perpendicular to a large white wall so that the outputs could be projected onto the wall for people to see as they walked along the mat.

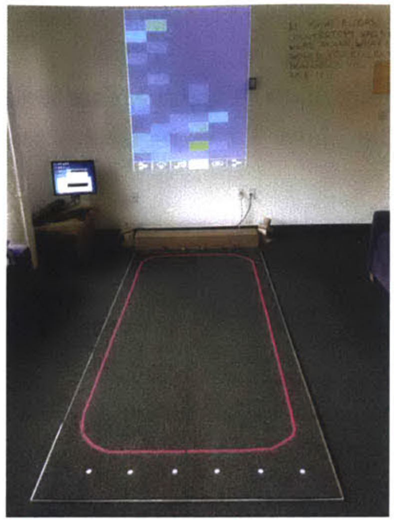


Figure 54: Member's Week 2014 demo space. The prototype was carpeted and the output was projected against the wall.

Two features were added to the Member's Week demo. The first was a pink line running near the edge of the system. The second are six white dots at the base of the mat (see Figure 54 above). The line serves as a path for a line following robot. This was a previous experiment to assess whether the system would be able to sense lighter objects such as mobile robots on the scale of a Roomba. Autonomous robots or actuated furniture within a space could be informed by the data coming from the floor or other surfaces.

5.6.1 Modes:

Six modes were programmed into the Member's Week demo. The purpose of each demo was to demonstrate the potential for use of the system for various applications ranging from healthcare to entertainment. Each was accessible by pressing one of the six dots, which were actually circles of Velcro positioned over six sensors at the base of the mat. These modes are raw data mode, light mode, music mode, heat map mode, orbit mode, and path plot mode.

Figure 55: Icons representing each mode the system could run; raw data, light, music, heat map, orbit, and path plot.

Raw Data Mode:

Raw data mode was simply a visualization of the raw data being received by Processing, with the analog values mapped to gray scale. Low forces registered as whiter, while higher forces tended towards black.

Figure 56: Raw data mode.

Light Mode:

To demonstrate the potential of such a system being paired with other intelligent devices within a home such as robotic furniture or appliances, RGB lights were programmed to change color depending on where a person was standing. One might imagine turning on lights by double tapping the floor and sliding a foot left or right to change the hue or brightness.

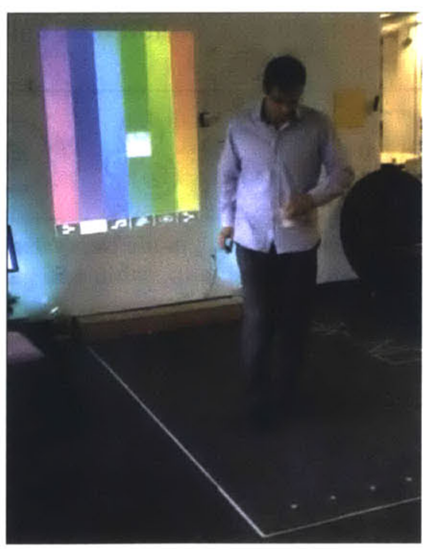


Figure 57: Light mode.

Music Mode:

To demonstrate the educational and creative uses for such a system, a music generating application was programmed that would play notes along a blues scale depending on where along the length of the mat someone was standing. Because the signal used was unprocessed, there was a fair amount of noise, so playing a normal scale would have created too much dissonance. A blues scale worked nicely because multiple notes could overlap and still sound pleasant. Future designs could replicate a piano keyboard interface or even stringed instruments using the matrix of sensors as the frets and strings.

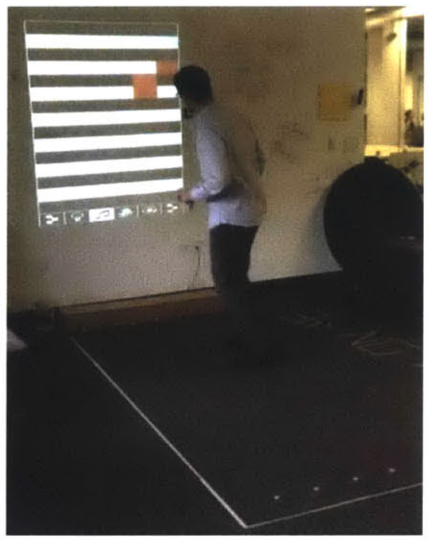


Figure 58: Music mode.

Heat Map Mode:

The heat map visualization is intended to show which tiles are stepped on the most. This application demonstrates the potential to use the system as a useful data input device. Gathering analytics about walking trends could have direct applications in retail, event planning and organization, and even manufacturing or logistics.

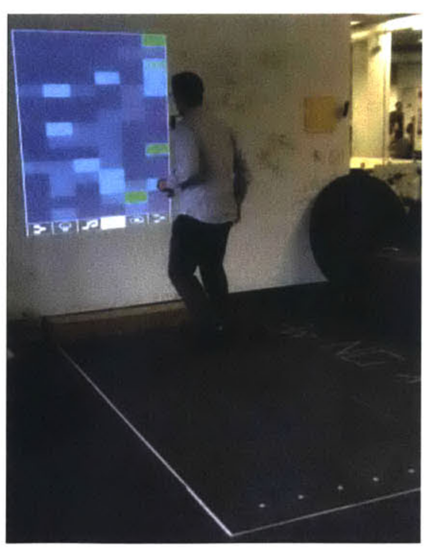


Figure 59: Heat map mode.

Orbit Mode:

Orbit mode's function was to show that the system could also be utilized for entertainment and fun. In this application anywhere the threshold is passed will instantaneously become a point of high gravity. Three circles that are affected by the gravity suddenly get pulled to the location and begin to orbit the point. Once the person moves, the circles follow and begin to orbit the other point. Dynamically walking around the mat creates the effect of having these circles following you wherever you walk.

Figure 60: Orbit mode.

Path Plot Mode:

Expanding upon the first path plot mode visualization, this mode also shows the pressure readings at each location in addition to the active tile and path walked around the mat.

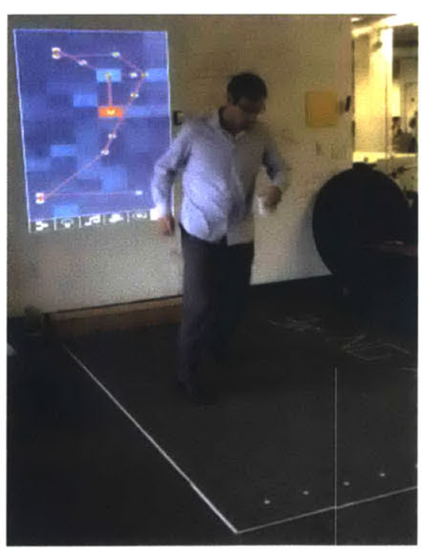


Figure 61: Path plot mode.

5.6.2 Closing Remarks

Having a system monitoring the path people take around their homes and how they behave at every stop along the way (i.e. kitchen, TV room, bedroom etc.) could give very rich and insightful information into the person's patterns of activities, preferences, and behaviors. This information can then be fed back into the system to optimize the experience for the person within the space. Just like the Nest thermostat learns its user's preferred temperature settings, a floor or surface based system could capture activity patterns and transform the furniture or ambiance according to what it thinks the user's intended next activity will be; thus creating informed home automation. As applied to elderly or disabled users (see Section 3.2), this system could indicate degree of mobility, activity, or cognitive state.

6. Thesis Prototype:

6.1 Overview:

After the functional success of the Member's Week prototype, the next step was to design a custom sensing surface and PCB that could be manufactured. Inspired by the design of Nan-Wei's capacitive floor prototype, conductive inkjet printing was chosen as the method to produce the traces of the FSR grid [40]. Also inspired by previous work of the Responsive Environments Group and a desire to minimize number of separate parts, the sensing sheet was designed to plug directly into a PCB using flat flex cable (FFC) connectors.

6.2 Technical Overview of the Conductive Inkjet Printing Process:

CIT Technology, who did the fabrication of the final sensor traces employs a process to print solid copper circuitry directly from digital files. The process first prints catalytic ink on a plastic substrate and then immerses the printed pattern in an electroless plating solution to create a layer of solid metal onto the pattern. As opposed to printed silver, this method which creates copper circuitry, can be directly soldered-to [60].

The process is roll to roll, so it is very appealing from a mass manufacturing point of view. The catalytic ink printer head has 1,000 nozzles, providing 360 DPI resolution. UV curing sets the ink as it moves through the printer.

The smallest track spacing possible is 3 pixels at 360 DPI and the smallest feature size is 220 microns (2 pixels at 360 DPI). The pattern for the final prototype, which included a ribbon cable section, required this level of resolution to fit all the sensor traces into a specific area.

All that is required is a Bitmap or DXF file of the pattern that needs to be printed. The one limitation is that the substrate material is 300mm and the available printable area is about 282mm. This dimensional constraint drove the final design and configuration of the prototype.

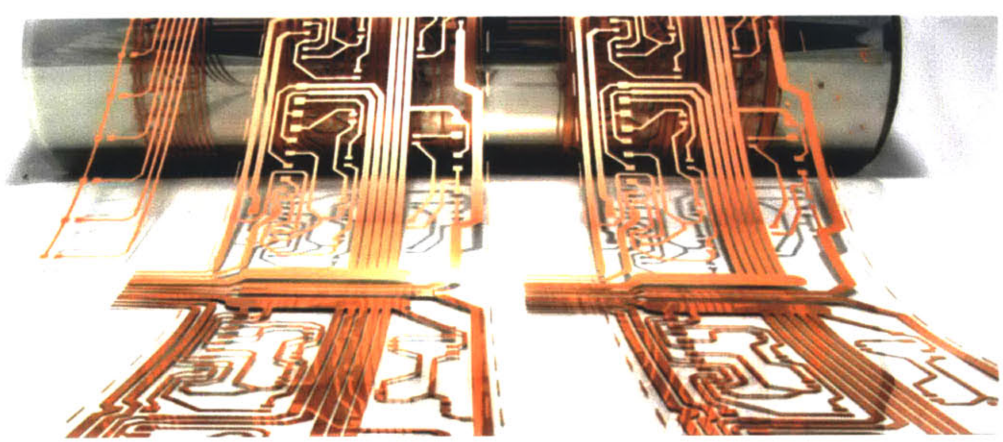


Figure 62: Example of inkjet printed electrical circuitry on a clear substrate. Image Source: CIT Technology

6.3 Conductive Inkjet Printing Experiments:

To understand how inkjet printing would influence the design of the FSR, a desktop printer hacked to print conductive ink was used to print out sample traces. The first set of sensor traces was designed to plug into a ten-channel JST FFC Connector with a pitch of 0.1" [79]. This meant that the circuitry from the previous prototype could be used to test the first inkjet printed sensors. The interlocking finger design was used for the pads and the sensors followed the same layout scheme used for the members week prototype. Each column consisted of sixteen sensors. Sixteen traces would go from each sensor pad to a channel on the multiplexer. A common power would connect all sensors and provide constant power to the sensors until their values are read. Unfortunately, the ink had a very high resistance and did not work well when connected to circuitry. Still, it was good to get a feel for how the sensors would appear, and this assisted in the sizing and creation the files used for the CIT Manufacturing Process [60].

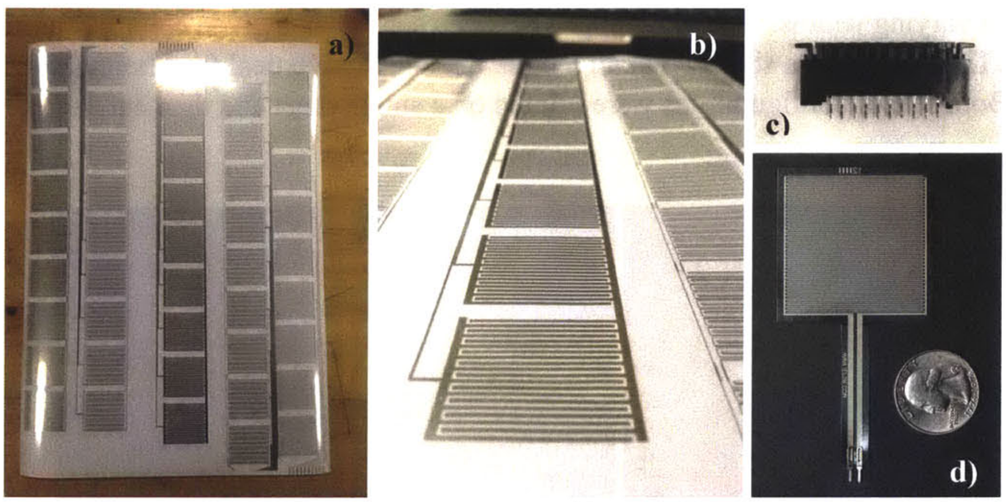


Figure 63: a) Page showing different print patterns and settings. b) Close up of printed pattern with interlocking fingers. c) JST FFC connector with 0.1" pitch pins. d) Off-the-shelf Interlink square FSR. Image Source: c) JST, d) Adafruit Industries

6.3.1 First Set of Inkjet Prints:

Using a combination of SolidWorks and Illustrator, two different versions of a sensing array were designed and sent to get printed; one with a square grid and one with a rectangular grid.

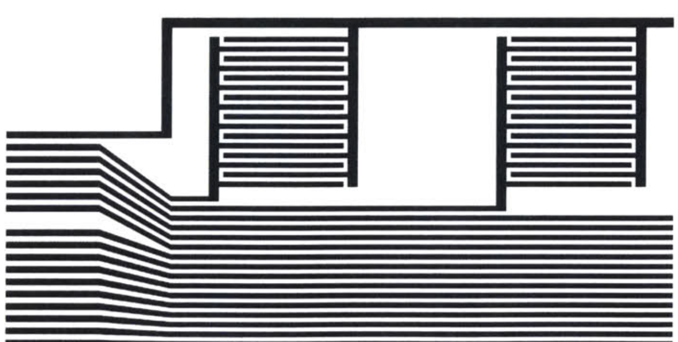


Figure 64:Close up of the traces and sensor pads for the first inkjet printed sheet.

Square Grid:

This sensor was designed to have nearly even spacing in the horizontal and vertical dimensions. Because of the layout, more sensors meant more traces which caused the spacing between columns to grow wider. Since sixteen sensors were required in each column and 1mm traces and gaps to ensure that there were minimal resistive losses over the length of the sheet, there was only room for four sensors across the material. The choice of 1mm traces was only done to ensure sources of error would be minimized for the purposes of debugging. Resistive losses over the length of the sheet were an unknown, so the traces were purposely made thicker.

Rectangular Grid:

This sensor configuration added another column by reducing the size of the traces/spaces to 0.5mm. This would also provide a good point of comparison to calculate the actual resistance of traces with varying widths. The vertical gap between sensors in a given column was also reduced. The effect was a shorter overall length, but a higher vertical (and horizontal) density of sensors. Both sensor configurations were printed on a single sheet and then cut to size upon arrival.

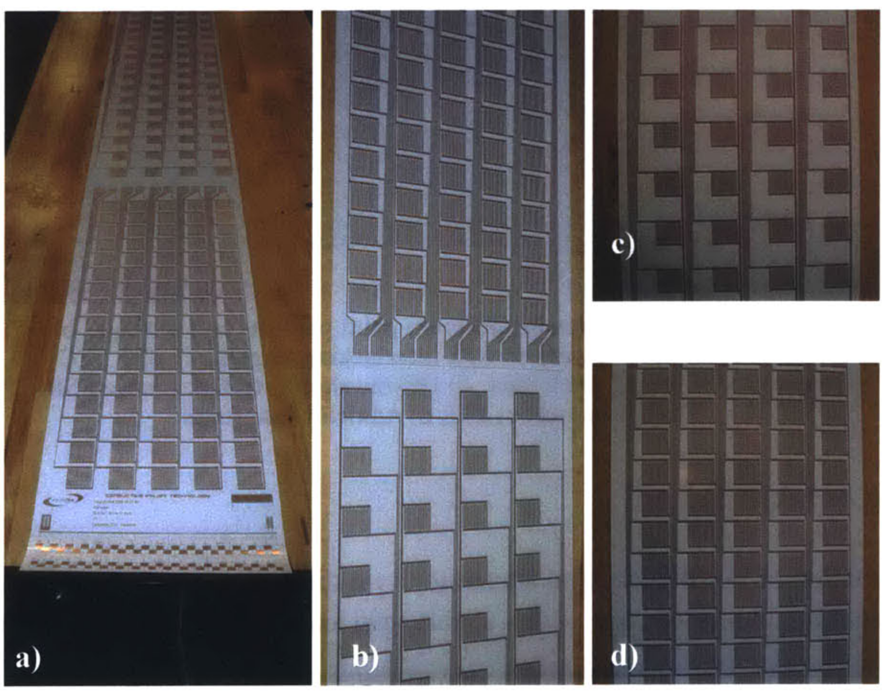


Figure 65: a) Entire inkjet printed sheet with both Square Grid and Rectangular Grid sheets. b) Close up of the Square and Rectangular Grid patterns. c) Square Grid. d) Rectangular Grid

6.3.2 Characterizing the Resistance:

Using a voltmeter, the resistance between two points along the longest trace was recorded. The distance between points was increased by 20cm until the end of the trace was reached. The results are tabulated below:

Table 1: Resistance vs. Length For 1mm Traces

Length (cm)	Resistance (Ohms)
20	5.1
40	10.2
60	15.27
80	20.35
94	24.13

Observing the results, the resistance per length can be approximated as 0.25Ω/cm

Table 2: Resistance vs. Length For 0.5mm Traces

Length (cm)	Resistance (Ohms)
20	9.45
40	19.45
58	28.81

Observing the results, the resistance per length can be approximated as 0.5 Ω/cm

The eventual thickness of the traces would be about 0.2mm, which would cause the resistance to be on the order of 1Ω/cm. The longest trace would be no greater than 2m (200cm) meaning that the resistance of the longest trace would be on the order of 200Ω, which is an acceptable amount of resistance for the circuitry.

6.3.3 Assembling the Square Grid Sensor:

The first step in creating the sensor was connecting the sheets to the existing circuitry. The square and rectangular grid sheets were first cut out of the main inkjet printed sheet. Tabs were cut at the end of each sheet to accommodate the FFC connectors. Two FFC connectors were required for each column, because the maximum number of positions was ten.

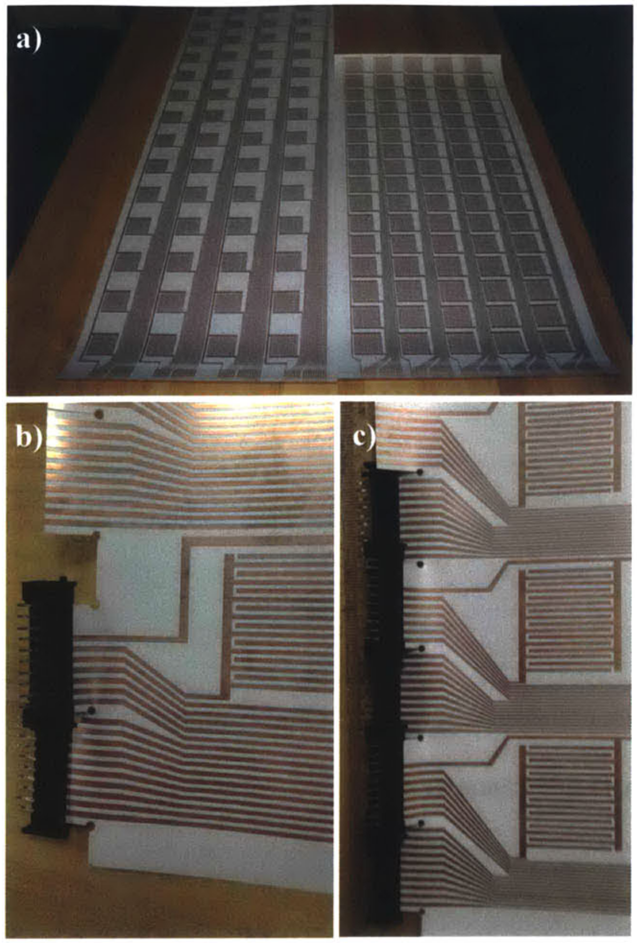


Figure 66: a) Square and Rectangular Grid sheets cut from the main sheet. b) Tabs cut in Square Grid sheet and FFC connectors placed. c) Tabs cut in Rectangular Grid sheet and FFC connectors placed.

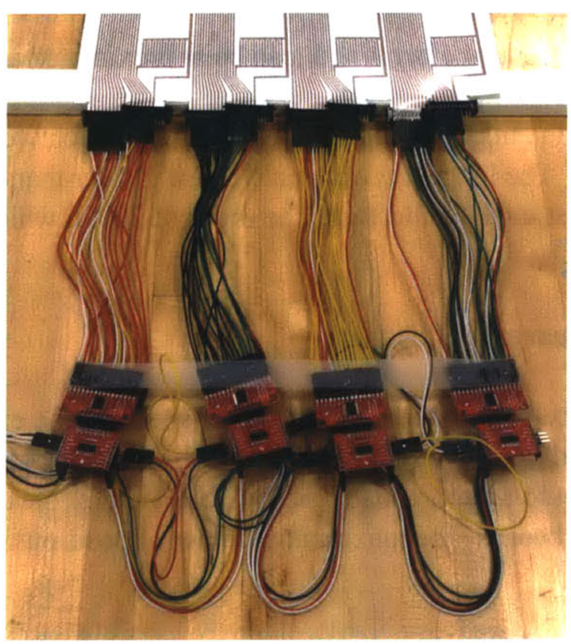


Figure 67: Square Grid sheet with FFC connectors plugged into the electronics from the Member's Week prototype.

Masking the Sensors:

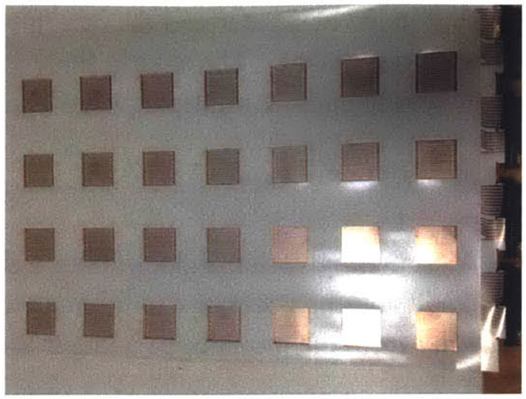


Figure 68: Square Grid sheet with paper masking.

To prevent shorting between adjacent traces, a mask needed to be applied to everything on the inkjet sheet besides the square sensor pads. Using a laser cutter, a masking layer was cut out of the art paper used in the previous prototype. This was placed over the inkjet printed sheet to allow access to only the FSR pads. The mask was first tested on the Square Grid sensor, but once verified for functionality, masks were cut for the Rectangular Grid sheet as well.

Testing the Inkjet Printed Sensors:

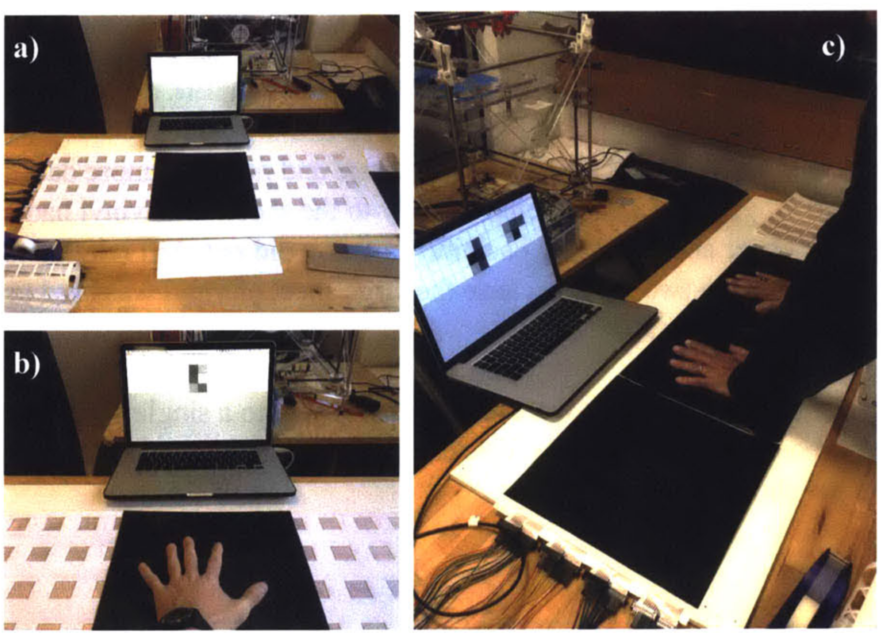


Figure 69: a) Square Grid sensor connected to the computer. b) Sensor activated and the output visualized on the computer. c) Testing two hands on the sensor.

After the masking layer was cut and firmly attached to the top of the inkjet sheet with tape, 1' x 1' squares of Velostat were placed on top of the masking layer to complete the FSR assembly. The master Arduino was connected to the computer and the raw data visualization Processing code was run to see how the system responded. The analog force values were visualized on the computer successfully and it was able to capture multiple points and quick actions.

6.3.4 Assembling the Rectangular Grid Sensor:

With the inkjet printed sensor sheet running with the electronics, a single completed sheet was assembled. This was to understand the steps necessary to fabricate one of the sensor sheet assemblies, which would eventually form the modules of a larger sensing system.

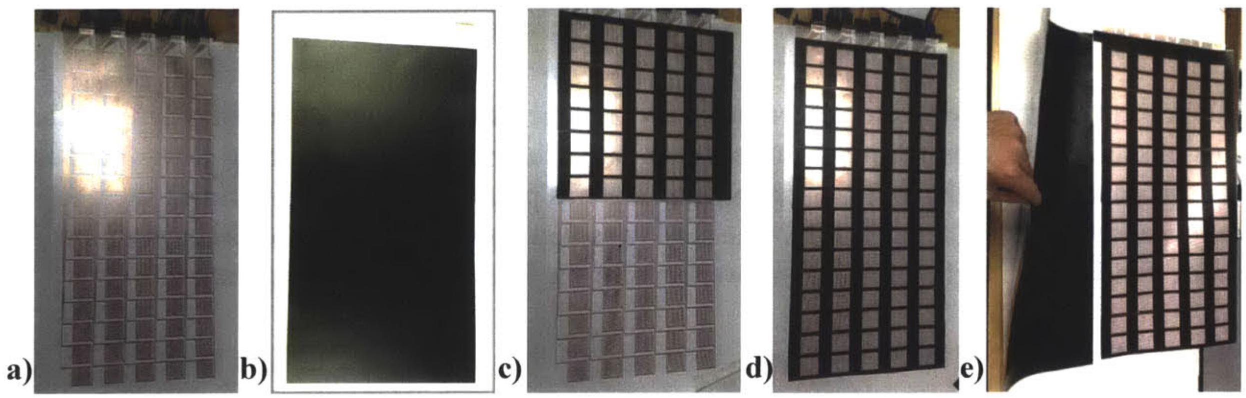


Figure 70: a) Rectangular Grid sheet. b) Large Velostat sheet. c) Half of the mask applied to the sheet. d) Complete mask applied. e) Velostat attached along the edge of the sheet. Once folded, the FSR matrix is complete.

The Rectangular Grid sheet was used as it was more dense and provided more data. A vinyl mask cut on a Roland vinyl cutter was adhered directly to the sheet. The vinyl had a lower profile than the poster paper and the sticky back meant that it did not shift relative to the traces. The mask had to be applied in two parts because the large sheet of vinyl was hard to peel and place at the same time.

Next, two sheets of the 1' x 1' Velostat were stuck side-by-side to a single sheet of vinyl. The vinyl served two purposes; first, to hold the Velostat in place and the second to provide a protective layer to the sheet assembly.

Lamination:

Ultimately, the above assembly was held closed with tape but lamination was also investigated as a method to secure the layers in place. Lamination would not only hold all the layers together but it would provide waterproofing and durability to the system. Since the sensing sheet would have to be open on one side to accommodate the connectors, any attempts at lamination delaminated after a few days because not all sides were sealed. A more likely solution would be in the assembly process, where each layer receives a coating of adhesive and the layers are firmly rolled together with a harder base layer and durable top protective layer.

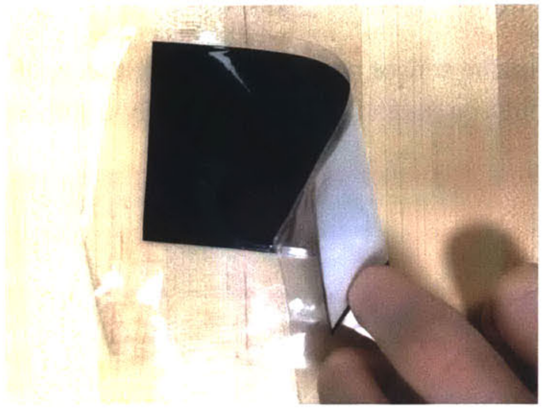


Figure 71: Laminated inkjet printed sensor and Velostat.

6.4 Final Prototype Hardware and Software:

6.4.1 Triangular Inkjet Printed Sensors:

After discussing the design of the sensor traces and pads with a colleague, he suggested that different shaped sensors should be investigated; particularly triangular sensors that overlap.

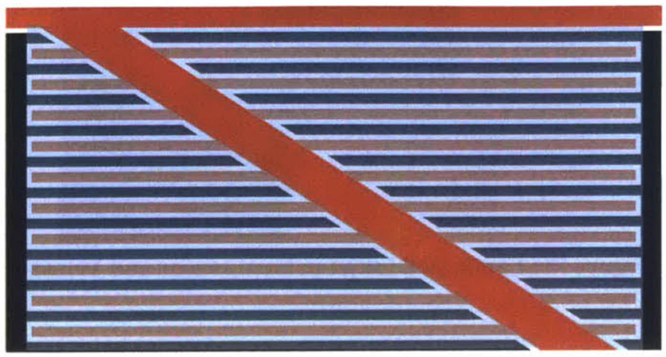


Figure 72: One cell of two triangular sensors. The red colored section is the common power. The black sections are what connect to multiplexer. The blue shading is the sensor area.

By sharing a common power, two adjacent sensors can sit side by side. Each "cell" would contain two sensors and occupy the same area as two square sensors, but each sensor would be twice as long. After some thinking and considering other designs, a triangular sensor configuration was chosen for its baseline functionality as well as added functionality in future applications.

The benefit of this type of configuration is that it can provide an additional dimension of sensing. In addition to the coordinate location in X and Y of an activated sensor, the ratio of activation of adjacent sensors can indicate where along the length of the cell the majority of the force is being applied. This interpolation provides a partial in-cell spatial awareness for the length dimension. Also, if the force occurs partially on one sensor and not its compliment, it would be possible to figure out specifically where the edge of the load ended.

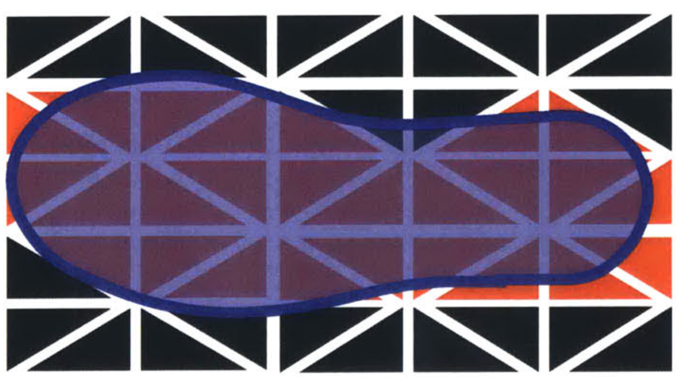


Figure 73: Sample section of triangular sensors being actuated by a foot. Note that within some cells of sensors, only one of the two is activated. This provides added accuracy to the output image.

The length of each strip to set to 2m, which was long enough to capture two or three footsteps of an average person walking. The sensor cells (containing two sensors) were 33.5mm x 64mm with a 2mm space between each cell in the vertical and horizontal. This size allowed 64 sensors to be placed in each column. Since each sheet is 282mm wide and each cell is 33.5mm wide, a maximum of four columns could be included on each sheet because the adjacent space was needed for the 64 traces.

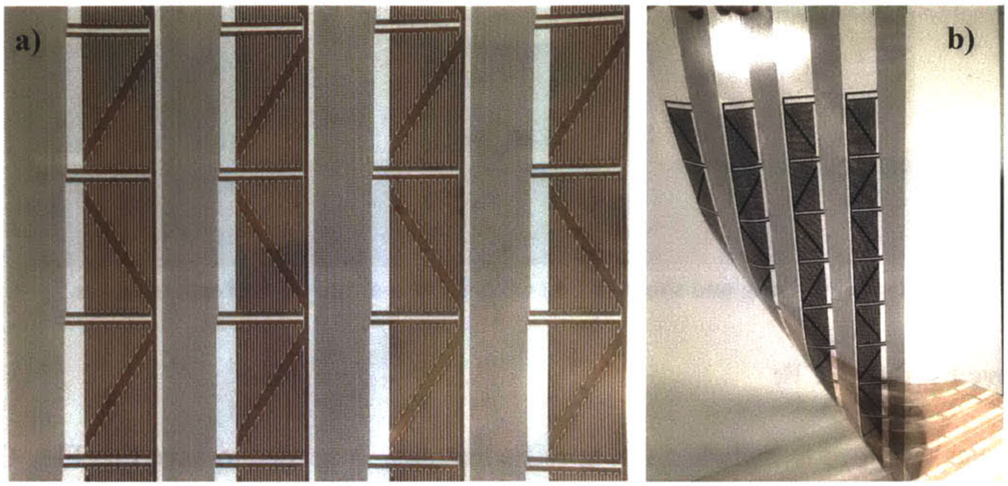


Figure 74: a) Close up of triangular sensor configuration. b) Alternate view of the inkjet printed triangular sensors. Note the long section of printed ribbon cable near the top of the image.

To solve this gap problem, each sensor module would include two sensing sheets, faced toward each other.

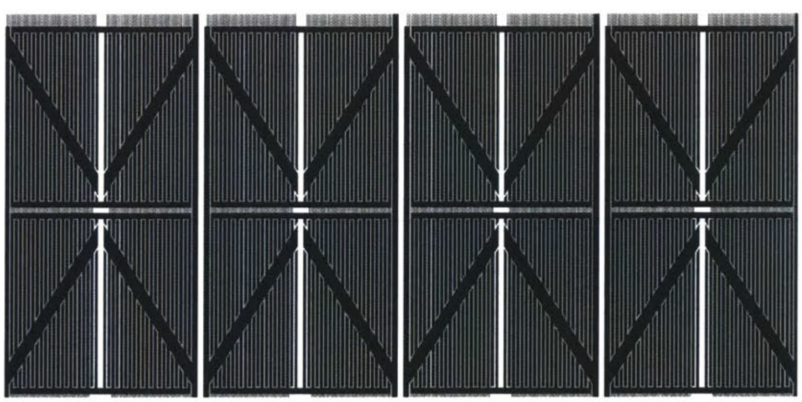


Figure 75: Overlapping sensor sheets demonstrating how the gaps of one sheet are filled by the sensors of the other.

The 64 traces were each 0.2mm thick with a 0.3mm gap between each. The ends of each set of traces would fit into a FFC connector with 68 channels and 0.5mm pitch (more on this in Section 6.4.2)). All of the traces fit within the width of a sensing cell (33.5mm) meaning that when the two sheets are faced towards each other, the cells would make an evenly repeating grid.

The traces on one set are longer than the other because of size limitations on the PCB. The PCB was designed to be about the width of the sheet assembly (300mm) and placing eight of the FFC connectors side by side would be longer than that length. Their placement had to be staggered by placing one row further away from the bottom of the board. To ensure that the sensors lined up when overlapped, the ribbon cable section of one sheet had to be longer to bridge the gap in the connector distance.

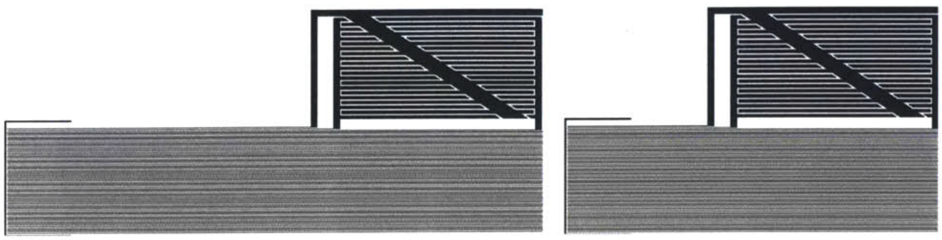


Figure 76: Long and short ribbon cable section of the printed sensor traces.

6.4.2 Slave Module PCB:

The PCB was designed to include a few specific features. It needed to have Flat Flex Cable (FFC) connectors for eight sets of traces representing two inkjet printed sheets. It needed its own microcontroller to handle the data collection and communicate with other PCBs in the assembly. It also needed all the shift registers and multiplexers necessary to collect the 512 signals per module. Five header pins were at either side of the board, and they carried power ground, SDA, SCL (see I²C section 6.4.3) , and a general signal (this was only used for debugging).

The schematic for a single board was created in Altium Designer [81]. It contained one Atmega328 microcontroller, 32 multiplexers, 8 FFC connectors and 8 shift registers in addition to some resistors capacitors and LEDs; see Appendix H for the PCB bill of materials and Appendix I for a close up of the schematic of the board.

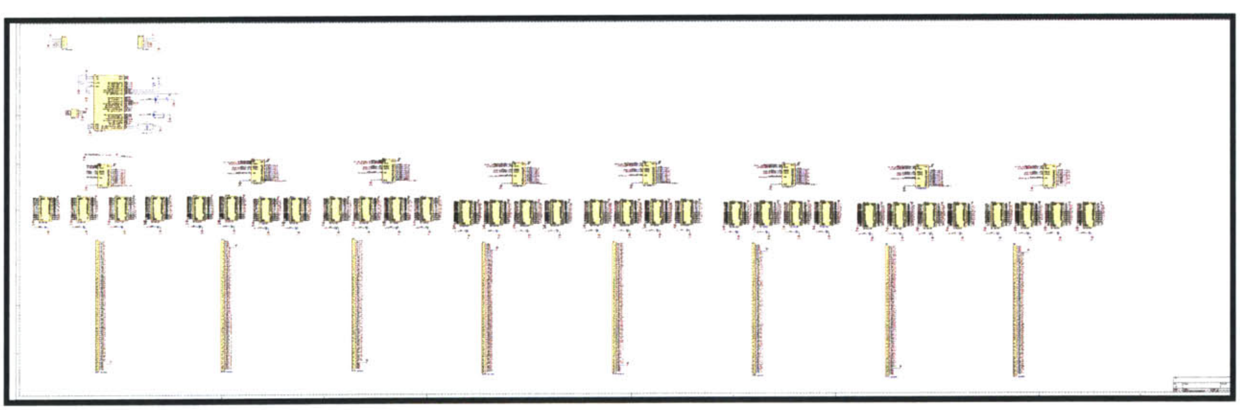


Figure 77: Schematic for a single slave module PCB.

A Freelancer, George Zarnescu in Romania helped to layout the board and generate files so that it could be manufactured. 4pcb in Aurora, Colorado fabricated and assembled the PCB and sent back ten completed boards.

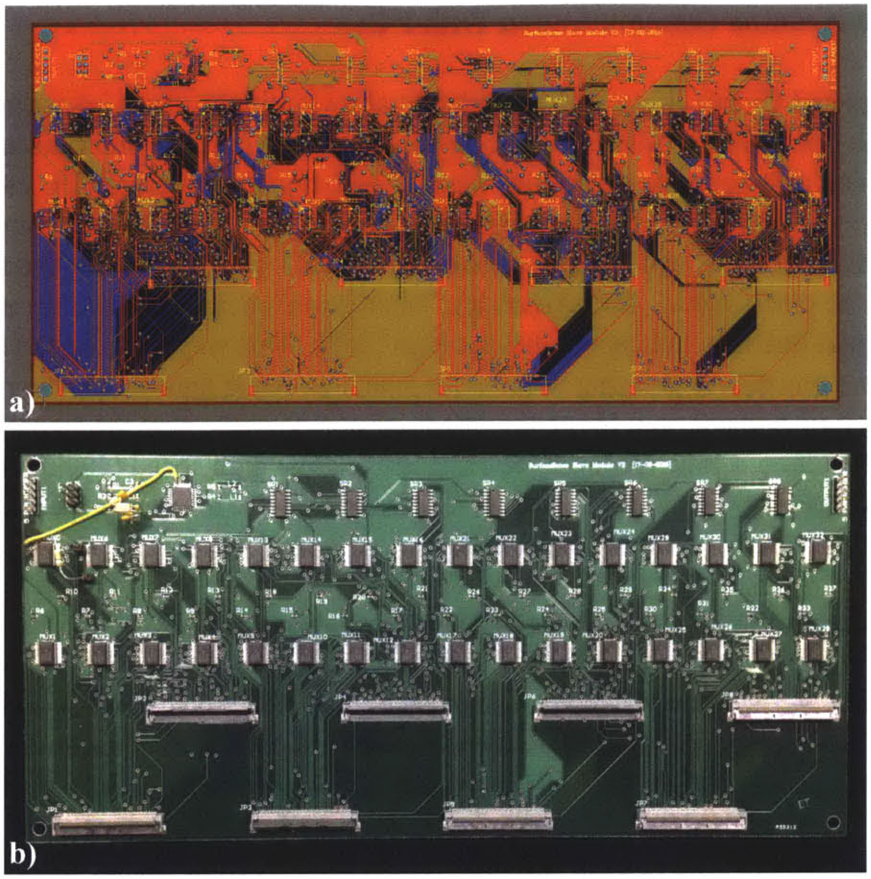


Figure 78: a) A rendering of the final three-layer board. b) Fabricated and assembled board.

The boards came out well but required two additional components before they could work. First, an oscillator crystal was soldered onto the board as they were out of stock at the fabrication facility. Second, the signal line of the multiplexers needed to be connected to the analog 0 channel on the Atmega (This connection was neglected in the schematic and therefore was not included in the fabrication).

Firmware was uploaded to the board using ISP (in system programmer) via a Arduino ISP board.

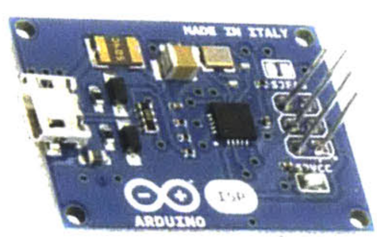


Figure 79: Arduino ISP board. Image Source: Amazon.

After uploading the firmware for the Arduino onto each board, they were tested by running the common blink.ino script, which would the built in LEDs. All were successful, demonstrating that the chip was wired and working properly.

6.4.3 Arduino-to-Arduino Communication With I^2C:

I^2C, or Inter-Integrated Circuit Protocol is intended to allow multiple "slave" devices/chips to communicate with one or more "master" devices/chips. It is intended for short distance communications and only requires two signal wires to exchange information. I^2C can support up to 1008 slave devices on the same bus and with multiple masters. Most devices communicate at 100kHz or 400kHz. Each I2C bus consists of two signals, SCL (clock) and SDA (data). The bus master generates the clock signal. I2C drivers are "open drain" meaning that "they pull the corresponding signal line low, but cannot drive it high" [68]. This way there can be no bus "contention" where one device is trying to drive the line high while another tries to pull it low. Each line has a pull up resistor on it (from the line to power) to make sure the signal is high when no device is pulling it low. Because the PCB is using the high speed (400kHz) protocol, 1.8kΩ resistors are used as the pull up resistors. Each slave device is given a specific address, which the master then uses to send and receive data.

6.4.4 Communication With an Assembly of Slave Modules:

As mentioned, SDA and SCL lines were built into each PCB. Each board serves as a slave and a single Arduino Uno serves as the master. Each board was assigned a unique address, which the master used to send and receive data. Each board was running a similar code to the Member's Week Prototype. Some of the loops were changed to account for more channels but it still just specifies what the registers send to each MUX to collect the data from channels 0-511. There are a few changes though that allow the code to be more efficient and communicate with the master Arduino. To reduce data overhead, the two byte analog value was mapped to a one byte value ranging from 1-254. The reasons for this range is that 0 and 255 are used as a case indicator, see below. The Slaves continuously collect the data until queried by the Master.

To get the data from each board the master first sends a request to each asking for one byte. There are two possible outcomes for this first request:

Case 1: no activated sensors
On the slave board, if after collecting the 512 sensor values, none are above a threshold value (in this case it was 50) then the slave sends a 0 in response to this first request.

Case 2: activated sensors
If after collecting the 512 sensor values, more than one are above 50, then the slave sends 255.

If the master receives a 0, then it knows that the specific board didn't have any activity, so it prints 512 ones to the Serial Monitor and moves on to the next slave board. If instead the master receives a 255, then it knows that the slave has data to send. The master then executes a loop that requests one byte from the specific slave 512 times. In response to each request, the slave sends the value in the sensor value array corresponding to the request number. Every loop the slave module increments the index. In this way, the master writes the 512 values from that slave one at a time to the Serial Monitor:

The master simply loops through every slave in order; 1, 2, 3, 1, 2, 3 etc. to create a continuous update of each board. This continuous method is good for visualizing activity in Processing, but for analyzing the data, the master code is slightly modified so that an external program such as MATLAB can send either a 1, 2, or 3 via Serial which tells the master which slave to request data from. The code for the slave and master modules can be found in Appendix A.

6.4.5 Sensor Sheet Assembly:

The inkjet printed sheets required a bit of finishing before they were incorporated into the sensor. Two versions were tested. The first was printed on a 100 μm thick white substrate. The second which made up the final sensor assembly was printed on 125 μm thick clear substrate. The second version was chosen because the thicker material felt more robust which was needed for the system to withstand the walking tests.

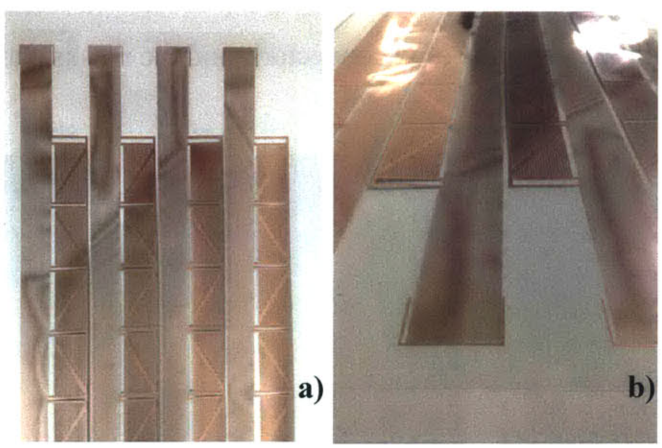


Figure 80: a) View of inkjet printed sheet upon arrival. b) Close up of guide border used to cut the flat flex cable section to the proper size.

Once the printed sensors arrived, traces needed to be cut to the size of the flat flex cable. In the inkjet file, a boarder around the traces was incorporated to indicate where to cut. One problem encountered was the fact that the FFC connectors are unidirectional. This meant that the ribbon cable must be placed face down (copper side down) to contact the pads. This was not a problem for the top printed sheet but the lower needed the traces to be flipped over. This flip was accomplished by cutting a 1cm section off the end, applying Z-conductive tape and then laying it back over the traces. This maintained the dimension of the ribbon section and allowed for the lower sheet to face up.

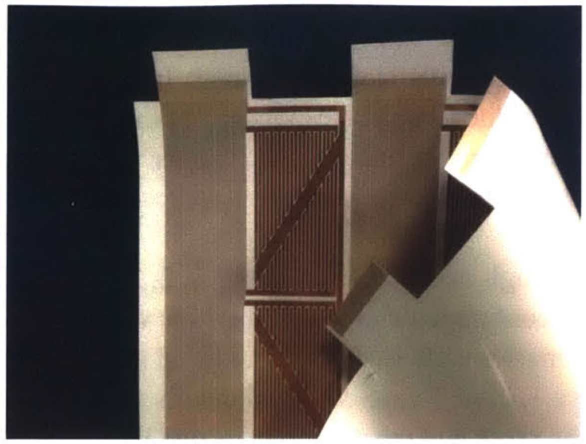


Figure 81: Cut and flipped ribbon cable section for the lower sensing sheet.

Once cut to size, triangular masks were created. The masks were made the same way as previous iterations, laser cutting poster board (and for one of the assemblies vinyl) and then taping the multiple sections together to make a 2m long mask. This was done twice for each slave module because of the top and bottom sensor sheets.

The final step was to cut a large sheet of Velostat. A roll of Velostat was purchased from 3M and from that roll a continuous sheet of 0.3m x 2m was cut.

The sheets were then plugged into the FFC connectors, and the masks and Velostat placed in between the inkjet layers.

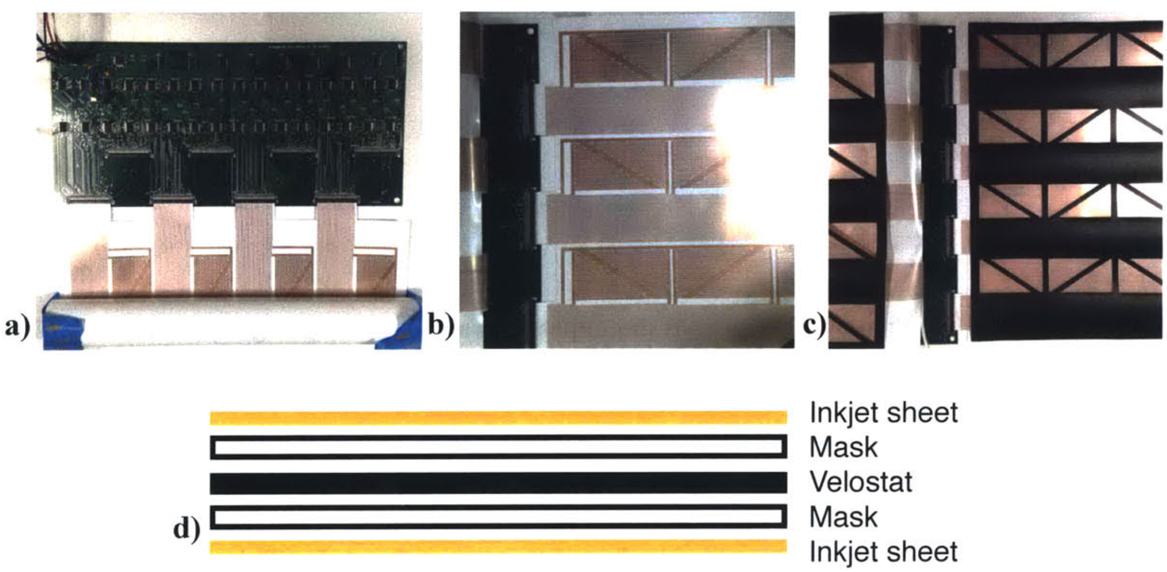


Figure 82: a) Bottom inkjet sheet plugged into the PCB. b) Top and bottom inkjet sheets plugged into the PCB. c) Masking layer applied to the top and bottom inkjet sheets. d) Assembled sensor sheet architecture (PCB not shown)

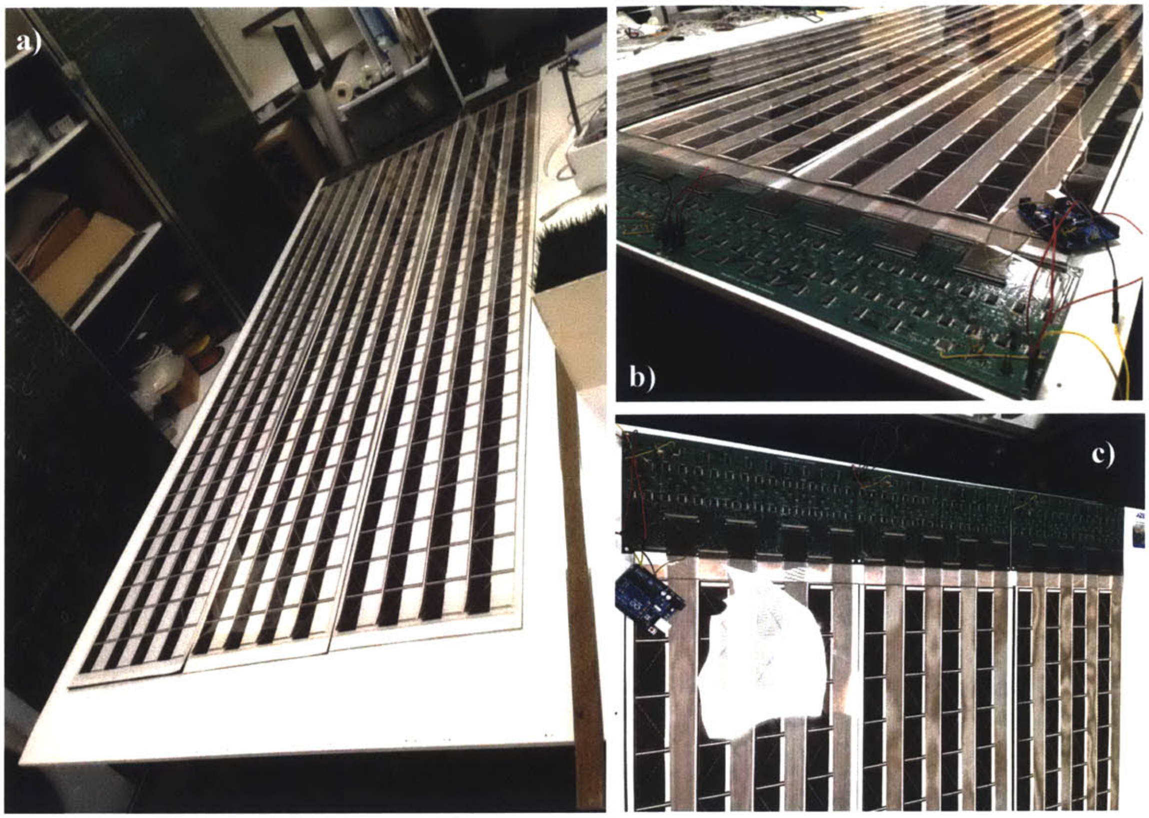


Figure 83: a) View of three slave module assemblies connected together. b) View of PCBs and master Arduino. c) Additional view of the PCB/sensor sheet interface.

The module assemblies were placed side by side on the PVC sheet from Member's Week and the sensing capabilities were tested out. See shape characterization experiment (Section 8.3) for images and explanation of the visualized pressure data.

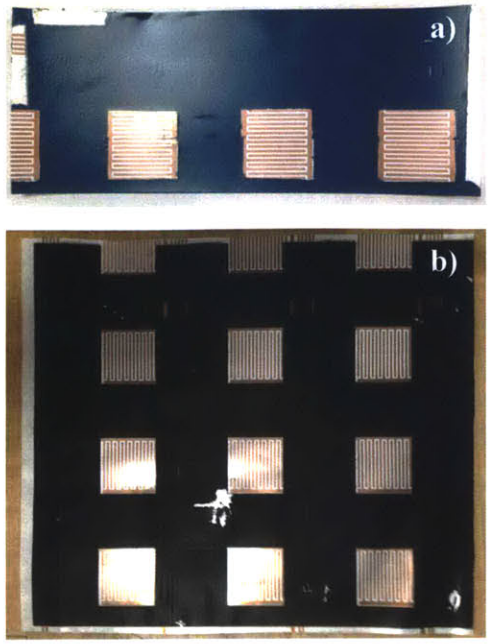


Figure 84: a) Primed and spray painted sensor sheet. b) Sensor sheet run through an office inkjet printer.

6.4.6 Masking Experiments:

Though cutting out masks from poster board or vinyl was effective, it would be a lot faster and effective to print a masking layer directly onto the surface. Two different methods for having a print on or paint on masking layer were investigated. For the first experiment a section of the Square Grid inkjet sheet was run through a standard office printer. For the second experiment different layers of primer and paint were spray painted onto sections of the Square Grid sheet. A variety of scratching, bending, and crinkling tests were then performed to assess how resilient the painted-on/printed-on layer was to wear and flexing.

The results of the masking experiments show that both printer ink and primer/paint are able to electrically insulate the traces. Primer and paint are more scratch and crinkle resistant. Because the inkjet printing process is print then plate, a sheet of inkjet printed sensor traces could be passed through either an additional printing or painting machine to not only create a more durable masking layer but even print out the FSR material itself. This technique is used in conventional FSR sheet manufacturing. See Appendix F for the results of the various tests performed during the masking tests.

7. Visualizations of Sensor Data:

7.1 Overview:

This thesis is an exploration of both the hardware and applications for pervasive surface-based sensing. Different applications have already been demonstrated primarily during the Member's Week 2014 Demo. This section focuses more on visualizations that could someday convey complex information such as physical and mental state of an elderly or at-risk individual to concerned parties such as family members or caretakers in the case of the elderly or disabled. The visualizations should not be just another graph though. They must be informative but not convoluted. Information, trends, and problems must be expressed clearly and be understood with little effort. For the two visualization objects created for this thesis, inspiration was drawn from ambient display philosophy as well as more academic human centered design principles.

7.2 Previous Work:

7.2.1 Ambient Orb: David Rose, Ambient Devices

Inspired by the crystal ball and its magical use in many stories around the world, the ambient orb is a device that, through subtle changes in color hue, conveys a variety of information instantaneously to the user. Information like stock market trends, traffic congestion, pollen forecasts, weather, windspeed etc. [34, 35].

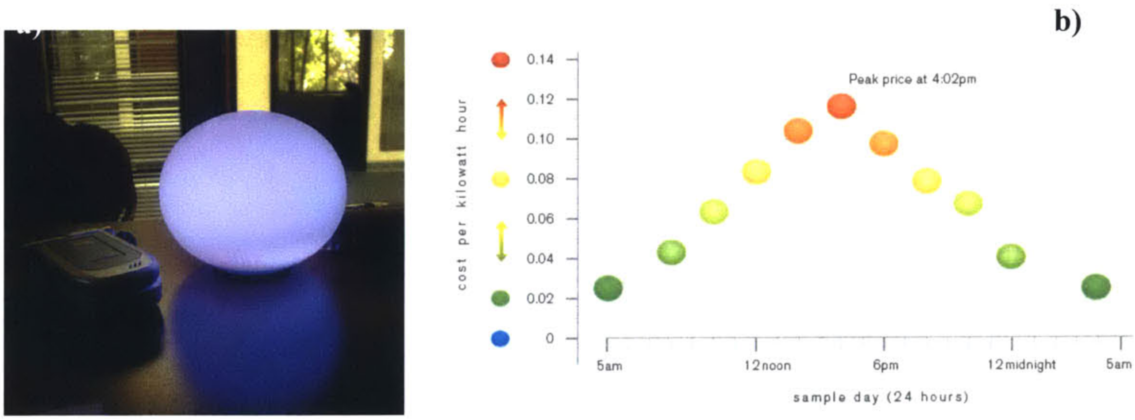


Figure 85: a) Ambient Orb. b) Visual of how the orb changes over the course of a day in response to price of electricity. Image Sources: a) DevX b) Ambient Devices

As he explained in his book on the Internet of Things, Enchanted Objects David says he was was "admiring the features of familiar objects, such as house clocks and compasses, and the simplicity of interacting with them" [36].

Rose states that he likes old clocks, old weather stations and barometers because "they are always there, politely waiting for you to notice them" [36]. Unlike phones and computers, "clocks respect your attention. They aren't interruptive. They have a calm presence. They don't require you to do anything to gather information from them. Don't need you to find an icon, launch an app or type in a URL" [36]. He was looking for a form that could display information in a "home-friendly" and "aesthetically pleasing way" [36]. The eureka moment came when he "realized that [he] could express information…as subtly changing colors" [36]. Trends in information could cause a shift of hue along a spectrum. A person would quickly come to know the color range and what it meant.

7.2.2 Digital Family Portraits: Professor Elizabeth Mynatt, Georgia Tech Institute for People and Technology

The digital family portrait was designed to "provide awareness of senior adults' day to day activities" which in turn "promot[es] peace of mind for extended family members" [9]. The portrait provides qualitative sense of the targeted individual's daily activity and well being. Sensor data collected about the individual drives the digital visualization. It is designed to be in a public space, either hung on a wall or placed on a mantle to blend in with the rest of the household items. The portrait displays representational data ranging from "general measurements of activity, to indications of the weather" in an attempt to capture the observations of someone who was living in the same house or next door to the individual. Mynatt states, "most awareness interfaces only provide a snapshot of the present. Since many questions about an elderly parent refer to trends over time, such as "is she becoming more socially isolated?" We provide representations of the past as well as the present." [9]. The frame is able to show current and long term trends about the individual, which is very useful to see if there is any shifting over time.

The frame design had a few requirements. It needed to convey "relevant information about a person's daily life, to support low level awareness of that person's well being" [9]. Trends over time needed to be depicted in the different sensing categories. The visualization should only provide qualitative views, to respect privacy. It should look aesthetically pleasing and not stand out amongst the rest of the home decorations. And finally, the visualization should be "emotionally appropriate," and convey negative information accordingly.

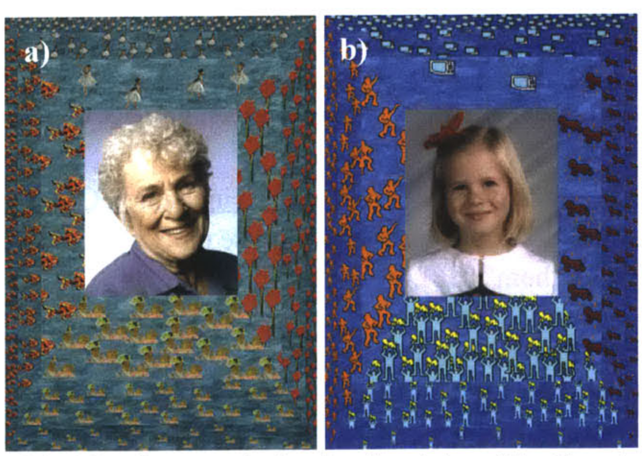


Figure 86: a) Digital family portrait showing the status of the Grandmother for her children. b) Portrait of the Granddaughter that is meant for the Grandmother to create a digital connection to her family. Image Source: Georgia Tech.

For the above visualizations, the past 11 days are divided across three bands on the frame. The band closest to the picture represents the current day, the second band represents an average of the three days prior to the current day, and the third band represents the average of the 7 days prior to the inner two bands; i.e. "the closer you are to the photograph, the more recent the information, and the less compressed it is" [9]. The different icons represent different sensing categories and the density indicates the measurement value. To provide more of a connection, the grandmother had a frame of her granddaughter placed in her home, and vice versa. In field trials, the daughter and grandmother would discuss their respective frames.

Two physical-digital visualizations were designed that would represent possible ways walking or activity data of a user might be conveyed to their family. The best aspects of the digital family portrait and ambient orb were used to inspire the design of the visualizations. From the portrait design, we really appreciated the familiarity of the picture frame as a common household object and the intuitive way that current and long-term data was displayed. From the ambient orb, we were inspired by its glanceability. Something that instantly conveyed the information one was interested in without requiring much mental conversion. We decided to design two other common household objects that wouldn't be obviously intelligent but rather would subtly display their respective information for those who were interested and were aware of them.

7.3 "Digital" Clock:

The first device was a "digital" clock face that changed to a data display interface when somebody walked close enough. A computer monitor served as the digital interface, and a CNC routed wooden tabletop provided a layer of aesthetics and framing of the interfaces. Ultrasonic sensors on the top and bottom of the wooden frame allowed the system to determine if someone was standing in front of the system. This would eventually trigger the visualization. Both the digital clock and data visualizations were done with Processing. An Arduino Uno on the back of the clock was connected to processing and handled the ultrasonic sensor input.

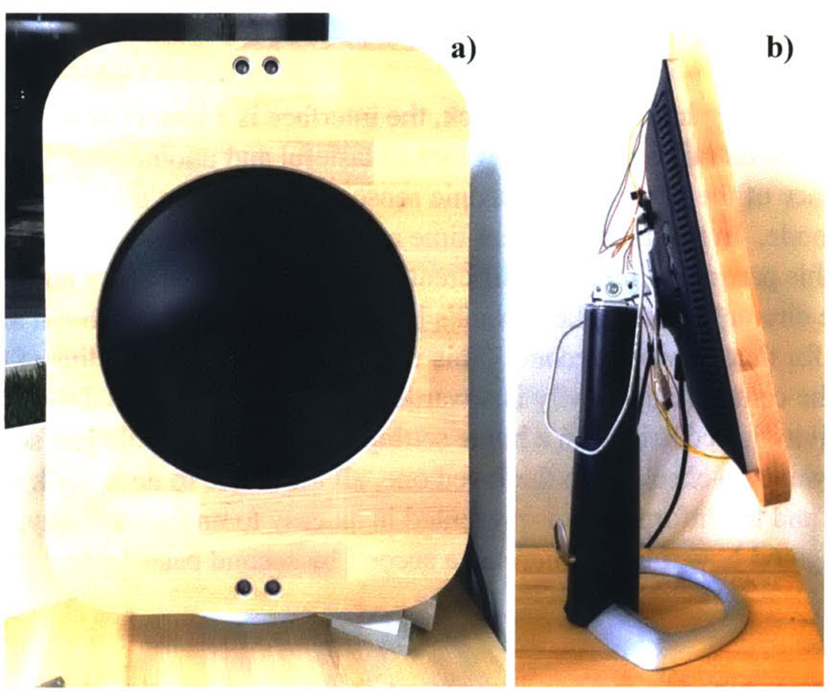


Figure 87: a) Clock face showing the routed frame, sensors, and black digital screen. b) Side view of the Clock showing the wires and Arduino.

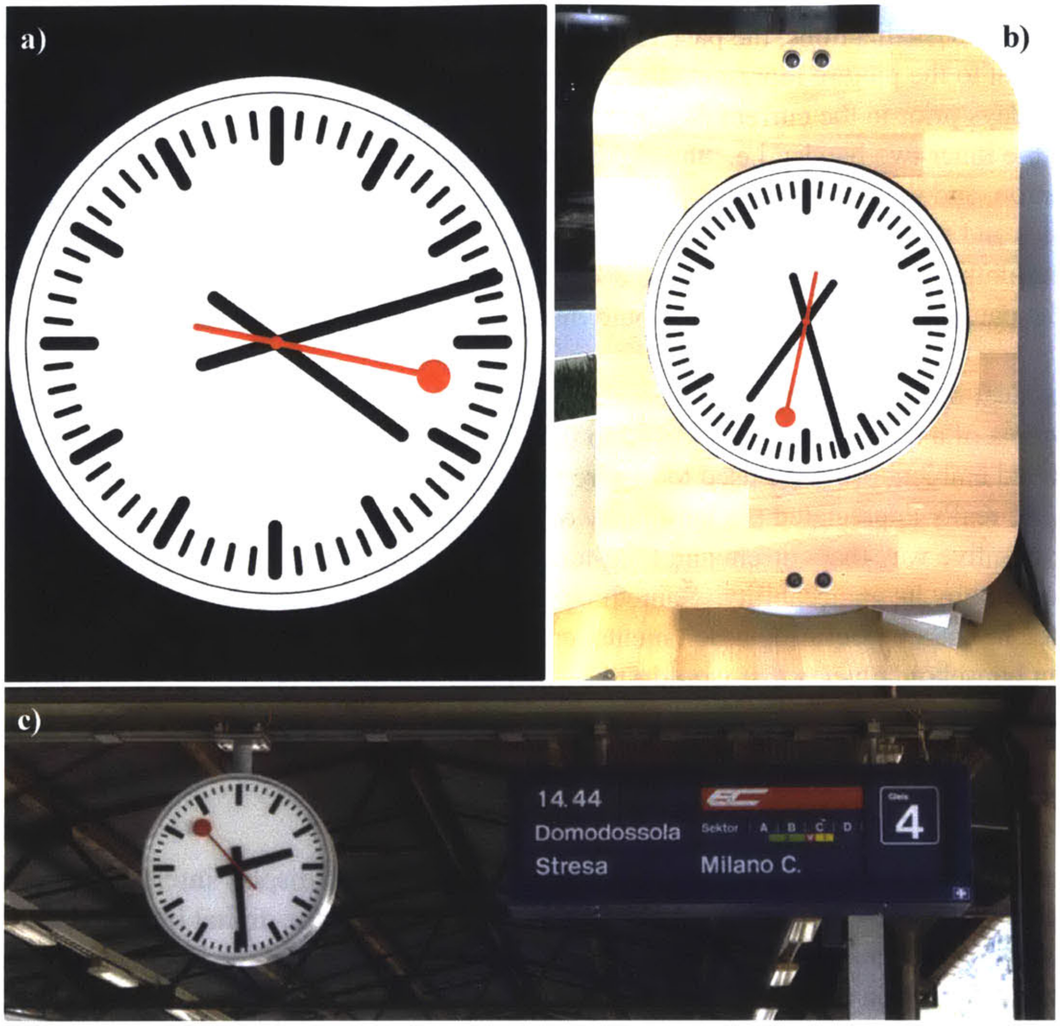


Figure 88: a) Default clock face inspired by the Swiss Railroad Clock Aesthetic. b) Default clock face on the physical display. c) Inspiration for the clock design. Image Source: c) Retours

7.3.1 Family Clock Interface

When nobody is standing in front of the clock, the interface is a simple wall clock. It was modeled after the Swiss railroad clock so it looked tasteful and unobtrusive. When somebody approaches the face of the clock, the ultrasonic sensors are triggered and the clock switches its display to data mode. In this mode the real-time activity information of the loved one are displayed. For this particular interface, different rooms of the loved one's house or apartment are listed around the circumference. The second's hand still goes clockwise but it also traces out the level of activity for that previous second. This value is mapped to the radius of the display (no activity is near the center, high activity is towards the edge) and it is filled the color of the room in which the activity is occurring. In a home setting, most people would just see a clock. If a family member wants to check in on the loved one, all they have to do is walk up to the clock, it changes screens and the information is presented in an easy to understand way; where is the activity occurring and how much is there on a second by second basis?

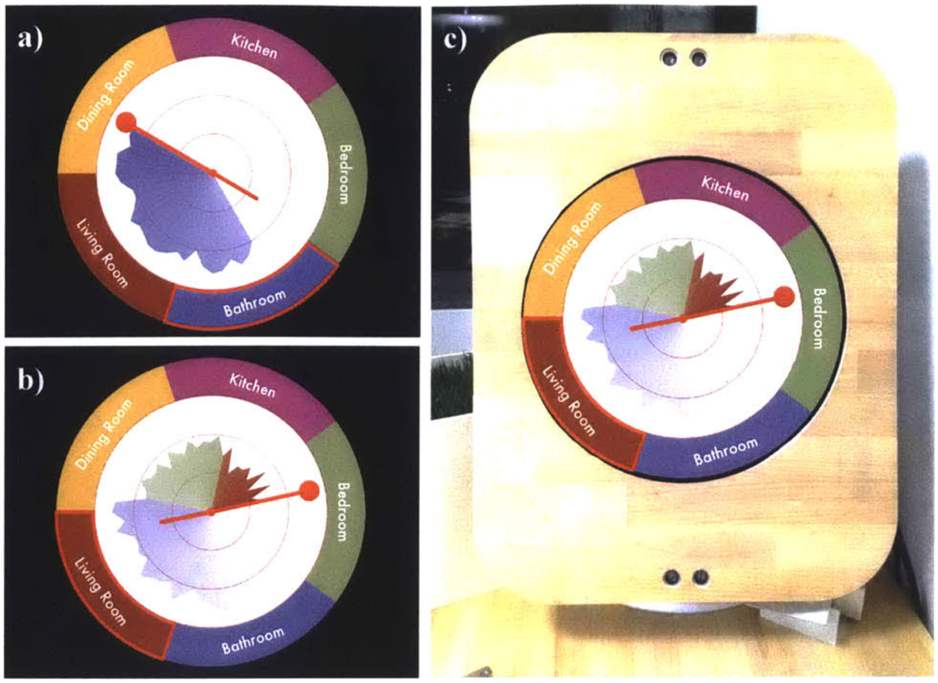


Figure 89: a) Data visualization showing one active room. b) Data visualization showing the active room and history over the last minute. c) Data visualization on the physical display.

7.3.2 Personal Clock Interface:

In Mynatt's paper, she talked about how an elderly individual and the concerned relatives would have complementary frames in their homes [10]. So while her family had a picture of her, the grandmother would have a picture of her family or grandchild and she could see how they were doing. We also thought it would be nice for the individual to have a visualization of their own data. But we didn't want to overburden them with information or unnecessary alerts. So we designed an interface that would appear on the elderly individual's clock. It shows when they wake up, what activities they have done, and how much activity occurred during the different intervals. On the below visualization, you can see when sunrise occurred (sun icon), when the person woke up (alarm clock icon). Other events include when the person left/returned to their room (door icon), when the sunset occurred (moon icon) and when the person went to bed (bed icon). A person looking at this visualization can easily see how much time they spent doing certain activities relative to the 24 hour day. Did they get out enough, did they sleep enough, these are things that could be easily inferred by looking at the visualization.

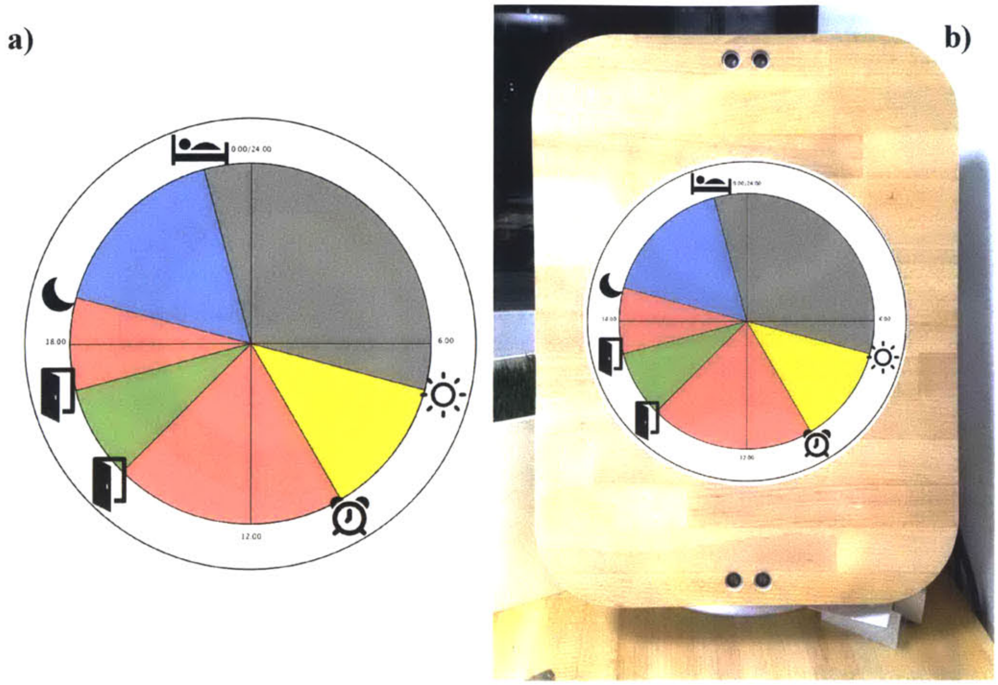


Figure 90: a) Data visualization for the individual showing at what times they did certain activities. b) Personal visualization on the physical display.

7.4 Potted Plant:

The second device was a transforming potted plant where the height and color of the grass depended on the average daily activity levels of the person being monitored.
In addition to the clock, which showed real time location and activity level, we needed a visualization that would show longer-term trends. We thought about another common household object, which is some kind of flowerpot or potted plant. Most people have some kind of potted plant in their home, as a centerpiece, personal garden, or pet project. Using a potted plant to convey the status of another living thing is an easy comparison to draw. If the plants aren't doing well, then you know something is wrong.

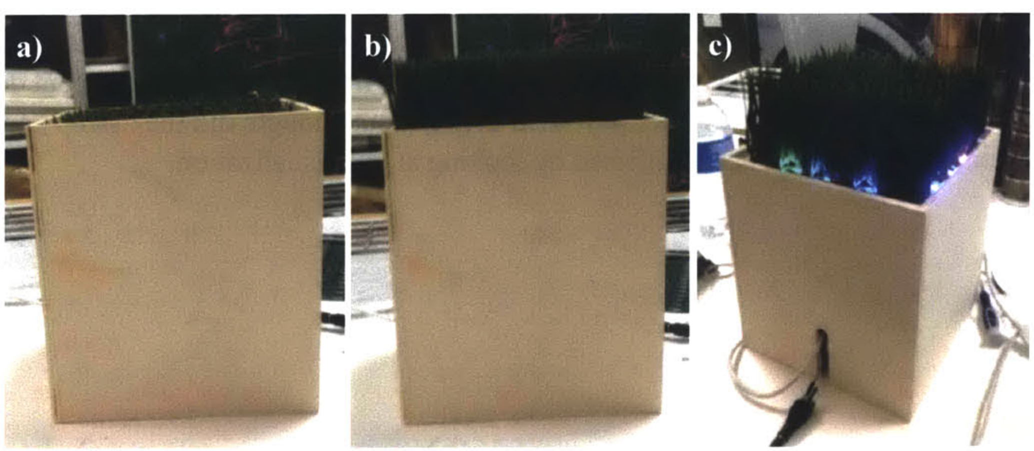


Figure 91: a) Grass short indicating low activity. b) Grass tall indicating high activity. c) Random colors to demonstrate the ability to highlight the grass with color.

With one or several plants sitting side by side, a relative would be able to survey the daily activity trends of the elderly individual. The grass would "grow" depending on how active the monitored person was that day. Colors can also add to the effect. Red-lit grass indicates that there was an issue or the activity was lower than normal. Similarly, Green-lit grass indicates that all is well. Other colors or motion that occur suddenly could indicate specific events both good and bad. For example, with seven in a line, one can see the daily activity of the individual over the span of a week. If the individual didn't feel well on Thursday, one would be able to see the days before and after to monitor both the progression of the illness and their recovery.

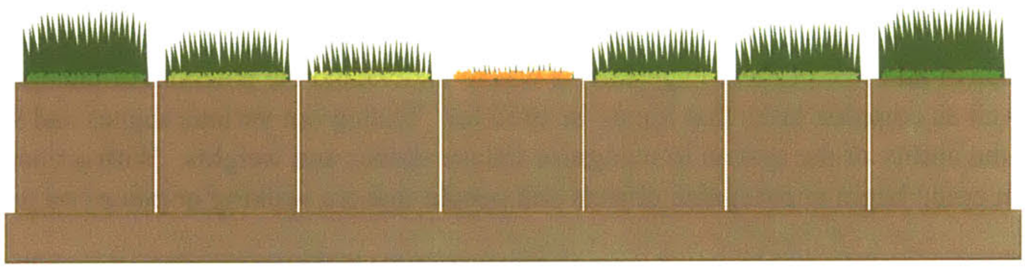


Figure 92: Seven planter boxes representing the seven days of the week. The short, orange colored grass indicates that activity was very low that particular day.

The grass is raised and lowered by a stepper motor driven lead screw, similar to how 3D printers raise and lower a build plate. A chain of Adafruit Neopixel LEDs provides the lighting. These LEDs allow individual control of each diode in the chain. A 12V power supply powers the motors and Arduino. The enclosure is made out of 1/4" plywood but could easily be ceramic, glass or other aesthetic materials.

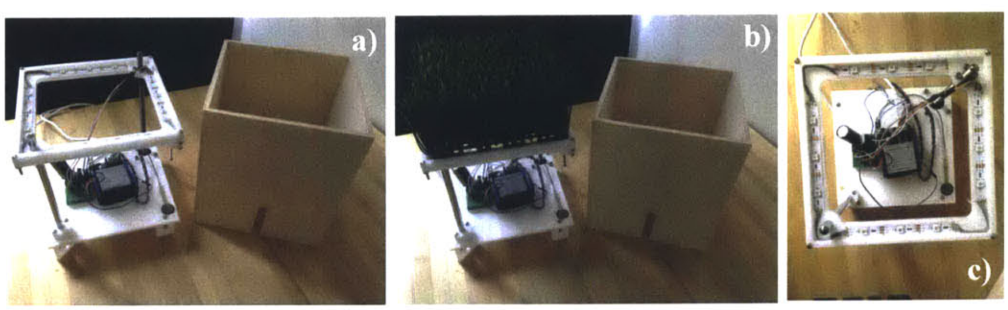


Figure 93: a) Exploded view of the Potted Plant system. b) Potted Plant system with artificial grass placed. c) Top view of the internals of the system. The LEDs can be seen around the perimeter.

8. Experiments and Verification:

8.1 Overview:

Three experiments were conducted on the system to assess and characterize its ability to capture different pressures, locations, and gait parameters. The first experiment was meant to characterize the spatial and pressure sensing characteristics of the system. The second was a walking test where people walked across the system and their video-recorded footsteps were compared to an algorithm processing the sensor's data. The third experiment tested the response and repeatability of one, two, or four sensors under known loading conditions.

The purpose of all three of these experiments was to understand the potential of the system for basic as well as complex tasks that it may be used for. Testing out various shapes and weights indicates the ability of the system to recognize unique shapes and weights. With proper training, the system could begin to recognize objects and people that are walking or placed on top of it.

8.2 Sensor Characterization:

To understand the sensitivity of each of the individual FSRs and to assess repeatability and response loading tests were conducted on one, two, and four sensors. These included simple load then measure, measure then remove loading, and measure then apply loading. This would give indications of repeatability as well as loading/unloading curves for the sensor. Because the analog output of 0-1023 was mapped to 1-254, the range of the outputs runs from 1-254.

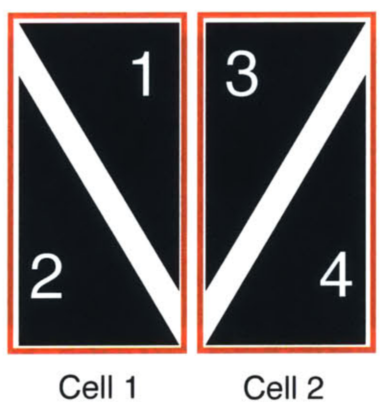


Figure 94: Diagram of two cells of sensors on the final prototype. Each cell contains two triangular FSRs

8.2.1 Single Sensor:

Three loading tests (with three trials of each) were conducted on the sensor at position 2 (see Figure 94 above) with a 250g weight. A container full of water was used to achieve the desired weight and a flat-bottomed LEGO applicator was used to apply the force on the single sensor. The dimensions of the small fixture were 1.5cm x 1.5cm. The force was applied to the largest open area of the triangular sensor.

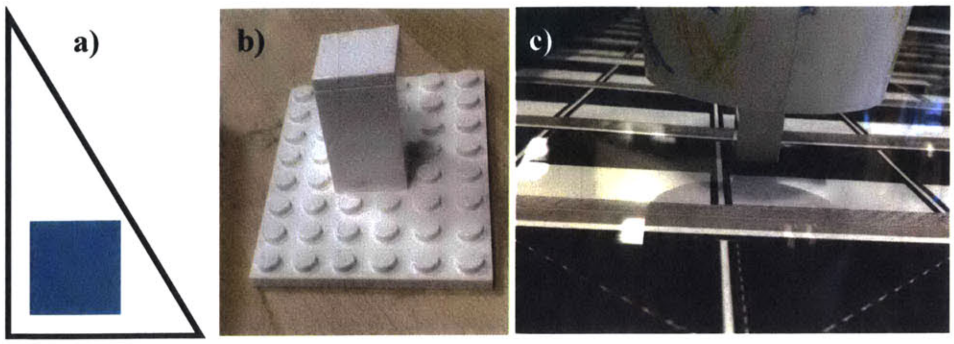


Figure 95: a) Location of force application. b) LEGO® force applicator. c) Water container and applicator loading a single sensor.

Static Loading:

The first test was static loading. The weight was placed onto the sensor and the steady state output was recorded. Between each trial the weight was removed entirely from the sensor and reapplied after a few seconds. This way the system was given time to return to its original state before loading. The three average values for each trial were about 191, 149, and 171 respectively.

Applied Loading:

The second test was applied loading. The data recording was started, and a few seconds later the weight was placed carefully and quickly onto the sensor. Between each trial, the weight was completely removed as well. A second order exponential fit was applied to each curve to understand the general shape of the response. The final value the sensor settled upon after loading was 172, 146, and 175 for the three trials respectively.

Unloading:

The third test was unloading. The recording began, and a few seconds later the weight was carefully and quickly removed from the sensor. No curves were fit to the output because it almost immediately dropped to zero once the weight was removed. The beginning values were about 150 and 110 for trial one and three respectively. Trial two did not record properly so it was removed.

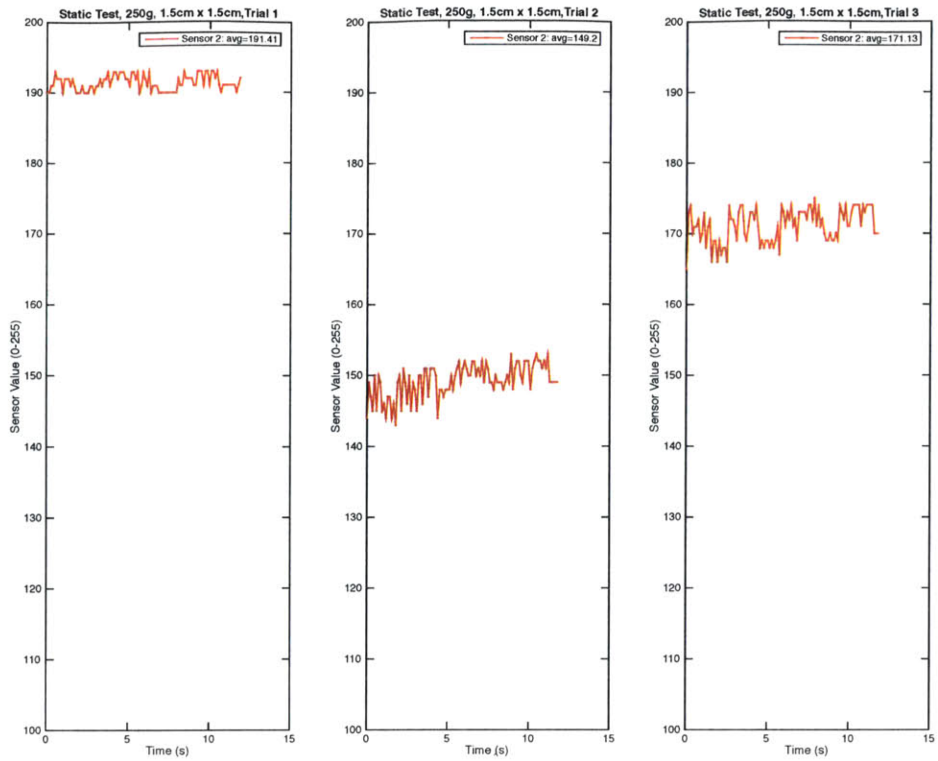


Figure 96: Static loading tests for a single sensor with 250g.

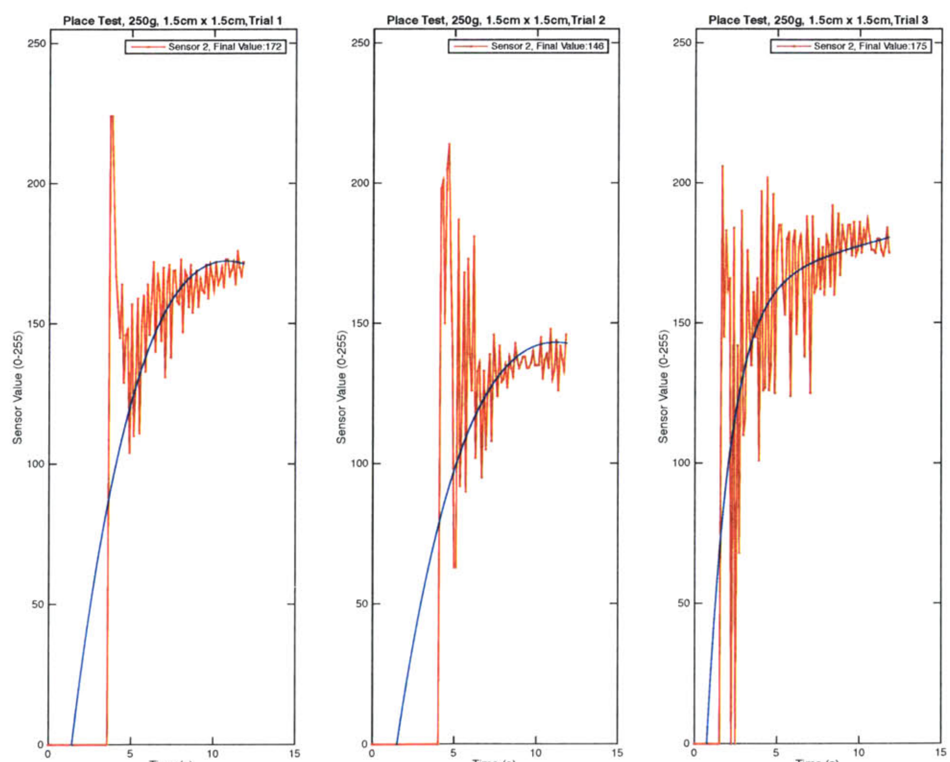


Figure 97: Applied loading tests for a single sensor with 250g. The red curve is the raw data and the blue is the fit curve.

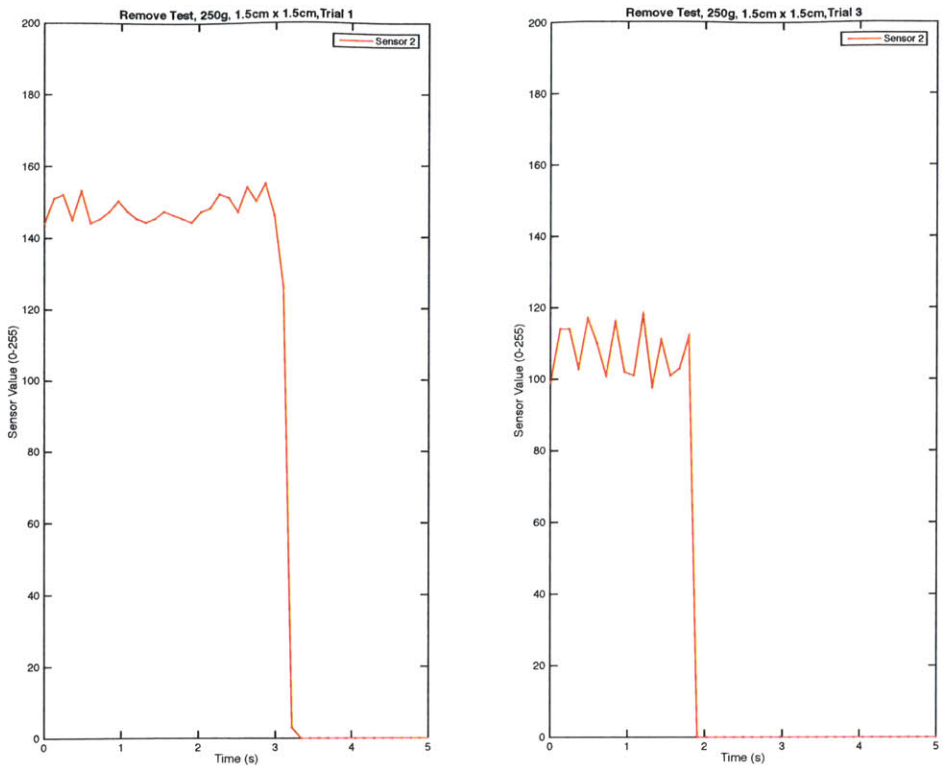


Figure 98: Unloading tests for a single sensor with 250g.

8.2.2 One Cell of Two Sensors:

After testing the individual sensor, multiple sensors were tested. Another LEGO® force applicator was created with the area of one cell (~ 6cm x 3cm). The force was applied over the sensors at positions 1 and 2. After trying one test, it was discovered that the spacers were preventing the perfectly flat bottom of the applicator from actuating the sensors. Similar to the shape tests conducted on the sensor mat (see Section 8.3), a layer of fuzzy Velcro was applied to the bottom to help actuate the sensor.

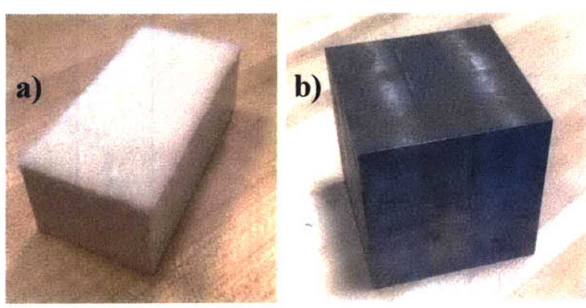


Figure 99: a) LEGO® force applicator for one cell. Fuzzy velcro covers the smooth surface. b) 7.47kg block of steel

A 7.47kg block of steel was used to apply the force. A static loading test and an unloading test were conducted on a single cell of two sensors. An applied loading test was not conducted because it was difficult to place the heavy weight quickly and accurately. It was easy enough to remove quickly though. The results are shown below. For the static loading, sensor in position

2 in trial one, and both sensors in positions 1 and 2 in trials two and three had an average recorded value of about 246. Sensor 1 in Trial 1 had an average value of 234.

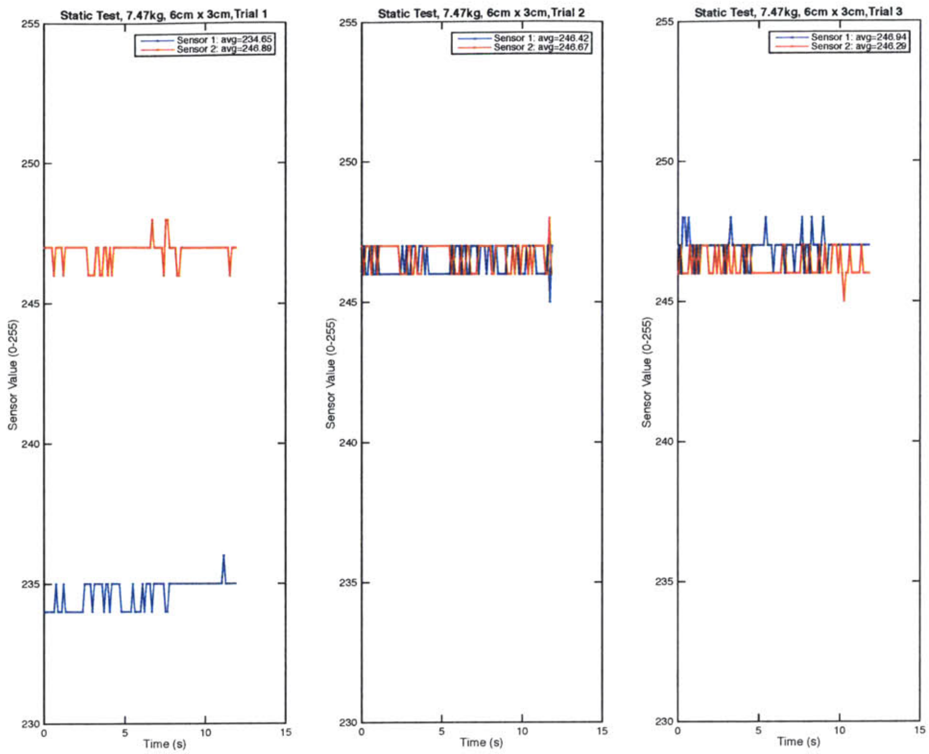


Figure 100: Static loading tests for a single cell with 7.47kg.

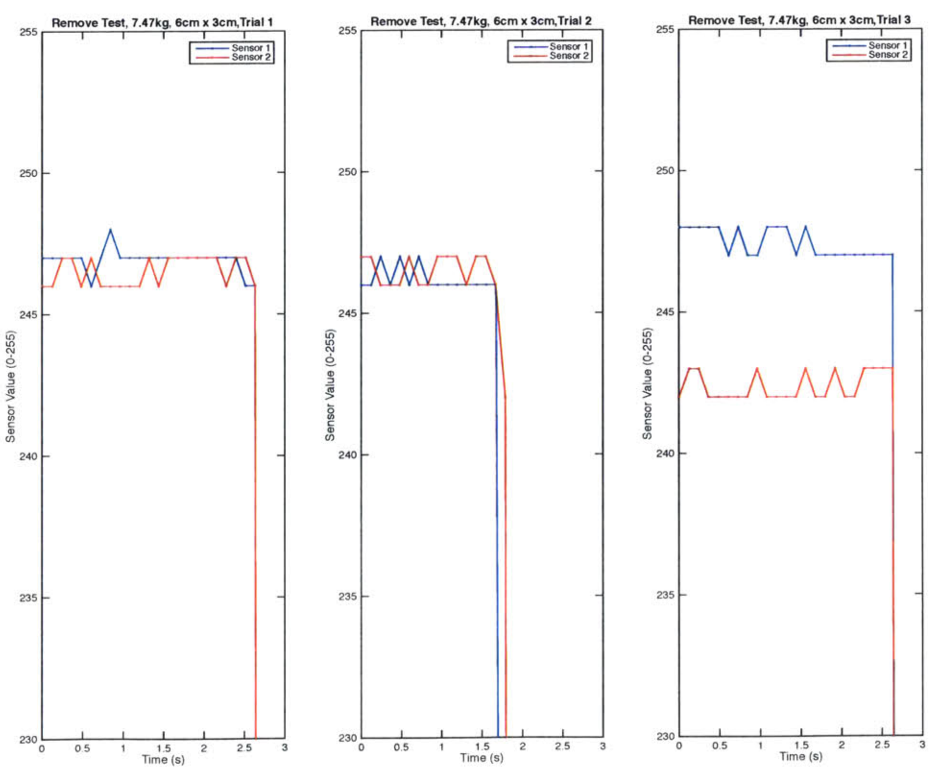


Figure 101: Unloading tests for a single cell with 7.47kg.

8.2.3 Two Cells containing Four Sensors:

The final test conducted was the loading of two adjacent cells with a total of four sensors. The only test that was performed was three trials of static loading.

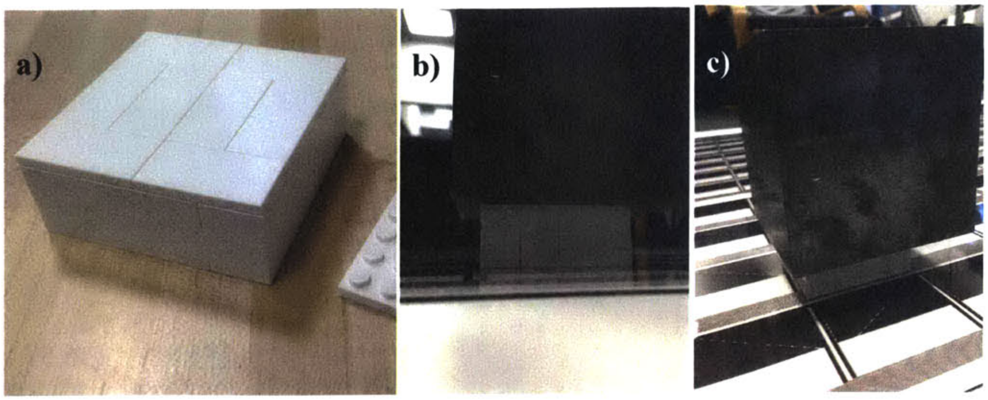


Figure 102: a) LEGO® force applicator for two cells. b) Steel block applying weight to the cells. c) Other angle of steel block applying weight.

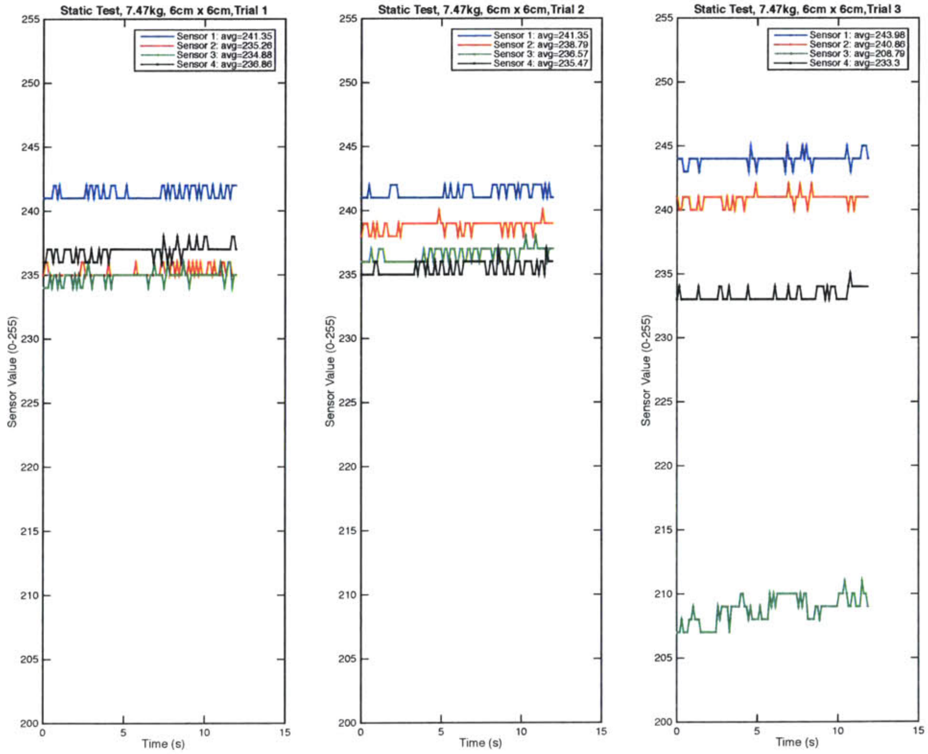


Figure 103: Static loading tests for two cells with 7.47kg.

Though there was some spread, the average values recorded hovered around the high 230s for all four sensors. Trial three had the most variability and it could be due to human error because trials one and two are very similar.

Table 3: Static loading values (1-254): One sensor with 250g, and one and two cells with 7.47 kg.

	One Sensor T1	One Sensor T2	One Sensor T3	One Cell T1	One Cell T2	One Cell T3	Two Cells T1	Two Cells T2	Two Cells T3
Sens. 1	191.41	149.20	171.13	234.65	246.42	246.94	241.35	241.35	243.98
Sens. 2				246.89	246.67	246.29	235.26	238.79	240.86
Sens. 3							234.88	236.57	208.79
Sens. 4							236.86	235.47	233.30

8.2.4 Discussion:

There are three different pressures that were tested for the single sensor, one cell, and two cell experiments respectively:

$$P_1 = 0.25kg \cdot \frac{9.8m}{s^2} \cdot \frac{1}{0.015m \cdot 0.015m} = 10.89kPa$$

$$P_2 = 7.47kg \cdot \frac{9.8m}{s^2} \cdot \frac{1}{0.06m \cdot 0.03m} = 40.67kPa$$

$$P_3 = 7.47kg \cdot \frac{9.8m}{s^2} \cdot \frac{1}{0.06m \cdot 0.06m} = 20.34kPa$$

And the average sensor value from the static tests were: V_1=170.58, V_2=244.64, and V_3=235.62

The static values showed good consistency between the trials for each of the weights tested. The graphs showed noise but little drift. The only exception was the first trial, where there was some drift in the values. This can be attributed to the to motion of the water inside the container as well as springiness of the sensing sheet (more on that later).

Explanation of Static Loading Outputs:

The recorded analog values do not correspond linearly to the weights tested but this is also expected. Though only two weights were tested, the fact that 10.89kPa returned an average raw value of 107.58 and 40.67kPa returned an average raw value of 244.64 indicates that the response is nonlinear. This observation is in agreement with the curves relating resistance to force for some of some of the off-the-shelf FSR sensors. Past 1kg of force the resistance values start to differ by only fractions of a kΩ. This in turn corresponds to smaller analog value differences in recorded higher forces.

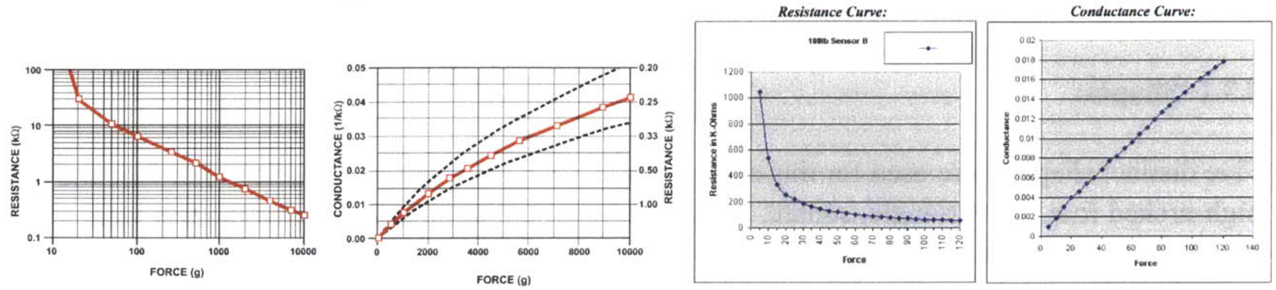


Figure 104: Force/Resistance and Force/Capacitance curves for Interlink and FlexiForce sensors. Image Sources Interlink and Tekscan

Explanation of Applied Loading/Unloading Outputs:

The easiest graphs to explain are the outputs for the unloading cases. In the sensor architecture a spacer sits between the Velostat and inkjet printed sheets. The trace sheet is thin but relatively rigid plastic so once something is no longer pushing the plastic through the spacer to contact the Velostat, it snaps back into place. If the inkjet printed sheet were laminated directly to the Velostat, we would see a more gradual return to zero because Velostat takes some time to return to it's original shape after loading. Most FSR designs incorporate a spacer for this reason.

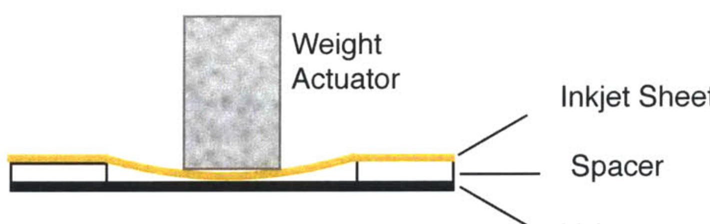


Figure 105: Sensor actuated by a weight.

For the applied loading (single sensor only), the fit curves show a relatively more gradual approach to the final value. This is because the velostat takes some time to compress, and to reach the final loaded level of compression. The noise is a combination of my hands being unable to perfectly imitate a step loading condition, as well as the water moving around the container. Because the oscillations continue over a span of several seconds, the motion of the water and subsequent rocking of the applicator on the sensor are more likely the cause of the noise.

8.3 Shape Characterization:

To understand the response of the system to different shapes and weights, several objects were created including flat and arched foot shapes and basic shapes such as a triangle, square, and circle to press into the mat under known loading conditions. The output images would be recorded and used to understand the performance of the system. Though no machine training or image analysis is done for this experiment, the results show that a well-designed algorithm would have distinct features that can be used for identification. Even without a computer monitoring the data, a user can interpret the shapes and relative weights of the objects activating the sensor. See Appendix G for all the output images from the shape characterization tests.

8.3.1 Shapes:

Five distinct shapes were used for this experiment: a triangle, a square, a circle, a flat foot and a high-arched foot. The triangle, square, and circle have similar dimensions and the feet were traced from the bottom of my shoe (US size 11) so it is slightly larger than my foot. The area of each shape is enough to incorporate multiple individual sensors so that they would be easier to identify when pressed into the system. This was done more for the benefit of the viewer. The foot shape had two distinct profiles. The first imitated a solid shoe bottom with little-to-no contour. The second imitated the profile of a person with a high arch walking barefoot. The goal for the different feet configurations was to demonstrate a difference in output under the same loading conditions.

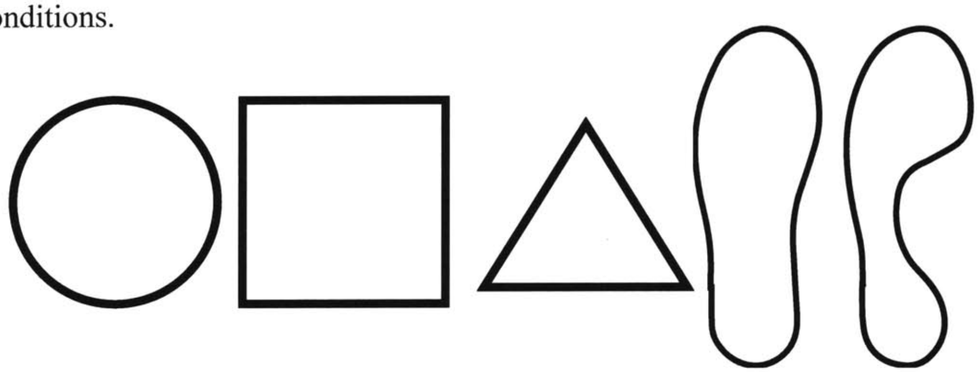


Figure 106: Shapes used in the characterization experiments. Circle, square, triangle, flat foot and high-arched foot.

Table 4: Dimensions for each of the shapes used in the characterization experiments.

	Width (mm)	Height (mm)	Area (m^2)	Area (ft^2)
Circle	268	268	0.072	0.773
Square	280	280	0.078	0.844
Triangle	300	233	0.035	0.376
Flat Foot	113	304	0.034	0.370
Arched Foot	113	304		

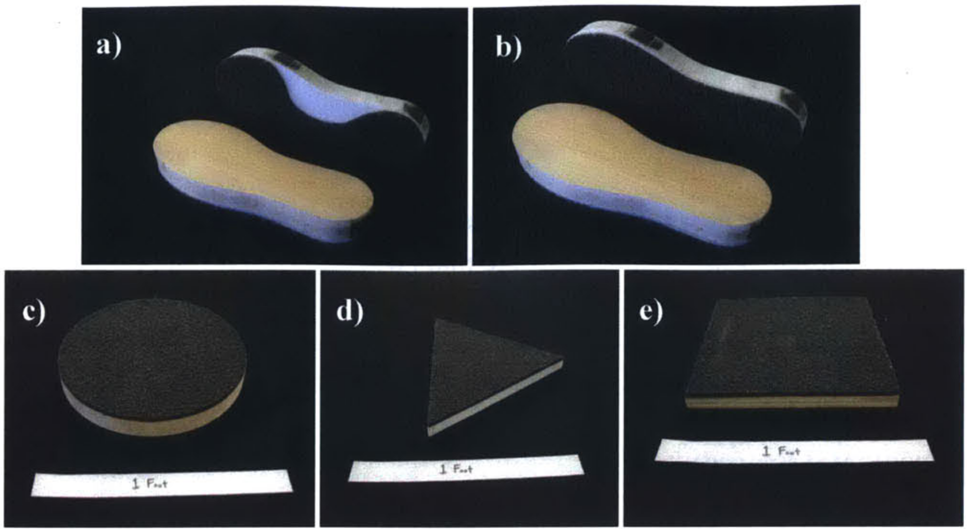


Figure 107: a) Arched foot profile. b) Flat foot profile. c) Circle profile. d) Triangle profile. e) Square profile. The carpeted side will be face down in the experiment. Note the 1' reference marker.

8.3.2 Force Distribution:

Because the FSRs include a spacer in their construction, a completely flat, non-compliant surface such as a slab of wood is unable to activate the sensors well. To provide better force distribution and compression of the inkjet sheets into the Velostat, a layer of 1/8" foam was added to the top of the mat. Stepping on the foam pushes it into spacer gap and provides better contact for the conductors and FSR material. In addition to the foam layer, beneath each shape a layer of 1/8" carpet was added. This provided additional compliance. Both these interventions were done to elicit a better response from the system. Future improvements to the design of the sensor would eliminate this need.

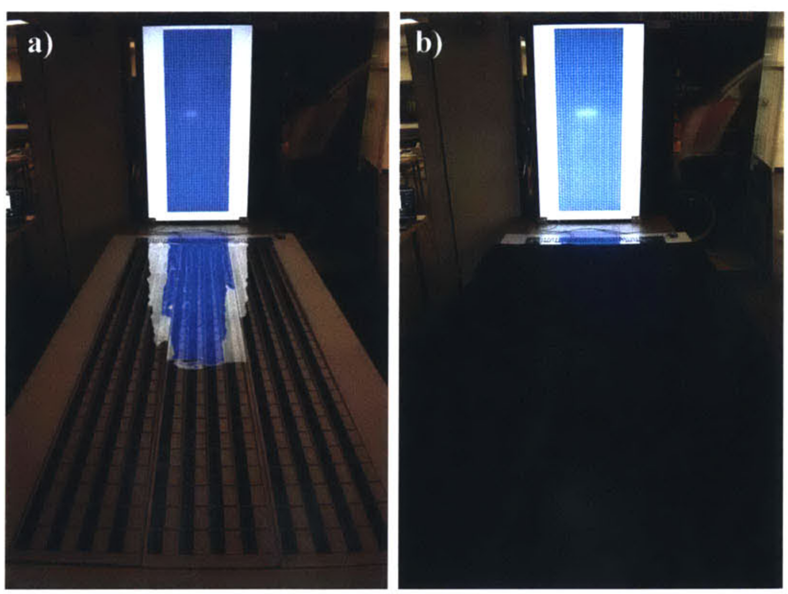


Figure 108: a) Sensor mat system placed on the table. b) Foam sheet covering the system.

8.3.3 Experimental Setup:

For the experiments the sensor mat system was placed on a large conference table in the Changing Places Group area. The head of the mat (where the PCBs were) was placed at the head of the table. Also at the head of the table, a flatscreen TV flipped 90° was used to display the visualizations. To allow better force distribution and activation of the sensors, I placed a 1/4" foam sheet over the mat and it was there for the duration of the experiments.

8.3.4 Experimental Procedure:

Weight:

Two possible loading conditions were incorporated into the experiment. Either one or two 5-gallon water cooler jugs provided the weight. Each weighs about 42lbs. Taken together, the weight is comparable to that of a small child. One was enough to effectively activate a group of sensors, and the second provided further activation and a second point of comparison. Either one or both of the water jugs were placed on a sheet of 3/4" wood to evenly distribute the weight to the object below.

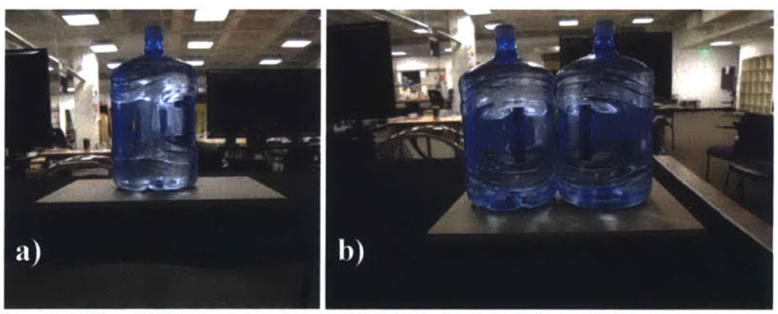


Figure 109: a) One water jug weighing about 42lbs. b) Two water jugs weighing about 84lbs.

Orientation:

Different orientations were given to the shapes for separate experiments. This was to determine whether the same shape would be visualized differently or not as well compared to another orientation. The feet were oriented differently to represent foot placement pathologies.

Foot:

For both flat and arch bottoms, the feet were placed at parallel, toe out, and toe in configurations. This choice in orientation had two purposes. The first was to demonstrate that different foot placement configurations had unique and recognizable outputs. The second was to anticipate diagnostic tools that could recognize walking pathologies such as atypical or asymmetric feet placement.

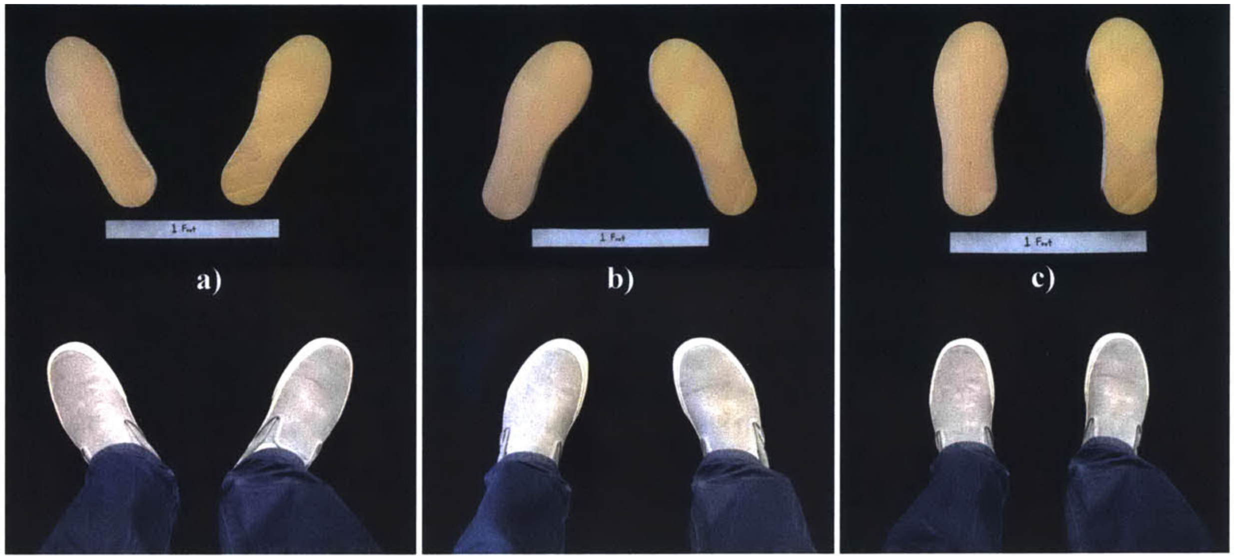


Figure 110: Experimental orientation and my own feet placement for a) Toe-out or "duck-footed" orientation. b) Toe-in or "pigeon-toed" orientation. c) Parallel feet orientation.

After the controlled foot shapes were measured, I replicated the flat and arched foot configurations myself by standing either with or without shoes on the system. This was to generate an output of a live person standing on the system. Weight is distributed differently and unevenly, and the weight itself is twice as much as the maximum weight condition for the experiments.

Square:

The square was placed at 0° and 45° angles relative to the long axis of the system to provide differing views of the same shape.

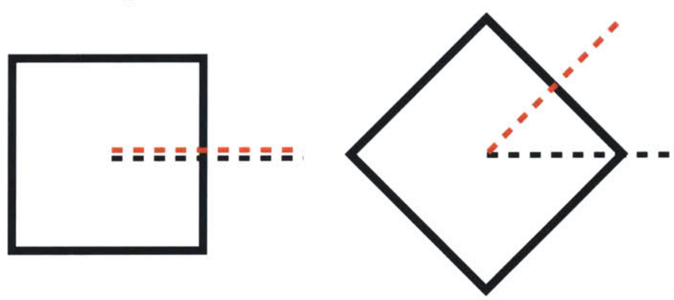


Figure 111: Square orientations for 0 and 45°.

Triangle:

The triangle was placed at 0°, 45°, and 90° angles relative to the long axis of the system to also show how the different outputs could still be recognized as triangular.

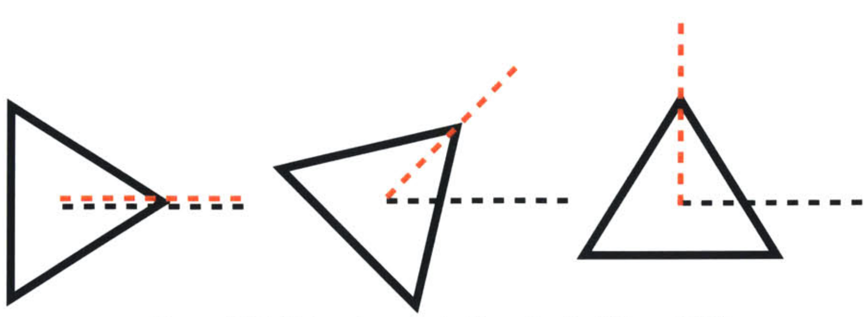


Figure 112: Triangle orientations for 0, 45°, and 90°.

Circle:

The circle did not have any other configurations. I was more interested in how a circular object would appear on a rectangular-gridded sensing system.

8.3.5 Triangular and Square Pixel Output Visualizations:

Each trial was visualized with both a triangular visualization and square visualization. The triangular visualization represented the true position of the triangular-shaped sensors. The square visualization placed the overlapping sensors side-by-side, thereby elongating the visualization lengthwise. Even though the triangular visualization is more true to the physical location and shape of the sensors, the square visualization is easier to look at and interpret. The color of the visualization was a mapping of the analog pressure value of the sensor from blue (no loading) to red (maximum loading). The effect is an intensity map that shows areas of high and low pressure, which is useful for identifying local high or low points on a surface pressed onto the system.

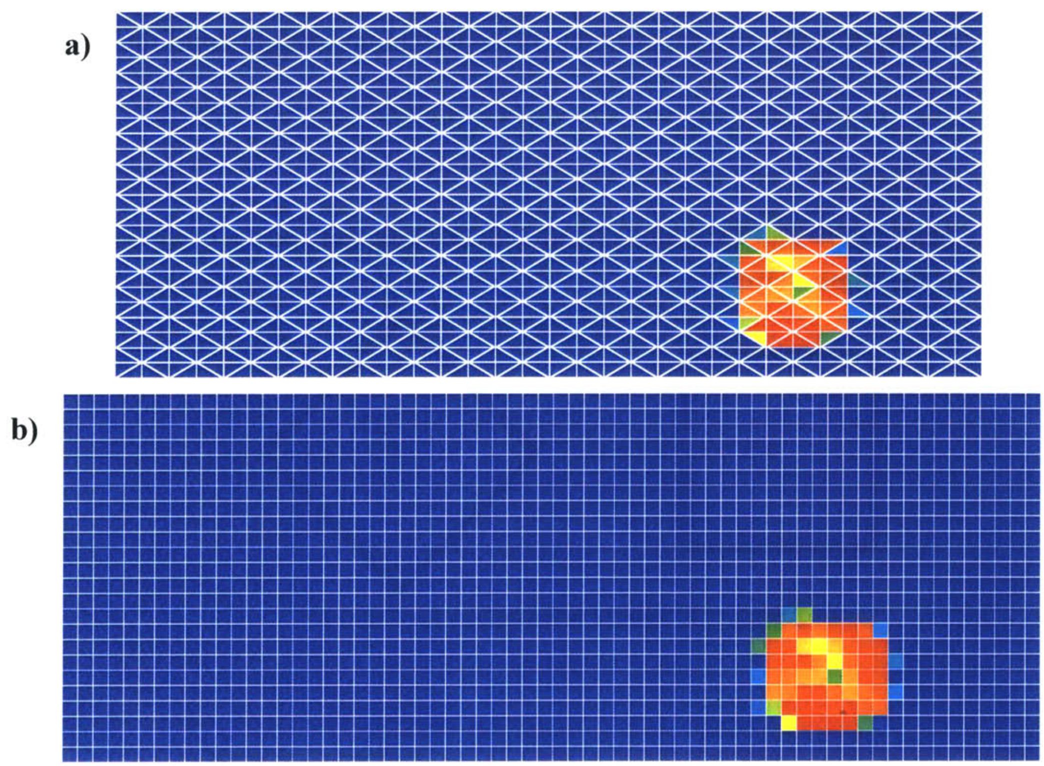


Figure 113: a) Circle object with triangular pixel visualization. b) Same object with the square pixel visualization.

8.3.6 Miscellaneous Activities:

After the controlled shape experiments were concluded, we were interested in collecting some sample outputs of common household or office activities. A colleague laid down, sat upright with legs straight and arms back, sat cross legged, and held a push up pose to see how these activities were visualized. Someone laying down is of interest because of the potential for sleep monitoring and pressure monitoring for those who are bedridden. The cross-legged or upright pose can indicate media viewing or stretching. And the push up pose can indicate exercise, which would be useful for exercise monitoring or physical fitness applications. For the last visualization, an office chair was placed on the system to see how the wheels were visualized. The pentagonal arrangement of the wheels showed up nicely. Sitting in the chair and capturing my corresponding footprints made it more clear that there was a person sitting in a chair. This could be useful for a variety of applications ranging from work health to activity recognition.

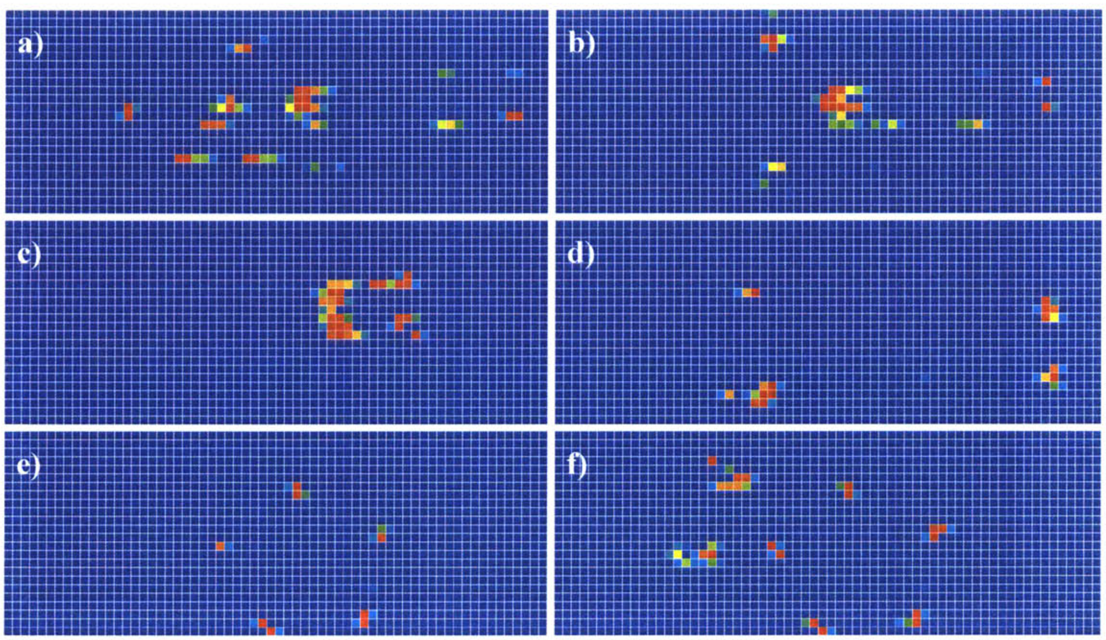


Figure 114: a) Person lying down. b) Person sitting upright on their hands. c) Person sitting crosslegged. d) Person performing a push up. e) Office chair. f) Person sitting in office chair.

8.3.7 Results and Analysis:

The visualizations demonstrated that shapes and relative pressures could be differentiated very well. In the pictures below, for the feet and shapes, the 40lbs loading had more cells closer to low/medium pressure (green, yellow, and orange). After placing a second water jug on the surfaces, the cells became more uniformly red. An edge effect is noticeable where even in the 80lbs loading case; there are cells of blue/green on the perimeter of the shape. This could be due to partial loading of the sensors at the edge of the shape.

Figure 115: Outputs of different shape tests.

The shapes appear relatively clearly and the angles of the triangle and square are fairly easy to discern. It is also easy to see the different foot orientations as well as the areas of high pressure on the balls of the feet and heel. The differences between arched and flat feet are very obvious, as the flat foot visualization has almost no variation within the output, most squares are uniformly colored.

Assessment of Pressure Sensitivity:

The square and circle had comparable areas while the triangle and foot had comparable areas half that value. At full loading, it can be seen that the circle and squares do not have uniform red cells throughout the activated area. As a contrast, full loading on the flat foot and triangular shapes produced more red throughout the area.

This can be explained by the saturation limit of the sensor. At 80lbs/~0.8ft^2, that creates a pressure of ~100psi. The sensors are inconsistent and the carpeted surface is uneven which is why some of the sensors are fully activated (saturated) and others are not, at least on the circle and square shapes. But the relatively high number of non saturated cells leads me to believe that 100psi is close but not quite at the saturation pressure. Looking at the single jug loading condition which has ~50psi, most of the squares are blue/green/yellow. This observation agrees with the stated limit of 100-200psi cited in the Interlink FSR integration guide.

Looking at the feet, for both the low and high loading condition, there are more red cells. This is probably due to higher local pressures under the feet. Together, the feet have the same area as the circle and square, but yet they show more red cells. The triangle, which has a similar area as a single foot shows a similar output for the loading conditions, which reinforces this assumption.

Side-by-side Person Comparison:

To show how two people of differing sizes would appear on the system, A 5'7" woman and a 6'3" man stood side by side on the mat. It's obvious who the footprints belong to based on the output size and pressures. In addition to having a larger foot size, the man applies more weight and therefore has a higher concentration of red squares representing full loading.

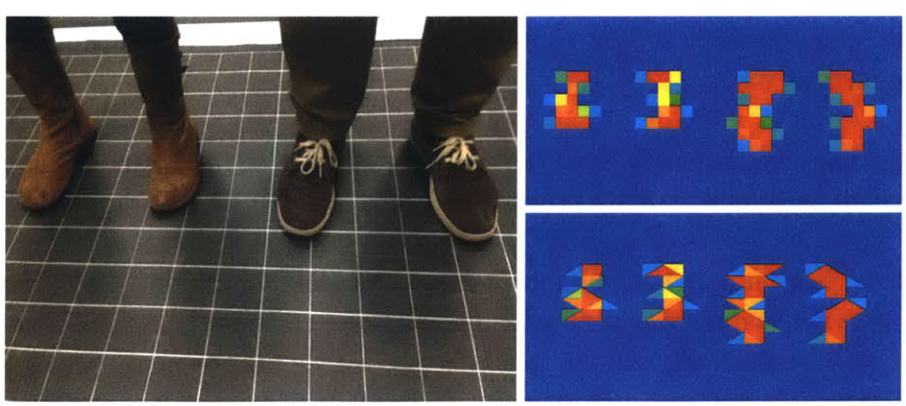


Figure 116: Man and woman standing side by side

8.4 Walking Test:

The walking test demonstrates the system's ability to track people at a small and large scale. By understanding the system's ability to recognize the position and time of footsteps, one can speculate on the diagnostic abilities of the system. If footsteps are accurately pinpointed, it is safe to assume that at scale, the system would be able to give general location of the user if specific footsteps are not needed. For more information on sensor size, see discussion and future work sections (Section 5, Section 6, and Section 9).

The experiment consisted of two measurements occurring in parallel. For one, a MATLAB script collects the sensor mat data as someone walks across the surface, analyzes the data frame by frame and outputs centroid coordinates representing the center of each foot step and the averaged time at which each step occurred. At the same time, two webcams pointed at different angles at the mat record video of the user walking. Grid lines drawn on the top surface of the mat allow one to look through the video and approximate where the centroids of each step were and at which time they occurred based on the timestamp of each frame of the video. For this experiment, the video feed served as ground truth. For each walking trial, these two sets of measurements are compared to assess their agreement.

A final set of experiments imposed artificial gait abnormalities on a user by making them wear an immobilizing knee brace, attaching a heavy weight to one foot, or requiring them to carry a heavy object. These values were compared to the normal gait values for that same user to understand the ability of the system to recognize abnormalities relative to the normal gait.

As a disclaimer, no COUHES were filed before conducting these experiments. All volunteers were gathered from either my research group or around the Media Lab.

8.4.1 Gait:

As described in the research motivations section (Section 3), features that predict both likelihood of falls as well as cognitive decline can appear in the gait and walking style of an elderly individual. Monitoring those parameters and identifying problematic trends as they occur would be immensely beneficial to the at risk individual, their family and their doctor. These experiments only quantify three aspects of gait but in theory could be extended to other important parameters as well.

While most people are familiar with what walking is, gait refers to the "manner or style of walking, rather than the actual walking process [17]. The gait cycle is the "time interval between the exact same repetitive events of walking" i.e. between steps on the same foot and it is generally measured beginning when one foot contacts the ground [gait website]. There are two phases of the gait cycle, stance phase and swing phase. Stance phase is about 62% of the cycle and includes all parts of the cycle that the foot is in contact with the ground. Swing phase therefore is the other 38% and includes the parts of the cycle where the foot has been lifted from the ground and is in the air before being planted for a new step. The cycle can be further divided into several sections. The figure below shows two representations of how the gait cycle can be subdivided.

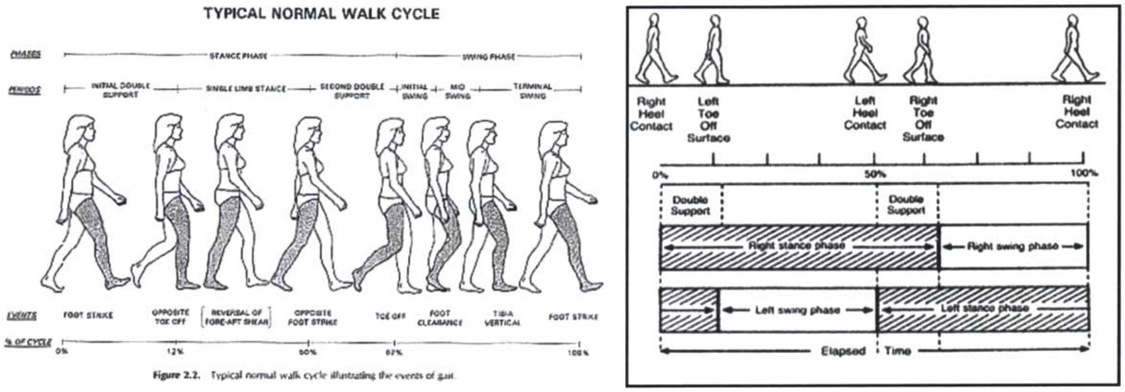


Figure 117: Two representations of the human gait cycle. Image Source: UTPA

In addition to the phases of gait, there are also parameters within each step that healthcare professionals and researchers study to understand an individual's walking health. Aspects such as step length, step width, step velocity, and maximum force are all used to understand how a person is walking [43]. The pressure profiles under each foot are also observed to see how a person's weight is distributed and whether they are putting more weight one on foot vs the other. Below is a sample output from a gait analysis conducted by a Tekscan.


Gait Table

Gait Table	Human-Flat-Feet
Number of Strikes	17
Cadence (steps/min)	84.9
Gait Time (sec)	8.49
Gait Distance (cm)	533.1
Gait Velocity (cm/sec)	62.8
Gait Velocity/Leg Length (LL/sec)	n/a

Differential Table

Differential Table	Human-Flat-Feet
Step Time Diff (sec)	-0.01
Step Length Diff (cm)	1.2
Step Velocity Diff (cm/sec)	2.3
Step Length/Leg Length Diff (ratio)	n/a
Step Width Diff (cm)	1.1
Stride Time Diff (sec)	0.01
Stride Length Diff (cm)	-3.0
Stride Velocity Diff (cm/sec)	-2.6
Maximum Force Diff (%BW)	n/a
Maximum Force Diff (kg)	-0.57
Impulse Diff (%BW*sec)	n/a
Impulse Diff (kg*sec)	0.70
Maximum Peak Pressure Diff (KPa)	-5
Foot Angle Diff (degree)	6
Gait Cycle Time Diff (sec)	0.01
Stance Time Diff (sec)	0.05
Swing Time Diff (sec)	-0.04
Single Support Time Diff (sec)	0.00

Gait Cycle Table

Gait Cycle Table (sec)	Human-Flat-Feet		
	Left	Right	Difference
Gait Cycle Time	1.40	1.41	0.01
Stance Time	0.92	0.97	0.05
Swing Time	0.48	0.44	-0.04
Single Support Time	0.45	0.45	0.00
Initial Double Support Time	0.24	0.25	0.00
Terminal Double Support Time	0.25	0.24	-0.00
Total Double Support Time	0.49	0.49	0.00
Heel Contact Time	0.58	0.64	0.06
Foot Flat Time	0.54	0.50	-0.04
Midstance Time	0.33	0.38	0.05
Propulsion Time	0.37	0.32	-0.05
Active Propulsion Time	0.15	0.08	-0.07
Passive Propulsion Time	0.22	0.25	0.02

Step-Stride Table

Step-Stride Table	Human-Flat-Feet		
	Left	Right	Difference
Step Time (sec)	0.71	0.70	-0.01
Step Length (cm)	44.0	45.2	1.2
Step Velocity (cm/sec)	62.0	64.2	2.3
Step Length/Leg Length (ratio)	n/a	n/a	n/a
Step Width (cm)	10.9	12.0	1.1
Stride Time (sec)	1.40	1.41	0.01
Stride Length (cm)	90.2	87.2	-3.0
Stride Velocity (cm/sec)	64.5	61.9	-2.6
Maximum Force (%BW)	n/a	n/a	n/a
Maximum Force (kg)	89.60	89.03	-0.57

Figure 118: Pressure output and gait cycle analysis by Tekscan Walkway™ gait analysis system. Image Source: Tekscan

8.4.2 Experimental Preparation and Setup:

Before conducting the walking tests, the system needed to be made more robust so that repeated walking would not damage any of the components. After aligning each layer of a single slave module assembly, scotch tape was used to tape the layers together. After each layer was secured to the layer either above or below it, the entire slave module assembly felt more secure and was far less prone to sliding when force was applied to it. With all three slave modules taped together, they were aligned on a large board of PVC and secured in place with gaffer's tape. Gaffer's tape was used because it is strong but does not permanently adhere to any of the surfaces it is applied to. This way, repairs or adjustments could be made easily. The corners of each slave modules were taped to the board and the seams between slave modules were taped together to prevent relative sliding. Next a 3/4" thick wooden board was cut that would protect the electronics as users were walking across the mat. The board was suspended across the width of the mat over the PCBs. The gap between the electronics and the board was provided by 1.5" x 1.5" cross section lengths of 80/20® aluminum extrusion. As was the case for the shape and pressure characterization experiments, a layer of foam was needed on top of the mat system to distribute force and activate the sensors better. The foam was placed over the mat and taped on all sides but the PCB side. The foam is stretchable which posed a problem when people were walking as the force of stepping caused the material to shift. To provide a better walking surface that would not stretch with walking, a continuous layer of gaffers tape was placed over the entire surface. With the sensor sheets secure and electronics covered, the next step was to lay out grid lines and set up the camera capturing system.

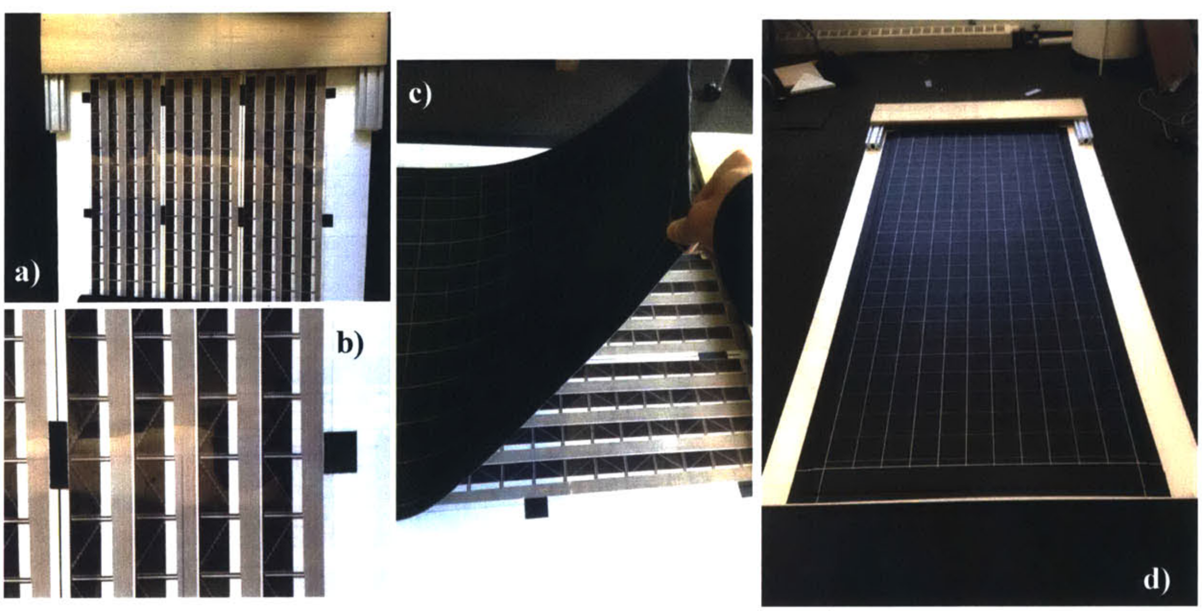


Figure 119: a) Slave modules taped in place with the wood plank protecting the PCB. b) Close up of the tape. c) Close up of the gaffer's tape/foam layer. d) View of the walking setup. Note the grid for the camera-based measurements

Camera System:

In order to have a ground truth comparison for the MATLAB generated positions and times, webcams were set up to capture each walking trial. Each video would be analyzed visually to determine the X and Y positions of each step as well as the time at which they occurred. To provide a reference for whoever analyzed the video feed, a grid was drawn on the top of the mat. The upper left and upper right sensors were used as a reference for the horizontal dimension, and the length of the strip (from the top sensor to the bottom most sensor) was used for the vertical dimension. The horizontal dimension was 880mm while the vertical dimension was 2.1m. A grid dimension of 80mm was selected because it was large enough to be visible on the camera and the horizontal dimension was divisible by this value. There were eleven markers in the horizontal and 26 in the vertical. There was a remainder of 2cm in the vertical as well. Because everything was referenced to the upper left corner of the mat, the extra 2cm did not matter.

Two wide angle (120°) webcams were used to record the video because they were able to capture a wide distance at close range. One webcam monitoring the vertical dimension was placed on a table adjacent to the mat. The webcam monitoring the horizontal was mounted half way up a vertical beam of 80/20® at the head of the mat. Together, these two cameras provided good visibility for the X and Y dimensions of the mat even with the presence of someone walking.

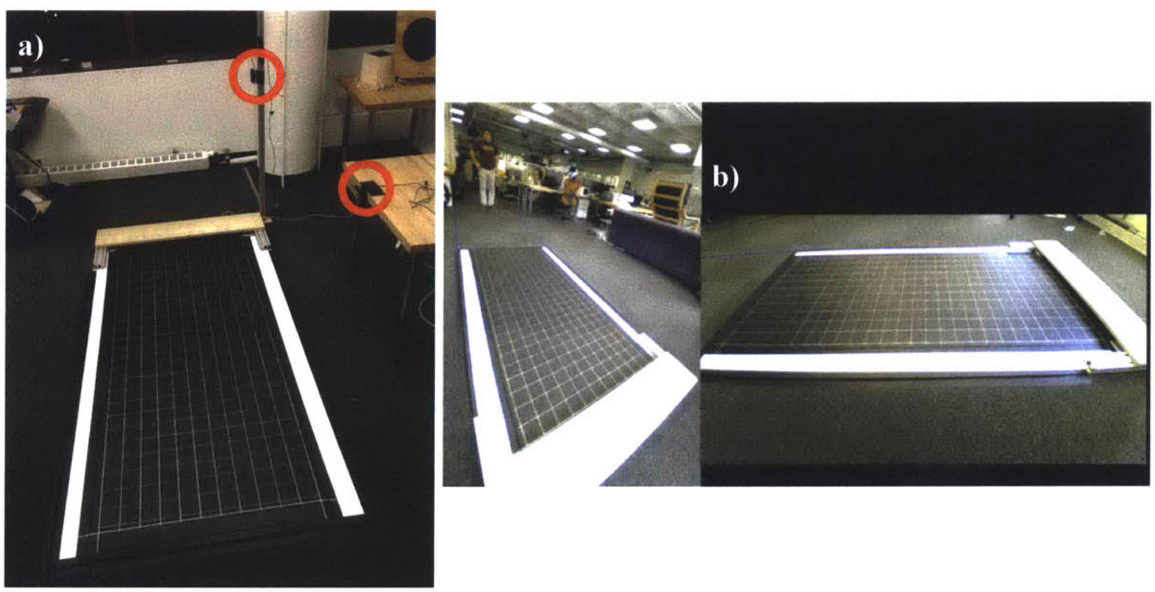


Figure 120: a) Walking experiment setup. The cameras are circled in red. b) The two camera angles that are recorded.

Because a variety of dimensions and directions will be reference, the figure below illustrates the meaning of each reference direction/dimension. The coordinate system is zeroed at the upper left corner of the mat looking from the bottom. The long dimension is Y, the smaller dimension is X. As will be explained in the video analysis section (Section 8.4.5), a footstep moving towards the head of the mat has upper, lower, left, and right around it to help identify the bounding box.

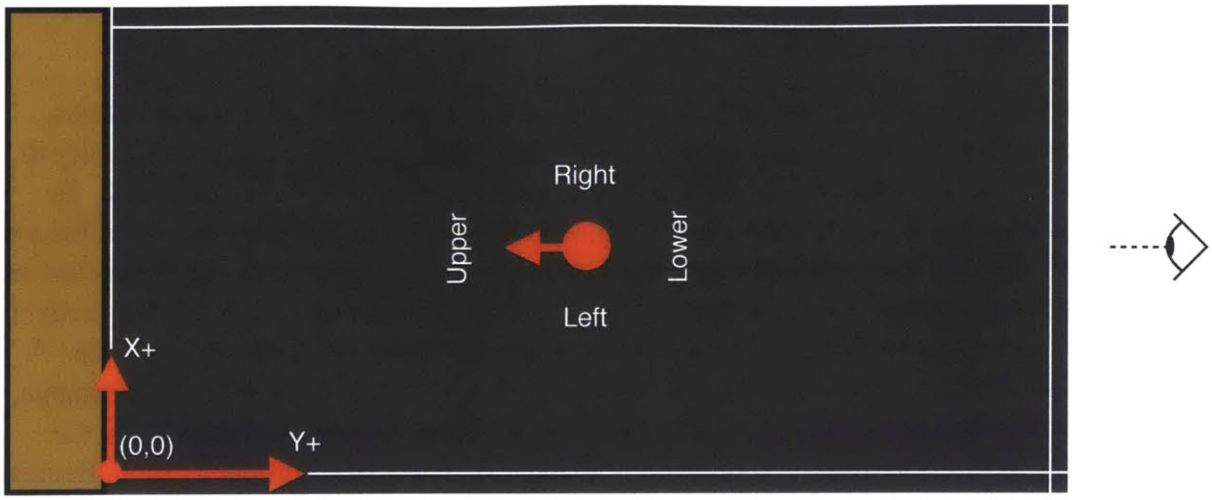


Figure 121: Reference directions and dimensions for the camera experiment. The coordinate system is zeroed at the upper left corner of the mat looking from the bottom (eye icon). The red circle with the arrow in the center indicates walking direction.

MATLAB Code Description:

For copies of the MATLAB code see Appendix C. A majority of the following description was provided by my UROP Maria Fabre, who designed and implemented in MATLAB the algorithms needed for this experiment.

MATLAB first begins by establishing a serial connection with the master Arduino. By repeatedly sending a 1, 2, and 3, to the master Arduino, MATLAB pulls the data from each of the three slave modules (see development section for how master Arduino handles data requests).

Gathering data and preprocessing: Each time information from the slave Arduinos are gathered, it is in the form of an array of 512, one byte numbers. From that array a matrix is created with the same dimensions as the whole mat and each index's value is the pressure exerted on the mat at that position. It takes one call of each slave (total of 3) to fill in this matrix representing the entire mat. This matrix is then processed into a binary matrix. Using a threshold of 50, all values below that become 0, all above become 1 (Code: *NewMainly.m* up to 111). After the matrix is created, we check that there is at least a point with a pressure above 50. In that case, then we continue with the analysis. If there is not, then we request for new data from the Arduino.

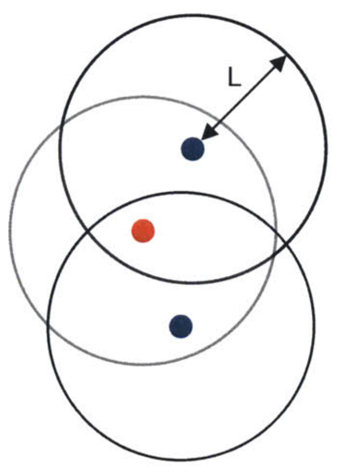
Timestamp: Each frame will have three different timestamps, these are the times when each of the slaves are checked. Through the explanations, a brief definition of how the timestamp is maintained for each point will be provided.

Clustering twice per frame:

1. **Kmeans Clustering:** First the matrix is inputted into the function *MakeCentroidT.m*, which uses the function *kmeans* from MATLAB in order to identify three distinct clusters. The number of clusters was chosen to be 3 because there are expected to be either one or two clusters representing a person walking. Since the mat is meant to only gather data from one user, we chose three clusters in order to take care of the two possible events and any noise that may occur due to step vibrations etc.. The output of this function will be the input of the next clustering function.

   Timestamp: Based on the coordinate of the centroid, we will define which of the three different timestamps provided by the Arduino corresponds to that centroid. Since all the points come from the same frame, then all the points within a cluster will have the same time unless there are points that fall within more than one of the slave modules. In that case, the centroid will define the timestamp region.

2. **Cluster:** The function *Cluster.m* will take as inputs the centroids of the three clusters identified by *MakeCentroidT.m*. We use a bounding area to help identify points within a single footstep, so an area of 10 units by 5 units is used. This comes out to roughly 1ft x .5ft. This function identifies the centroids that are within a radius of 10 units (foot length, L) and finds the centroid of those found centroids to create unique centroid for that event. We will check which of the three centroids has the most centroids within the range specified, and make all the centroids identified to be within that region a cluster (Figure 122 below).



The Cluster.m function will make all three points part of the cluster since the region of one of the centroids encloses the rest.

Figure 122: Diagram of clustering scheme. Each centroid gets a radius of 10. The centroid who's radius encloses the most other centroids is chosen.

A table with those centroids is returned, *T*, and a separate table with all the centroids that were not identified to be within the region is also returned, *N*. Thus, the *Cluster.m* function is called recursively by giving it as input those centroids that were not within the region, *N*, and stopping until the function returns an empty *N*. This iteration is executed in *NewMainly.m* line 141 to 144. **Timestamp:** it does not change for each centroid. ***CentroidPoints.m*** is called in order to find a centroid of the points in *T* at each iteration. It will just find the average on the x-coordinate, y-coordinate, and **timestamp** for all the points. Thus the output of *CentroidPoints.m* at each iteration is appended to a table that will contain a unique centroid per cluster.

Interpreting relation between frames:

Base case: If it is the first frame with at least a cluster of pressure above 50, we will save those centroids in a table, *Points*, and move to the next data gathering.

Frames after base case, determining direction: We will compare all the centroids one by one to each element in *Points, pointA*. If the distance between *pointA* and the centroid of the new frame is less than one unit away in both x- and y-coordinate, and it is the second frame, then the *pointA* will just be updated to the average of both centroids. If the points are within the length of the foot (10 units) within each other in both directions, then we can define the direction of that event, footstep. The geometry is shown in Figure (), this code is in *InsideArea.m,* it will define a new region smaller than a circle with radius of the length of the foot that will enclose the rest of the points in future frames that are within this footstep.

If the new centroid is not within the distance specified, then similarly to *Cluster.m,* a list of the new centroid and *pointA* is returned if they were close, and if they were not, then *pointA* and the new point are returned separately. This is done iteratively until we have gone over all the new centroids.

Determine θ: 2 Cases

Case 1) new point (2) **above** original point (1)

$$\theta=\tan^{-1}\left(\frac{x_2-x_1}{y_2-y_1}\right)$$

Case 2) new point (2) **below** original point (1)

$$\theta=\pi-\tan^{-1}\left(\frac{x_2-x_1}{y_2-y_1}\right)$$

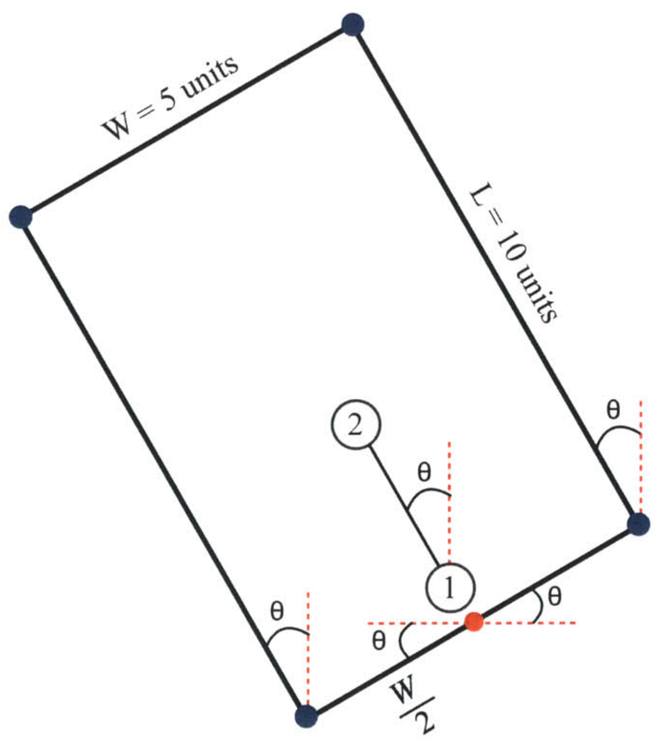


Figure 123: Diagram of calculating theta from the new centroid and bounding box coordinates.

Frames after finding directionality: If a new centroid is found to be within a point that already had a directionality, then that must mean that the new centroid is within the four coordinates identified by the angle as shown in Figure 2. This is done with convex hull, if it is defined by five points, then the point is outside the region, and if it is defined by four points, then the point is within the four original points. If a new centroid is not within the area of any of the previous points, then it becomes a new point that we should find a new area point to define directionality and so on.

A footstep is over: once point does not find any centroids to append to its list in one of the next frames, then that "event" is over. Thus we find the centroid of all the points that were identified within that region and that will be the centroid of that footfall event.

Final Clustering: In the end of all the frame reading, we find clusters of the clusters outputted once an event is done as a final step to identify the overall centroids of the steps. This takes care of any incorrect assumption such as the length not being big enough or assuming an event is over when it is not. This is executed by the code described before: *Cluster.m* and *CentroidPoints.m.*

MATLAB Results: At the end of each session of data collection, MATLAB returns a table with the X and Y coordinates of each of the footsteps as well as the timestamp of each. It also returns a binary image showing the location and profile of every step taken. A blue point indicates the centroid of each step. Also visible in some cases are green points representing centroids used to calculate the blue final centroid.

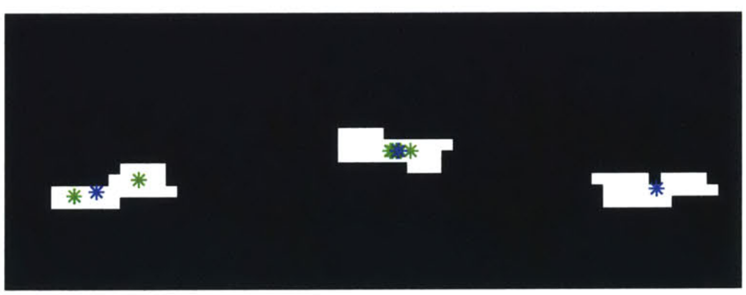


Figure 124: MATLAB output showing location of each footstep during the recording session. The blue markers are the main centroid of the step, while the green markers were used to calculate the blue centroid.

8.4.3 Experimental Procedure for General Walking Tests:

Two laptops were needed to run the system and experiment. The first laptop was connected to the mat and reading in the data. It was also running the MATLAB script that analyzed the incoming data. The second laptop was connected to the two WebCams. Using EvoCam video recording software, recordings from both webcams was captured at the same time and saved as a file for each trial [80]. These laptops for positioned on the table adjacent to the mat. I would sit that this table while running the experiments.

A walkway area leading up to the mat was marked with blue tape. The length of the space between the tape start line and the mat was about two meters. This was done to allow the subject to achieve a natural gait before walking onto the mat.

Running the General Walking Experiment:

Subjects were taken to the start line and given the same speech and demonstration.

"You will start here (indicating the blue tape line), and I will be sitting at the computer managing the software. I will start the code which takes a few seconds to initialize. When I say go, I would like you to walk at a natural pace straight ahead across the mat and step off the other side. If you need to step over the wooden plank that's fine if you need to step onto the wooden plank that is also fine. It doesn't matter which foot you begin with or how many steps you take."

While reciting the speech I demonstrated to the subject how I wanted them to walk across the mat by doing so myself.

I would then go over to the table and prepare the video recording and MATLAB script. I would first begin the video recording. Next, I started the MATLAB script. The MATLAB scripts have a five second delay at the beginning to allow the Arduino to reset. After this delay the code printed, "READY." At this point I would tell the subject to go. The subject would walk across the mat to the other side and the second or two later I would stop the video recording. The MATLAB script stopped automatically after it was done collecting and analyzing the data. I would run three trials for each person to account for faulty data. After each trial I would save a text file containing all of the data, A PDF showing the image of the steps taken, and a copy of the X and Y coordinates and timestamp of each foot step and paste into it and Excel file. I would also label and save the videos and placed them into a folder on the other computer. Ten people walked on the system, seven males and three females.

8.4.4 Experimental Procedure for Gait Abnormality Tests:

In addition to the general walking experiments, whose main purpose was to demonstrate the accuracy and repeatability of the system in analyzing the parameters of someone's gait, also of interest was the ability of the system to identify abnormal gait. Using one subject, four separate walking tests were conducted. The first test was of normal walking, this was the control. For the second test the subject wore a hinged knee brace, but using a piece of Velcro the motion of flexion was constrained to only a few degrees. This would represent a case where a user is recovering from a joint injury and they are either using crutches or a brace that would affect their natural walking gait. For the third test, a twenty pound weight was taped to the subject's right foot. The weight was difficult to move and created an imbalance between the right foot and the left foot. This would mimic a scenario representing a stroke victim or someone with a neuromuscular disorder that would create weakness in one or both of the legs or feet. For the final test, the user carried a five gallon jug of water across the mat. Though this test case did not represent a medical condition, it was of interest to see how doing physical activities such as lifting a heavy object what affect the user's gait and whether that change is perceptible.

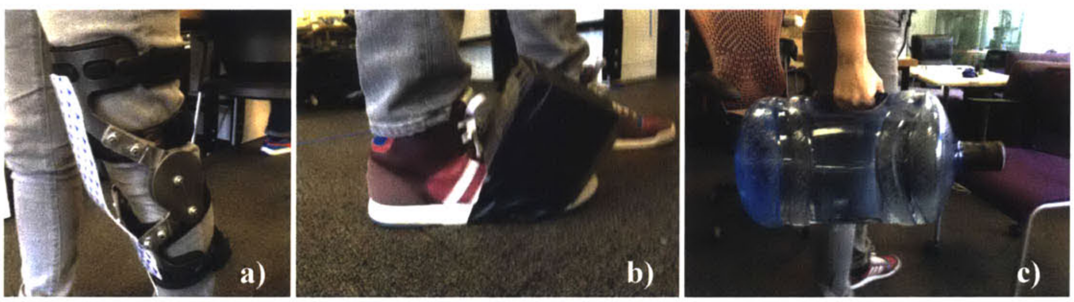


Figure 125: a) Subject wearing a constrained knee brace. b) Weight taped to the subject's right foot. c) Subject carrying a five-gallon water jug.

No video was taken in the abnormal gait experiments, and only the MATLAB code outputs including the images and data were use in the analysis.

8.4.5 Analysis and Discussion of General Walking Tests:

MATLAB Output:

The output from MATLAB included an image of the steps taken, the centroids of each step and the times at which those steps occurred. So no further processing was needed.

Video Output:

Three measurements were needed from the video. The X and Y position of each step and the time at which those steps occurred. Using those values one could calculate a variety of useful gait parameters.

Step Times:

The MATLAB code operates by taking a set of frames representing a single foot step, averaging the timestamps of each frame and returning that as the "averaged" time at which that step occurred. The same needed to be done for the videos.

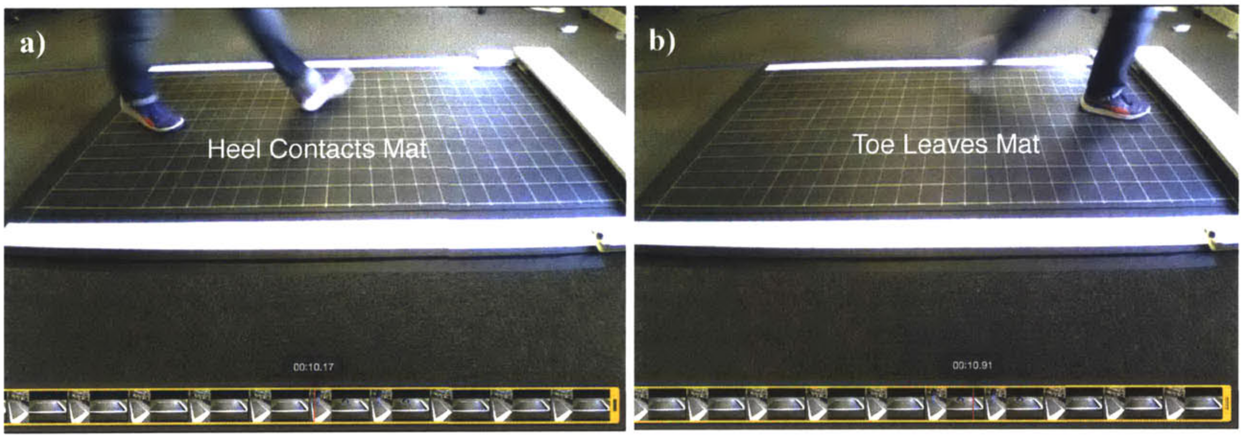


Figure 126: Finding the time of each step. a) First time occurs when the heel first contacts the mat. b) Second time occurs when the toe of the same foot leaves the mat.

The times were determined by the frame timestamp in each video, which had millisecond resolution. These two times were averaged to give the approximate time of the center of the step so that the comparison is more true to the MATLAB determined time.

Step Position:

For the step positions in the video, the steps were determined by finding the center of the rectangle of grids squares the foot occupied at its flattest point in the step. Using the X and Y grids as guides, a box was drawn around the foot. The center of that box was approximated as the centroid position of the footstep. Sometimes the foot did not land in the center of a set of squares so few of the potential cases are highlighted below. The red lines show where the bounding box would be drawn.

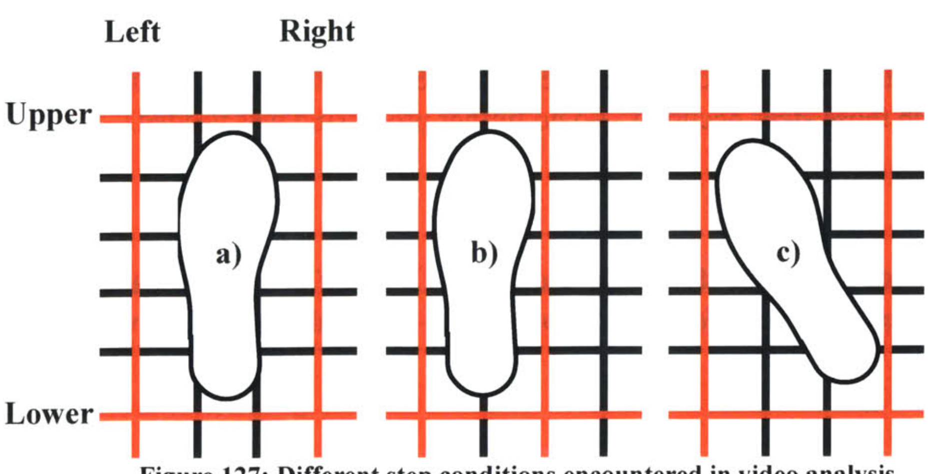


Figure 127: Different step conditions encountered in video analysis. a) Step between two vertical lines. b) Step on the middle of a vertical line. c) Step at an angle across vertical lines.

The X and Y position was calculated by finding the center of the bounding box created by the Upper, Lower, Left and Right lines. Note from (fig whatever) that the Lower line is higher value than the Upper line. And the Right line is a higher value than the Left line.

$$X = \frac{Left+Right}{2}$$

$$Y = \frac{Upper+Lower}{2}$$

A majority of subjects took three steps on the mat; there was only one trial where somebody took four, and one where somebody took two. Those trials were discarded. For both MATLAB and the video analysis, the was organized into the following format:

Table 5: Format of MATLAB and Video data outputs.

Step #	Time (hr:min:sec)	X Coordinate (index)	Y Coordinate (index)
Step 1	H:M:S	X	Y
Step 2	H:M:S	X	Y
Step 3	H:M:S	X	Y

To understand how the MATLAB determined values differed from the video determined values, the absolute value of the difference between the X, Y, and normalized time values of each step for MATLAB and the video was calculated. This allowed us to see the magnitude of the discrepancies and on which step they were occurring.

The time values were normalized by subtracting the time of the first step from each of the subsequent steps. This gave the time between the second and the first, and the third and the first steps.

So rather than: $(H{:}M{:}S_1, H{:}M{:}S_2, H{:}M{:}S_3)$

Where H, M, and S are hours, minutes and seconds, the values become:

$$(0, H{:}M{:}S_2 - H{:}M{:}S_1, H{:}M{:}S_3 - H{:}M{:}S_1)$$

This was done to both the video and MATLAB time values. It was decided that the first step would serve as the absolute reference point and subsequent times would be relative to that point. This is why the first time for both video and MATLAB is 0.

Next, the X and Y coordinates of both the MATLAB and video outputs needed to be converted into their real world position. The conversion of the video coordinates was easy. Because each grid line was spaced 8cm apart, the coordinates only needed to be multiplied by eight to get the centimeter position relative to the upper left corner of the mat.

The absolute position of the MATLAB coordinates was a bit more difficult. The mat is comprised of three slave module assemblies; each module is 8 sensing cells wide. As shown in the (section), the sensing "cells" are actually two adjacent triangular sensors overlapping each other. Each cell is 33.5mm x 64mm with a 2mm gap between adjacent cells in the horizontal and vertical dimensions.

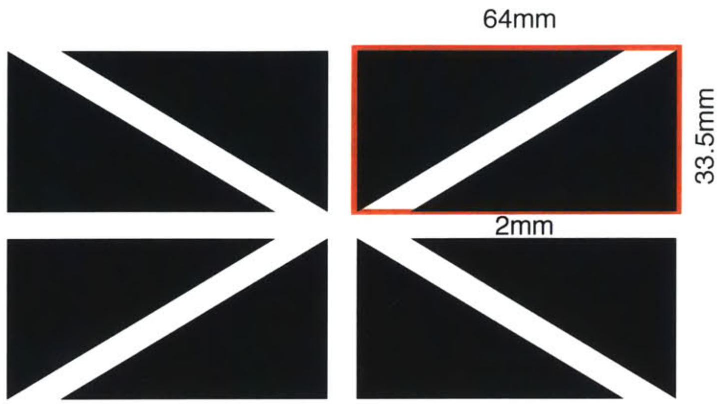


Figure 128: Diagram of a set of sensors on the system. A single cell is outlined in red. The dimensions are indicated.

If one were assuming that each sensor was activated at the centroid, the centroid of each triangular sensor (as for any right triangle) is 1/3 the length of its base and 1/3 the length of its height.

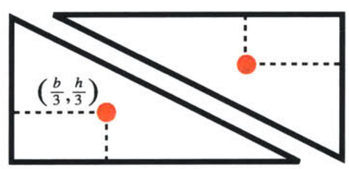


Figure 129: Location of the centroid of a right triangle.

But looking at the true centroid of every individual sensor, one can see (above figure) that there are differing distances between centroids both in the horizontal and vertical dimensions. To simplify the calculation, I assumed that each cell was actually two rectangles, of 33.5mm x 31mm with a 2mm gap in the vertical between sensors.

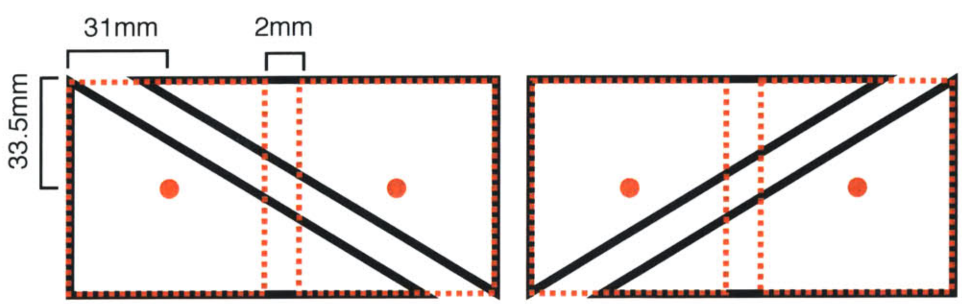


Figure 130: Centroid approximation for each cell of two sensors. This ensures the centroids form a rectangular grid.

When calculating the true dimensions of each centroid, there were a few spaces to take into account. The 2mm spaces between adjacent cells as well as a 17mm gap between the sheets in the assembly.

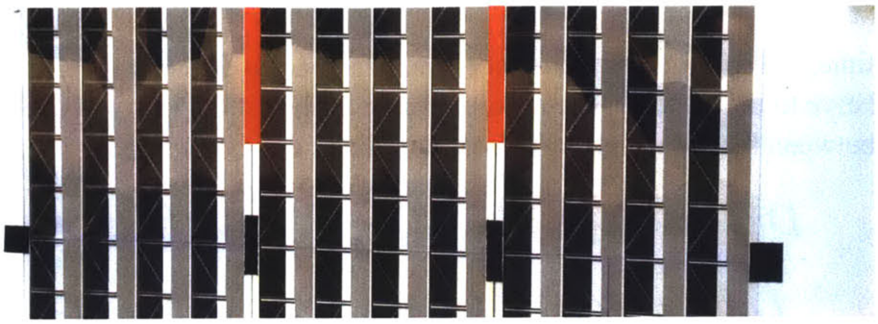


Figure 131: Gap between slave modules in the mat assembly.

A pseudo-function was used to determine the position of each sensor taking into account that some would include the 17mm gap in the horizontal once or twice depending on the coordinate.

It should be noted that MATLAB matrix indices begin with one rather than zero. So the first coordinate would actually be (1,1) rather than (0,0).

After working out the math, here are the formulae to convert the X and Y coordinates MATLAB returns into the physical location:

Since the Y dimension does not depend on the 17mm gaps, it had a single formula:

$$y_{true} = \frac{31}{2} \cdot y_{coordinate} + (y_{coordinate} - 1) \cdot \left(\frac{31}{2} + 2\right)$$

Where every number is in millimeters.

If the x coordinate is from 0.5 - 8.5 then the position is calculated as:

$$x_{true} = \frac{33.5}{2} \cdot x_{coordinate} + (x_{coordinate} - 1) \cdot \left(\frac{33.5}{2} + 2\right)$$

If the x coordinate is from 8.5 - 16.5 then the position is calculated as:

$$x_{true} = \frac{33.5}{2} \cdot x_{coordinate} + 17 + (x_{coordinate} - 1) \cdot \left(\frac{33.5}{2} + 2\right)$$

If the x coordinate is from 16.5 - 24.5 then the position is calculated as:

$$x_{true} = \frac{33.5}{2} \cdot x_{coordinate} + 34 + (x_{coordinate} - 1) \cdot \left(\frac{33.5}{2} + 2\right)$$

So if one were looking at the first sensor, having a coordinate of (1,1), the X and Y positions in cm would be:

$$\left(\frac{33.5}{2}, \frac{31}{2}\right)$$

Unlike the time, the coordinates were not normalized because the distances were already absolute relative to the system. The absolute value of the differences was used to understand the agreement between the video and MATLAB coordinates at every step.

$$Diff_x = (|X1_v - X1_m|, |X2_v - X2_m|, |X3_v - X3_m|)$$

$$Diff_y = (|Y1_v - Y1_m|, |Y2_v - Y2_m|, |Y3_v - Y3_m|)$$

Where the subscript *v* indicates the video value and the subscript *m* indicates the MATLAB value. The mean and standard deviation of the coordinate and time differences are summarized below:

Table 6: Mean and standard deviation for the distance and time differences for each step. Since the time at Step 1 was 0 for both video and MATLAB, only Step 2 and Step 3 are listed.

	Diff S1 X (cm)	Diff S2 X (cm)	Diff S3 X (cm)	Diff S1 Y (cm)	Diff S2 Y (cm)	Diff S3 Y (cm)	Diff S2 t (cm)	Diff S3 t (cm)
Mean (cm)	1.62	1.55	1.17	4.73	4.04	2.79	0.19	0.26
Standard Deviation (cm)	1.36	1.14	0.99	3.62	3.63	2.06	0.11	0.16

Discussion:

As can be seen in the table, there is fairly close agreement between the MATLAB generated centroids and the video analyzed centroids. Especially in the X dimension, the average differences were in the 1-2cm ranges with a standard deviation of the same. Though this means that there was also a large spread in the values, the values themselves were still relatively low. The Y differences were on average larger, with a comparable standard deviation.

The normalized times for step 2 and 3 had poor agreement. Surveying all the subjects, each step took anywhere from 0.35 to about 0.75 seconds. The average difference of 0.19s and 0.26s for step 2 and step 3 respectively indicates that at the low end of the step times, the difference is almost the value of an entire step.

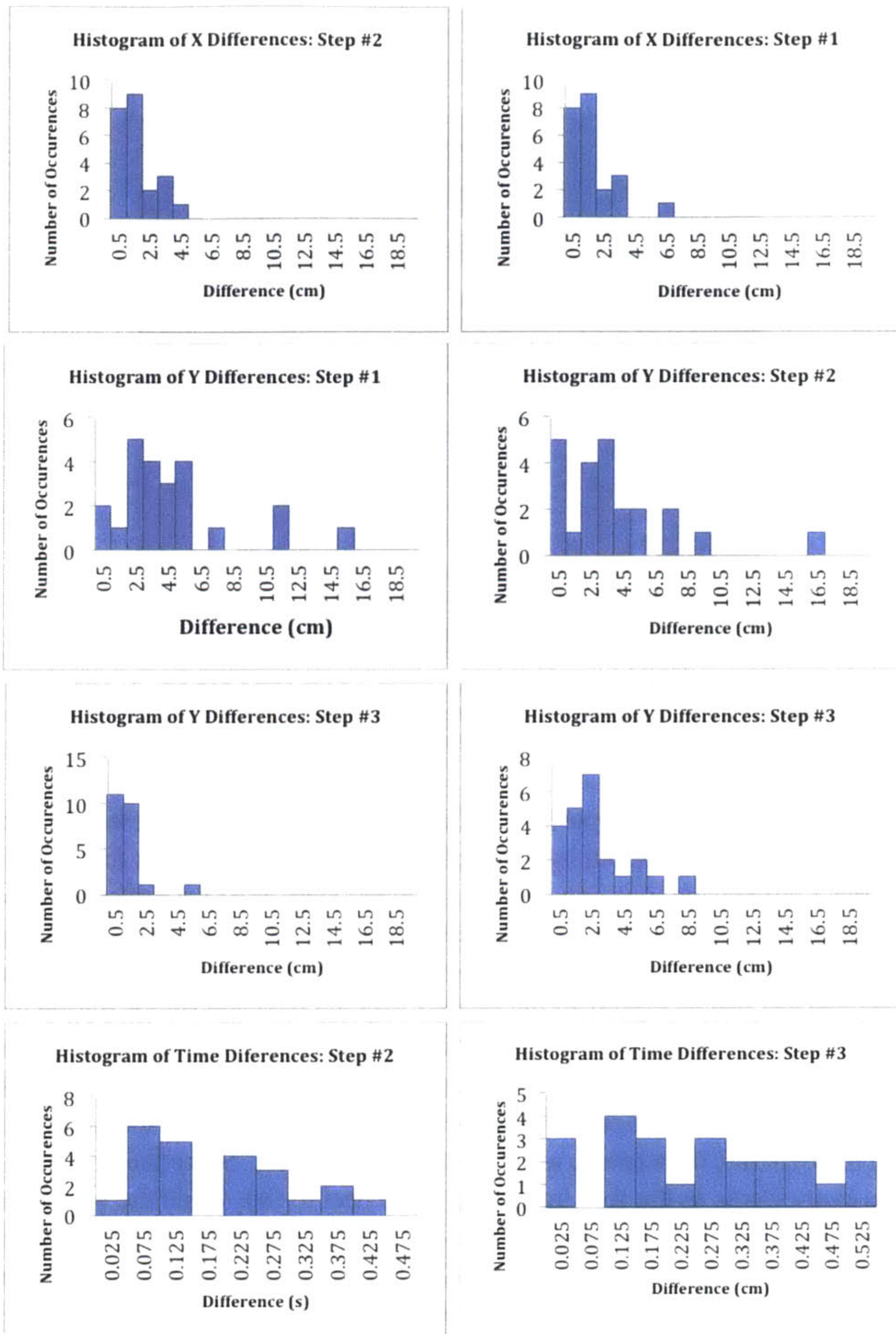


Figure 132: Histograms for the distance and time differences for each step listed in Table 6 above.

Centroid Discrepancy:

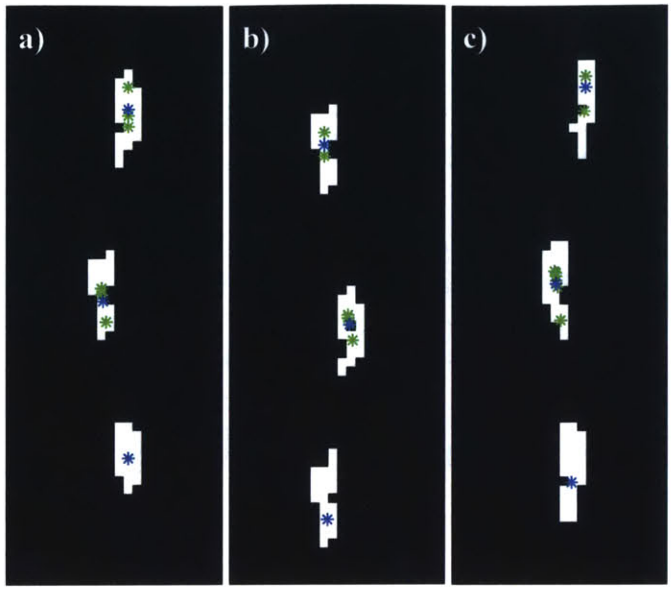


Figure 133: Three separate walking trials for the same person. a) The blue centroids are all centered in the step. b) The blue centroid in Step 1 is closer to the heel. c) The blue centroid in Step 3 is closer to the toe.

The observed discrepancies seen in positions of each footstep can be understood by looking at the above images, which were output by MATLAB for the same person on three separate trials. Starting with the X dimension, one can see that for every step, MATLAB places the blue centroid in the center of the white space representing the footprint. This is probably why the average error between the video and MATLAB X positions is on the order of 1 cm. Looking at the Y dimension one can see that in Figure #b on the first step the blue centroid is placed closer to the heel, and in Figure #c on the third step, MATLAB places the blue centroid closer to the toe. This feature occurs on a few other trials among different subjects. Since the MATLAB code generates a composite image by stitching together multiple frames of activated sensors, errors seem to occur most in the dynamic dimension, which is the Y dimension. Since the length of the foot area increases every frame as the foot transitions from heel to flat, while the width stays within the same range, the centroid of the length of the foot seems more prone to errors. Another source of error could be the predefined search area of 10 x 5 sensors (300mm x 150mm). Since the code is only looking for points within this region at every step, people who have a larger foot than this will have some data truncated. This could shift the position of the centroid as well. 150mm is more than enough to capture most foot widths so data in this dimension would less likely be lost. While this hypothesis has not been verified, surveying the various images of the trials shows that the centroids are often aligned in the width dimension but not always in the length. The video determined centroids are less prone to error because it simply involves taking the center of the bounding box around the foot, which will give a very close approximation of the actual centroid of the step.

Time Discrepancy:

In MATLAB, every frame read has its own time stamp. These time stamps are averaged to give the approximate time of the middle of the step. For the video analysis, the times are determined in a similar way, where the time at which the heel makes contact and the time at which the toe leaves the floor are averaged to give the time of the middle of that step. Both methods are prone to error. Both involve averaging which doesn't give the true value of the center of the step, only an approximation. As demonstrated by the centroid calculations, the MATLAB method has been shown to be error prone. The video method is equally error prone because every stage of the step is not captured. Since 30fps is slower than many of the leg swing motions that occur in walking, some of the finer events are missed. The frame where the heel seems to strike could be the actual moment or it could be or a few milliseconds after depending on when the frame started. At this time, because the video is more consistent, the times from the video can be trusted more, but neither (MATLAB vs. video) was a particularly accurate method for this experiment.

8.4.6 Analysis and Discussion of Gait Abnormality Tests:

The values from the abnormal walking tests were calculated the same way as the general walking test values. Since there was no video taken, the values are compared to themselves. Rather than the coordinates and times at each step, the difference for X, Y, and time are taken between subsequent steps. Since some of the abnormalities cause changes in gait, of interest is the step length, step width, and step times rather than just the coordinates and event time.

Table 7: Mean and standard deviation of step length and width for the normal and various abnormal gait walking tests.

	Normal L (cm)	Normal W (cm)	Knee L (cm)	Knee W (cm)	Ankle L (cm)	Ankle W (cm)	Carry L (cm)	Carry W (cm)
Mean (cm)	82.51	10.86	76.97	15.55	48.76	16.24	58.52	17.29
Standard Deviation (cm)	11.47	3.6	16.96	4.71	7.59	5.01	6.6	3.8

Table 8: Mean and standard deviation of step times for the normal and knee walking tests.

	Normal T1	Normal T2	Knee T1	Knee T2	Knee T3
Mean (cm)	0.43	0.53	0.45	0.78	0.36
Standard Deviation (cm)	0.16	0.18	0.18	0.16	0

Table 9: Mean and standard deviation of step times for the ankle and weight carrying tests.

	Ankle T1	Ankle T2	Ankle T3	Ankle T4	Carry T1	Carry T2	Carry T3
Mean (cm)	0.72	0.84	0.92	1.21	0.58	0.51	0.44
Standard Deviation (cm)	0.002	0.21	0.34	0	0.09	0.18	0.07

Table 10: Average number of steps for the normal and abnormal gait tests.

	Normal	Knee	Ankle	Water
Avg Number of Steps	3	3.25	4.33	4

For each of trials described in the introduction, a set of lengths, widths, and times were analyzed. The obvious common feature between the three abnormal cases is that there was a reduction in the average step length and an increase in the average step width. The weight attached to the ankle and water-jug-carrying cases both required 4 or more steps. Like the general walking tests, the standard deviation of the width values was less than the length values. The times for the most part had low standard deviations meaning there was more agreement between time values among different trials of the same abnormality case.

Wearing a knee brace didn't seem to significantly slow the time between steps. Carrying the jug of water showed some time increase but not much, while attaching a weight to one ankle did slow down the walking pace a comparatively large amount.

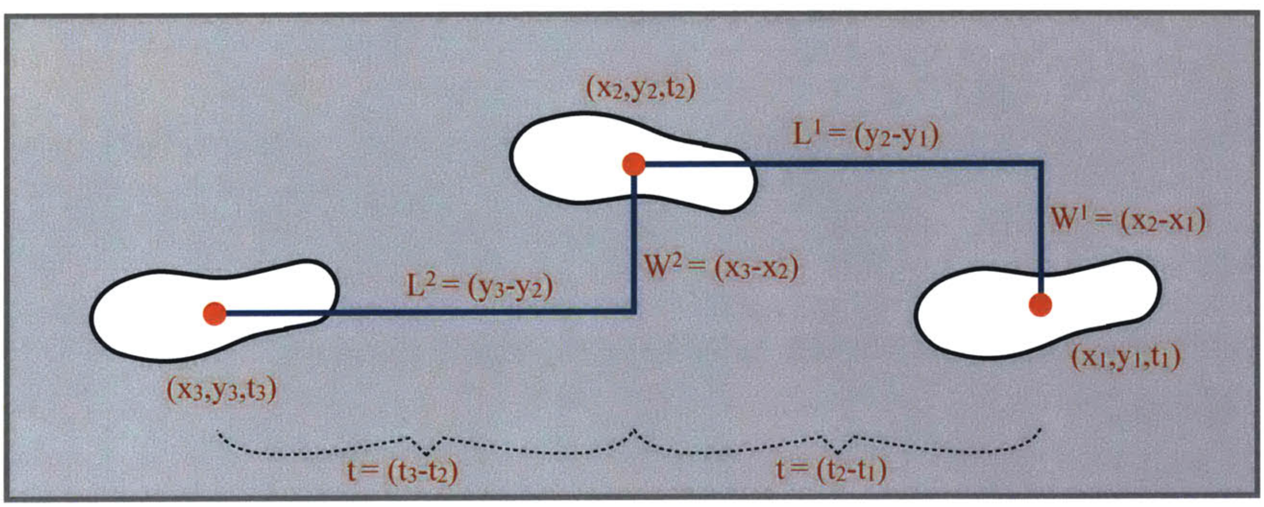


Figure 134: Diagram indicating how step width, length and time was calculated for the abnormal walking experiments.

8.4.7 Closing Remarks:

In the generalized walking experiments an algorithm analyzing the data from the system was demonstrated who's output showed close agreement to the values gathered from video feed of the same walking tests. Though there was some error in some of the dimensions and times of each step, we are confident that refinement of the algorithm would yield times closer to the actual values. The abnormal gait tests demonstrated that the system is able to recognize different average step widths, lengths, and times. Especially for the case of the weight attached to the ankle, the system was able to show a large variation between the collected values and the normal values. Monitoring for changes in gait over a long period of time is the main motivation behind constructing this system. Since the system is able to recognize different gait parameters, more refinement could allow it to determine whether a problem such as a stroke or injury to the limb has occurred. By comparing how the person is currently walking to a a set of known normal values for the individual, the system could keep a constant eye out for any slow deviations or sudden shifts.

9. Future Work:

9.1 Wireless Connectivity:

Currently to collect data, an assembly of sensor module slaves needs to be connected to a master Arduino, who is then connected to a computer via USB. As described in the Section 6.4.3, the slaves send their data to the master via I^2C and the master then forwards that data to the computer via serial. The next step for the system's communication would be to enable different assemblages of sensor modules to be placed around a home and wirelessly communicate their data. Usually Bluetooth is used as a simple wireless protocol, but I needed something that was faster and higher throughput so XBee modules using the ZigBee wireless standard were investigated as a potential wireless solution.

9.1.1 XBee:

From Sparkfun: "XBee modules are embedded solutions providing wireless end-point connectivity to devices. These modules use the IEEE 802.15.4 networking protocol for fast point-to-multipoint or peer-to-peer networking. They are designed for high-throughput applications requiring low latency and predictable communication timing" [69]

Figure 135: XBee module with wire antenna. Image Source: SparkFun

9.1.2 ZigBee:

From the ZigBee Alliance: "ZigBee is the only open, global wireless standard to provide the foundation for the Internet of Things by enabling simple and smart objects to work together, improving comfort and efficiency in everyday life" [70]. The ZigBee Alliance "Promotes worldwide adoption of ZigBee as the leading wirelessly networked, sensing and control standard for use in consumer, commercial and industrial areas. [70]

9.1.3 Using XBee With the Thesis Prototype:

Running the XBee on the Arduino to create a wireless serial connection with the computer is very simple. In the Arduino code, a SoftwareSerial is defined that uses two of the digital pins on the Arduino [76]. On the computer side, plugging in another XBee to the USB creates allows the computer to communicate wirelessly with the XBee on the Arduino. The XBee registers as an additional Serial Port on the computer, and by connecting to it, a wireless connection is established between the mat and computer as shown below.

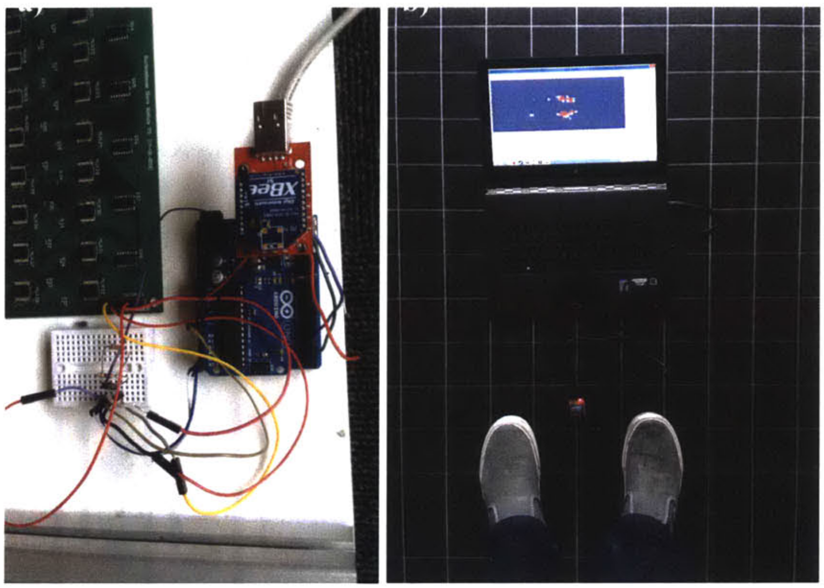


Figure 136: a) XBee connected to the master Arduino at the end of the slave module assembly. b) The complimentary XBee connected to the computer via USB. The visualization on the computer shows the presence of my feet. This data is being sent wirelessly.

An XBee has a unique serial number that it can be identified by, but it does not need to know the serial number of an other XBee in order to communicate with it. Instead, both XBees are set to talk on the same network by designating a common network ID, or PAN ID, which is a 16-bit number. Then, each xBee chooses two 16-bit addresses. The first, referred to as the MY address, designates the address with which an xBee will transmit information out. The second, or the DL address, is the address from which an xBee will receive information. Lastly, the baud rate, or the transmission rate, must be the same for both devices.

The code used was a modified version of a two-way communication script from smith College [71]. We adapted it for use on the master Arduino to transmit the data it received from all the slaves. There was some difficulty receiving data on a Mac OS X machine, but the protocol worked fine on a Windows 8 machine.

Future designs of the board would allow for the attachment of an XBee/battery master that would clip on the end of an assembly of slave modules, power the boards, collect the data and send it to a central location wirelessly. Various algorithms could also be designed to optimize the use of wireless and essentially put the mat in sleep mode until activity is detected.

9.2 Alternative Sensor Configurations:

As described in Section 5.3 a very common approach for reading or writing to a matrix of sensors or components is using a row and column configuration. For the final prototype, a single-sided sensor sheet was originally desired, but we later realized that another sheet was needed to fill in the gaps created by all the traces to each sensor. Though the prototype worked as intended, the number of traces necessary could be reduced by a factor of seventeen if a row and column configuration was used. Rather than requiring 1536 connections for each sensor (24*64), only 88 are needed (24 + 64) if a row and column scheme was used.

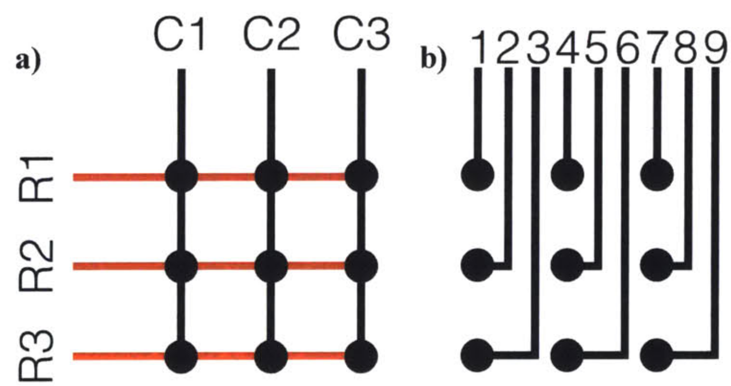


Figure 137: a) Row and column configuration for reading nine sensors. Only six pins are needed; R1, R2, R3, C1, C2, and C3. b) One-sensor-per-pin configuration. To read every location nine pins are needed.

In addition to reducing the number of multiplexers needed by 26 compared to the existing PCBs, the reduction of connections can allow the boards to be made smaller to incorporate more unobtrusively into the different application locations.

9.2.1 Different Sensor Densities in the Same Assembly:

For the prototype described in this thesis, a single sensor size was used on every module. This was done to achieve smaller-than-foot resolution at any point on the mat. But in a real-world setting, it may not be necessary to have that fine a resolution. For example, why would a bathroom need high resolution if one generally move around very little in a bathroom? A hallway on the other hand is a prime candidate for a more dense array of sensors because a hallway is a good location to pick out several footsteps to analyze. A solution could be to have modules that have differently sized sensors. These different modules can still be connected together, but rather than having uniform sensing size which would be more expensive, the density varies depending on location. Alternatively, different densities and sensor sizes can be designed into individual modules (Figure 138b). Incorporating fine and rough sensing capabilities into a single module would allow the designer the freedom to use whichever data they desired. if a fine mesh is wanted, all the sensors would be operational. If they want want rough positioning, then the fewer larger sensors would be activated.

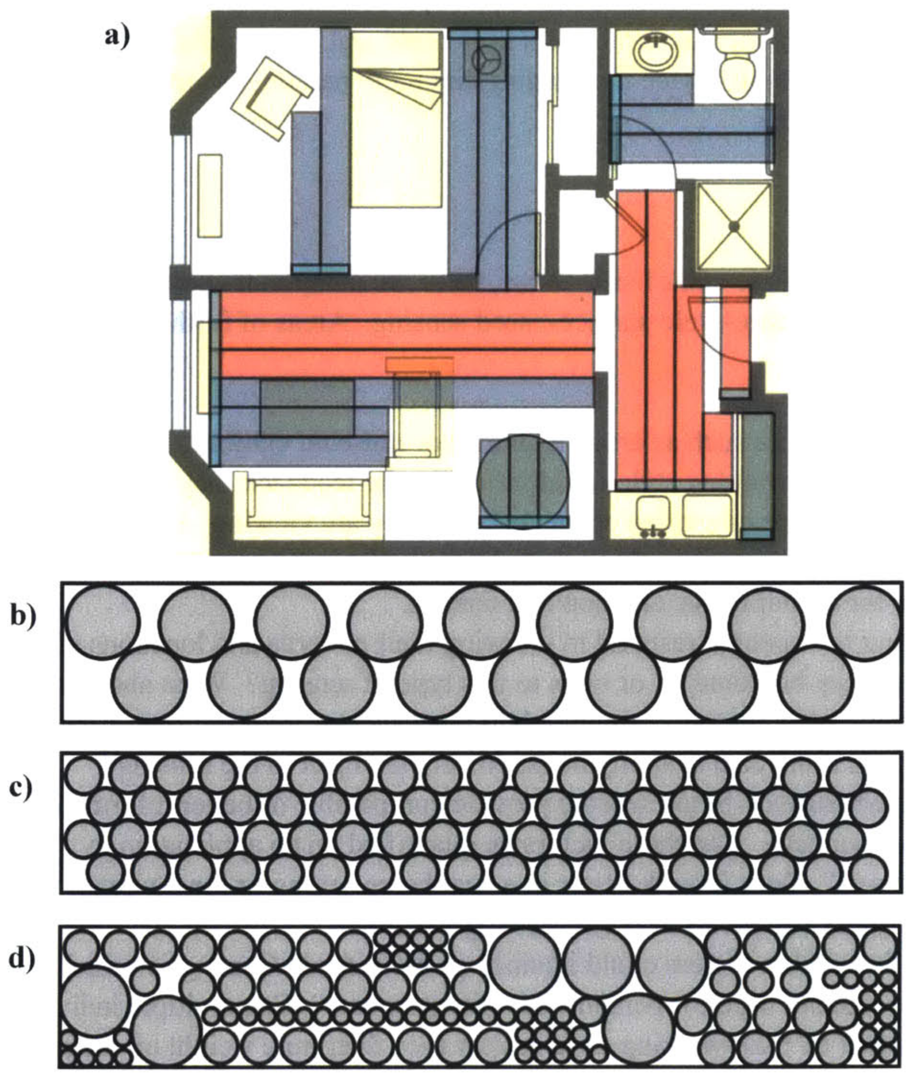


Figure 138: a) Floor plan of an apartment showing multiple assemblies of sensor modules placed at various locations. The denser sensors are red and are placed in the major walkways and hallways. The less dense sensors are blue because high resolution is not needed in those location, just presence detection. b) Sparse sensor module with large sensors. c) Dense sensor module with small sensors. d) Example of sensor module with multiple sensor sizes.

9.3 User Testing with Multiple Surfaces:

Something we would like is to understand how one might interact and use such a system in a real-world setting. Of interest would be manufacturing multiple surfaces and placing them around a volunteer's apartment. Using a survey or even camera-based system to understand which activities are being performed, we would like to see if the system could generate useful activity and movement information about the user. If certain locations were tagged as "kitchen" or "refrigerator", how often would the system be able to correctly ID activity based on location and duration? Can the system be trained to recognize these activities? How nuanced an activity is the system able to record and positively ID? We would also like to design a few applications ranging from entertainment to smart-element control that a user could test. We would like to see if using this system as an input device is engaging and more natural, or rather more burdensome and difficult.

9.4 Everything Else

Besides the more immediate work listed above, there are many more potential areas for investigation related to large-scale surface-based sensing. Areas of further exploration could include:

- Immersive experiences such as art installations or museum exhibits. The location of the person and their walking style could influence the feedback they receive from the art or it could trigger explanation of the material they are viewing.
- Performance, such as a sensing floor or stage for ballet and dance performances or an input device for digital music creation and control.
- Understanding how people respond to knowing their patterns and locations of walking. Are they bothered by or open to this type of sensing? What about more public locations such as stores or public buildings such as train stations, airports or, libraries? Even if the data is anonymous, do people object to the sensing?
- Testing whether surface-based sensors provide comparable or better information compared to a camera. Can these factors be quantified on an application by application basis? Does it work better in a healthcare setting or are the retail applications more beneficial over the existing solutions?
- Seeing whether such a system could identify an individual. Can a gait-based password be reliably used for personal security purposes? Can multiple individuals be differentiated by their walking style or how they distribute weight on their feet?
- How can these surfaces be used to help those with physical disabilities? Are there tasks or current interactions that could be made easier with this type of sensing?
- Keeping track of items around home or space. The location of objects like keys, important medicine, or documents are given to the user by the system.
- Capturing diet information by identifying plates on a table and tracking weight over time to indicate rate of food consumption.

- Knowing how many guests someone is having over and robotically setting the table or moving the furniture around to accommodate them.
- Preparing appliances, such as turning on the shower or starting the oven if the system knows that the user is walking towards it with the intention of using it.
- Identifying children in a classroom and knowing which activities they enjoy and objects they play with the most. This could allow for more tailored teaching and provide the teachers with information about what subjects most interest the student.
- Monitoring presence of doctors and nurses in a hospital room. If there is uncertainty in when medicine was administered, the system could provide a history of the room for that day that could be compared to the doctor's charts.

10. Conclusion:

It will not be long before pervasive sensing and ubiquitous computing become a utility rather than a novelty. The falling costs of components, ease of manufacturing, and a culture that is more aware of and dependent on connectivity will enable distributed intelligence to be placed in many more unique locations. Soon there will no longer be a need to quantify yourself because the environment will quantify it for you. Sensors will passively collect information about a user and generate optimizations and interventions that will improve the individual's life, experiences, and interactions with the objects and people around them.

This work describes the design and construction of a system for equipping surfaces with coordinate and pressure sensitivity. Such a system would fit into both current and future home environments. Not only would it provide an additional source of information about the user, but it would also enable a new means with which to interact and control various elements of the current and future connected home environment.

Current work in the field of connected devices, the internet of things, and more specifically surface-based person sensing was described in this thesis. Several applications where such a system could have a large impact were explored and tested. Problems related to aging in place and related gait analysis, daily activity monitoring, and fall detection could benefit greatly from the deployment of large-scale, surface-based sensing in the home. Secondary areas such as entertainment, digital presence, and education and fitness could use the sensor as a hardware platform on which to design and build interactive and meaningful applications.

The development process for the final prototype was discussed at length, providing insight into the design choices made at every step and the reasons for doing so. The final prototype, including the sensor construction, hardware design, software control and communication were described in detail to encourage others to take the design forward or use it for their own purposes.

The experiments conducted on the completed system showed promising results. The sensor characterization tests indicated good repeatability of measurement for the individual sensors as well as a fast response time to loading and unloading. Multiple sensor tests demonstrated that higher weight values could be measured by distributing the load over multiple sensors. The shape characterization tests demonstrated that feet positioning and shape orientation are easy to distinguish in the visualizations. The different weights loading each shape was also recognizable. The walking tests demonstrated fairly good agreement between the algorithmically determined footstep parameters and the video determined parameters. The abnormal gait tests demonstrated that the system is able to report back different gait parameters as compared to the normal.

Though not discussed in the development section, the final cost of the system was somewhat expensive, but with dedicated sensor sheet manufacturing and circuit fabrication the cost could certainly be reduced. The complete cost of parts for the final assembly of three modules was on the order of $5000 (conservative estimate). Thus the system, covering an area of $2m^2$ or about $21.5ft^2$, costs roughly $\$233/ft^2$.

The visualizations, though not connected to the system, anticipated a time when the information gathered from pervasive sensing will need to be conveyed to different groups of people in a clear and intuitive manner. From family members to doctors to the individuals themselves, ambient and familiar visualization will enable complex sensor results to be interpreted quickly and dealt with appropriately.

Future work will explore the use of multiple surfaces within a single space to see if the data and interfaces they provide are an improvement or detriment as compared to existing methods. Different sensor architectures and communication methods are also of interest so that future versions of the sensor are more unobtrusive, durable, and multifunctional. Many other ideas were also presented in the hope that others will be inspired to prototype and test new applications and features using the system.

Based on it's potential to positively contribute to multiple areas, particularly those related to elderly or disabled populations, as well as its simplicity of construction and manufacturing processes used, the author feels that the described prototype could be the basis for a system to provide awareness and utility to the often overlooked, yet highly trafficked surfaces of a space.

References:

Aging in Place:

1. Swartz, Angela. "Lively Sensors Let Relatives Keep Track of Seniors." SFGate. April 2, 2014. Accessed May 4, 2015. http://www.sfgate.com/technology/article/Lively-sensors-let-relatives-keep-track-of-seniors-5368327.php.

2. Eisenberg, Anne. "A 'Smart' Home, to Avoid the Nursing Home." The New York Times. April 5, 2001. Accessed May 4, 2015. http://www.nytimes.com/2001/04/05/technology/a-smart-home-to-avoid-the-nursing-home.html.

3. Taub, Eric. "The Technology for Monitoring Elderly Relatives." The New York Times. July 28, 2010. Accessed May 4, 2015. http://www.nytimes.com/2010/07/29/garden/29hometech.html?_r=0.

4. Humphrey, Michael. "How Will Millenials Face Their Parents' Aging? With Wearable Tech And Human Touch." Forbes. August 31, 2014. Accessed May 4, 2015. http://www.forbes.com/sites/michaelhumphrey/2014/08/31/how-will-millennials-face-their-parents-aging-with-wearable-tech-and-human-touch/.

5. Stunt, Victoria. "Use of Surveillance Tech to Monitor Seniors at Home on Rise - Technology & Science - CBC News." CBC News. March 9, 2014. Accessed May 4, 2015. http://www.cbc.ca/news/technology/use-of-surveillance-tech-to-monitor-seniors-at-home-on-rise-1.2535677.

6. "Trends in Aging: Wearable Tech and Sensors for Seniors." HL7 Standards. September 8, 2014. Accessed May 4, 2015. http://www.hl7standards.com/blog/2014/09/18/trends-aging-wearable-tech/

7. Herz, J.C. "Headline: Wearables Are Totally Failing the People Who Need Them Most." Wired.com. November 6, 2014. Accessed May 4, 2015. http://www.wired.com/2014/11/where-fitness-trackers-fail/.

8. Mynatt, Elizabeth D., Essa, Irfan, and Rogers, Wendy. "Increasing the Opportunities for Aging in Place." *CUU '00 Proceedings on the 2000 conference on Universal Usability*, 2000, pp 65-71

9. Mynatt, Elizabeth D., Rowan, Jim, Craighill, Sarah and Jacobs, Annie. "Digital family portraits: supporting peace of mind for extended family members." *CHI '01 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2001, pp. 333-340.

10. Mynatt, Elizabeth D., and Rowan, Jim. "Cross-Generation Communication via Digital Picture Frames." 2002.

11. Cahn-Weiner, Deborah A., et. al. "Cognitive and neuroimaging predictors of instrumental activities of daily living." The Journal of the International Neuropsychological Society. 2007. Accessed May 8, 2015. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2877031/

12. 2015 Alzheimers Disease Facts and Figures. alz.org. Accessed May 8, 2015. http://www.alz.org/facts/overview.asp

13. OnKöl. Accessed May 8, 2015. http://www.onkol.net/

14. Healthy Places Terminology. CDC.gov. Accessed May 8, 2015. http://www.cdc.gov/healthyplaces/terminology.htm

15. Hoffman, Jan. "Watchful Eye in Nursing Homes." The New York Times. November 18, 2013. Accessed May 8, 2015. http://well.blogs.nytimes.com/2013/11/18/watchful-eye-in-nursing-homes/

16. Ludden, Jennifer. NPR. Wired Homes Keep Tabs On Aging Parents. August 24, 2010. Accessed May 8, 2015. http://www.npr.org/templates/story/story.php?storyId=129104664.

Gait, Mobility Disorders, and Falls:

17. "Phases of the Normal Gait Cycle." Phases of the Normal Gait Cycle. Accessed May 4, 2015. http://faculty.utpa.edu/rafree/res/biomechanics/Virtual Biomechanics Laboratory/phases_of_the_normal_gait_cycle.html.

18. Abellan van Kan G, Rolland Y, et. al. "Gait speed at usual pace as a predictor of adverse outcomes in community-dwelling older people an International Academy on Nutrition and Againg (IANA) Task Force." The Journal of Nutrition, Health & Aging. 2009. Accessed May 4, 2015. http://www.ncbi.nlm.nih.gov/pubmed/19924348.

19. "Osteoarthritis." Centers for Disease Control and Prevention. Accessed May 4, 2015. http://www.cdc.gov/arthritis/basics/osteoarthritis.htm.

20. Montero-Odasso M, Shapira M, et. al. "Gait velocity in senior people. An easy test for detecting mobility impairment in community elderly." The Journal of Nutrition, Health & Aging. 2004. Accessed May 4, 2015. http://www.ncbi.nlm.nih.gov/pubmed/15359349.

21. "Osteoarthritis." Arthritis Foundation. Accessed May 4, 2015. https://www.arthritis.org/conditions-treatments/disease-center/osteoarthritis/.

22. Bejek Zoltan, Robert Paroczai et. al. "Gait Parameters of Patients with Osteoarthritis of the Knee Joint." Physical Education and Sport. 2006. Accessed May 4, 2015. http://facta.junis.ni.ac.rs/pe/pe2006/pe2006-02.pdf.

23. Sarmini, Mahmoud M.D. "Understanding Normal & Pathological Gait." Accessed May 4, 2015. http://www.medschool.lsuhsc.edu/physical_medicine/PPT/Normal_Pathological_Gait.ppt.

24. Brach JS, berlin JE, et. al. Too much or too little step width variability is associated with a fall history in older persons who walk at or near normal gait speed. The Journal of Neuroengineering and Rehabilitation. July 26, 2005. Accessed May 4, 2015. http://www.ncbi.nlm.nih.gov/pubmed/16042812.

25. Hafner, Katie. "Bracing for the Falls of an Aging Nation." The New York Times. November 2, 2014. Accessed May 4, 2015. http://www.nytimes.com/interactive/2014/11/03/health/bracing-for-the-falls-of-an-aging-nation.html.

26. "Falls Among Older Adults: An Overview." Centers for Disease Control and Prevention. Accessed May 4, 2015. http://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html

27. "What is Osteoarthritis?" National Institute of Arthritis and Musculoskeletal and Skin Diseases. Accessed May 4, 2015. http://www.niams.nih.gov/Health_Info/Osteoarthritis/osteoarthritis_ff.asp.

28. Graf, PhD(c), MS, RN, GCNS-BC, Carla. "The Lawton Instrumental Activities of Daily Living (IADL) Scale." Try This: Best Practices in Nursing Care to Older Adults. 2013. Accessed May 8, 2015. http://consultgerirn.org/uploads/File/trythis/try_this_23.pdf

Sensing and Tracking of People:

29. Pirsiavash, Hamed and Ramanan, Deva. "Detecting Activities of Daily Living in First-person Camera Views." Accessed May 4, 2015. http://www.ics.uci.edu/~hpirsiav/papers/adl_cvpr12.pdf

30. Shackleton, John, VanVoorst, Brian, and Hesch, Joel . "Tracking People with a 360-Degree Lidar." Accessed May 4, 2015. http://www.velodynelidar.com/lidar/hdlpressroom/pdf/papers/journal_papers/Tracking%20People%20with%20a%20360-Degree%20Lidar.pdf

31. Tarantola, Andrew. "MIT's "Kinect of the Future" Can Track You Through Walls." Gizmodo. October 15, 2013. Accessed May 4, 2015. http://gizmodo.com/mits-kinect-of-the-future-can-track-you-through-wall-1443947631.

32. Hnat, Timothy W., Griffiths, Erin, Dawson, Ray, and Whitehouse, Kamin. "Doorjamb: Unobtrusive Room-level Tracking of People in Homes using Doorway Sensors. Accessed

May 4, 2015. http://www.cs.virginia.edu/~whitehouse/research/buildingEnergy/hnat12doorjamb.pdf

33. Li, Jia, and Radke, Richard J. "Using Time-of-Flight Measurements for Privacy-Preserving Tracking in a Smart Room." Accessed May 4, 2015. http://www.ecse.rpi.edu/~rjradke/papers/jia-tii13.pdf

Wearables, Internet of Things, and Connected Devices:

34. "Ambient Products." Ambient Devices. Accessed May 4, 2015. http://www.ambientdevices.com/about/consumer-devices

35. "Glanceable Information." Ambient Devices. Accessed May 4, 2015. http://www.ambientdevices.com/technology/glanceable-information

36. Rose, David. "The Origin of the Ambient Information Orb." In Enchanted Objects: Design, Human Desire, and the Internet of Things, 75-78. New York, NY: Scribner, 2014.

37. Iseman, Luke. "Space-Age Gardening: Aquaponics, Hydroponics, and Aeroponics." Make. May 2, 2014. Accessed May 4, 2015. http://makezine.com/2014/05/02/space-age-gardening-aquaponics-hydroponics-and-aeroponics/

38. Wood, Molly. "At the International CES, the Internet of Things Hits Home." The New York Times. January 4, 2015. Accessed May 4, 2015. http://www.nytimes.com/2015/01/05/technology/international-ces-the-internet-of-things-hits-homes.html

39. "Taking Measure of the Quantified Self Movement." CBS News. March 23, 2014. Accessed May 4, 2015. http://www.cbsnews.com/news/taking-measure-of-the-quantified-self-movement/.

Smart Surfaces Previous Work (Physical and Digital):

40. Gong, Nan-Wei, Hodges, Steve, and Paradiso, Joseph A. "Leveraging Conductive Inkjet Technology to Build a Scalable and Versatile Surface for Ubiquitous Sensing." UbiComp '11 *Proceedings of the 13th international conference on Ubiquitous computing*, 2011, pp 45-54.

41. Richardson, Bruce, Leydon, Krispin, Fernstrom, Mikael, and Paradiso, Joseph A. "Z-Tiles: building blocks for modular, pressure-sensing floorspaces." *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, 2004, pp. 1529-1532.

42. Paradiso, Joseph A., Abler, Craig, Hsiao, Kai-yuh, and Reynolds, Matthew. "The magic carpet: physical sensing for immersive environments." *CHI '97 Extended Abstracts on Human Factors in Computing Systems*, 1997, pp. 277-278.

43. Tekscan. Accessed May 4, 2015. https://www.tekscan.com/

44. "Tactonic Floor Tiles." TactonicTech. Accessed May 4, 2015. http://tactonic.com/floorTiles.html

45. Weissmann, Ben, Linder, Natan, Kubat, Rony, and Maes Pattie. "Lens: A Javascript SDK for Building Web-Based Augmented Reality Applications on the LuminAR Platform." *Presented as poster at EECScon 2013*. Cambridge, MA, USA., Apr 2013.

46. Bränzel, Alan et. al. "GravitySpace." Hasso-Plattner-Institute. 2013. Accessed May 4, 2015. http://hpi.de/baudisch/projects/gravityspace.html

47. Sandler, Seth. "How to Make a Multitouch Surface." Accessed May 4, 2015. http://sethsandler.com/multitouch/

48. "Woven Pressure Sensor Matrix." How to Get What You Want. Accessed May 4, 2015. http://www.kobakant.at/DIY/?p=4296.

49. "How Elsi is Installed." Elsi Technologies. Accessed May 5, 2015. http://www.elsitechnologies.com/index.php/en/how-it-works/how-elsi-is-installed

50. Clark, Patrick. "Innovation: Floor Tiles That Can Monitor the Health of the Elderly." Bloomberg.com. March 20, 2014. Accessed May 5, 2015. http://www.bloomberg.com/bw/articles/2014-03-20/intellimat-flooring-measures-health-based-on-footstep-patterns.

51. Voisen, Sean. "Designing a Pressure-sensitive floor." Sean Voisen. Accessed May 5, 2015. http://sean.voisen.org/blog/2013/08/designing-pressure-sensitive-floor/

FSR Design and Components:

52. "FSR 101 - THE BASICS." Sensitronics. Accessed May 4, 2015. http://www.sensitronics.com/fsr101.htm.

53. "Force Sensing Resistor Integration Guide and Evaluation Parts Catalog." Datasheet 90-45632 Rev. D, Interlink Electronics, 546 Flynn Road, Camarillo, CA, 93012, USA.

54. "FlexiForce® Sensors User Manual." RevF, Tekscan, 2005.

55. Dribin, Dave. "Keyboard Matrix Help." Jun 24, 2000. Accessed May 4, 2015. http://www.dribin.org/dave/keyboard/one_html/

56. "Shift Registers." SparkFun. Accessed May 4, 2015, https://learn.sparkfun.com/tutorials/shift-registers?_ga=1.188875883.1169809045.1410492748

57. "Arduino: Individually Control Shift Register Pins." bildr.blog. August 23, 2011. Accessed May 4, 2015. http://bildr.org/2011/08/74hc595-breakout-arduino/

58. "Muxing Around With The CD74HC4067 + Arduino." bildr.blog. February 1, 2011. Accessed May 4, 2015. http://bildr.org/2011/02/cd74hc4067-arduino/

59. "SparkFun Analog/Digital MUX Breakout - CD74HC4067." SparkFun. Accessed May 4, 2015. https://www.sparkfun.com/products/9056

60. "Conductive Inkjet Printing." cit technology. Accessed May 4, 2015. http://cittechnology.com/flexible-circuits/conductive-inkjet-printing.html

61. Arduino Mega. Arduino. Accessed May 4, 2015. http://arduino.cc/en/Main/arduinoBoardMega

62. analogRead(). Arduino. Accessed May 4, 2015. http://arduino.cc/en/Reference/analogRead.

63. Fraden, Jacob. Handbook of Modern Sensors. 3rd ed. N.p.: Springer, 2004. Print.

64. "3M Velostat Conductive Bags and Films." 3M United Kingdom. Retrieved from.http://www3.3m.com/catalog/uk/en002/electronics_mfg/static_control_packaging/node_JV20CN7XFBbe/root_K3BHNB8005gv/vroot_5KR9K2C2KZge/gvel_JQGPTT9723gl/theme_uk_staticcontrolpackaging_3_0/command_AbcPageHandler/output_html.

65. "Force Sensitive Resistor (FSR)." Adafruit. Accessed May 4, 2015. https://learn.adafruit.com/force-sensitive-resistor-fsr

66. "Intro to Copper Tape." chibitronics. Accessed May 5, 2015. http://chibitronics.com/education/

67. "Force Sensitive Resistors (FSRs)." Open Music Labs. Accessed May 5, 2015. http://imgur.com/gallery/BNDgonT

68. "I2C." SparkFun. Accessed May 8, 2015. https://learn.sparkfun.com/tutorials/i2c.

69. "XBee Buying Guide." SparkFun. Accessed May 8, 2015. https://www.sparkfun.com/pages/xbee_guide

70. ZigBee. Accessed May 8, 2015. http://www.zigbee.org/

71. Thiebaut, D. "Tutorial: Arduino and XBee Communication." Smith College. Accessed May 8, 2015. http://cs.smith.edu/dftwiki/index.php/ Tutorial:_Arduino_and_XBee_Communication

Software and Miscellaneous:

72. SolidWorks. Dassault Systèmes.

73. MATLAB. MathWorks.

74. Processing. processing.org.

75. Adobe Illustrator. Adobe Systems.

76. Arduino. arduino.cc

77. Seeed Studio. http://www.seeedstudio.com/

78. Ewing, Phillip. "Interactive Phototherapy: Integrating Photomedicine Into Interactive Architecture." Thesis, Master of Science in Architecture. 2015.

79. JST. Accessed May 15, 2015. http://www.jst.com/home2.html

80. EvoCam. Evological. http://www.evological.com/evocam.html

81. Altium Designer. Altium. http://www.altium.com/altium-designer/overview

82. Atlas Scientific. https://www.atlas-scientific.com/.

Appendix A: Arduino Code

Below is the code used on the Master Arduino and Slave Sensor Module PCBs. The Master Code allows the Arduino to communicate with the Slaves and request data via I^2C. The Slave Code allows the PCBs to collect the 512 sensor values on it's module and send it to the Master one at a time when requested.

Master Code:

```
#include <Wire.h>

int incomingByte = 0;

unsigned long initial;
unsigned long final;

String ids[]={"one","two","three"};
int count=0;
int c;

void setup(){

 Wire.begin();       // join i2c bus (address optional for master)
 TWBR = 1;
 Serial.begin(115200); // start serial for output


 Wire.beginTransmission(0x10); // transmit to device #4
 Wire.write(100);           // sends one byte
 Wire.endTransmission();    // stop transmitting

 Wire.beginTransmission(0x20); // transmit to device #4
 Wire.write(100);           // sends one byte
 Wire.endTransmission();    // stop transmitting

 Wire.beginTransmission(0x30); // transmit to device #4
 Wire.write(100);           // sends one byte
 Wire.endTransmission();    // stop transmitting

 delay(2000); //give the master a second to boot up


}


void loop()
{
 ///UNCOMMENT FOR PROCESSING
   request(0x10,0);

   request(0x20,1);

   request(0x30,2);

//UNCOMMENT FOR MATLAB
//if (Serial.available() > 0) {
//   // read the incoming byte:
//   incomingByte = Serial.read();
//
//   if (incomingByte=='1'){
//     //get data from each of the 4 mats. Comment out each block as needed.
//     //Serial.print("one ");
//     request(0x10,0);
//     //delay(50);
//   }
```

```
//
//   if (incomingByte=='2'){
//     //get data from each of the 4 mats.  Comment out each block as needed.
//     //Serial.print("one ");
//     request(0x20,1);
//     //delay(50);
//   }
//
//   if (incomingByte=='3'){
//     //get data from each of the 4 mats.  Comment out each block as needed.
//     //Serial.print("one ");
//     request(0x30,2);
//     //delay(50);
//   }
//
//}
//*********
//Serial.print("two ");
//request(0x20);
//delay(50);

//Serial.print("three ");
//request(0x30);
//delay(50);
//
//Serial.print("four ");
//request(0x40);
//delay(50);


}

//******BYTE MODE
void request(int ID,int printid){

  //initial = millis();

  Wire.requestFrom(ID, 1);   // request 6 bytes from slave device #2
  while(Wire.available()){
  c = Wire.read(); //look for the first 255
  }
switch (c){
  case 0:
     Serial.print(ids[printid]);
     Serial.print(" ");
     for(int i=0;i<512;i++){
     Serial.print(1);
     Serial.print(" ");
    }
    Serial.println();
  break;

  case 255:
    Serial.print(ids[printid]);
    Serial.print(" ");
    for(int i=0;i<512;i++){
     Wire.requestFrom(ID, 1);   // request 6 bytes from slave device #2

     while(Wire.available()){

     int c = Wire.read();
     Serial.print(c);
     Serial.print(" ");
    }
    }
     Serial.println();
   break;
}
}
```

Slave Code:

```
//http://www.arduino.cc/en/Reference/PortManipulation
#include <Wire.h>

int values[512]; // array to hold the values for each collection
int index;

int threshold=50;

#define SER_Pin 4  // SER_IN
#define RCLK_Pin 3 // L_CLOCK
#define SRCLK_Pin 2 // CLOCK

//How many of the shift registers - each board has 8
#define number_of_74hc595s 8

int LED=8; // alert led
int LED2=9;


//do not touch
#define numOfRegisterPins number_of_74hc595s * 8

boolean registers[numOfRegisterPins];

int request_event_number=0;

// index of each pin on the shift registers. The order is mixed up
// because the sensors overlap
int register_pin_vals[8][8]={

  {8,9,10,11,12,13,14,15},  //sr1
  {0,1,2,3,4,5,6,7},        //sr0
  {24,25,26,27,28,29,30,31}, //sr3
  {16,17,18,19,20,21,22,23}, //sr2
  {40,41,42,43,44,45,46,47}, //sr5
  {32,33,34,35,36,37,38,39}, //sr4
  {56,57,58,59,60,61,62,63}, //sr7
  {48,49,50,51,52,53,54,55}, //sr6

};

// 4 bit channel address for the mux channel

int muxChannel[16][4]={
  {0,0,0,0}, //channel 0
  {1,0,0,0}, //channel 1
  {0,1,0,0}, //channel 2
  {1,1,0,0}, //channel 3
  {0,0,1,0}, //channel 4
  {1,0,1,0}, //channel 5
  {0,1,1,0}, //channel 6
  {1,1,1,0}, //channel 7
  {0,0,0,1}, //channel 8
  {1,0,0,1} , //channel 9
  {0,1,0,1}, //channel 10
  {1,1,0,1}, //channel 11
  {0,0,1,1}, //channel 12
  {1,0,1,1}, //channel 13
  {0,1,1,1}, //channel 14
  {1,1,1,1}  //channel 15
```

```
};

int ID=0x30;

void setup(){

  // join i2c bus at this address 0x10,0x20,0x30 etc.
  Wire.begin(ID);

  // do something if it is requested from the master
  Wire.onRequest(requestEvent);

  Wire.onReceive(receiveEvent); // register event

  // pin modes, set digital pins as outputs
  pinMode(SER_Pin, OUTPUT);
  pinMode(RCLK_Pin, OUTPUT);
  pinMode(SRCLK_Pin, OUTPUT);
  pinMode(LED,OUTPUT);
  pinMode(LED2,OUTPUT);

  digitalWrite(LED2,LOW);

  clearRegisters();
  writeRegisters();
  getMat();
}


void loop(){
// read the mat data and save to the array
getMat();
}

void(* resetFunc) (void) = 0;//declare reset function at address 0

// dont mess with this section.  It collects the values from each sensor
void getMat(){

  int count=0;

  for(int i=0;i<number_of_74hc595s;i++){

    //this writes all the EN pins of every multiplexer (4 per shift register) off

    for(int j=0;j<4;j++){

    for(int k=0;k<number_of_74hc595s;k++){
      setRegisterPin(register_pin_vals[k][4],1);
      setRegisterPin(register_pin_vals[k][5],1);
      setRegisterPin(register_pin_vals[k][6],1);
      setRegisterPin(register_pin_vals[k][7],1);}

    for(int m=0;m<16;m++){ //each switch on the MUX

          setRegisterPin(register_pin_vals[i][0],muxChannel[m][0]);
          setRegisterPin(register_pin_vals[i][1],muxChannel[m][1]);
          setRegisterPin(register_pin_vals[i][2],muxChannel[m][2]);
          setRegisterPin(register_pin_vals[i][3],muxChannel[m][3]);
          setRegisterPin(register_pin_vals[i][4+j],0);//turn off each EN to engage the MUX

      writeRegisters();  //MUST BE CALLED TO DISPLAY CHANGES

     values[count]=map(analogRead(A0),0,1023,1,254);    // byte mode
     //values[count]=map(analogRead(A0),0,1023,0,1022);     // analog mode

     count+=1;

       }
     }
   }
```

```
}


//set all register pins to LOW
void clearRegisters(){
  for(int i = numOfRegisterPins - 1; i >= 0; i--){
    registers[i] = LOW;
  }
}


//dont mess with this -- this is the commands to switch shift registers and multiplexers
void writeRegisters(){
//SER_Pin 4   //SER_IN
//RCLK_Pin 3  //L_CLOCK
//SRCLK_Pin 2 //CLOCK

//PORTD = B10101000; // sets digital pins 7,5,3 HIGH


  PORTD = B00000000; //LOW --RCLK LOW

  for(int i=numOfRegisterPins -1;i>=0;i--){
    PORTD = B00000000; //LOW --SRCLK LOW

    int val = registers[i];

    if (val==HIGH){
      PORTD = B00010000;
    }                    //SER PIN HIGH OR LOW
    if (val==LOW){
      PORTD = B00000000;
    }

    PORTD = B00000100; //SRCLK HIGH
  }
 PORTD = B00001000; //RCLK HIGH


}

//set an individual pin HIGH or LOW
void setRegisterPin(int index, int value){
  registers[index] = value;
}

void receiveEvent(int howMany)
{
  while(Wire.available()) // loop through all but the last
  {
    int c = Wire.read(); // receive byte as a character

    if (c=100){
      digitalWrite(LED2,HIGH);
      resetFunc(); //call reset
    }
  }

}

//******BYTE MODE
void requestEvent(){
//
//
digitalWrite(LED,HIGH);
//
//
if(request_event_number==0){
       //getMat();
     if(ID==0x20){
       values[366]=1;
       values[367]=1;}
```

```
    if(ID==0x30){
     values[132]=1;
     values[133]=1;
     values[275]=1;
     values[276]=1;}

     int numval=0;
     for(int i=0;i<512;i++){
       if(values[i]>threshold){
         numval+=1;}}

     if(numval>0){
     Wire.write(255);
     request_event_number+=1;}

     else{
       Wire.write(0);
       request_event_number=0;}
}


else{

    if(ID==0x20){
     values[366]=0;
     values[367]=0;}

    if(ID==0x30){
     values[132]=0;
     values[133]=0;
     values[275]=0;
     values[276]=0;}

    Wire.write(values[request_event_number-1]);

    request_event_number+=1;

    if(request_event_number>512){
     request_event_number=0;}

  }

digitalWrite(LED,LOW);
}
```

Appendix B: Processing Code

Both Processing Codes below connect to the Master Arduino and receive a continuous feed of the values for Slaves 1, 2, and 3. These values from each Slave are used to color either squares or triangles corresponding to the sensor's position on the mat. The color goes from blue to red corresponding to values 1-254 (no force - high force). The only difference between the below two codes is that the Square Pixel Visualization elongates the relative positions of the sensors, while the Triangle Pixel Visualization is a 1:1 display of the sensor position.

Square Pixel Visualization

```
import processing.serial.*;
import processing.pdf.*;

Serial arduinoPort;

int rows=8;    //Size of matrix to hold the data. String from arduino is rows*columns long
int columns=64;
int space=75;
int space2=500;
//GENERATES COLORS FOR HEAT MAP
int[][] strip=new int[3][255*4];
String[] List=new String[rows*columns];

String[] row0=new String[columns];
String[] row1=new String[columns];
String[] row2=new String[columns];
String[] row3=new String[columns];
String[] row4=new String[columns];
String[] row5=new String[columns];
String[] row6=new String[columns];
String[] row7=new String[columns];

//String[][] row_s=new String[rows][columns];


String[][] mat=new String[rows][columns];

float tile_diameter=1.9*20;//cm
float tile_diameter2=1.9*20;

float vert_gap=tile_diameter;
float horiz_gap=tile_diameter;
//float horiz_gap=0;

FloatList xpeak;
FloatList ypeak;

StringList list;

String index;

int limit=700;


void setup() {

  for (int i=0; i<rows*columns; i++) {
   List[i]="0";
  }

  xpeak = new FloatList();
  ypeak = new FloatList();
  list = new StringList();
```

```
 size(displayWidth,displayHeight);

 //GENERATE HEAT MAP COLOR MATRIX
 for (int i=0; i<255; i++) {
  strip[0][i]=255;
  strip[1][i]=i;
  strip[2][i]=0;
 }
 for (int i=0; i<255; i++) {
  strip[0][i+255]=255-i;
  strip[1][i+255]=255;
  strip[2][i+255]=0;
 }
 for (int i=0; i<255; i++) {
  strip[0][i+255*2]=0;
  strip[1][i+255*2]=255;
  strip[2][i+255*2]=i;
 }
 for (int i=0; i<255; i++) {
  strip[0][i+255*3]=0;
  strip[1][i+255*3]=255-i;
  strip[2][i+255*3]=255;
 }



 frameRate(100);
 //arduinoPort=new Serial(this, "/dev/tty.HC-06-DevB-5", 9600); //Connect to arduino
 arduinoPort=new Serial(this, "/dev/tty.usbmodem1411", 115200); //Connect to arduino

 arduinoPort.bufferUntil('\n');
 delay(100);
// background(255);
// beginRecord(PDF, "test.pdf");
}

void draw() {


}

void keyPressed()
{

 switch (key) {
  case 'c':
   background(0);
   break;
  case 'q':
   endRecord();
   exit();
   break;
  default:
   break;
 }

}

void serialEvent(Serial p) {  // Reads string of incoming data from Arduino in string form
 //print("serial");

 String inString=p.readStringUntil('\n');
 if (inString != null) {
  inString = trim(inString);
  String[] inlist = splitTokens(inString, " ");

  list.clear();
  for (int i=0; i<inlist.length; i++) {
   list.set(i, inlist[i]);
  }
```

```
   if (list.size()==rows*columns+1) {

    index=list.get(0);
    list.remove(0);

      for(int i=0;i<list.size();i++){
       List[i]=list.get(i);}
       drawTiles(List,index);}}

 }



void drawTiles(String[] List,String index) {

 rectMode(CORNER);

 arrayCopy(List, 64*0, row0, 0, 64);
 arrayCopy(List, 64*1, row1, 0, 64);
 arrayCopy(List, 64*2, row2, 0, 64);
 arrayCopy(List, 64*3, row3, 0, 64);
 arrayCopy(List, 64*4, row4, 0, 64);
 arrayCopy(List, 64*5, row5, 0, 64);
 arrayCopy(List, 64*6, row6, 0, 64);
 arrayCopy(List, 64*7, row7, 0, 64);


 row1=reverse(row1);
 row3=reverse(row3);
 row5=reverse(row5);
 row7=reverse(row7);

 String row_s[][]= {row0, row1, row2, row3, row4, row5, row6, row7};

 row_s[5][42]="1";
 row_s[5][43]="1";


 for (int i=0; i<rows; i++) {
  for (int j=0; j<columns; j++) {

    int col=int(row_s[i][j]);

    int colordep=int(map(col, 1, 254, 254*4-1, 0));
    strokeWeight(1);
    fill(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
    stroke(strip[0][colordep], strip[1][colordep], strip[2][colordep]);

    if(index.equals("three")){
    rect(space+j*(horiz_gap), tile_diameter*0*rows+space2-i*(vert_gap), tile_diameter2, tile_diameter);}

    if(index.equals("two")){
    rect(space+j*(horiz_gap), tile_diameter*1*rows+space2-i*(vert_gap), tile_diameter2, tile_diameter);}

    if(index.equals("one")){
    rect(space+j*(horiz_gap), tile_diameter*2*rows+space2-i*(vert_gap), tile_diameter2, tile_diameter);}

  }
 }

}
```

Triangular Pixel Visualization

```
import processing.serial.*;
import processing.pdf.*;

Serial arduinoPort;
```

```
int l=25;
int h=15;
int x=400;
int y=0;

int bigspace=25;
int col;
int colordep;

int rows=8;    //Size of matrix to hold the data.  String from arduino is rows*columns long
int columns=64;
int space=8;
//GENERATES COLORS FOR HEAT MAP
int[][] strip=new int[3][255*4];
String[] List=new String[rows*columns];

String[] row0=new String[columns];
String[] row1=new String[columns];
String[] row2=new String[columns];
String[] row3=new String[columns];
String[] row4=new String[columns];
String[] row5=new String[columns];
String[] row6=new String[columns];
String[] row7=new String[columns];

//String[][] row_s=new String[rows][columns];


String[][] mat=new String[rows][columns];

float tile_diameter=22;//cm

float vert_gap=tile_diameter;
float horiz_gap=tile_diameter;
//float horiz_gap=0;

FloatList xpeak;
FloatList ypeak;

StringList list;

String index;

int limit=200;


void setup() {
background(0);
  for (int i=0; i<rows*columns; i++) {
    List[i]="0";
  }

  xpeak = new FloatList();
  ypeak = new FloatList();
  list = new StringList();
  size(displayWidth, displayHeight);


  //GENERATE HEAT MAP COLOR MATRIX
  for (int i=0; i<255; i++) {
    strip[0][i]=255;
    strip[1][i]=i;
    strip[2][i]=0;
  }
  for (int i=0; i<255; i++) {
    strip[0][i+255]=255-i;
    strip[1][i+255]=255;
    strip[2][i+255]=0;
  }
  for (int i=0; i<255; i++) {
    strip[0][i+255*2]=0;
```

```
   strip[1][i+255*2]=255;
   strip[2][i+255*2]=i;
  }
  for (int i=0; i<255; i++) {
   strip[0][i+255*3]=0;
   strip[1][i+255*3]=255-i;
   strip[2][i+255*3]=255;
  }



 frameRate(100);
 //arduinoPort=new Serial(this, "/dev/tty.HC-06-DevB-5", 9600); //Connect to arduino
 arduinoPort=new Serial(this, "/dev/tty.usbmodem1411", 115200); //Connect to arduino
//
 arduinoPort.bufferUntil('\n');
 delay(100);

 //beginRecord(PDF, "test.pdf");


}

void draw() {

}

void keyPressed()
{

 switch (key) {
   case 'c':
     background(0);
     break;
   case 'q':
     endRecord();
     exit();
     break;
   default:
     break;
 }

}

void serialEvent(Serial p) {   // Reads string of incoming data from Arduino in string form
 //print("serial");

 String inString=p.readStringUntil('\n');
 if (inString != null) {
   inString = trim(inString);
   String[] inlist = splitTokens(inString, " ");

   list.clear();
   for (int i=0; i<inlist.length; i++) {
     list.set(i, inlist[i]);
   }

     if (list.size()==rows*columns+1) {

      index=list.get(0);
      list.remove(0);

         for(int i=0;i<list.size();i++){
           List[i]=list.get(i);}
           drawTiles(List,index);}}

 }
```

```
void drawTiles(String[] List,String index) {

  rectMode(CORNER);

  arrayCopy(List, 64*0, row0, 0, 64);
  arrayCopy(List, 64*1, row1, 0, 64);
  arrayCopy(List, 64*2, row2, 0, 64);
  arrayCopy(List, 64*3, row3, 0, 64);
  arrayCopy(List, 64*4, row4, 0, 64);
  arrayCopy(List, 64*5, row5, 0, 64);
  arrayCopy(List, 64*6, row6, 0, 64);
  arrayCopy(List, 64*7, row7, 0, 64);


  row1=reverse(row1);
  row3=reverse(row3);
  row5=reverse(row5);
  row7=reverse(row7);

  String row_s[][]= {row0, row1, row2, row3, row4, row5, row6, row7};
 // row_s[4][42]="1";
   row_s[5][42]="1";
  row_s[5][43]="1";
  mat=row_s;

  if(index.equals("one")){
    draw_triangles(mat,"one");}

  if(index.equals("two")){
    draw_triangles(mat,"two");}

  if(index.equals("three")){
    draw_triangles(mat,"three");}

}

void draw_triangles(String[][] mat, String index){

 int val=300;
  stroke(0);

 strokeWeight(1.5);
  // y=400;
 if(index=="one"){
  y=600;
  x=val;}

 if(index=="two"){
  //y=480;
  y=483;
  x=val;}

 if(index=="three"){
   //y=360;
   y=366;
   x=val;}
for(int j=0;j<8;j+=2){
x=val;
y=y-25-4;
for(int i=0;i<64;i+=4){

  col=int(mat[j][i]);
  colordep=int(map(col, 1, 254, 254*4-1, 0));
  fill(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  stroke(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  triangle(x+0,y+0,x+l,y+0,x+0,y+h);

  col=int(mat[j][i+1]);
  colordep=int(map(col, 1, 254, 254*4-1, 0));
  fill(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  x=x+10-9;
  stroke(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
```

```
  triangle(x+0,y+h,x+l,y+0,x+l,y+h);

  col=int(mat[j][i+2]);
  colordep=int(map(col, 1, 254, 254*4-1, 0));
  fill(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  x=x+25;
  stroke(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  triangle(x+0,y+0,x+0,y+h,x+l,y+h);

  col=int(mat[j][i+3]);
  colordep=int(map(col, 1, 254, 254*4-1, 0));
  fill(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  x=x+10-9;
  stroke(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  triangle(x+0,y+0,x+l,y+0,x+l,y+h);
  x=x+bigspace;
 }
 x=30;
 y=y-5+5;
 x=val;
 for(int i=0;i<64;i+=4){

  col=int(mat[j+1][i]);
  colordep=int(map(col, 1, 254, 254*4-1, 0));
  fill(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  stroke(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  triangle(x+0,y+0,x+l,y+0,x+0,y-h);

  col=int(mat[j+1][i+1]);
  colordep=int(map(col, 1, 254, 254*4-1, 0));
  fill(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  x=x+10-9;
  stroke(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  triangle(x+0,y-h,x+l,y+0,x+l,y-h);

  col=int(mat[j+1][i+2]);
  colordep=int(map(col, 1, 254, 254*4-1, 0));
  fill(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  x=x+25;
  stroke(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  triangle(x+0,y+0,x+0,y-h,x+l,y-h);

  col=int(mat[j+1][i+3]);
  colordep=int(map(col, 1, 254, 254*4-1, 0));
  fill(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  x=x+10-9;
  stroke(strip[0][colordep], strip[1][colordep], strip[2][colordep]);
  triangle(x+0,y+0,x+l,y+0,x+l,y-h);
  x=x+bigspace;
 }
 x=30;
 }

 }
```

Appendix C: MATLAB Code

Below are the scripts and functions used to find the centroids of a set of footsteps as well as the time at which each step occurs. See Section 8.4.2 for a detailed description of the MATLAB Code.

NewMainly.m

```
clc;   % Clear the command window.
close all; % Close all figures (except those of imtool.)
imtool close all; % Close all imtool figures.
clear; % Erase all existing variables.
workspace; % Make sure the workspace panel is showing.
closeSerial;
clf;

% comPort=('/dev/tty.usbmodem1451'); %arduino com port for mac
%comPort=('COM4'); %arduino com port for windows
black=false;
%s=serial(comPort);

%set parameters of the arduino
% set(s,'DataBits',8);
% set(s,'StopBits',1);
% set(s,'BaudRate',115200);
% set(s,'Parity','none');
% set(s,'Terminator', 'CR/LF');
% set(s,'InputBufferSize',2048);
% fopen(s);

%figure(1),clf


%parameters of the incoming data set
width=5;
columns=8;
rows=64;
num_values=rows*columns;
threshold=300;
GlobalMat=[];
blah = zeros(5);
No=[];
%allocate space
mag=zeros(1,num_values);

array=zeros(1,512);
Clock=zeros(1,9);

%while true
pause(5);
%while true
trial=0;
run=0;
fid = textread('table.txt', '%s','delimiter', '\n');
[finalrun, param]=size(fid);
while run<finalrun

  for j=1:3
    index=char(num2str(j));


    meas=fid{run+j};
    %meas=fscanf(fid,'%c');

    B = strsplit(meas,','); %separate the values by space

    %C(~cellfun('isempty',C));

    for i=1:512
%       i
```

```
%        B(i)
%        j
        array(i)=str2double(B(i))-1;
    end
    for k=513:515
        Clock((j-1)*3+k-512)=str2double(B(k));
    end
    Q=array;


    %C2 = cell2mat(C);
    %C = C(1:end-1);

    % if length(C)==rows*columns
    %count=count+1;
    % mag=str2double(C);



    mat = vec2mat(Q,rows)'; %turn magnitude vector to matrix
    % mat = fliplr(mat);

    mat(:,2)=flipud(mat(:,2));
    mat(:,4)=flipud(mat(:,4));
    mat(:,6)=flipud(mat(:,6));
    mat(:,8)=flipud(mat(:,8));

    %mat=flipud(mat);

    GlobalMat=[GlobalMat,mat];
    %GlobalMat=mat;


end


GlobalMat(18,14)=0;
GlobalMat(19,14)=0;
GlobalMat(4,19)=0;
GlobalMat(5,19)=0;
GlobalMat(12,21)=0;
GlobalMat(13,21)=0;
GlobalMat(19,21)=0;
GlobalMat(20,21)=0;
GlobalMat(1,8)=0;
GlobalMat(1,16)=0;
GlobalMat(1,24)=0;
GlobalMat(42,4)=0;
GlobalMat(44,6)=0;
GlobalMat(GlobalMat<50)=0;
GlobalMat(GlobalMat>=50)=1;

%GlobalMat(:,1)=[];
m=int2str(123);
Matrix=GlobalMat;              %up to centroid=[...] we find a centroids
if run==0
    M=GlobalMat;
else
    M(GlobalMat==1)=1;
end
Matrix=Matrix/sum(Matrix(:));  %with the matrix and if it is empty we do not find the
[m,n]=size(Matrix);            %centroids of the image
[I,J]=ndgrid(1:m,1:n);
centroid=[dot(J(:),Matrix(:)), dot(I(:),Matrix(:))]
if isnan(centroid(1))==0    %found centroids on image
    T=MakeCentroidT(GlobalMat, Clock);
else                 %T=[] if there were not centroids on matrix
    T=[];
end
if and(isempty(T), trial==0)   %If start with empty images
    GlobalMat=[];
    run=run+3
```

```
   continue;                %keep reading without doing anything else

end

if trial==0                %if start with a non-empty image
   [T,N]=Cluster(T);            %find first cluster
   C=CentroidPoints(T);          %find centroid of cluster
   points=C;                   %points will be first centroid found
   while isempty(N)==0            %while we haven't found all clusters
      [T,N]=Cluster(N);
      C=CentroidPoints(T);
      points=[points; C];

   end

else   %if we are not at the start and we still have centroids
   if isempty(T)==0
      Ref=T;
      [T,N]=Cluster(T);  %find first cluster
      C=CentroidPoints(T); %find centroid of first cluster
   else
      Ref=[];
      T=[];
      C=[];
      N=[];
   end

   while isempty(N)==0    %find rest of clusters
      [T,N]=Cluster(N);
      X=CentroidPoints(T);

      C=[C; X];         %all centroids of the clusters of this image

   end

   for j= 1:size(points)      %for all the centroids of the first image

      b=0;

      if exist('A', 'var')==1
         [si,r]=size(A);

         if si>=j
            if isempty(A{j,1})==0
               b=1;
            end
         end
      end
      if j==1
         if isempty(Ref)
            Refe=true
         else
            Refe=false
         end
         if and(b==1, not(Refe))

            [A{j,1}, A{j,2}, A{j,3}, A{j,4}]=insideArea(C, points(j,:), points(j,4:5), A{j,3});
         elseif not(Refe)

            [A{j,1}, A{j,2}, A{j,3}, A{j,4}]=insideArea(C, points(j,:), points(j,4:5));

         end
         if not(Refe)
            No=A{j,4};
            centroids=A{j,1};
         end
      else
         if and(black==true, Refe)
            %trial=trial+1;
            trial=0;
            black=false;
            continue;
```

```
            end
            if or(isempty(No)==0, Refe)

               if or(b==1, and(b==1,Refe))
                  cent=A{j,1};
                  [dim, q]=size(cent);
                  if not(Refe)
                     [A{j,1}, A{j,2}, A{j,3}, A{j,4}]=insideArea(No, points(j,:), points(j,4:5), A{j,3});
                  end
                  centroids=A{j,1};
                  final=[centroids;cent(2:dim,:)];
                  A{j,1}=final;
                  lof=A{j,4};
                  if or(size(lof)==size(No), Refe)
                     x=final{:,4};
                     y=final{:,5};
                     LastCent=CentroidPoints(final);

                     if exist('EventCent','var')
                        [difp, f]=size(EventCent);
                        same=false;
                        for num=1:difp
                           if and(EventCent{num, 4}==LastCent{1, 4}, EventCent{num, 5}==LastCent{1, 5})
                              same=true;
                           end
                        end
                        if same==false
                           EventCent=[EventCent; LastCent];
                        end
                     else
                        EventCent=LastCent;
                     end
                     if Refe

                        black=true;
                     end

                  end
               else
                  [A{j,1}, A{j,2}, A{j,3}, A{j,4}]=insideArea(No, points(j,:), points(j,4:5));
                  centroids=A{j,1};

               end
               No=A{j,4};

            else
               CentroidPoints(points(j, :));
            end
         end

      end
      if exist('No','var')
         points=[points;No];
      end

   end
   %%%%IMAGE ZONE


   GlobalMat=[];
   trial=trial+1
   run=run+3
end

EventCent=[points; EventCent];
if isempty(EventCent)==0
[ECT,ECN]=Cluster(EventCent);            %find first cluster
FC=CentroidPoints(ECT);           %find centroid of cluster
PFC=FC;
while isempty(ECN)==0     %find rest of clusters
   [ECT,ECN]=Cluster(ECN);
   FC=CentroidPoints(ECT);
```

```
   PFC=[PFC; FC];
end
PFC=sortrows(PFC,3)
writetable(PFC, 'centroidTable.txt');
Name='NoImrandom.jpeg';
hold on;
M(42,4)=0;
M(44,6)=0;
imwrite(M, Name);
J = imread(Name,'jpg');
BW = im2bw(J);
imshow(BW)

hold on;

plot(EventCent{:,5}, EventCent{:,4}, 'g*')
plot(PFC{:,5}, PFC{:,4}, 'b*')
hold off
[leng, width, rate, avl, avw, avr, finalrate]=Measurements(PFC)
dlmwrite('MeasureTable.txt', [leng, width, rate, avl, avw, avr, finalrate])
else
   'NO FOOTSTEPS TO ANALYZE'

end
```

MakeCentroidT.m

```
function T=MakeCentroidT(M, Clock)

%Matrix=randomFootstep(64, 8, 10, 5);
 [I,J]=find(M==1)
 rng(1);
 %opts = statset('Display','final');
 if length(I)>4

 [idx3,C3] = kmeans([I,J],3);

     c=Clock;
        Centroid=zeros(3,5);
        if C3(2)<=8
           Centroid(1,1:3)=c(1:3);
       Centroid(2,1:3)=c(1:3);
       Centroid(3,1:3)=c(1:3);
        elseif C3(2)<=16
           Centroid(1,1:3)=c(4:6);
       Centroid(2,1:3)=c(4:6);
       Centroid(3,1:3)=c(4:6);
        else
           Centroid(1,1:3)=c(7:9);
       Centroid(2,1:3)=c(7:9);
       Centroid(3,1:3)=c(7:9);
        end

       Centroid(:,4:5)=C3(:,:);

        Centroids=Centroid;
if isempty(Centroids)==0
   T=array2table(Centroids, 'VariableNames', {'hour';'minutes';'seconds';'centroidsx';'centroidsy'});

else
   T=[];
end
 else
    T=[];
 end
```

Cluster.m

```
function [centroids,N]=Cluster(T)
```

```
length=10; %size of foot
centroids=[];
N=[];
siz=size(T,1);
number=zeros(siz); %maybe 1,siz
max=-1;
maxele=0;
for k=1:siz
centroid=[T(siz-k+1,4), T(siz-k+1,5)];  %make it the centroid=point that will define our area we want to be checking

    a=centroid{1,1};            %a is the x of the centroid
     b=centroid{1,2};           %b is the y of the centroid


for i=k:siz
  if i==k
    continue
  else
     point=[T(siz-i+1,4) T(siz-i+1,5)];
     x=point{1,1};               %x is the x coordinate of the point we are looking at in the table
      y=point{1,2};
     if or(or(and(a < x, (a+length)>x),and(a>x, (a-length)<x)), a==x)   %check if point we are looking at is
                                                        %within a rectangle of 10 by 11
          if or(or(and( b<y, (b+length)>y), and(b>y, (b-length)<y)), b==y)

             number(k)=number(k)+1;
          end
     end
  end
end
for i=1:k-1
  if i==k
    continue
  else
     point=[T(siz-i+1,4) T(siz-i+1,5)];
     x=point{1,1};               %x is the x coordinate of the point we are looking at in the table
      y=point{1,2};
     if or(or(and(a < x, (a+length)>x),and(a>x, (a-length)<x)), a==x)   %check if point we are looking at is
                                                        %within a rectangle of 10 by 11
          if or(or(and( b<y, (b+length)>y), and(b>y, (b-length)<y)), b==y)

             number(k)=number(k)+1;
          end
     end
  end
  if max<number(k)
     maxele=k;
     max=number(k);
  end
  end

end
if siz>1
centroid=[T(siz-maxele+1,4), T(siz-maxele+1,5)];  %make it the centroid=point that will define our area we want to be checking
    centroids=T(siz-maxele+1,:);
    a=centroid{1,1};            %a is the x of the centroid
     b=centroid{1,2};
for t=1:siz

  if t==maxele

    continue
  else
     point=[T(siz-t+1,4) T(siz-t+1,5)];
     x=point{1,1};               %x is the x coordinate of the point we are looking at in the table
      y=point{1,2};
     if or(or(and(a < x, (a+length)>x),and(a>x, (a-length)<x)), a==x)   %check if point we are looking at is
                                                        %within a rectangle of 10 by 11
          if or(or(and( b<y, (b+length)>y), and(b>y, (b-length)<y)), b==y)

             centroids=[centroids; T(siz-t+1,:)];
          else
```

```
            N=[N;T(siz-t+1,:)];
          end
      else
        N=[N;T(siz-t+1,:)];
      end
  end
end
else
  centroids=T;
end
```

CentroidPoints.m

```
function [C]=CentroidPoints(T)
T
siz=size(T,1);
countx=0;
county=0;
counttime=0;
sumx=0;
sumy=0;
sumtime=0;
times=[];
for k=1:siz
   countx=countx+1;
   county=county+1;
   counttime=counttime+1;

centroid=[T(siz-k+1,4), T(siz-k+1,5)];  %make it the centroid=point that will define our area we want to be checking

     sumx=sumx+centroid{1,1};            %a is the x of the centroid
      sumy=sumy+centroid{1,2};            %b is the y of the centroid
      sumtime=sumtime+T{siz-k+1,1}*3600+T{siz-k+1,2}*60+T{siz-k+1,3};
      times=[times, T{siz-k+1,1}*3600+T{siz-k+1,2}*60+T{siz-k+1,3}];
end
x=sumx/countx;
y=sumy/county;
%time=sumtime/counttime;
%time=median(times);
time=T{1,1}*3600+T{1,2}*60+T{1,3};
hours=floor(time./3600);
rest=mod(time,3600);
minutes=floor(rest./60);
seconds=mod(rest,60);
C=[hours, minutes, seconds, x, y]
C=array2table(C, 'VariableNames', {'hour';'minutes';'seconds';'centroidsx';'centroidsy'})
```

InsideArea.m

```
function [centroids, centroid, angle, No]=insideArea(T, centroids, centroid, angle)
% once table with all the centroids is found it will go one by one. Needs as input the table T
 %Table structure: hour  minutes    seconds   centroidsx   centroidsy
 length=10; %size of foot
 width=5;  %size of foot
 thre=2;

for j=1:size(T,1)
   siz=size(T,1);
   point=T(siz-j+1,:);  %define point by line being read on the table
   if and(j==1, not(exist('centroid','var')))                    %if it is the first point
       %if we input the centroid

         centroid=[T(siz,4), T(siz,5)];  %make it the centroid=point that will define our area we want to be checking
         centroids=T(siz,:);        %matrix that will contain all the centroids that are within the square


              %for plotting the first two points
   else
      a=centroid{1,1};             %a is the x of the centroid
      b=centroid{1,2};             %b is the y of the centroid
```

```
x=point{1,4};              %x is the x coordinate of the point we are looking at in the table
y=point{1,5};              %y is the y coordinate of the point we are looking at in the table
ex=0;
if exist('angle','var')==1
if islogical(angle)==0
  ex=1;
  changey=sin(angle)*width/2;
      changex=cos(angle)*width/2;
      ax=length*sin(angle);
      by=length*cos(angle);
      coord1=[a+changex, b+changey];
      coord2= [a-changex, b-changey];
      coord3=[coord1(1,1)-ax, coord1(1,2)+by];
      coord4=[coord2(1,1)-ax, coord2(1,2)+by];
end
else
  angle=true;               %angle will become a double, but in order to easily det if angle was calculated
               % or not we will use another
               % classtype

end

if and(height(centroids)==1, ex==0)      %if no point has been added to our centroid besides the first
  if or(or(and(a < x, (a+length)>x),and(a>x, (a-length)<x)), a==x)   %check if point we are looking at is
                                    %within a rectangle of 10 by 11
    if or(or(and( b<y, (b+length)>y), and(b>y, (b-length)<y)), b==y)

      if and(or(or(and(a < x, (a+1)>x),and(a>x, (a-1)<x)), a==x), or(or(and( b<y, (b+1)>y), and(b>y, (b-1)<y)), b==y))   %check if point we
are looking at is
                                    %within a rectangle of 10 by 11
      c=point(1,1:3);
       cent=[c,centroid];
       Closepoints=[cent; point];
       centroid=CentroidPoints(Closepoints);
       centroids=centroid;
       centroid=centroid(1, 4:5);
      else
        centroids=[centroids; point];          %if it is is then it will be our second point
                                    %that will define the direction of the footstep


       X=(x-a)/(y-b);     %define angle at which the second point is identified

       angle=atan(X);
       if y<b
       angle=pi-angle;
       end
       %define the 4 points of the rectangle
       changey=sin(angle)*width/2+thre;
       changex=cos(angle)*width/2+thre;
       ax=length*sin(angle);
       by=length*cos(angle);
       coord1=[a+changex, b+changey];
       coord2= [a-changex, b-changey];
       %'point 2 passed'
       coord3=[coord1(1,1)-ax, coord1(1,2)+by];
       coord4=[coord2(1,1)-ax, coord2(1,2)+by];
      end
    else
       if exist('No','var')==0
       No=point;
    else
       No=[No;point];

      end
    end
    else
       if exist('No','var')==0
       No=point;
    else
       No=[No;point];
    end
```

```
      end

    else
      %if we are not working with the first nor the second point we
      %will use the convex hull idea to check if the point is within
      %the four vertices of the rectangle defined in the end
      xs=[coord1(1,1); coord2(1,1); coord3(1,1); coord4(1,1); point{1,4}];
      ys=[coord1(1,2); coord2(1,2); coord3(1,2); coord4(1,2); point{1,5}];

      K=convhull(xs,ys);
      si=size(K);
      mx=0;

      for i=1:si(1,1)

        if K(i,1)>mx
          mx=K(i,1);

        end

      end
      %if convex hull finds only 4 points in the border then the
      %point must be within the rectangle
      if mx==4
        centroids=[centroids; point];
      else
        if exist('No','var')==0
          No=point;
        else
          No=[No;point];
        end
      end
    end


  end
end
if exist('angle','var')==0
  angle=true;
end
if exist('No','var')==0
    No=[];
  end
%plotting the points.
% if height(centroids)>1
% x=[coord1(1,1); coord2(1,1); coord3(1,1); coord4(1,1)];
% y=[coord1(1,2); coord2(1,2); coord3(1,2); coord4(1,2)];
% k=convhull(x,y);
%
% end
```

Appendix D: Lawton Instrumental Activities of Daily Living Scale

Activities of daily living test that is administered to elderly individuals.

try this:® general assessment series

Best Practices in Nursing Care to Older Adults

From The Hartford Institute for Geriatric Nursing, New York University, College of Nursing

Issue Number 23, Revised 2013

Editor-in-Chief: Sherry A. Greenberg, PhD(c) MSN, GNP-BC
New York University College of Nursing

The Lawton Instrumental Activities of Daily Living (IADL) Scale

By: Carla Graf, PhD(c), MS, RN, GCNS-BC, University of California, San Francisco

WHY: The assessment of functional status is critical when caring for older adults. Normal aging changes, acute illness, worsening chronic illness, and hospitalization can contribute to a decline in the ability to perform tasks necessary to live independently in the community. The information from a functional assessment can provide objective data to assist with targeting individualized rehabilitation needs or to plan for specific in home services such as meal preparation, nursing and personal care, home-maker services, financial and medication management, and/or continuous supervision. A functional assessment can also guide the clinician to focus on the person's baseline capabilities, facilitating early recognition of changes that may signify a need either for additional resources or for a medical work-up (Gallo & Paveza, 2006).

BEST TOOL: The Lawton Instrumental Activities of Daily Living Scale (IADL) is an appropriate instrument to assess independent living skills (Lawton & Brody, 1969). These skills are considered more complex than the basic activities of daily living as measured by the Katz Index of ADLs (See *Try this*: Katz Index of ADLs). The instrument is most useful for identifying how a person is functioning at the present time and for identifying improvement or deterioration over time. There are 8 domains of function measured with the Lawton IADL scale. Historically, women were scored on all 8 areas of function; men were not scored in the domains of food preparation, housekeeping, laundering. However, current recommendations are to assess all domains for both genders (Lawton, Moss, Fulcomer, & Kleban, 2003). Persons are scored according to their highest level of functioning in that category. A summary score ranges from 0 (low function, dependent) to 8 (high function, independent).

TARGET POPULATION: This instrument is intended to be used among older adults, and may be used in community, clinic, or hospital settings. The instrument is not useful for institutionalized older adults. It may be used as a baseline assessment tool and to compare baseline function to periodic assessments.

VALIDITY AND RELIABILITY: Few studies have been performed to test the Lawton IADL scale psychometric properties. The Lawton IADL Scale was originally tested concurrently with the Physical Self-Maintenance Scale (PSMS). Reliability was established with twelve subjects interviewed by one interviewer with the second rater present but not participating in the interview process. Inter-rater reliability was established at 0.85. The validity of the Lawton IADL was tested by determining the correlation of the Lawton IADL with four scales that measured domains of functional status, the Physical Classification (6-point rating of physical health), Mental Status Questionnaire (10-point test of orientation and memory), Behavior and Adjustment rating scales (4-6-point measure of intellectual, person, behavioral and social adjustment), and the PSMS (6-item ADLs). A total of 180 research subjects participated in the study, however, few received all five evaluations. All correlations were significant at the 0.01 or 0.05 level. To avoid potential gender bias at the time the instrument was developed, specific items were omitted for men. This assessment instrument is widely used both in research and clinical practice.

STRENGTHS AND LIMITATIONS: The Lawton IADL is an easy to administer assessment instrument that provides self-reported information about functional skills necessary to live in the community. Administration time is 10-15 minutes. Specific deficits identified can assist nurses and other disciplines in planning for safe hospital discharge.

A limitation of the instrument includes the self-report or surrogate report method of administration rather than a demonstration of the functional task. This may lead either to over-estimation or under-estimation of ability. In addition, the instrument may not be sensitive to small, incremental changes in function.

FOLLOW-UP: The identification of new disabilities in these functional domains warrants intervention and further assessment to prevent ongoing decline and to promote safe living conditions for older adults. If using the Lawton IADL tool with an acute hospitalization, nurses should communicate any deficits to the physicians and social workers/case managers for appropriate discharge planning.

MORE ON THE TOPIC:

Best practice information on care of older adults: www.ConsultGeriRN.org.

Gallo, J.J., & Paveza, G.J. (2006). Activities of daily living and instrumental activities of daily living assessment. In J.J. Gallo, H.R. Bogner, T. Fulmer, & G.J. Paveza (Eds.), *Handbook of Geriatric Assessment* (4th ed., pp. 193-240). MA: Jones and Bartlett Publishers.

Graf, C. (2006). Functional decline in hospitalized older adults. *AJN, 106*(1), 58-67.

Graf, C. (2008). The Lawton Instrumental Activities of Daily Living Scale. *AJN, 108*(4), 52-62.

Lawton, M.P., & Brody, E.M. (1969). Assessment of older people: Self-maintaining and instrumental activities of daily living. *The Gerontologist, 9*(3), 179-186.

Lawton, M.P., Moss, M., Fulcomer, M., & Kleban, M. H. (2003). *Multi-level assessment instrument manual for full-length MAI.* North Wales PA: Polisher Research Institute, Madlyn and Leonard Abramson Center for Jewish Life.

Pearson, V. (2000). Assessment of function. In R. Kane, & R. Kane (Eds.), *Assessing Older Persons. Measures, Meaning and Practical Applications* (pp. 17-48). New York: Oxford University Press.

The Lawton Instrumental Activities of Daily Living Scale

A. Ability to Use Telephone

1. Operates telephone on own initiative; looks up and dials numbers ... 1
2. Dials a few well-known numbers ... 1
3. Answers telephone, but does not dial ... 1
4. Does not use telephone at all ... 0

B. Shopping

1. Takes care of all shopping needs independently ... 1
2. Shops independently for small purchases ... 0
3. Needs to be accompanied on any shopping trip ... 0
4. Completely unable to shop ... 0

C. Food Preparation

1. Plans, prepares, and serves adequate meals independently ... 1
2. Prepares adequate meals if supplied with ingredients ... 0
3. Heats and serves prepared meals or prepares meals but does not maintain adequate diet ... 0
4. Needs to have meals prepared and served ... 0

D. Housekeeping

1. Maintains house alone with occasion assistance (heavy work) ... 1
2. Performs light daily tasks such as dishwashing, bed making ... 1
3. Performs light daily tasks, but cannot maintain acceptable level of cleanliness ... 1
4. Needs help with all home maintenance tasks ... 1
5. Does not participate in any housekeeping tasks ... 0

E. Laundry

1. Does personal laundry completely ... 1
2. Launders small items, rinses socks, stockings, etc. ... 1
3. All laundry must be done by others ... 0

F. Mode of Transportation

1. Travels independently on public transportation or drives own car ... 1
2. Arranges own travel via taxi, but does not otherwise use public transportation ... 1
3. Travels on public transportation when assisted or accompanied by another ... 1
4. Travel limited to taxi or automobile with assistance of another ... 0
5. Does not travel at all ... 0

G. Responsibility for Own Medications

1. Is responsible for taking medication in correct dosages at correct time ... 1
2. Takes responsibility if medication is prepared in advance in separate dosages ... 0
3. Is not capable of dispensing own medication ... 0

H. Ability to Handle Finances

1. Manages financial matters independently (budgets, writes checks, pays rent and bills, goes to bank); collects and keeps track of income ... 1
2. Manages day-to-day purchases, but needs help with banking, major purchases, etc ... 1
3. Incapable of handling money ... 0

Scoring: For each category, circle the item description that most closely resembles the client's highest functional level (either 0 or 1).

Lawton, M.P., & Brody, E.M. (1969). Assessment of older people: Self-maintaining and instrumental activities of daily living. *The Gerontologist, 9*(3), 179-186.

Figure 139: Lawton Instrumental Activities of Daily Living Scale.
Image Source: Hartford Institute for Geriatric Nursing.

Appendix E: Tekscan Walkway Gait Analysis System

Tekscan Walkway gait analysis system informational brochure.

Tekscan®

Walkway™
Gait Analysis System

The Walkway™ is the only platform-based gait analysis system that captures force and plantar pressure data, PLUS temporal (time) and spatial (distance) gait parameters. Walkway is a versatile, research-validated tool that can be used alone or integrated with other systems, such as EMG or 3D motion capture. Walkway is...

- **Accurate and reliable** – validated by leading researchers for a variety of applications.
- **Best-in-class for profiling anatomical landmarks** – highest resolution sensors on the market.
- **Fast and easy to use** – capture multiple footsteps in a single pass with no setup required.
- **Low profile** – thinnest platform on the market minimizes gait changes, even in small children.

Selection Guide and Specifications

	Standard Resolution (Base)	Standard Resolution (High Speed)	High-Resolution (Pediatric)
Sensor Model	3150	3150E	7101
Sensor Technology	Resistive	Resistive	Resistive
Resolution	1.4 sensels per cm²/ 9.2 sensels™ per in²	1.4 sensels per cm²/ 9.2 sensels per in²	4 sensels per cm²/ 25 sensels per in²
Platform Height	0.6 cm / 0.2 in	0.6 cm / 0.2 in	0.6 cm / 0.2 in
Electronics Included	1 Evolution® Handle per sensor & 1 Multi-Port Powered USB Hub	1 VersaTek® Cuff per sensor & 1 VersaTek 2-Port Hub for every 2 sensors	2 VersaTek Cuffs & 1 VersaTek 2-Port Hub per sensor
Maximum Scan Speed	100 Hz	440 Hz	185 Hz
Connection Type	USB 2.0	USB 2.0	USB 2.0
Pressure Ranges	40, 75, 125 psi (276, 518, 862 kPa)	40, 75, 125 psi (276, 518, 862 kPa)	40, 75, 125 psi (276, 518, 862 kPa)
Power	USB BUS	0.35 A, 100 – 200 VAC, 50 – 60 cycles	USB BUS

DATA SHEET

Walkway Gait Analysis System – Software Features

Tekscan's advanced, proprietary Walkway software helps you analyze your data in a variety of ways. All Tekscan software works with current Windows based operating systems. Software features include:

- Automated calculation of gait parameters displayed in table
- Automated 3-Segment foot model
- Pressure profiles (visual pressure distribution)
- Force vs. time graphs
- Patient/subject database
- Side-by-side comparison for pre-and post-recordings
- Peak/stance average view
- Single or multiple passes in a recording
- Export recorded movies as AVI or ASCII files
- External triggering and data synch

Walkway Software features automatic placement of strike boxes.

Selection Guide

Standard Resolution

Standard Resolution Walkways are appropriate for evaluating adults and older children. They are available with either Evolution or VersaTek electronics, depending on the scanning rate needed for a given application.

Configurations – Standard Resolution

	Overall Dimensions		Sensing Area	
	Centimeters	Inches	Centimeters	Inches
WE2 or WV2*	93.5 x 58.4	36.8 x 23.0	87.2 x 36.9	34.3 x 14.5
WE3 or WV3*	158.6 x 58.4	62.5 x 23.0	130.8 x 36.9	51.5 x 14.5
WE4 or WV4*	207.1 x 58.4	81.5 x 23.0	174.4 x 36.9	68.6 x 14.5
WE5 or WV5*	243.84 x 58.4	96.0 x 23.0	218.0 x 36.9	85.8 x 14.5
WE6 or WV6*	287.0 x 58.4	113.0 x 23.0	261.5 x 36.9	103.0 x 14.5
WE7 or WV7*	330.2 58.4	130.0 x 23.0	306.0 x 36.9	120.5 x 14.5

**E is for Evolution electronics (100 Hz max scanning) and V is for VersaTek electronics (440 Hz max scanning).*

High Resolution

High Resolution Walkways are best for evaluating small children. It is also suitable for evaluating adults. All High Resolution Walkways operate with Tekscan's patented VersaTek electronics, which provide faster scan rates of up to 185 Hz.

Configurations – High Resolution

	Overall Dimensions		Sensing Area	
	Centimeters	Inches	Centimeters	Inches
HRV2	99.3 x 58.4	39.0 x 23.0	97.5 x 44.7	38.4 x 17.6
HRV3	148.5 x 58.4	58.5 x 23.0	146.3 x 44.7	51.5 x 14.5
HRV4	207.0 x 58.4	81.5 x 23.0	195.1 x 44.7	76.8 x 17.6
HRV5	256.5 x 58.4	101.0 x 23.0	243.8 x 44.7	96.0 x 17.6
HRV6	304.8 x 58.4	120.0 x 23.0	292.6 x 44.7	115.2 x 17.6
HRV7	330.2 x 58.4	139.0 x 23.0	341.4 x 44.7	134.4 x 17.6

Tekscan

To view the complete computer requirements visit www.tekscan.com/computer-requirements. Specifications subject to change.

CALL TODAY FOR A DEMONSTRATION!

+1.617.464.4281 | 1.800.248.3669 | info@tekscan.com | www.tekscan.com/medical

DS_RevA_020215

Figure 140: Tekscan Walkway Gait Analysis System. Image Source: Tekscan.

Appendix F: Masking Tests

Sheet Labels:

- O – original copper sheet, no coating
- OP – first paint sheet (2 layers of paint)
- OPP – first paint + primer sheet (5 layers of primer, 2 layers of paint)
- N1 – 1 layer of primer
- N2 – 2 layers of primer
- N3 – 3 layers of primer
- P1 – 1 layer of primer + 1 layer of paint
- P2 – 2 layers of primer + 1 layer of paint
- P3 – 3 layers of primer + 1 layer of paint
- Expected – the values that would have been idea for every sheet

resistance - Used the following points (1, 2, 3, 4, 5) on the copper sheet. (a,b) means the resistance between points a and b was measured. Resistance was measured in ohms.

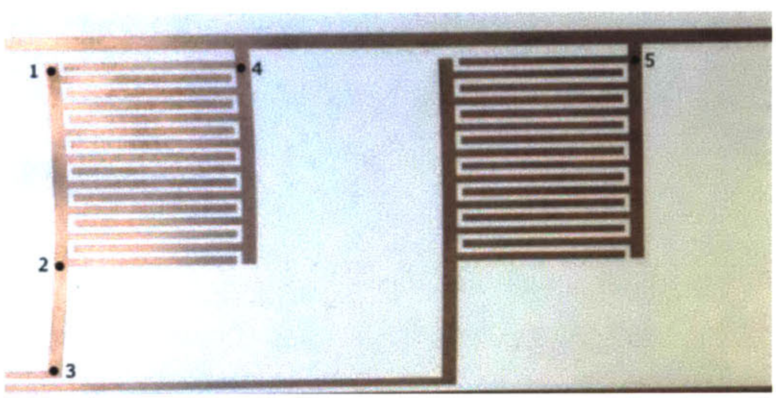


Figure 141: Points for testing resistance before and after masking with paint or printer ink.

slide – "a -> b" means that the two leads were placed on points a and b, then the lead on point a was slid (along the tracings) to point b to see if there was any conductance.

n = no conductance, y = conductance

crinkle – Crinkled the sheet a certain number (specified on spreadsheet) of times to see if the coat on the sheet was affected.

rub – Rubbed the sheet 40 times to see if the coat was affected. Did not test this on N1/N2/N3 because coat was not visible.

scratch – Scratched the sheet to see if the coat was affected.

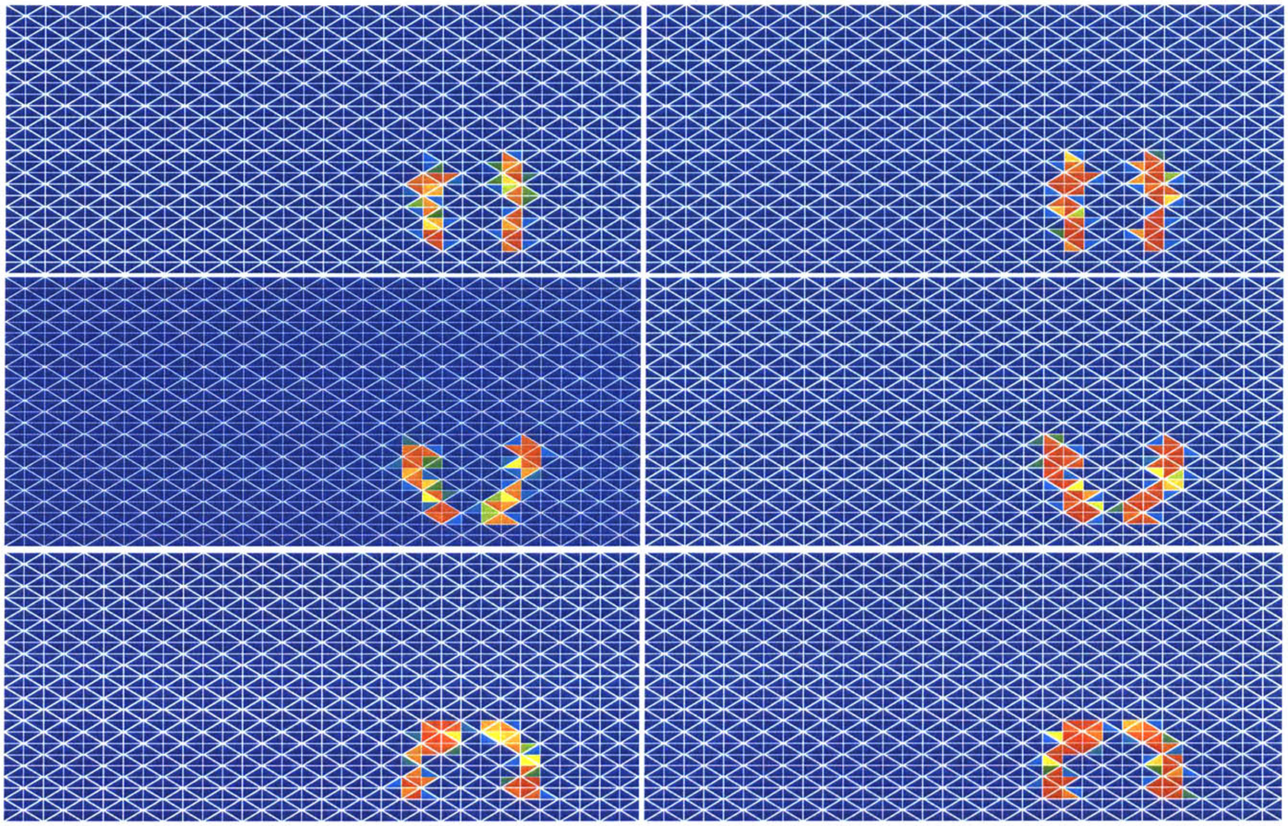
- **horiz** – scratched thumb horizontally 20 times
- **vert** – scratched thumb vertically 20 times
- **metal** (to break) – counted the number of scratches it took to break through the coat with a sharp metal object

Table 11: Results of resistance, slide, crinkle, rub, and scratch tests.

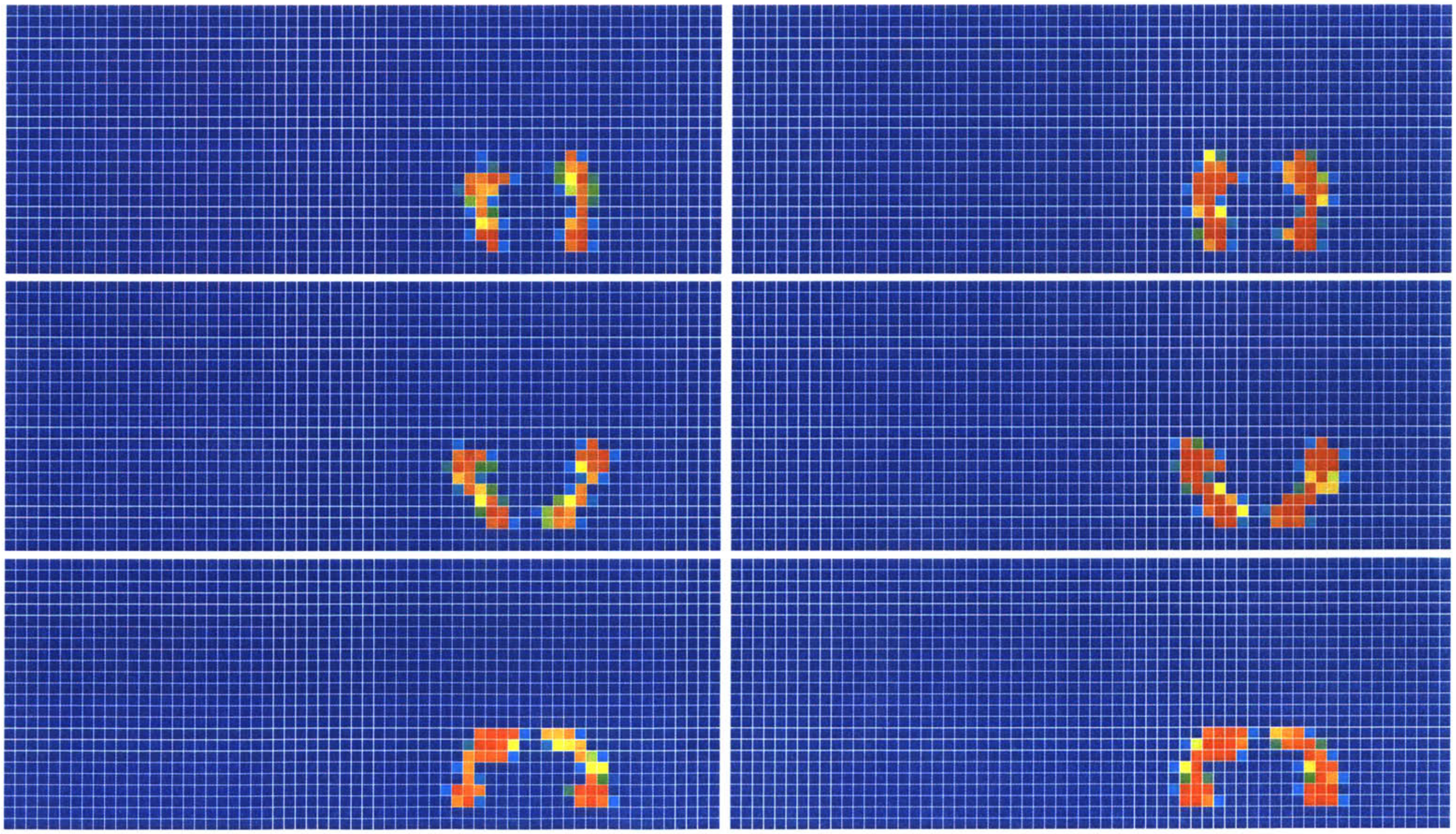
	resistance (in ohms)				slide	comments	crinkle			rub	scratch		
	(1,2)	(1,3)	(2,3)	(1,4)	2->3		10	20	40	thumb	horiz	vert	metal (to break)
O	0.7	1.3	0.5	∞	n								
OP	0.8	∞	∞	∞	n		1 flake	many cracks, 2 flakes	many flakes	residue	scuff (more)	slight lines	10
OPP	1	∞	∞	∞	n		none	none	some lines	nothing	scuff	slight lines	20
N1	0.6	1.7	0.6	∞	y	non-uniform conductance near 3	none	bend on plastic	< same	-	slight lines	break	1
N2	0.6	∞	0.6	∞	y	^same	none	bend	< same	-	none	break	2
N3	0.6	∞	∞	∞	n		none	bend	< same	-	none	break	2
P1	0.6	∞	∞	∞	n		none	bend	< same	nothing	scuff	break	1
P2	0.6	∞	∞	∞	n		none	none	bend	nothing	scuff	break	4
P3	0.6	∞	∞	∞	n		none	none	bend	nothing	scuff	break	5
expected	0.7	∞	∞	∞	n		none	none	none	nothing	none	none	20

Appendix G: Shape Characterization Test Images

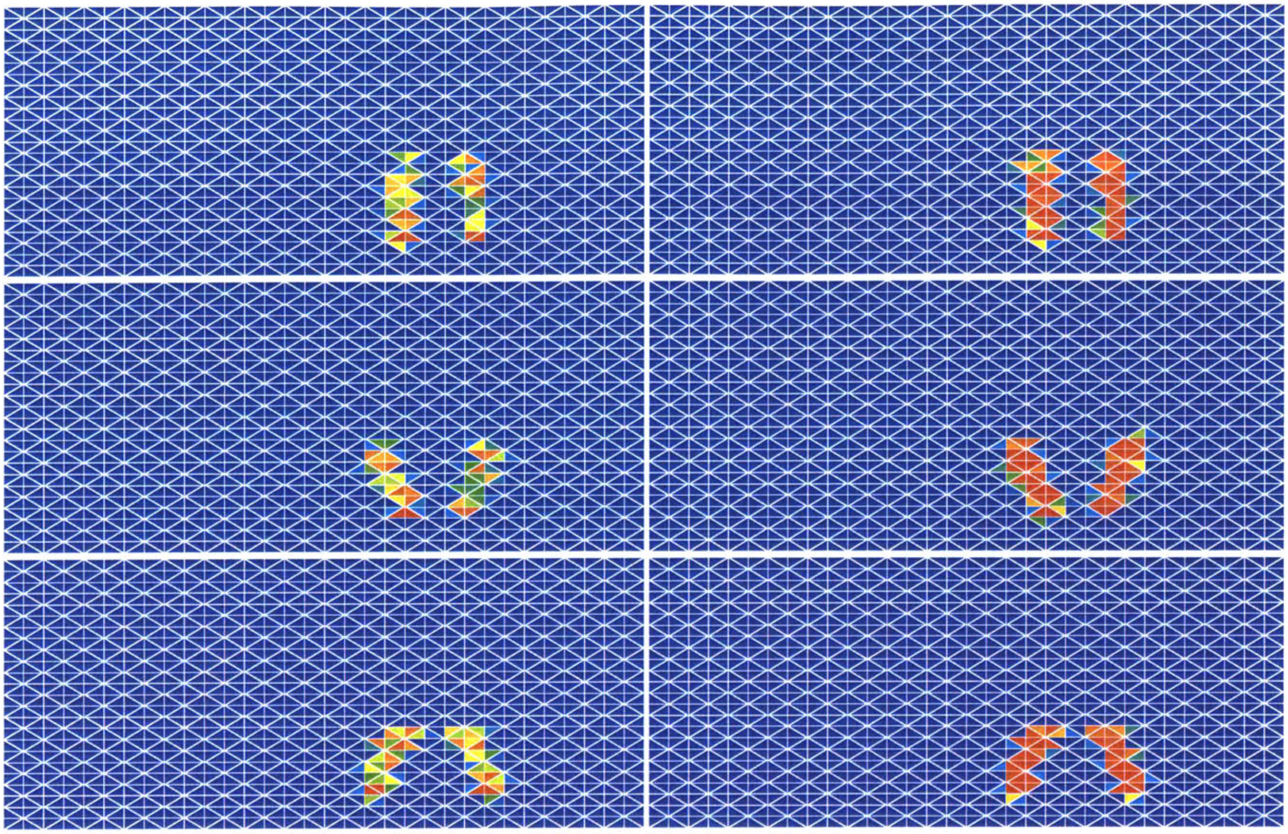
Triangle Pixel, Arched Foot, 40lbs Left Column, 80lbs Right Column



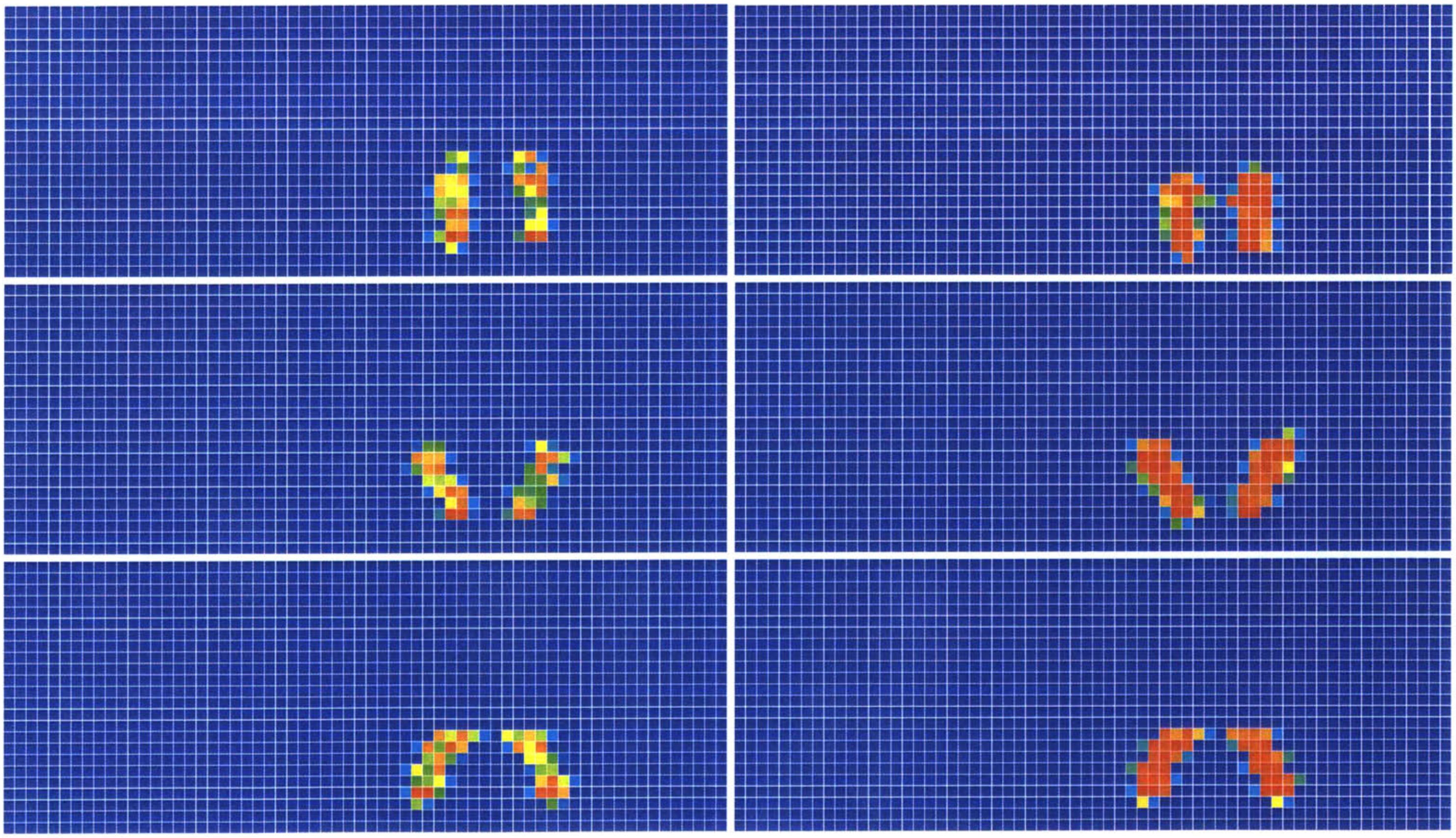
Square Pixel, Arched Foot, 40lbs Left Column, 80lbs Right Column



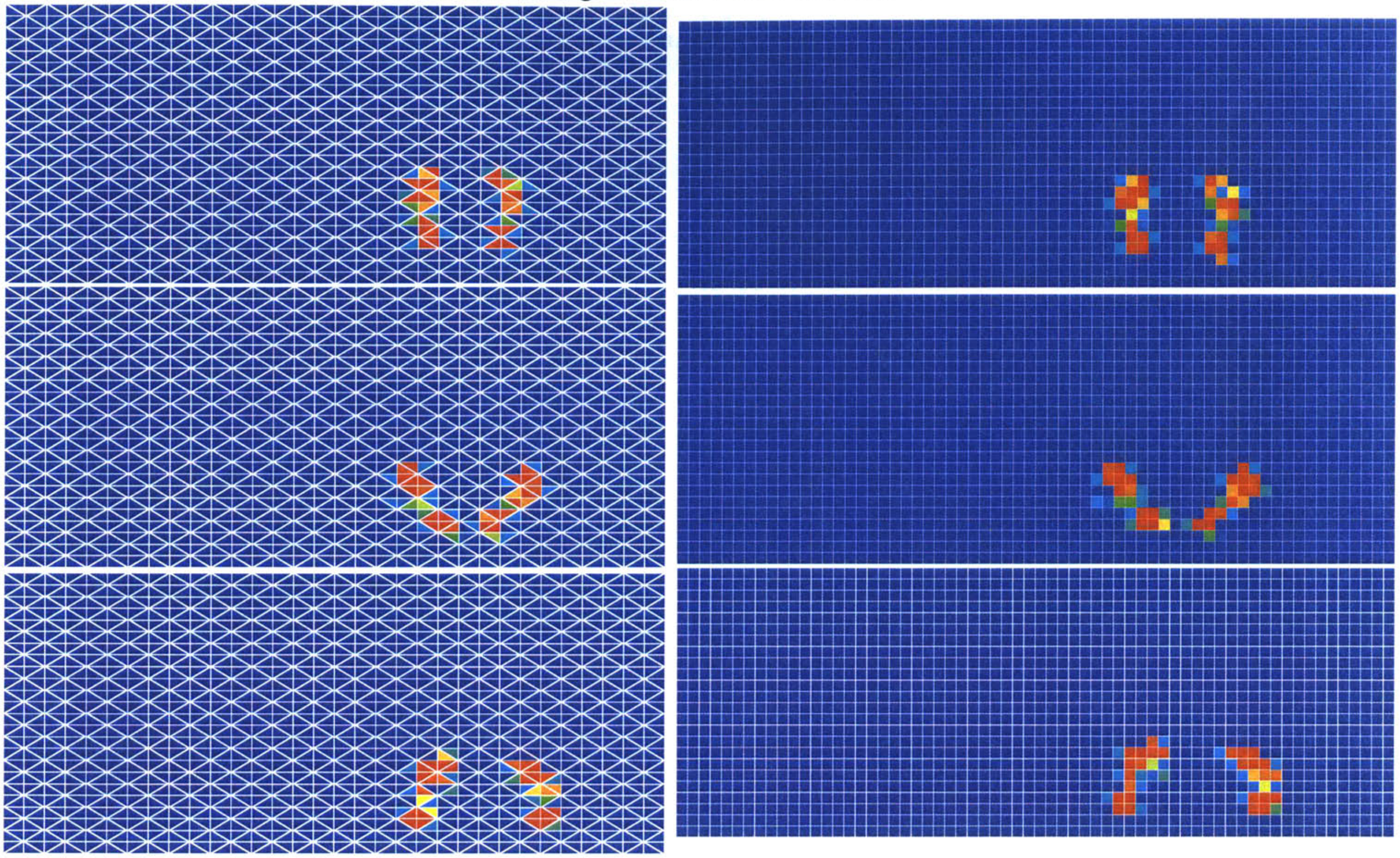
Triangle Pixel, Flat Foot, 40lbs Left Column, 80lbs Right Column



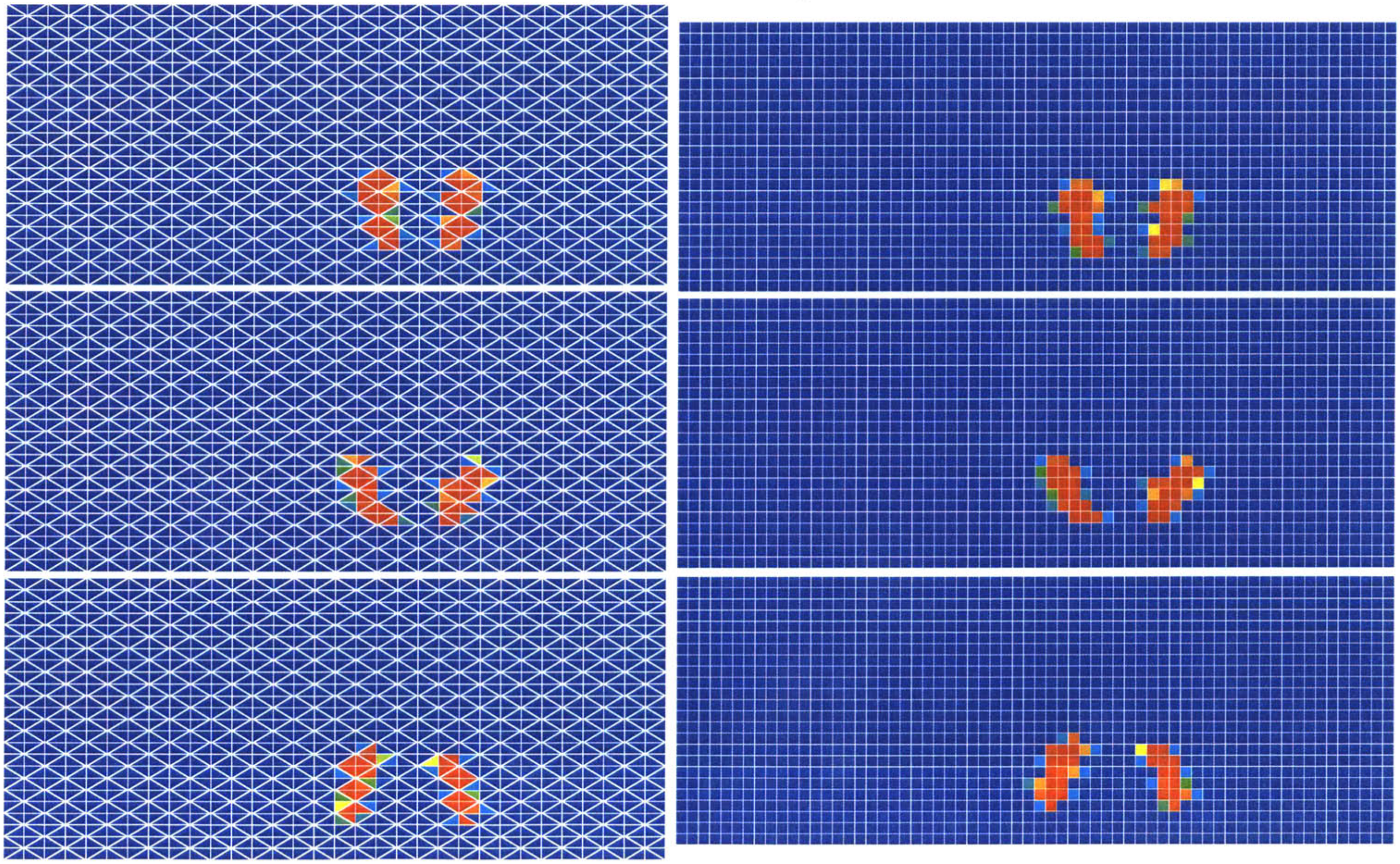
Square Pixel, Flat Foot, 40lbs Left Column, 80lbs Right Column



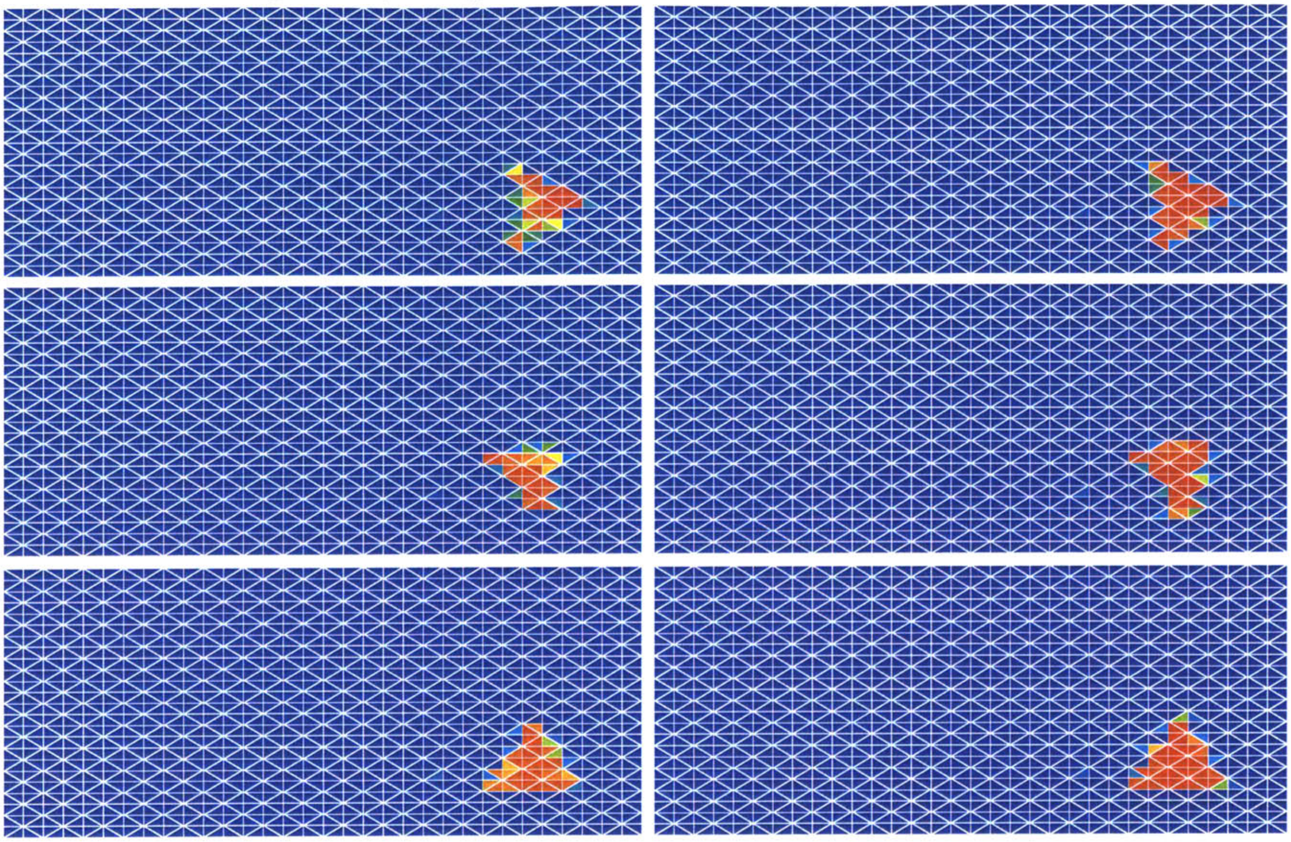
Me Standing Without Shoes On, 165lbs



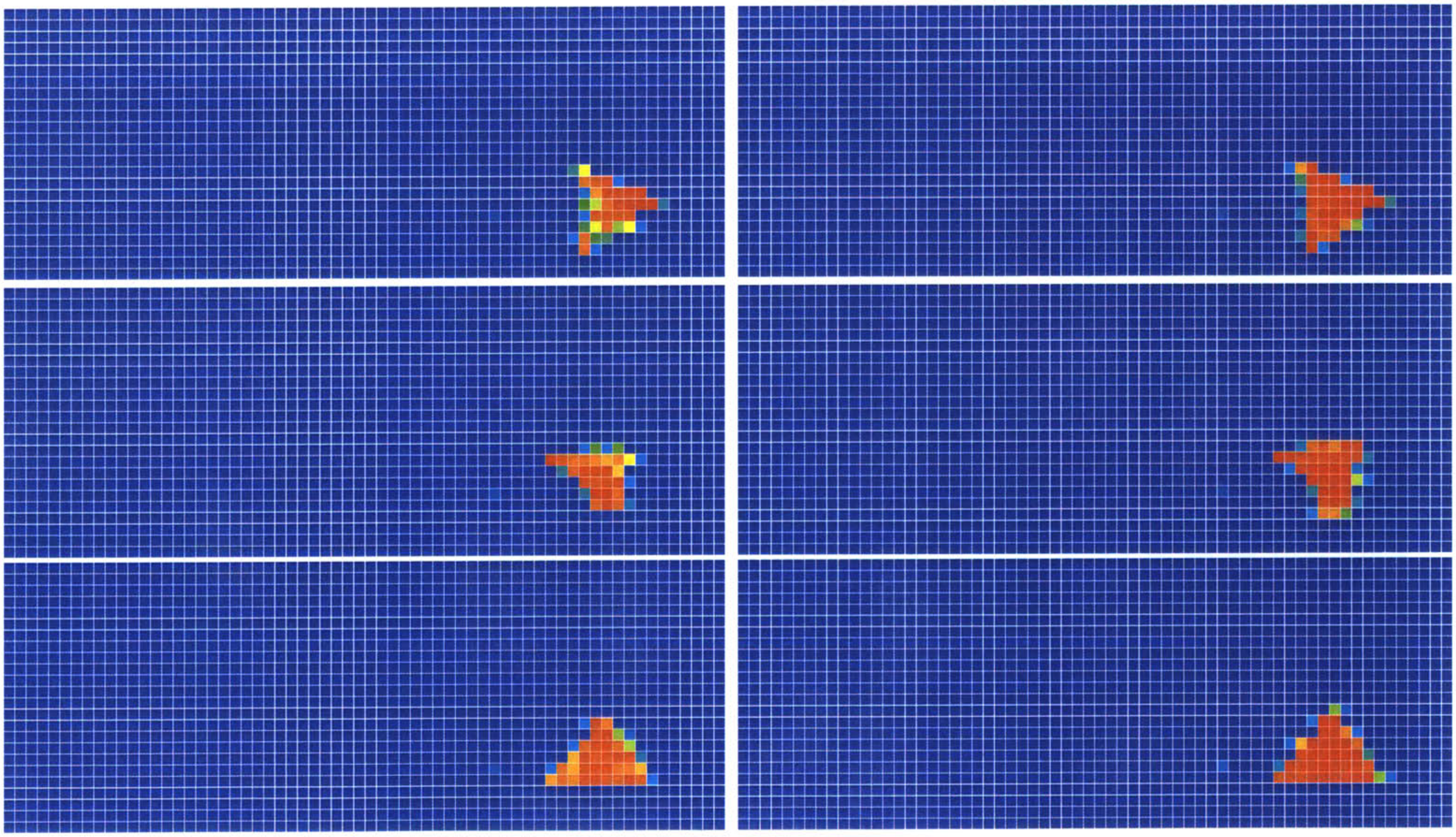
Me Standing With Shoes On, 165lbs



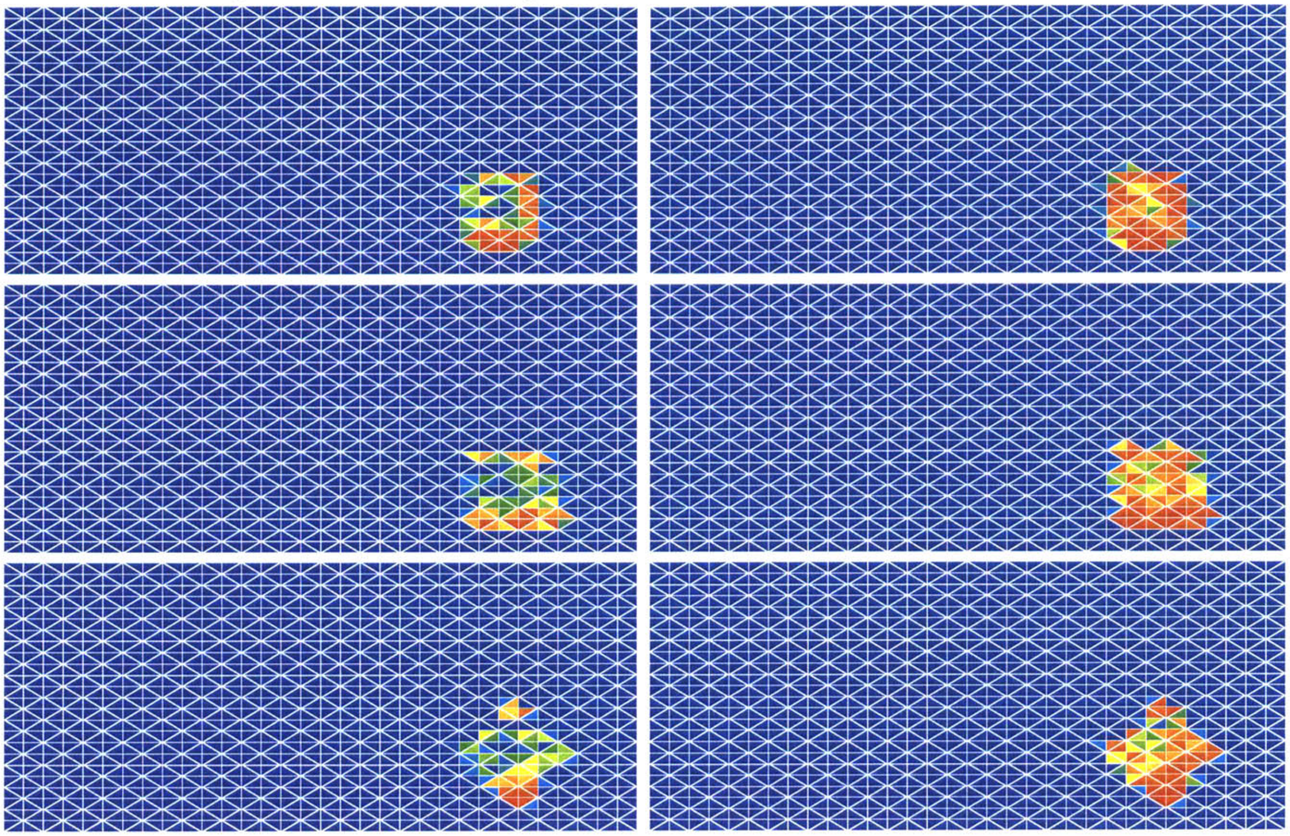
Triangle Pixel, Triangle, 40lbs Left Column, 80lbs Right Column



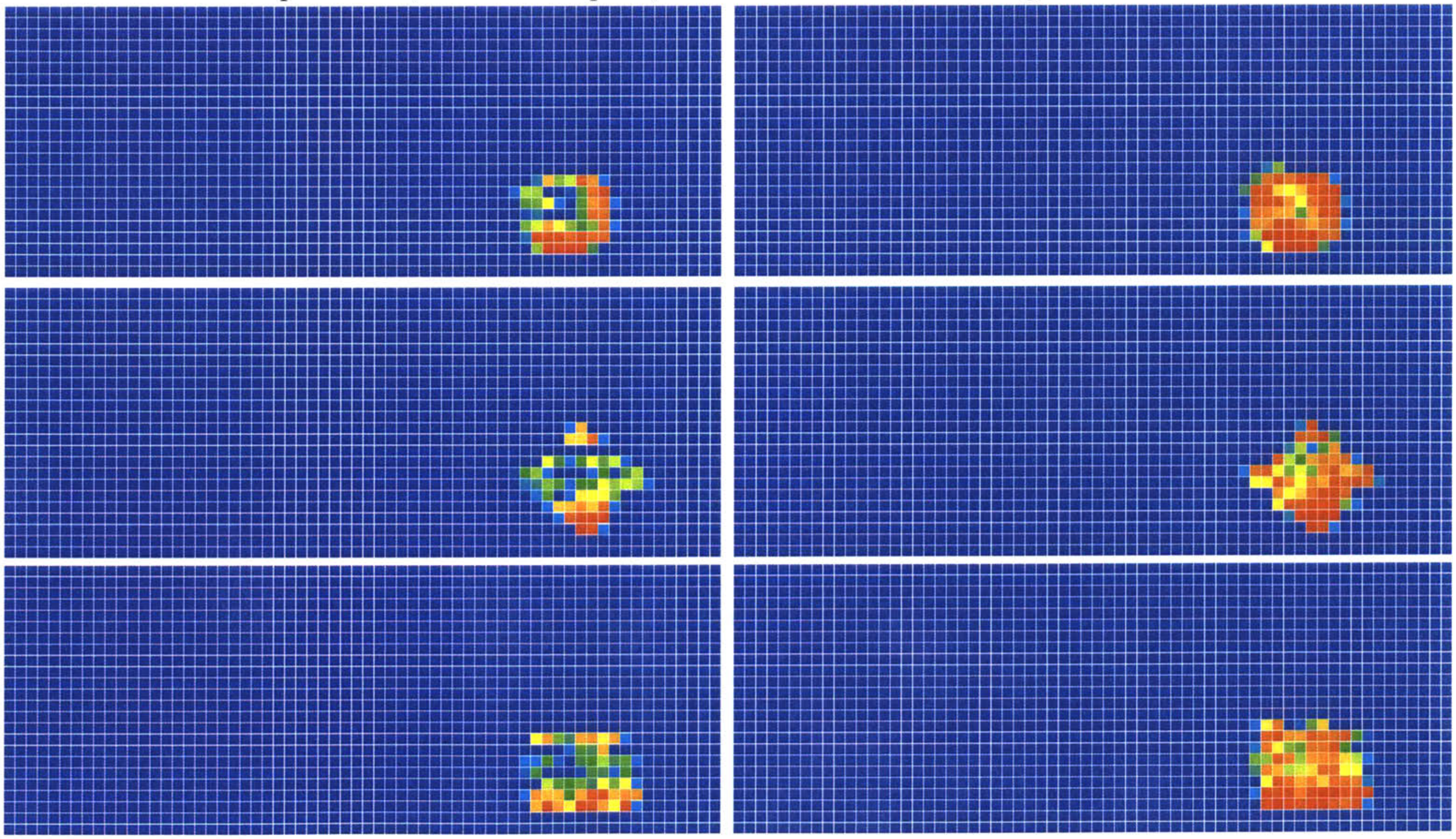
Square Pixel, Tringle, 40lbs Left Column, 80lbs Right Column



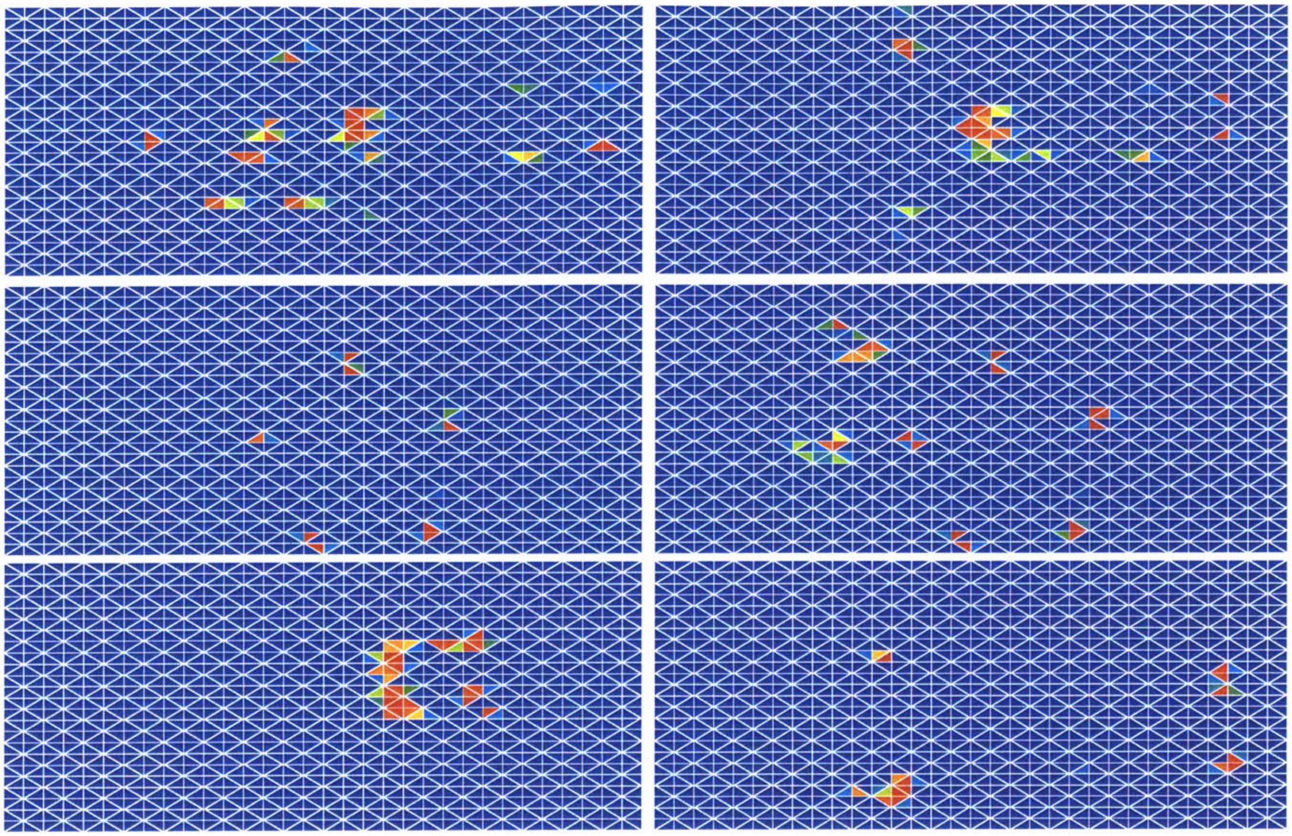
Triangle Pixel, Circle and Square, 40lbs Left Column, 80lbs Right Column

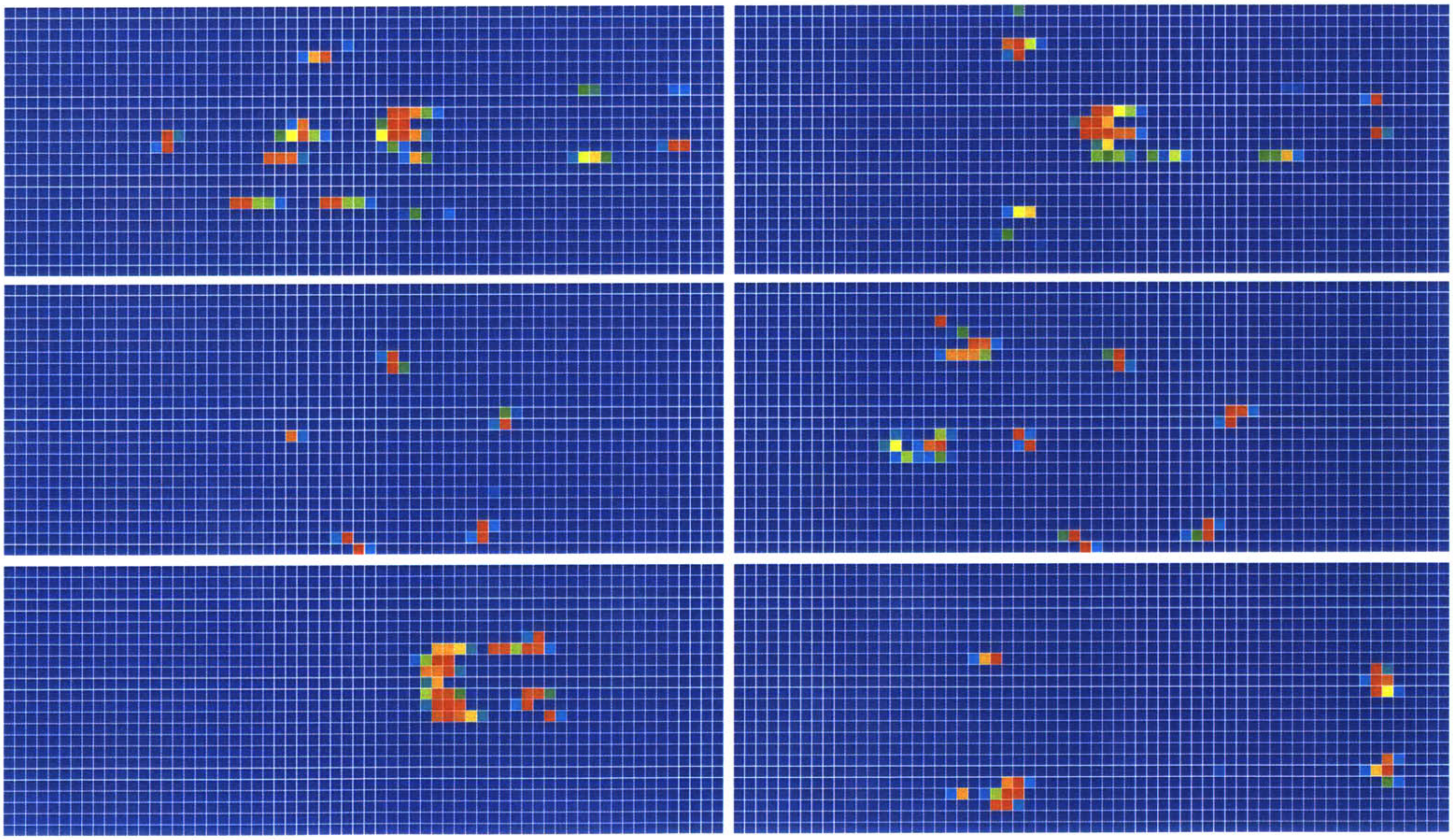


Square Pixel, Circle and Square, 40lbs Left Column, 80lbs Right Column



Activities Described in Section 8.3.6





Appendix H: Slave PCB Bill of Materials

Table 12: Bill of Materials for Slave PCB.

#	LibRef	Particular Site Part Number	Manufacturer Part Number	PartType	Description	Footprint	Quantity
1	ATmega328P-AU	ATMEGA328-AURCT-ND	ATMEGA328-AUR	ATmega328P-AU	8-bit AVR Microcontroller, 32KB Flash, 1KB EEPROM, 2KB SRAM, 32-pin TQFP, Industrial Grade (-40°C to 85°C)	Quad32-Atmega_Microcontroller	1
2	Capacitor	490-1532-1-ND	GRM188R71C104KA01D	0.1uF	SMD Cap 1206	603	2
3	Header 68	HFV068CT-ND	FH28E-68S-0.5SH(05)	68 PIN HEADER	CONN FFC BOTTOM 68POS 0.50MM R/A	FH28D-68S-0.5SH-FFC-Hirose	8
4	Header 5	609-3462-ND	68000-105HLF	5 PIN HEADER	Header, 5-Pin	HDR1X5	2
5	LED	160-1827-1-ND	LTST-C193TBKT-5A	BLUE	LED SMD 1206/0603	603	1
6	LED	160-1830-1-ND	LTST-C193KRKT-5A	RED / ORANGE	LED SMD 1206/0603	603	1
7	CD74HC4067	296-9226-1-ND	CD74HC4067SM96	CD74HC4067	1:16 multiplexer	SSOP24_M	32
8	Resonator SMD	COM-00094	SPK-5032-16MHZ	Resonator SMD	CRYSTAL 16MHZ 18PF SMD	ECS-160-18-23A-EN-TR	1
9	Header 3X2	952-1921-ND	M20-9720345	Header 3X2	Header, 3-Pin, Dual row	HDR2X3	1
10	Resistor	RMCF0603JT330RCT-ND	RMCF0603JT330R	330	SMD Res	603	2
11	Resistor	RMCF0603FT10K0CT-ND	RMCF0603FT10K0	10K	SMD Res	603	35
12	SN74HC595D	MM74HC595MXCT-ND	MM74HC595MX	MM74HC595M	8-Bit Shift Register with 3-State Output Registers	SOIC16_M	8
							94

#	LibRef	Particular Site Part Number	Manufacturer Part Number	PartType	Description	Footprint	Quantity
1	ATmega328P-AU	ATMEGA328-AURCT-ND	ATMEGA328-AUR	ATmega328P-AU	8-bit AVR Microcontroller, 32KB Flash, 1KB EEPROM, 2KB SRAM, 32-pin TQFP, Industrial Grade (-40°C to 85°C)	Quad32-Atmega_Microcontroller	1
2	Capacitor	490-1532-1-ND	GRM188R71C104KA01D	0.1uF	SMD Cap 1206	0603	2
3	Header 68	HFV068CT-ND	FH28E-68S-0.5SH(05)	68 PIN HEADER	CONN FFC BOTTOM 68POS 0.50MM R/A	FH28D-68S-0.5SH-FFC-Hirose	8
4	Header 5	609-3462-ND	68000-105HLF	5 PIN HEADER	Header, 5-Pin	HDR1X5	2
5	LED	160-1827-1-ND	LTST-C193TBKT-5A	BLUE	LED SMD 1206/0603	0603	1
6	LED	160-1830-1-ND	LTST-C193KRKT-5A	RED / ORANGE	LED SMD 1206/0603	0603	1
7	CD74HC4067	296-9226-1-ND	CD74HC4067SM96	CD74HC4067	1:16 multiplexer	SSOP24_M	32
8	Resonator SMD	COM-00094	SPK-5032-16MHZ	Resonator SMD	CRYSTAL 16MHZ 18PF SMD	ECS-160-18-23A-EN-TR	1
9	Header 3X2	952-1921-ND	M20-9720345	Header 3X2	Header, 3-Pin, Dual row	HDR2X3	1
10	Resistor	RMCF0603JT330RCT-ND	RMCF0603JT330R	330	SMD Res	0603	2
11	Resistor	RMCF0603FT10K0CT-ND	RMCF0603FT10K0	10K	SMD Res	0603	35
12	SN74HC595D	MM74HC595MXCT-ND	MM74HC595MX	MM74HC595M	8-Bit Shift Register with 3-State Output Registers	SOIC16_M	8
							94

Appendix I: Slave PCB Schematic Detail

The shift register/four multiplexer/68-pin connector section is repeated eight times per board.