Graspables:
Grasp Recognition as a User Interface

Brandon Thomas Taylor

B.Sc. Electrical and Computer Engineering
University of Texas at Austin, May 2006
B.Sc. Radio, Television and Film
University of Texas at Austin, May 2006

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
September 2008

Author **Brandon Thomas Taylor**
Program in Media Arts and Sciences
August 29, 2008

Certified by **V. Michael Bove, Jr.**
Principal Research Scientist
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by **Deb Roy**
Chair, Departmental Committee on Graduate Studies
Program in Media Arts and Sciences

Graspables:
Grasp Recognition as a User Interface

Brandon Thomas Taylor

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on August 29, 2008
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences at the Massachusetts
Institute of Technology

Abstract

The Graspables project is an exploration of how measuring the way people hold and manipulate objects can be used as a user interface. As computational power continues to implemented in more and more objects and devices, new interaction methods need to be developed. The Graspables System is embodied by a physical set of sensors combined with pattern recognition software that can determine how users hold a device. The Graspables System has been implemented in two prototypes, the Bar of Soap and the Ball of Soap. User studies have been conducted demostrating the effectiveness of the Graspables System and a variety of applications have been developed to demonstrate its utility.

Certified by **V. Michael Bove, Jr.**
Principal Research Scientist
Program in Media Arts and Sciences
Thesis Supervisor

Graspables:
Grasp Recognition as a User Interface

Brandon Thomas Taylor

Thesis Reader **John Maeda**
Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Graspables:
Grasp Recognition as a User Interface

Brandon Thomas Taylor

Thesis Reader **Joseph Paradiso**
Associate Professor of Media Arts and Sciences
Steinhardt Program in Media Arts and Sciences

Acknowledgements

Thanks to my advisor, Mike Bove, for ideas, support and allowing me to hang around the lab for a couple of years.

Thanks to John Maeda and Joe Paradiso, my thesis readers, for their advice and help in keeping things in perspective.

Thanks to: Jeevan Kalanthi for his mystical electronic insights. Quinn Smithwick for consistantly bailing me out when time was running low. Dan Smalley and Matt Adcock for their pattern recognition expertise. Jim Barabas and Ana Luisa Santos for their extensive testing. Takashi Okomoto for his publishing skills. Jim Trankelson et al. for inspirational Friday night brainstorming.

Additional thanks to Kristin Hall, Tom Lutz, Linda Peterson, and everyone one else at the lab.

Lastly, thanks to all involved with the CELab, Digital Life, and Things That Think consortia at the Media Lab for supporting this research.

Contents

List of Figures

1

Introduction

Perhaps the best place to start in explaining the Graspables project is to define what exactly is meant by grasp-recognition. For the Graspables, the goal was to explore how basic manipulations of an object can contain useful information. In other words, what can you learn from the way people hold an object? Can you distinguish whether a user wants to make a phone call or just look up a contact by the way they hold their phone? Can a golf club predict a slice if it is gripped improperly?

In pursuing these questions, the Graspables project was constrained by the desire to have a system that could be realistically implemented in existing objects and devices. This led us to shy away from approaches that require elaborate sensing environments or expensive input devices. The hope was that the right combinations of sensors and software could give objects a enhanced understanding of their users' actions without limiting portability or affordability.

Another key aspect of the research was the focus placed on objects themselves. Instead of focusing on just creating a new interface method or a specfic type of controller, we were very interested in understanding and exploring how people interact with a variety of different objects. Our view was that how people interact with a coffee cup potentially has as much to say about grasp-recognition and interaction as how they hold modern electronics. Thus, we wanted a system that could be implemented into arbitrary geometeries.

1.1 Motivation

The origin of the Graspables project can be traced back to a high level discussion of ways to improve multi-function handheld devices. It was then that someone tossed out the idea that an ideal multi-function device would need to be capable of two things: it would need to automatically know what users want to do with it and it would need to be able to alter its affordances accordingly. When it wasn't being used, the device would simply appear to be an undifferentiated block, like a bar of soap.

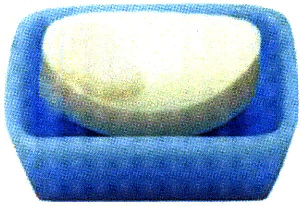

Fig. 1-1 The Ideal Handheld Device?

While the Graspables may not completely fulfill this vision, the idea of creating devices that implicitly understand users' intentions– without the need for menus and direct commands– was the launching point for the project. As the project evolved, emphasis shifted away from multi-fuction handhelds to how manipulating objects in general can be used as an interface. Throughout this process, motivations came from a variety of fields, ranging from handheld accessories to virtual reality controllers to sensate skins and more.

1.2 Thesis Structure

This thesis will seek to explain the Graspables project and provide context for the work it entails. Chapter 2 explores projects and fields of study that relate to or provide inspiration for the Graspables project. Chapter 3 will discuss the design concepts used in the Graspables. This chapter will also entail detailed discussions of the Bar of Soap and Ball of Soap as implementations of the Graspable system. Chapter 4 will discuss software design aspects of the Graspables System. It will also give a detailed explanation of how the classification techniques were explored using an early user study. Chapter 5 will discuss applications of the Graspable system and explore specific implementations. Chapter 6 will explain conclusion drawn form the work on the Graspables as well as future work to be considered.

1.2.1 Background

The problems that the Graspables seek to explore intersect with a variety of projects and research fields. From kinesiology and ergonomics to sensate skins and virtual reality, there is, not surprisingly, a lot of interest in different aspects of how people interact with objects.

Chapter 2 will review areas of research that are relevant to the development of the Graspables project. This chapter will explore how the goals behind the Grapsables are relevant to such fields and how implementations of grasp-recognition can provide them with a novel tool. Finally, the

Graspables implementations will be explicitly compared to other research devices.

1.2.2 Hardware

Obviously, how users interact with an object is largely dependent upon what the object is and what they are trying to accomplish with it. One aspect of the Graspables was to explore how the physical geometry of a device impacted what it could be used to respresent. Chapter 3 discusses the two prototype implementations of the Graspables System, the Bar of Soap and the Ball of Soap. While the general design ideas and their evolution through prototype iterations will be provided here, more detailed descriptions of the hardware can be found in Appendix A.

1.2.3 Software

Similar to how the physical nature of the devices impacts how users will interact with them, interpretation of grasps and manipulation is dependent upon on how the device is being used. Chapter 4 will discuss the software methods that were used throughout the Graspables project.

1.2.4 Applications

The applications represent how the Graspables project is presented to users. This chapter provides a detailed description of the specific interfaces and objectives that have been created to study how grasp-recognition can be used as an interface.

1.2.5 Conclusions

The final chapter of this thesis will give a brief assesment of the Graspables project and the use of grasp-recognition in general. It will also discuss future implementations and studies that would be useful.

1.3 Thesis Terminology

Throughout this thesis a variety of terms will be used to describe different aspects of the research. The term “Graspables” will be used when discussing the ideas and goals of the project as a whole. The “Graspables System”, on the other hand refers to the combination of sensors and software that are actually used to create the grasp-recognition interface. The “Bar of Soap” and the “Ball of Soap” refer to specific hardware implemenations of the Graspables System.

2

Background: Objects and Interfaces

In a film class it was once suggested that good editing was one of the least appreciated aspects of film. The reasoning being that if an editor has done his job correctly, the audience will never notice his work.

In many ways this applies to the design of user interfaces as well. When well designed, interactions becomes so second nature that people think of the interface and function as a single entity. At the other end of the spectrum are the VCRs blinking "12:00" and other devices whose operations seem to be more work than they are worth.

This chapter will seek to explain some of the motivations behind developing the Graspables as a new user interface. It will then discuss some of the fields of study that are relevant to the development of grasp recognition. Lastly, this chapter will compare the Graspables to other specifically relevant projects.

2.1 Ubiquitous Computing

Nearly twenty years ago, Mark Weiser coined the term Ubiquitous Computing to describe the idea of a vast network of computing devices interacting unobtrusively to enhance productivity. While the profliteration and dispersion of computational power has certainly occurred, it has not yet "anish[ed] into the background" [34, p.94].

Projects like the Graspables fit into the realm of Ubiquitous Computing by trying to expand the ways in which computers are controlled. By developing grasp-recogntion as a user interface, it is hoped that users can be presented with a more natural method of interacting with devices. Instead of seeing a device and trying to imagine how its menu system and buttons are mapped, grasp-recognition can leverage users' intuitions about how devices should be used for certain functions.

2.1.1 New Sensing Modalities

Aside from the abundance of touchscreens, there really has not been a dramatic change in computer interfaces since the mouse and Graphical User Interface (GUI) took over. However, recent trends indicate that new interaction methods are starting to gain a foothold.

Voice recognition is becoming more and more common, with the market topping one billion dollars in 2006 and growing steadily since[8]. The Nintendo Wii has become the top selling current generation video game system[30], no doubt largely because of its inovative accelerometer and optical sensing based controller. Niche markets from motion capture suits to brain wave readers (see Figure 2-1) further indicate new opportunites to break away from old interface methods.

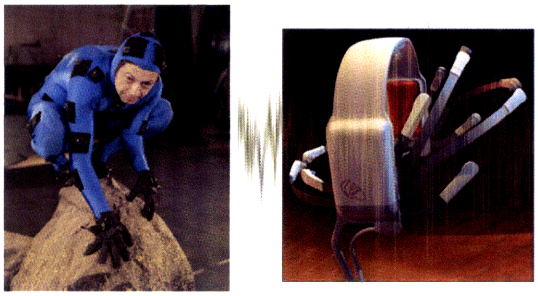

Fig. 2-1 (A) Motion Capture Suit Used in Lord of the Rings[3] (B)Emotiv EEG Headset Controller[2]

2.1.2 Affective Computing

Affective Computing focuses on how computers can sense and respond to emotional cues. Studies, ranging from to detecting facial expressions[33] to controlling user posture[4], have shown that behaviors normally overlooked by computer systems can provide valuable information about the user. By recognizing these cues, Affective Computing hopes to make interfaces less frustrating and more focused on users' emotional needs. While the Graspables project is not designed to interpret emotions, it does share the goal of using multimodal sources of information to make computational systems that respond in a more intuitive manner.

2.1.3 Physical Computing

The area of Physical Computing is interested in incorporating physcial objects into computational systems to enhance user interactions. Studies have been performed demonstrating how certain computerized tasks can be more easily accomplished when properly modeled by physical devices[12]. Early work on the concept of Graspable User Interfaces suggested that by "facilitating 2-handed interactions, spatial caching, and parallel position and orientation control" physical computing could provide a richer interface than virtual, graphics-based interfaces[13].

A grasp-recognition based interface would, by virtue of its nature, capitalize on these advantages. Rather than creating controls through arbitrary key-mappings, the physical nature of an implementation of the Graspables System would provide suggestive physical affordances and passive haptic feedback by approximating and representing real-world objects.

2.1.4 Interfaces for Portable Devices

Portable device interfaces provide a distinct challenge for designers. For more complex portable systems, there is a natural desire to mimic the operational semantics of the computer as much as possible. People are accustomed to the window metaphor of most desktop computer GUIs, so it makes sense to leverage this knowledge to some extent. While some portable devices, such as the Sony Vaio VGN-UX490N, shown in Figure 2-2, really try to recreate the interface of a desktop computer, in general the sizes of portable devices make this impractical.

A common approach in portable devices is to imitate the clicking and dragging functions of a mouse with a touchscreen. Full keyboards are often implemented either in the form of physical buttons or virtual ones. Both approaches have drawbacks. Physical buttons are necessarily small and always present, even when an application only needs a subset of the keys. Virtual buttons on the other hand, provide no tactile feedback which can render them unusable to certain groups of users.

These issues have led researchers to explore other interaction methods that may end up being more appropriate for handheld devices. A common example is the use of accelerometers in many cameras and phones for switching between portrait and landscape views. Another approach is to capitalize on the inherent mobility of handheld devices by exploring gestures as an interaction method. Studies have explored using gestures

Fig. 2-2 The Sony Vaio VGN-UX490N, a 4" x 6" Portable Computer that Attempts to Preserve the Desktop's Interface

for things such as the detection of common usage modes[21] to the mapping of functions to relative body positions[5]. While it is hard to predict what new interfaces will catch on, successes like the iPhone's Multi-Touch display[1] provide encouragement for continuing research.

2.2 Object Representation and Simulation

In implementing the Graspables system into the Bar of Soap and Ball of Soap, we were interested in how the devices' geometries impact what objects they can easily represent.

The most common input devices, mice, keyboards and even video game controllers generally sacrifice representation in favor of more robust, general controls. Over time, these systems develop a semantic of their own (think how similar all video game controllers are or how people expect to be able to click icons in graphical interfaces) and people hardly even think about the control metaphors. However, there are exceptions.

Tablet and stylus systems exist to better bridge the gap between writing or drawing and computers. Video games often have speciallized periperials such as steering wheels or guns. These examples highlight the importance of objects for some tasks. While there is likely no way to completely avoid the tradeoff between robust and representative controls, it is certainly worth exploring how new interfaces can create more literal interactions and of what value these may be.

2.3 Adaptive User Interfaces

Adaptive User Interfaces rely on artificial intelligence techniques to train systems to better respond to specific users and uses. As would be expected, adaptive user interfaces have typically been focused on human interaction that have subtle variations across populations. Speech recogntion [6] and hand writing recognition [7] have long been the focus of such research. However, more recently other applications such as synthetic music controllers[22] and gesture recognition for mobile phones[15] have also made use of these algorithms.

Given the wide variations in hand sizes across populations and the open ended nature of possible interaction methods, Adaptive UI's are relevant to the development of grasp-recognition. Even for situations where prescribed grasps would be desirable, Adaptive UI's could provide useful techniques for testing and understanding potential pitfalls in the interface.

2.4 Sensing Grasps

This section will look at areas of research that are relevant to the development of grasp-recogntion hardware.

2.4.1 Whole-Hand Interfaces

The area of whole-hand interfaces has long been viewed as a potential method for creating more natural human-computer interactions. While such interfaces can be applied in place of the standard keyboard or mouse for general computing[23], they are more commonly considered as a way of interacting in virtual environments. Unfortunately, these systems typically require complicated setups usually involving either a data glove[9] or some form of machine vision system[25].

While there is obviously some overlap between this research and the goals of the Graspables project, there are a couple of key distinctions. The first and most obvious is the difference in levels of complexity and robustness. While a whole-hand interface could potentially perform grasp recognition tasks, the systems are significantly more expensive and less portable than the Graspables System. Secondly, the Graspables project places a higher interest in exploring how various object affordances impact user manipulations, whereas whole-hand interfaces typically focus on hand gestures independently of objects.

2.4.2 Sensate Skins

In robotics research there is a strong interest in being able to mimic the dense and varied sensing capabilities of human skin[20]. While building the sensor arrays poses significant challenges by itself, there is also the related issue of making sense of the sensor data.

Early on in the Graspables project, implementing the system in a modular form that could be connected in arbitrary geometries was considered. The Tribble, a device developed by the Responsive Environments group at the MIT Media Lab[19], provided inspiration for how such distributed sensing nodes could be combined and used as a single sensing unit. While this approach was eventually abandoned in favor of implementing the Graspables as a limited set of complete devices, the concepts behind the sensate skins provided an excellent starting place for grasp-recognition.

2.5 Interpreting Grasps

This section will discuss a few specific projects that use touch sensing as a means to infer information about user intentions.

2.5.1 The Huggable

The Huggable, shown in Figure 2-3, is a robotic Teddy Bear being designed by the Personal Robots group at the MIT Media Lab to provide therapeutic interactions similar to those of companion animals. In order to accomplish this, its necessary that the Huggable be "capable of properly detecting the affective content of touch"[32]. The Huggable is thus equipped with an array of sensors that detect the proximity of a human hand, measure the force of contact and track changes in temperature. The data from these sensors is then processed to distinguish interactions such as tickling, poking or slapping[31].

From a technical perspective, the goals of the Huggable are very similar to those of the Graspable System. Both seek to identify and understand

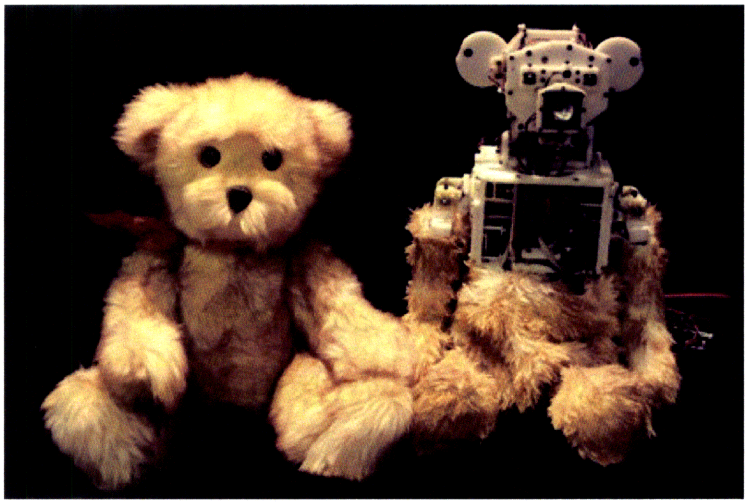

Fig. 2-3 The Huggable, Shown with and without its Skin[14]

the ways users manipulate an object. In many ways, the Huggable could be viewed as a sophisticated example of a grasp-recognition system. That said, there are obvious differences between the Graspables System described in this thesis and the hardware/software system of the Huggable. The sensing hardware of the Huggable, for example, relies on dense arrays of Quantum Tunneling Composite (QTC) force sensors and broad electric field sensors for touch sensing, whereas the Graspables are implemented with a dense set of capacitive sensors. Similarly, the Huggable has been demonstrated using more computationally complex classification techniques than would be ideal for the Graspables.

2.5.2 The Tango

The Tango is a whole-hand interface designed by the Multisensory Computation Laboratory at Rutgers for the manipulation of virtual 3D objects[24]. The device, shown in Figure 2-4, is a hand-sized spherical object with a 3-axis accelerometer and an 8x32 capacitive sensing grid housed in a compressible dielectric material. The Tango is calibrated to detect variations in pressure from which a simplified hand model can be estimated. The

Tango uses spherical harmonics to create a rotationally invariant map of pressures[18]. These pressure maps can then be reduced using principal component analysis and classified using K-nearest neighbors. A 3D virtual environment in which the Tango was used to manipulate virtual objects was also developed.

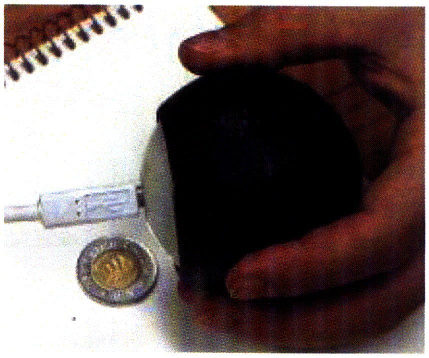

Fig. 2-4 The Tango Device[24]

While the Tango clearly shares certain objectives with the Graspables, there are significant differences in their respective implementations. First, the grid structure of the capacitive sensors and the classification software of the Tango would not directly translate to other device geometries, severely limiting the number of objects it could represent. Also, since the Tango is actually attempting to infer general hand poses it requires additional constraints, such as single hand use. In the end, while the sensing techniques and software analyisis provide interesting references, the goals of the Tango require a significantly different approach than those of the Graspables.

2.5.3 SAIT Grip Pattern Recognition

When development began on the first version of the Bar of Soap, a similar study was being conducted by the Samsung Advanced Institute of Technology (SAIT)[10, 17]. After receiving encouraging results from an initial study in which painted gloves where used to create image maps of grasp patterns, a prototype device was built for real-time grasp detection. The SAIT device, shown in Figure 2-5, contained a 3-axis accelerometer and 64 capactive sensors. A user study was performed to try and classify 8 different use modes with the device.

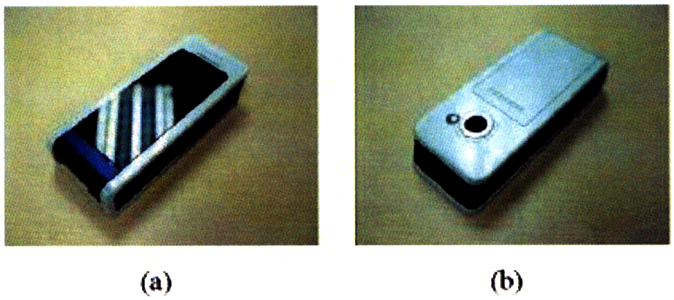

Fig. 2-5 The SAIT Prototype Device[10]

The results from the SAIT study match up well with those of the intial Bar of Soap study (see Section 5.1) which is encouraging for the larger concept of grasp-recognition. The SAIT device uses non-binary capacitive sensors, a different sensor layout on a device of different physical dimensions, a unique set of use modes and different classification techniques from the Bar of Soap. Even so, the SAIT device was able to correctly classify 75%-90% of modes across users, which is very similar level of accuracy that the Bar of Soap acheived. It is worth noting that the subjects in the SAIT were instructed to hold the device in a certain way, unlike the Bar of Soap study, where the users were only given a use mode with no suggested grasp. Even so, the study provides independent validation of the Graspables concept, if not of the Bar of Soap's particular implementation.

3

Graspables Hardware Implementations

The Graspables System is a hardware and software platform capable of detecting how a user is manipulating a device. This chapter will describe the hardware used in the Graspables System. This chapter will also discuss the design of two specific implementations of the Graspables System, the Bar of Soap and the Ball of Soap.

3.1 Hardware Goals

The goal of the Graspables System is to develop a set of sensors capable of measuring relative hand position and orientation. The system needs to be flexible enough to accomodate distinct sensor layouts for objects with different physical geometries. It is also important that the system be able to process and transmit data in real-time.

3.2 General Hardware

The hardware within the Graspables System can be divided into two functional groups. The first group consists of components that provide infrastructure for the Graspables. This group regulates power, processes data and communicates with other devices or computers. The second group consists of the sensing components. Whereas the first group is consistant across different implementations, the sensors vary depending on the needs of the realized system.

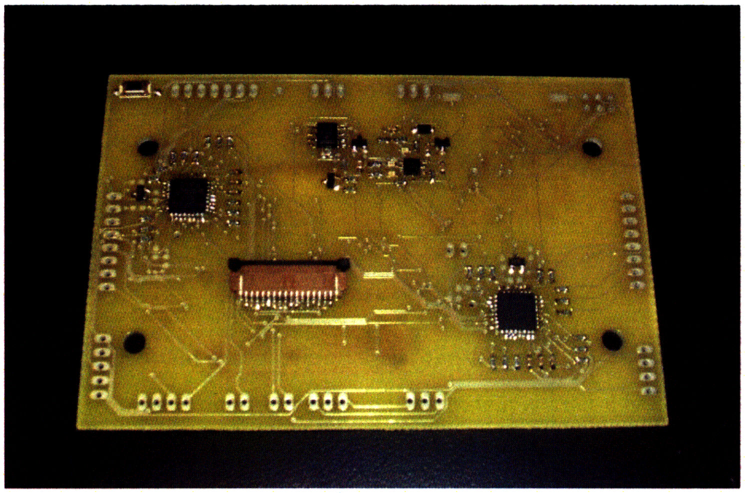

Fig. 3-1 A Partially Populated PCB for the Bar of Soap

3.2.1 Controlling and Communicating

The Graspables are envisioned as devices that can operate independently or as periperials for a computer. The key infrastructure features that are needed for any instantiation of the Graspables system are: (1) a controller to collect and process data, (2) a method of sending and receiving data from other computers and devices and (3) a battery or power supply. For all the prototypes discussed in this thesis the Atmel Atmega644 ICs were used as the controller, BlueGiga WT12 bluetooth chips provided communication and 1000mAh Lithium Polymer batteries provided power (see Appendix A for more details). These specific components were chosen solely for reasons of convenience and could easily be replaced by comparable components.

3.2.2 Sensors

Unlike the control and communication components, the sensing hardware was customized for each implementation. However, some general principles still apply.

Early on we made the desision to use capacitive sensoring as the method for measuring how users hold the Graspables[27]. Specifically, we chose QT60248 Qmatrix touch sensor ICs for the implementations[26]. These chips allow for independent design and placement of up to 24 capacitive sensors. Each sensor can then be individually calibrated to ensure consistant response across the device. Lastly, a 3-axis accelerometer was incorporated in order to determine the device's orientation and relative motion.

3.3 Prototype Implementation

In order to test the Graspables System, it was necessary to have a an object to implement the system in. The Bar of Soap, a rectangular box designed to approximate the size of modern handheld multi-function devices, was the first prototype for the Graspables System. Three versions of the Bar of Soap were built, each improving upon the sensing system and the prototype's functionality as well. The Ball of Soap was then built to expore grasp recognition in other scenarios

3.4 The Bar of Soap

The Bar of Soap was designed to explore how grasp-recognition could be of use in a variety of modern handheld devices. It also served as a test bed for the Graspables System's hardware. The following sections will describe each of the three versions.

3.4.1 Version 1

The first Bar of Soap prototype, shown in Figure 3-2, is a 4.5x3x1.3 inch rectangular box containing a 3-axis accelerometer and 48 capacitive sensors encased in an 1/8th inch thick shell of transparent, vacuum-formed PETg plastic. A microcontroller in the device samples these sensors and reports the results to a PC via Bluetooth. This device did not contain a battery and thus had to have a power cord connected to its sensor-less face.

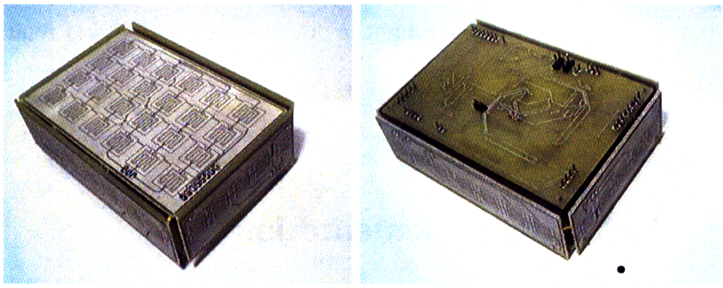

Fig. 3-2 The Front and Back of the Bar of Soap V1 without the PETg Shell

The capacitive sensors were created by printing interdigitated copper traces (See Figure 3-3) on a printed circuit board (PCB). The sensors are approximately a quarter of square inch in area, roughly the size of a fingertip. These sensors were controlled by Qprox QT60248 chips, each of which can support 24 buttons. The sensors were evenly spaced on five of the six faces of the prototype with 24 sensors on the back, 7 on each longer side and 5 on each shorter side. For this version, no capacitive sensors were placed on the front of the device, with the intention of having it represent a display surface. Parameters in the Qprox chip were adjusted so that each sensor would respond when a finger was placed directly above it, but not above a neighboring sensors.

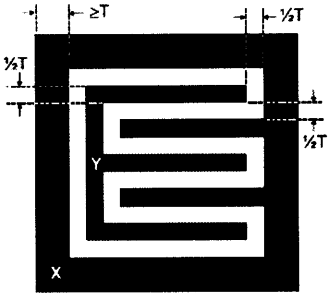

Fig. 3-3 The PCB Layout Schematic for the Bar of Soap's Capacitive Sensors

3.4.2 Version 2

The initial feasibility study, discussed in Section 5.1, provided encouraging results from our first prototype. However, there were various limitations that needed to be addressed. Thus, we built a second prototype, shown in 3-4.

The most obvious change was the addition of a cholesteric LCD screen. These screens employ a bi-stable display which only requires power when the image is changed, greatly increasing battery life. We felt that the lack of a display surface in the first version severely limited what we could explore with the Bar of Soap. By adding a display, we were able to indicate to the users the layout of various affordances corresponding to different modes that Bar of Soap could represent. Thus, if we wanted to explore how users grasp a camera, we could display a facsimile of a camera display instead of just a sensorless face (see Figure 3-5). This also allowed us to examine how different layouts of virtual buttons on the display would impact how users interact with the device.

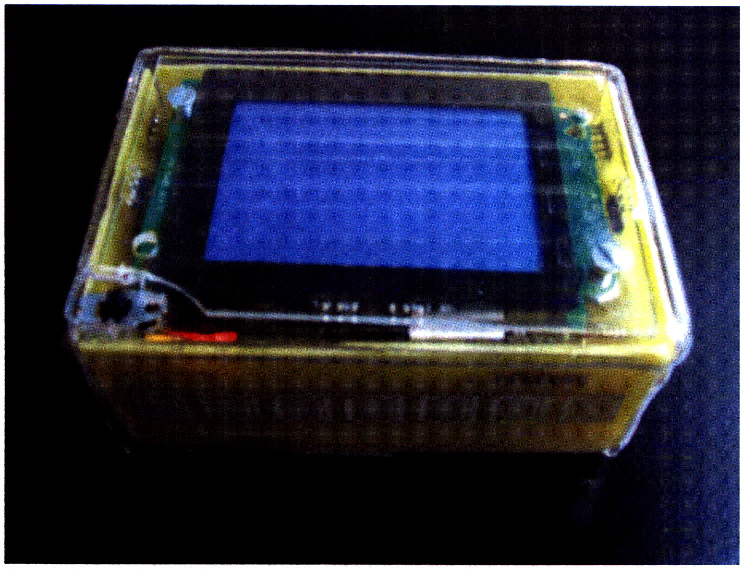

Fig. 3-4 The Second Version of the Bar of Soap

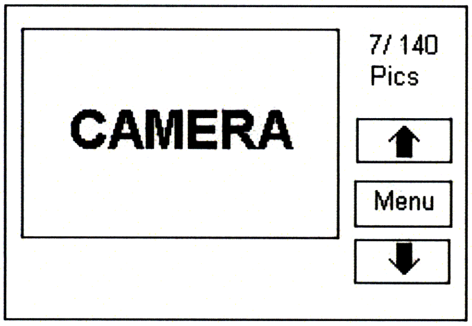

Fig. 3-5 A Sample Screen Image from the Bar of Soap

In addition to the screen, a battery was added to the second version so that the device could operate without a power cable tethering it.

3.4.3 Version 3

Whereas the second prototype had been a modification of the original PCB, the third version of the Bar of Soap was created from a new circuit board design. Seen in Figure 3-6, the Bar of Soap V3 has screens on both the front and back surfaces. The battery charging and power circuit were redesigned to include a power button and an additional 24 capacitive sensors were added so that all the surfaces had capacitive touch sensors. Lastly, a transparent acrylic case was used in placed of the vacuum-formed PETg shell of the previous versions.

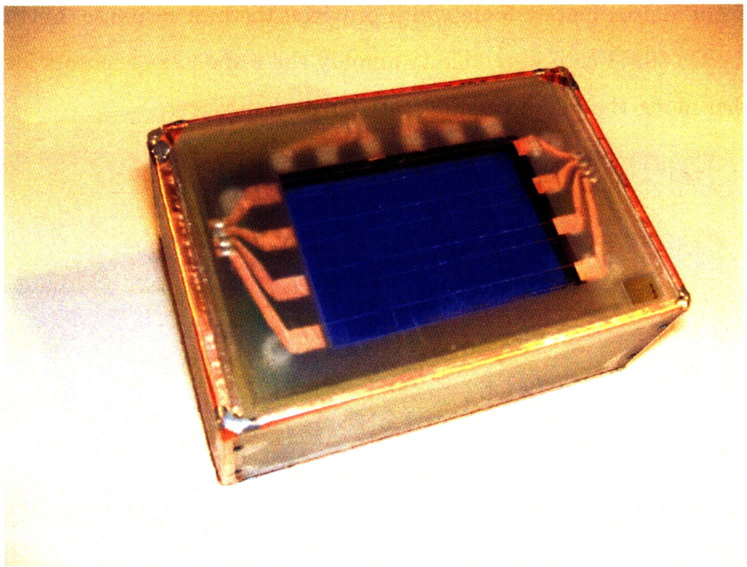

Fig. 3-6 The Final Bar of Soap V3 Prototype

The primary reasons for the development of the third version were to provide more flexibility in what could be done with the display and to make the device more uniform and symmetric in its sensing abilities. Transparent capacitive sensors were developed and placed over both screens. This allowed the display surfaces to also function as sensing surfaces, which in turn allowed the Bar of Soap to better emulate functional devices with interactive touchscreens. The additional screen on the back side and accompanying touch sensors also gave us the ability to treat the Bar of Soap not just as a multi-function handheld, but as a generic rectangular sensing device with two customizable faces.

The transparent sensors were created by placing thin film coated with Indium Tin Oxide (ITO) on opposite sides of a clear piece of acrylic, as shown in Figure 3-7. ITO is a transparent conductive material that fills the role of the interdigitated copper traces (see Figure 3-3) of the other capacitive sensors. By connecting the outer layer of ITO to the pulsed 'X' line of the QT60248 chip, a user's finger will cause enough coupling with the ITO to create a measurable effect on the other, inner layer of ITO. By testing a variety of ITO shapes and sizes, it was determined that strips of approximately 5mm width provided the best response. Settings in the QT60248 chip were able to amplify the responses of sensors located further along the ITO to compensate for its resistance.

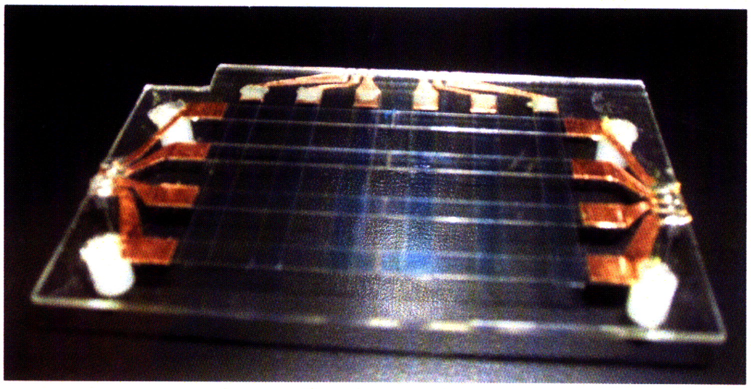

Fig. 3-7 Clear Capacitive Sensors Made with ITO Coated Film

3.5 The Ball of Soap

As the Bar of Soap evolved from an exploration of ways to improve multi-function handhelds into a more general platform to explore grasp-recognition, we began to consider the limitations of its physical form. While a small rectangular box provides an adequate representation of many handheld electronics, it has inherent limitations.

3.5.1 Version 1

In order to explore interactions with different physical forms, the Ball of Soap was developed. Since a truly spherical object would create difficulties in laying out the capacitive sensors used in the Bar of Soap, we built the Ball of Soap as a small rhombicosidodecahedron. This 62-sided Archimedean solid, shown in Figure 3-8, provides flat surfaces near the quarter square inch size and surface density of the Bar of Soap's sensors when the overall diameter approaches 3 inches.

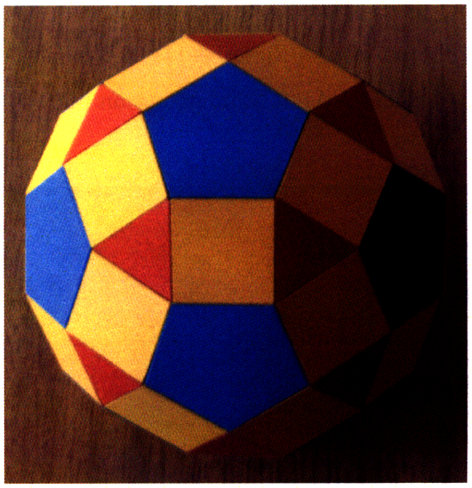

Fig. 3-8 A Small Rhombicosidodecahedron

The surface structure of the Ball of Soap prevented the simple printing of interdigitated copper traces used as sensors on the Bar of Soap. Instead, adhesive copper pads were cut and attached to the faces with wires running to circuit boards inside the ball. We explored using a variety of trace shapes for each of the different face geometries (see Figure 3-9), but found that only using the smallest, triangle arrangement provided a more consistent response across the capacitive sensors.

As can be seen in Figure 3-10, the small Rhombicosidodecahedron shape also allowed the Ball of Soap to be separated into three sections for easier assembly. The two end pieces are identical and each contain a Qprox chip

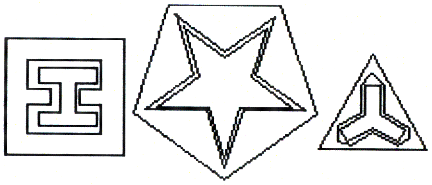

Fig. 3-9 Capacitive Sensor Layouts Tested on the Ball of Soap

that controls the 23 capacitive sensors on its surface. The center piece has 16 faces, 15 of which have capacitive sensors and one that houses the power button and programming interface. Inside the Ball, attached to the center piece is the main circuit board housing the microcontroller and accelerometer. As can be seen in Figure 3-11, this version of the Ball had a cable that ran power and UART communication lines to the main circuit board.

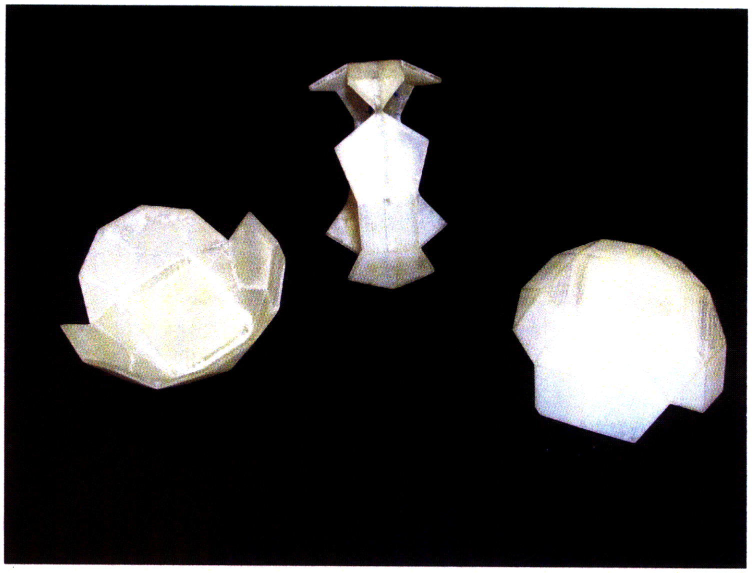

Fig. 3-10 The Three Sections of the Ball of Soap

Finally, the shell of a baseball was wrapped around the outside of the Ball of Soap. While this particular covering was chosen specifically for the application described in Section 5.2, it could take on any number of appearances as long as the capacitive sensor traces are not left exposed to contact.

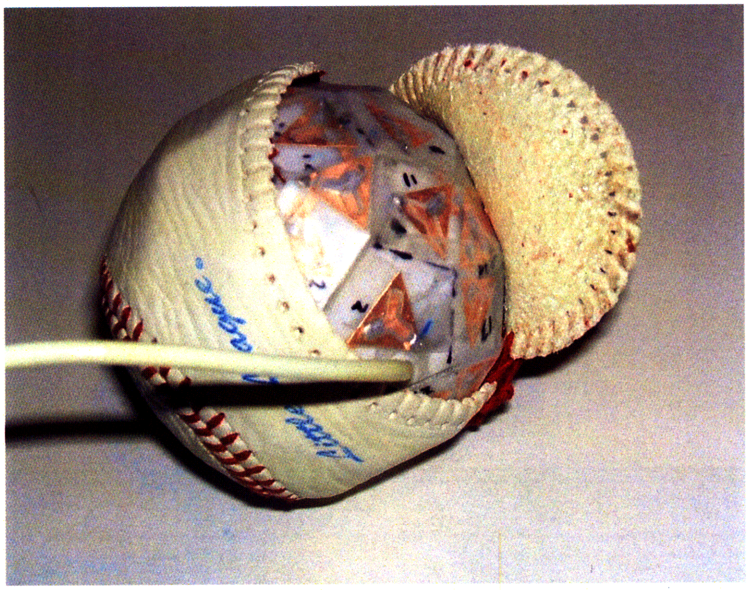

Fig. 3-11 The First Version of the Ball of Soap with its Baseball Shell

3.5.2 Version 2

As with the first Bar of Soap, the Ball of Soap V1 was built as much to test the sensing hardware as it was to study manual interactions. While the power cord reduced the complexity and made programming and testing the device easier, it certainly interfered with how the device could be held.

The second version used the same circuit board as the first, only this time the bluetooth chip and battery were attached. Figure 3-12 shows the inside of the Ball of Soap in its final assembled state.

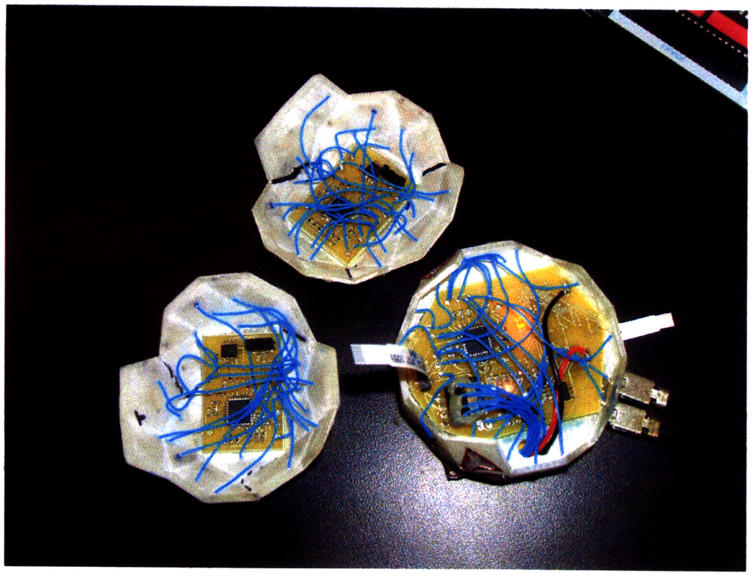

Fig. 3-12 The Insides of the Second Version of the Ball of Soap

4

Graspables Software Design

Any implementation of the Graspables hardware is only as useful as the software that makes sense of it. This section will describe the methodology used in selecting appropriate pattern recognition algorithms.

4.1 General Approach

This chapter will focus on general principles of the Graspables pattern recognition software. These principles will be illustrated with specific examples and data from the initial feasibility study performed with the first version of the Bar of Soap (described in Section 3.4.1). This should provide a general overview of how developing grasp-recognition as a user interface was approached. Specific variations implemented for different applications will be discussed in Chapter 5.

4.2 Feasibiliy Study

With the first Bar of Soap, we were essentially testing two hypotheses at once. First, we were hypothesizing that users naturally grasp and interact with multi-function handhelds in measurably distinct ways depending on the functions they are using at a given time. Secondly, we were hypothesing that the first Bar of Soap was capable of measuring the differences between these interactions.

4.2.1 Study Objectives

When we first began exploring the idea of a grasp-based sensing system, much of our work focused on finding adequate sensor resolution and pattern recognition techniques. The first user study, performed with the screen-less first version of the Bar of Soap allowed us to explore these issues and provided valuable data for improving the device. In this study, we had users treat the Bar of Soap V1 as a mutlti-function handheld device and measured how they interacted with it for five different functionality modes.

4.2.2 Data Collection

Users were seated with the Bar of Soap in front of them on a table. They were told that they would be given a specific functionality mode (e.g. "the device is a phone") and that they should then pick up the device and interact with it however they saw fit until instructed to set it back down. Users were also informed that they should treat the front, sensorless face of the of the device as the location of any display that they would expect from such a device. It is important to note that no suggestion of appropriate grasps was given to the user either before or during the tests.

After giving these instructions to the user, we would begin recording the data stream from the Bar of Soap and then verbally indicate what functionality mode the device should be treated as. Once the user had established a relatively stable pose with the device, we would label and save the data sample and have the user place the device back on the table. We would then repeat this process with each user until we had data samples from each of the five tested functionality modes: camera, gamepad, PDA, phone, remote control. Typical pose examples for each of the five poses are shown in Figure 4-1.

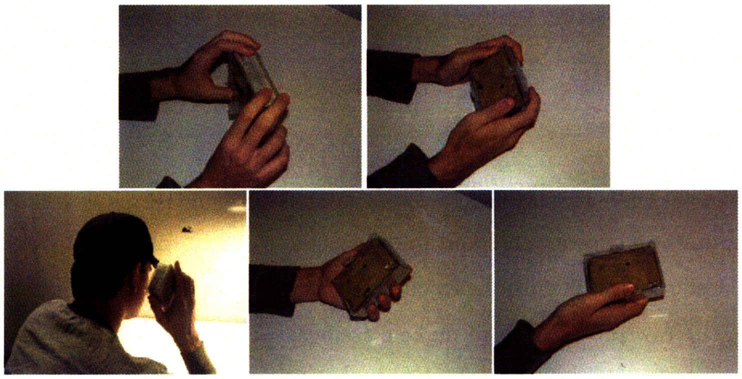

Fig. 4-1 Example Grasps for Each of the Five Functionality Modes (clockwise from top left): Camera, Gamepad, Remote Control, PDA, Phone

Since, in this study, we were interested only in the way the user grasps the device and not in the dynamics of how they pick it up, each data sample was trimmed to a single measurement of the 48 capacitive sensors and 3 accelerometer axes. This was done by simply averaging over the final four measurements in each sample to smooth over any minor changes in the way the user held the device.

Using this data collection process we generated two distinct data sets, each containing 39 grasp samples for each of the five functionality modes. The first set was obtained from a single user to see how consistently one person held the device. For the second set, 13 different users held the device to see how grasps varied across a population.

4.2.3 Training and Testing Sets

With the two data sets, we were able to explore both how well we could distinguish a single user's grasps and also how well grasps generalize across a population. To study a single user, we trained the classifiers using 29 of the 39 samples for each mode. The final 10 samples were used to test the classifiers. To test how well these grasps held across a population, we used the 39 samples from multiple users as the test data.

4.3 Feature Selection

With each capacitive sensor and accelerometer axis acting independently, computational constraints and limited amounts of training data can lead to classification difficulties. To circumvent this, features were created from the raw sensor data to make classification more feasible. The feature selection varied depending on the application.

4.3.1 Feature Selection and the Feasibility Study

In order to analyze the data, we first had to explore methods of reducing the 51 independent sensor readings to a more manageable feature space. We explored many different techniques, including Principal Component Analysis (PCA)[11], Fisher Linear Discriminant Analysis, data subsets (such as only using accelerometer data), and grouping capacitive sensors. Of the methods we tried, the one that provided the highest recognition rates across multiple users was by grouping sensors on each face and accounting for symmetries.

With this method, the 48 capacitive sensors formed six sensor groups (two 5-sensor sides, two 7-sensor sides, and two 12-sensor halves of the back face) with the value of each sensor group being a count of the active sensors in that group. Since the device should not have a preferred orientation (aside from the sensor-less face) we grouped the buttons according to accelerometer orientation rather than their physical location. Thus, if the device were rotated by 180 degrees, the capacitive sensors on the left side would always be treated as the left side group, even though the specific sensors would move. Finally, to give equal weight to both the accelerometer and sensor groups, all the data were normalized on a scale of 0 to 1. This feature reduction method will be referred to as "Rational Reduction".

4.3.2 Discriminability and Sensor Impact

To try and quantify what impact different sensors had in the Bar of Soap's ability to distinguish grasps, we created additional data sets that ommitted data from certain sensors. Using these data sets we calculated the discriminibility of the five modes. The discriminibility[11], d, between two classes, i and j, is defined as

$$d_{ij} = \frac{\hat{\mu}}{\left(\sqrt{\hat{\mu}^T \Sigma_i \hat{\mu}^T} \sqrt{\hat{\mu}^T \Sigma_j \hat{\mu}^T}\right)}$$

where

$\hat{\mu} = \mu_i - \mu_j$,

μ_i is the mean for class i,

and Σ_i is the covariance matrix for class i

and indicates how distinguishable different classes are. A lower number indicates that the classes will appear more similar to each other in the classification algorithm.

Table 4.1 shows the discriminibilities for the the data using only accelerometer data. Table 4.2 shows the discriminibiilities when only the capacitive

	Cam	Game	PDA	Phone	Remote
Cam	0	3.15	3.33	0.28	8.32
Game	-	0	0.70	3.99	1.20
PDA	-	-	0	4.18	3.59
Phone	-	-	-	0	10.2

Table 4.1 Class to Class Discriminabilities Using only Accelerometer Data

	Cam	Game	PDA	Phone	Remote
Cam	0	2.63	4.45	5.34	2.72
Game	-	0	5.85	6.38	5.86
PDA	-	-	0	1.56	5.39
Phone	-	-	-	0	5.77

Table 4.2 Class to Class Discriminabilities Using only Capacitive Sensor Data

sensor data is used. The particularly low discriminibilities between the Camera and Phone classes and between the Gamepad and PDA mode in Table 4.1 indicate modes for which the orientation of the device are very similar. The relatively low value for discriminability between the Phone and PDA classes in Table 4.2, on the other hand, shows that users tended to grip their phones and PDAs in similar fashions.

	Cam	Game	PDA	Phone	Remote
Cam	0	3.96	5.99	4.60	7.34
Game	-	0	6.82	8.49	5.74
PDA	-	-	0	5.33	5.85
Phone	-	-	-	0	9.48

Table 4.3 Class to Class Discriminabilities Using the Rationally Reduced Feature Set

Table 4.3 shows the discriminabilities using all the sensors and reducing them to the feature set described in Section 4.3.1. While it may not be surprising that the more inclusive data set provides better discriminabilities on average, these tables do give some indication of the relative roles that different sensors are playing. The relatively high discriminabilties across Table 4.3 also provides encouragement that the feature reduction is not detrimental to the classification process.

4.3.3 Feature Selection and Classification Rates

The data from the feasibility study was reduced to different feature sets and then run through a variety of classification techniques (discussed in Section 4.4.1) to compare how they impacted classification. The classification results from using Fisher Linear Discriminants and the Rational Reduction technique (see Section 4.3.1) to reduce the data are discussed in this section.

	Single User Testing	Multi-User Testing
Templates	97.5	61.5
Neural Nets	97.5	59.0
K-Nearest	92.5	63.6
Parzen	95.5	66.7
Bayes	92.5	64.1
GLD	90.0	58.0

Table 4.4 Classification Rates Using Fisher Linear Discriminant Reduced Feature Set

	Single User Testing	Multi-User Testing
Templates	82.2	75.4
Neural Nets	92.4	79.0
K-Nearest	95.0	75.8
Parzen	95.4	72.3
Bayes	95.0	79.0
GLD	87.5	70.3

Table 4.5 Classification Rates Using Rationally Reduced Feature Set

Table 4.4 shows the classification rates from a variety of classification techniques when using Fisher Linear Discriminant to reduce the data. Table 4.5 shows the classification rates for the Rationally Reduced data. As you can see, both feature reduction methods acheived high accuracy across the classifiers when trained and tested on a single user. However, when training the data on a single user and testing on a population, the results favor the Rationally Reduced data significantly. This seems to indicate that the Fischer Linear Discriminant is preserving more idiosyncracies of the individual that do not apply widely.

4.4 Pattern Recognition

In exploring a variety of classifiers, high recognition accuracy was not the only consideration. Since the classification would ultimately need to be determined by the device in real-time, a lower complexity calculation would be preferable. Additionally, since the five modes we selected are essentially arbitrary, it was important that the classification technique be easily adapted to other modes.

4.4.1 Classification Techniques

A wide variety of classification techniques were explored to evaluate the ability of the Bar of Soap to distinguish grasps. A neural network was trained and tested using 10 iterations of randomized, leave N out validation for a number of network nodes[11]. K-Nearest Neighbors was implemented for K values one to twenty, each with three different tie-breaking algorithms. Parzen Windows, another non-parametric classification technique in which labeled points contained within a hypersphere around the sample point are compared, was tested for a variety of hypersphere volumes and voting schemes. Multicategory Linear Discriminant functions and Naive Bayesian classifiers were also implemented. Lastly, Template Matching, which simply compares the distance from data samples to the mean values of the different classes, was tested.

4.4.2 Classification Rates and Results

From the methods we tested, the simple Bayesian classifier provided a good mix of recognition accuracy, ease of implementation and adaptability. The Bayesian classifier works by assuming that each mode can be represented as a Gaussian distribution in the feature space. The feature space, in this case, is a nine-dimensional space where each dimension represents an accelerometer axis or capacitive sensor group. Thus, any way in which the Bar of Soap is grasped will provide data that corresponds to a single point, x, in this feature space. This point can then be input into a discriminant function, $g_i(x)$, one function for each mode, as shown in equation 4.1. The discriminant function that returns the highest value is chosen as the most likely mode for the given grasp, x[11].

$$g_i(x) = -\frac{1}{2}x^T{\Sigma_i}^{-1}x + {\Sigma_i}^{-1}\mu_i x - \frac{1}{2}{\mu_i}^T{\Sigma_i}^{-1}\mu_i - \frac{1}{2}\ln(|\Sigma_i|) \qquad (4.1)$$

where

x is the vector of the reduced data from a sample grasp,
μ_i is the mean for class i,
and Σ_i is the covariance matrix for class i

This method provides the advantages that the mode can be determined with relatively simple arithmetic and that the only parameters that must be stored are the means and covariances of the different modes. Additionally, it would not require extensive work to update these stored parameters to allow the device to adapt to specific users' grasps.

4.5 Learning Algorithms

One of the fundamental problems with training a classifier is determining the optimal amount of training. If there is not enough training data, the classifier is likely to have difficulties as new data variations are introduced. On the other hand, too much training data can lead to overfitting the training data. In a system designed for multiple users, this can be especially problematic as the end user's behavior can varies significantly from the training data.

A common method for overcoming these deficiencies is to adopt a learning mechanism in the device. While doing so requires extra initial effort from the user to train the device to their behavior, it can add a lot of flexibiity to a device.

Among the classifiers we tested, some lend themselves more easily to learning than others. The non-parametric methods, such as K-Nearest Neighbor and Parzen Windows can easily adopt new data points by simply expanding the database of known points. The main drawback to these learning methods is that classification time is dependent upon the size of the known database, so each learned point increases the computation time.

4.6 Gesture Recognition

By itself, static grasp recognition might have a rather limited set of uses. However, by incorporating gesture recognition, hand movements and relative location can be determined. Instead of only analyzing static poses, dynamic actions can be recognized.

Increasingly, gesture recognition using acclerometers and other sensors is being incorporated into handheld device interfaces[29, 16]. Much like single decision classification techniques, there are a variety of ways to implement gesture recognition. Hidden Markov Models provide a very flexible means of gesture recognition. HMMs model a gesture as a series of states with statistical transition rates and observable characteristics[28]. After training the models, the probability of an observed sequence can be calculated on the fly.

5

Applications & Analysis

In the process of developing the Graspables System, applications were always a consideration. While it is hoped that the grasp-recognition technique is general enough to be applied to many other scenarios, specific objectives strongly influenced the design of the prototypes. This chapter will discuss the applications that have been developed for the Graspables implementations. For each application, details will be provided about how the classification techniques work and what kind of user experience they provide.

5.1 The Bar of Soap as a Multifunction Handheld Device

Following the model of the feasibility study (see Section 5.1), the third version of the Bar of Soap was used to implement a natural mode switching handheld device. The addition of two screens and battery power made for a more realistic handheld model. The additional 24 capacitive sensors on the front face also provided denser coverage for the grasp-recognition system.

5.1.1 Data Collection

As with the original feasibility study, users were seated with the Bar of Soap in front of them on a table. They were informed that the device should be assumed to take on a certain functionality which would be indicated by the screen. The screen images used in the study can be seen in Figure 5-1. The users were asked to pick up and interact with the device as they saw fit given the interface that appeared on the screen. No demonstrations were provided.

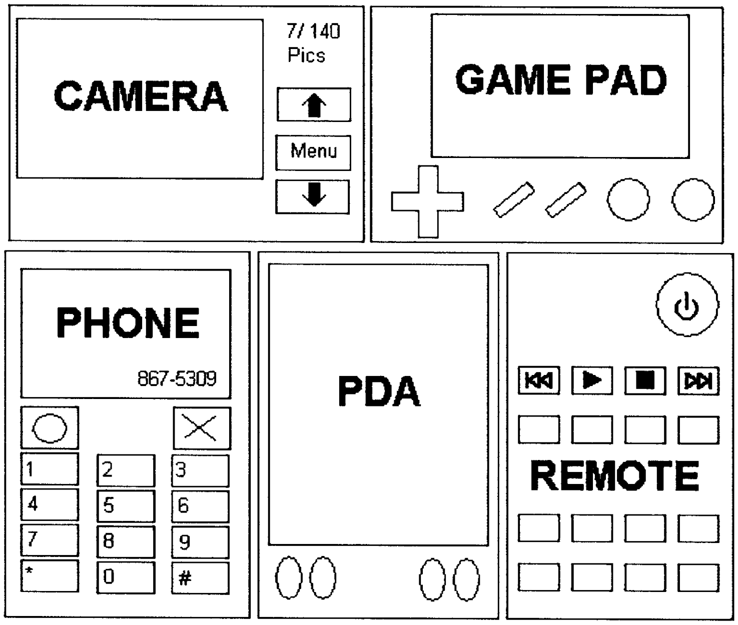

Fig. 5-1 Screens Displayed on the Bar of Soap Indicating Functionality Mode (clockwise from top left): Camera, Gamepad, Remote Control, PDA, Phone

A desktop computer collected the raw sensor data at a rate of 100Hz via a Bluetooth connection with the Bar of Soap. This connection also allowed commands to be sent to the Bar of Soap to select the appropriate screen and label the sensor data. Data was collected in groups of 10 modes at a time. This allowed each of the 5 modes to be randomly displayed along with a mirrored image of each mode. Additionally, at the beginning of each training session, the upward facing screen of the Bar of Soap was alternated. This precaution, along with the mirroring of the screens, was taken to reduce the chance of any orientational bias appearing in the data.

5.1.2 Feature Selection

The feature selection followed that of the original Bar of Soap feasibility study. Capacitive sensors were grouped together according to which face they were on, with the sensors on the screens being divided into two halves. These active sensor counts, along with the accelerometer readings, were all normalized between zero and one, giving a total set of 11 features.

5.1.3 Data Processing

A naive Bayesian classifier was used to distinguish the functionality modes. Before this could be done, features were ordered to maintain consistant relative positions. This insured that when the Bar of Soap was held in a certain position, the left edge was always treated as the left edge, even if the device was flipped about one of its axes. After training the Baysian classifier on a desktop computer, the classifier was simple enough to be run in real time on the Bar of Soap's microcontroller.

5.1.4 User Interface

The Bar of Soap as a handheld provides a very self-contained demonstration of the potentials of grasp recognition. The device passively senses its orientation and the position of a user's hands, then displays the an virtual interface corresponding to the most appropriate mode. For demonstration purposes the sampling and classification routine is performed every three seconds, but it could easily be triggered by some sort of gesture. Figure 5-2 shows the Bar of Soap as a handheld application in use.

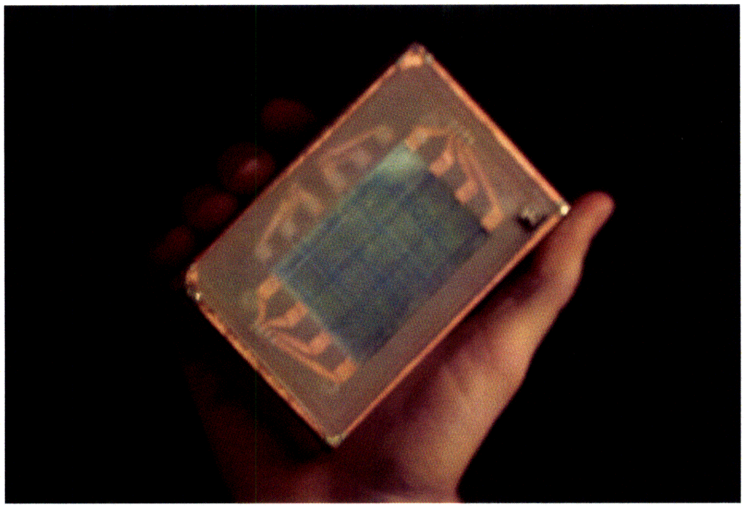

Fig. 5-2 The Bar of Soap as a Multi-Function Handheld

5.2 The Ball of Soap as a Baseball Pitch Detector

In baseball, subtle differences in the way the pitcher grips the ball have a profound effect on the outcome of the pitch. Thrown correctly, a slider can become nearly unhittable as it darts away from the batter at the last second. However, the slightest error in its delivery can see the pitch landing hundreds of feet in the wrong direction.

Given the importance of fine finger manipulations on pitching, this seems an ideal scenario for the Graspables system. A baseball that can detect how it's being held could be extremely useful in training players to throw certain pitches or diagnose potential delivery issues. On the other hand, baseball video games could use such a device to provide a method of pitch selection that is more realistic and engaging than pushing a button on a controller.

5.2.1 Data Collection

When using the Bar of Soap to represent a multifunction handheld device, there was an interest in how interactions vary across populations depending on the assumed mode of the device. This led to a situation where how users held the device came to define the grasp that was associated with a certain mode. For baseball, this is not the case. While individual grasps may vary slightly from pitcher to pitcher, in general the outcome (pitch type) is mapped directly to a certain, set hand position. Thus, instead of collecting data from a variety of users who were allowed to hold the device as they please, training data was aquired by having a single user appropriately hold the Ball of Soap for a set of pitch types. An example grasp is shown in Figure 5-3. Due to the four-way symmetry of a baseball, training data was collected for each pitch being held in the four different orientations.

5.2.2 Feature Selection

The Ball of Soap faces some classification issues unique from the Bar of Soap. While it has slightly fewer capacitive sensors, its rotation symmetry can create some additional difficulties. The Tango[24] (see section 2.5.2) dealt with this issue by using spherical harmonics to create a classification technique. However, for this specific application, the seams of a

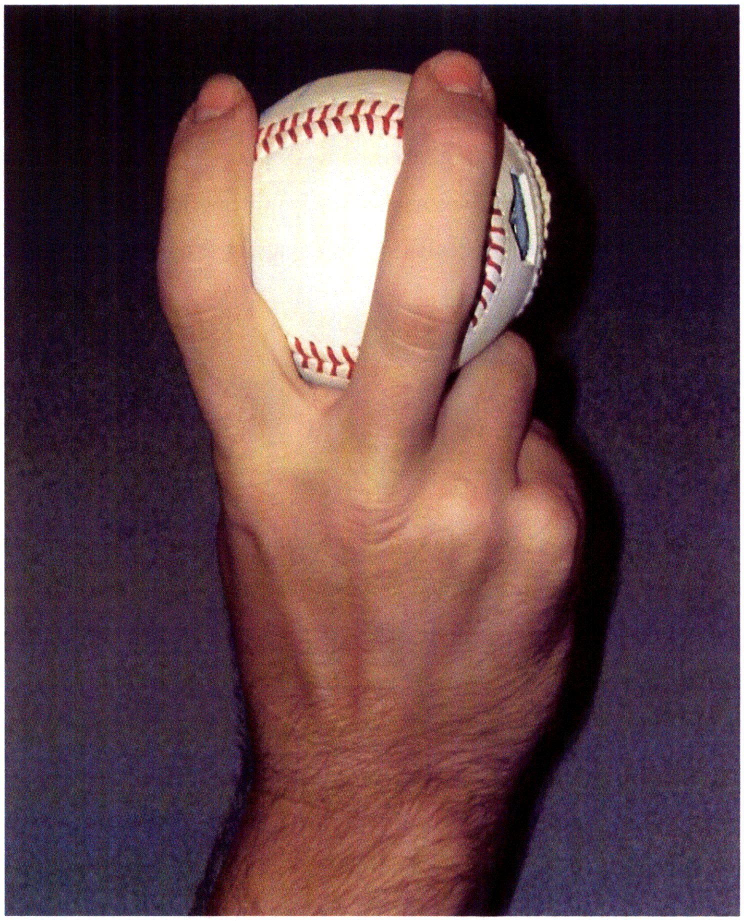

Fig. 5-3 The Ball of Soap Being Grasped as a Fastball

baseball reduce the symmetries to only four.

Like with the Bar of Soap as a multifunction handheld, the capacitive sensors were grouped in order to reduce the size of the feature space. Instead of grouping by sides, as in the Bar of Soap, capacitive buttons are grouped around the 12 pentagonal faces on the small rhombicosidodecahedron. Each pentagonal face is surrounded by 5 square faces, shared with a single other pentagonal face, and 5 triangle faces, shared with two other pentagonal faces (see Figure 3-8). These faces are weighted inversely to the number of groups they inhabit with an active pentagonal face receiving a weight of 6, a square 3 and a triangle 2. Thus each of the twelve groups creates a feature with a value between 0 and 31 depending on the number of activated faces.

5.2.3 Data Processing

The pitch recognition application for the Ball of Soap operates very similarly to the multi-function mode switching application for the Bar of Soap. The capacative sensors are grouped and processed as discussed above, then Bayesian discriminants are calculated. Each of the four pitch orientations are treated as a separate class. Thus, for N pitch types, 4xN determinants are calculated.

The classification routine is triggered by a throwing gesture. This is accomplished by continually sampling the sensors, but not calculating the discriminant functions until the accelerometer values surpass a threshold. When the threshold is crossed, the discriminants are calculated using the current capacitive sensor data and acclerometer data that was sampled a few cycles before the trigger.

5.2.4 User Interface

The Pitch Detection application operates as a Matlab script. Upon activation, Matlab opens a serial port for communication with the Ball of Soap and a screen presents the user with a pitcher ready to throw. The user then grips the Ball appropriately for the desired pitch and makes a throwing gesture. The acceleration of the Ball triggers the classification routine, which in turn triggers an animation taken from Nintendo's Mario Super Sluggers videogame to display the selected pitch (see Figure 5-4).

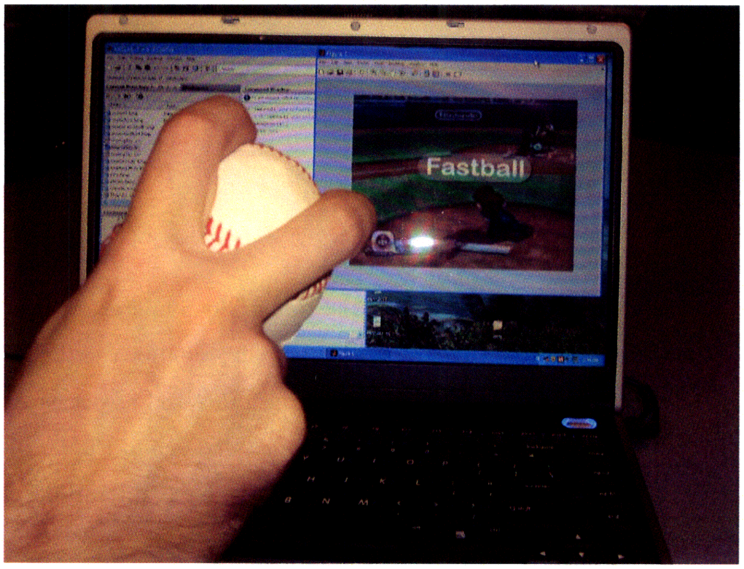

Fig. 5-4 The Ball of Soap as a Pitch Selector Application in Action

5.3 The Bar of Soap as a Generic Controller

The concepts contained within the Graspables project are not just limited to recognizing a small number of static grasps. Any object equipped with the sensors necessary for grasp-recognition can also be used for more standard computer interface objectives. A virtual Rubik's cube application was developed to demonstrate how the Bar of Soap can be used to represent and control a virutual object.

The idea is that sliding a finger across different faces of the Bar of Soap would trigger the rotation of the corresponding part of the Rubik's cube. While it might be possible to detect sliding gestures in a simpler manner, it was desired that the method be generalizable to more complex gestures. Thus, we chose to implement Hidden Markov Models[28] to detect the sliding gestures.

5.3.1 Data Collection

In order to train the Hidden Markov Models, data was collected and labeled as a single user slid his finger over the face of the Bar of Soap. The sliding gesture was recorded in both directions along each edge of the Bar of Soap and along the outermost rows and columns of sensors on the two faces. While the sliding gesture was being recorded on a specific side, no particular attention was placed on how the user was holding the Bar of Soap. This insured that data about manipulations that were not sliding gestures was also recorded.

5.3.2 Feature Selection

The sliding gestures were modeled using a left-right Hidden Markov Model (see Figure 5-5). The states represented the position of the finger activating either a single capacative sensor or two as it slides between them. The number of states in the model depended upon the number of capacative sensors on the side that was being modeled. In addition to the left-right HMM's modelling the sliding gestures, ergodic models exist to model general, non-sliding interactions.

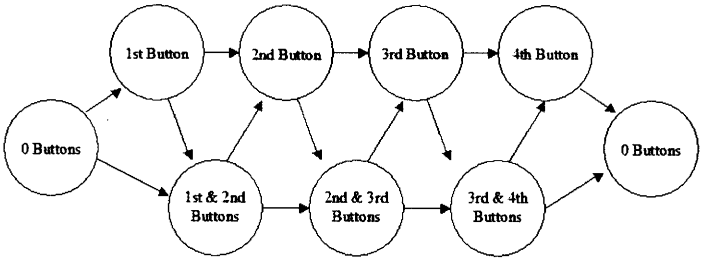

Fig. 5-5 A Left to Right HMM Model of a Four Button Sliding Gesture

These models are trained using the raw sensor data as observation sequences. The sliding models are trained using the corresponding sliding gestures. The general ergodic model is trained using the data from sliding gestures that do not correspond to the modeled area. For example, the data set that represents a sliding gesture along the short edge of the Bar of Soap is used to train the ergodic model of the long side.

5.3.3 Data Processing

For the virtual Rubik's cube, the Bar of Soap merely transmits raw sensor data to a computer for processing and interpretation. The sequences of activated capacitive sensors are then broken up into appropriate observation sequences corresponding to the different gesture models. The trained models are used to calculate the probability of observing such a sequence. If a sequence has a higher probability of being observed given one of the sliding gesture models, a sliding event is triggered.

The sensor data is also used to overcome the fact that the single row of sensors on the edges of the Bar of Soap cannot determine which end of the Rubik's Cube should be rotated (see Figure 5-6). In the event of a sliding gesture on the edge of the Bar of Soap, the end correspoding to whichever face has fewer activated touch sensors will be rotated.

5.3.4 User Interface

This application, developed as a Matlab script, streams raw sensor data from the Bar of Soap via Bluetooth. A graphical version of a Rubik's cube, shown in Figure 5-7, is displayed on screen that it mapped to the Bar of Soap's orientation as determined from the accelerometer data.

To rotate an end of the Rubik's cube, the user simply slides a finger over a row of capacitive sensors on the Bar of Soap. If the sliding gesture is performed on either of the larger faces, the rotation will occur in the direction of the sliding gesture and on the corresponding end of the virtual cube. To rotate the end along the axis that is mapped perpendicular to the front and back face, the user simply places their hand over one of the large faces and slides their finger along one of the edge faces in the direc-

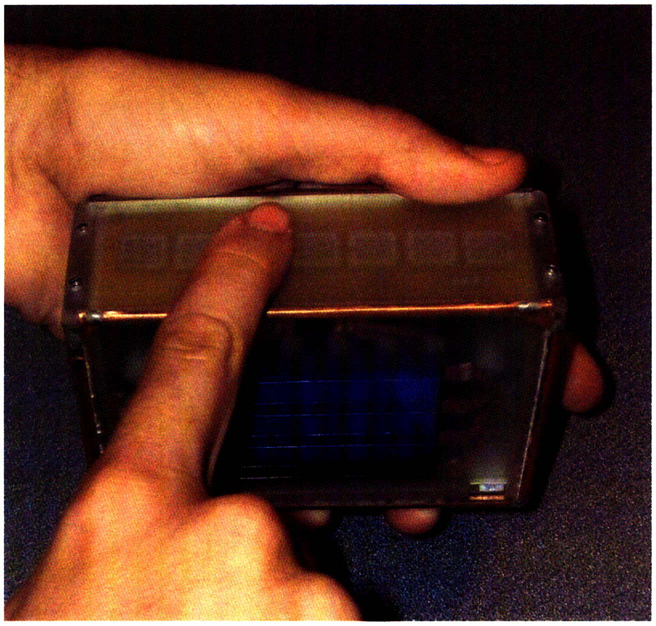

Fig. 5-6 Ambiguous Mapping of the Rubik's Cube Solved by Hand Detection

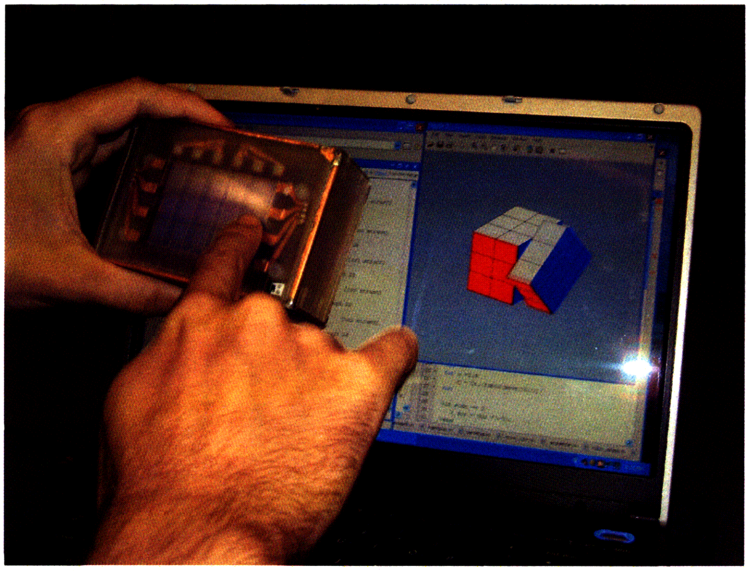

Fig. 5-7 Screen Shot of the User Interface for the Bar of Soap as a Generic Controller

tion of desired rotation. The virtual cube will interpret the covered side as the stationary side of the cube and rotate the opposite side.

6

Conclusion

The work described in this thesis was undertaken with the goal of developing an intuitive and effective interface for handheld devices. As the Graspables system was developed and tested, it became apparent that there was no reason to confine grasp recognition to the realm of portable devices. However, the goals of an intuitive and effective system remained.

One could argue that the fact that people seeing the Bar of Soap for the first time implicitly understand the meaning of “Hold it like a camera” demonstrates its intuitiveness. However, in many ways this is a better indictment of the pervasiveness of similarly designed commercial products than of the intuitiveness of a sensor covered box. As for effectiveness, the statistical nature of pattern recognition techniques automatically blur the idea with needs for acceptable error rates and the such.

On the other hand, this work does give a strong indication that grasp recognition certainly can implemented in an effective and intuitive way. The Graspables prototypes show that the hardware requirements for grasp recognition are acheivable. The user studies offer evidence that the software can perform at an acceptable level. Combine these results with a wide variety of potential applications and it would seem that the Graspables definitely succeeded in its goals.

6.1 Future Work

The prototypes discussed in this thesis only represent a small fraction of the potential implementations of the Graspables System. Another implementation that was discussed and would be worth developing is a stylus prototype. In the graphics art world alone, pencils, paintbrushes, erases and wands could all be represented by different ways of grasping a stylus.

For more of departure from the work in this thesis, implementing Graspables into existing devices would be interesting. Using the handheld device mode switching that was demonstrated by the Bar of Soap in a fully functional handheld would be worth studying. Questions about when to trigger the classification algorithms, what error rates would be acceptable to user, and the general effectiveness of the natural mode switching would be better explored by longer studies with functional devices. There is also the open question of how form factor variations in comercial handhelds would impact the Graspables System's effectiveness.

Applying what has been learned from the existing applications to other scenarios also has potential. Can the Graspables System be used as a safety check to ensure that power tools are being operated properly? What could be gained by expanding the scale of the system from handheld objects to whole body-sized arrays?

There is also room to perform further tests to improve the reliability and robustness of the system. Optimizing sensor densities could be valuable. Exploring how environmental factors such as humidity impact the capacitive sensors could improve system reliability. Exploring additional inputs such as pressure sensors could be beneficial. The software and classification routines could always benefit from more training data.

Appendix A

Schematics and PCB Layouts

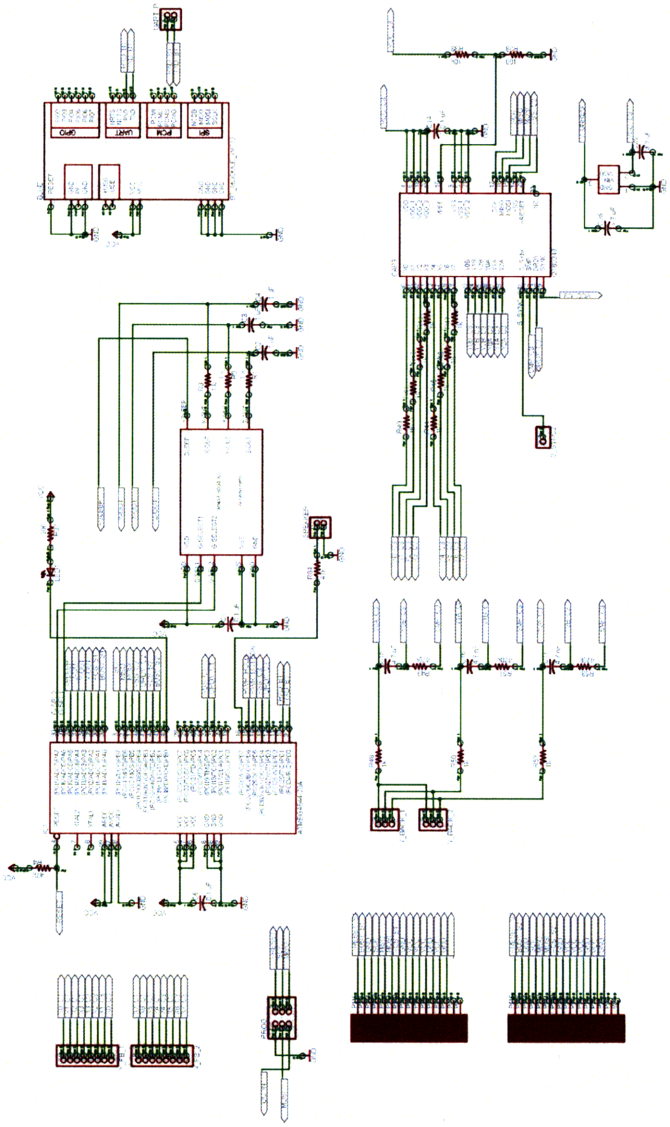

Fig. A-1 Schematics for the Bar of Soap V3 Control Circuit

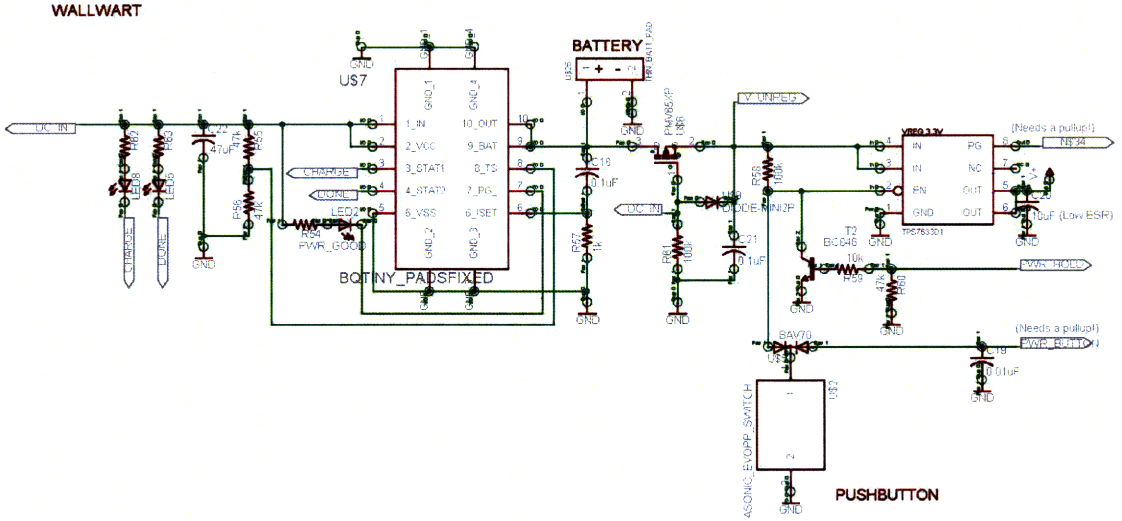

Fig. A-2 Schematics for the Bar of Soap V3 Power Circuit

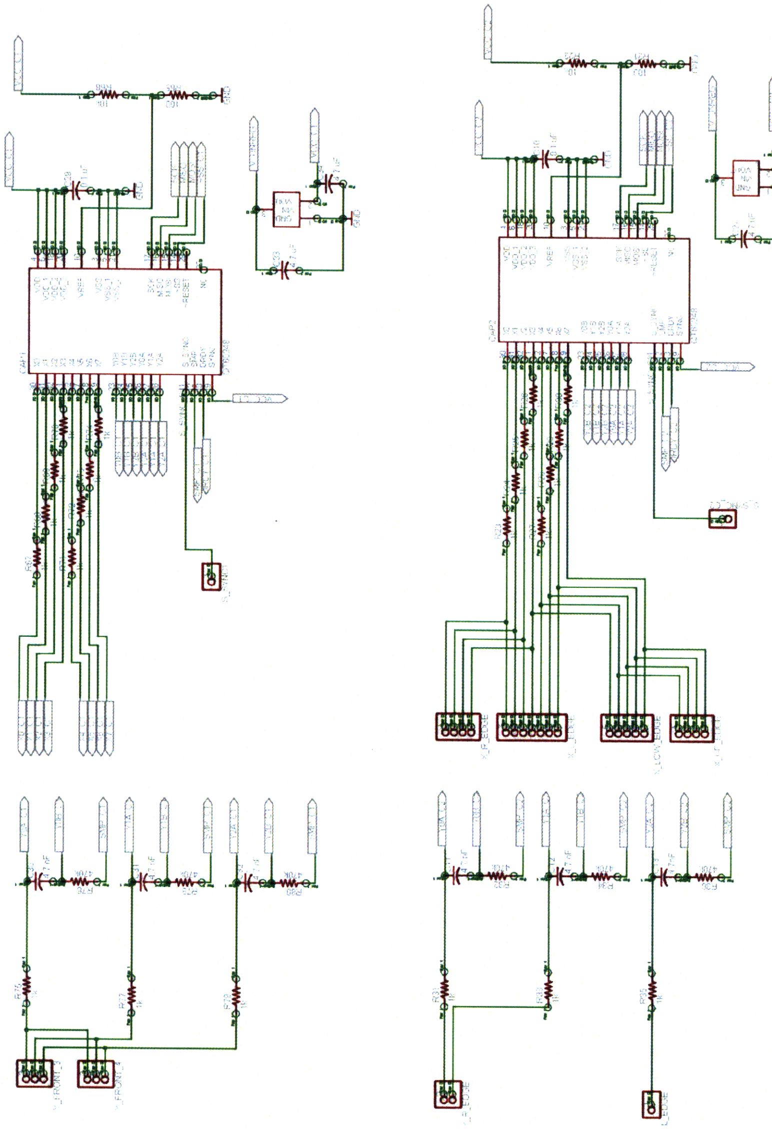

Fig. A-3 Schematics for the Bar of Soap V3 Capacitive Sensors Circuit

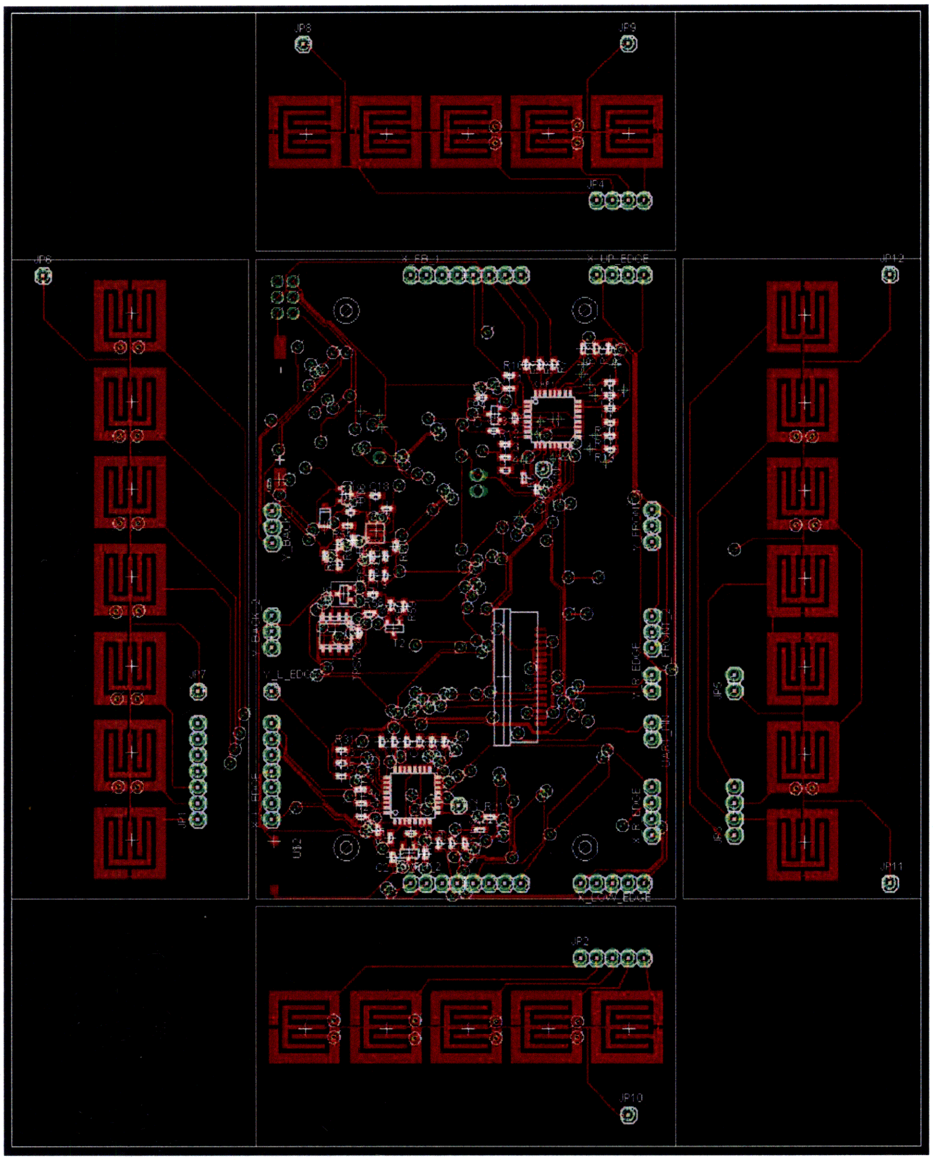

Fig. A-4 Top Layer of the Bar of Soap V3 PCB

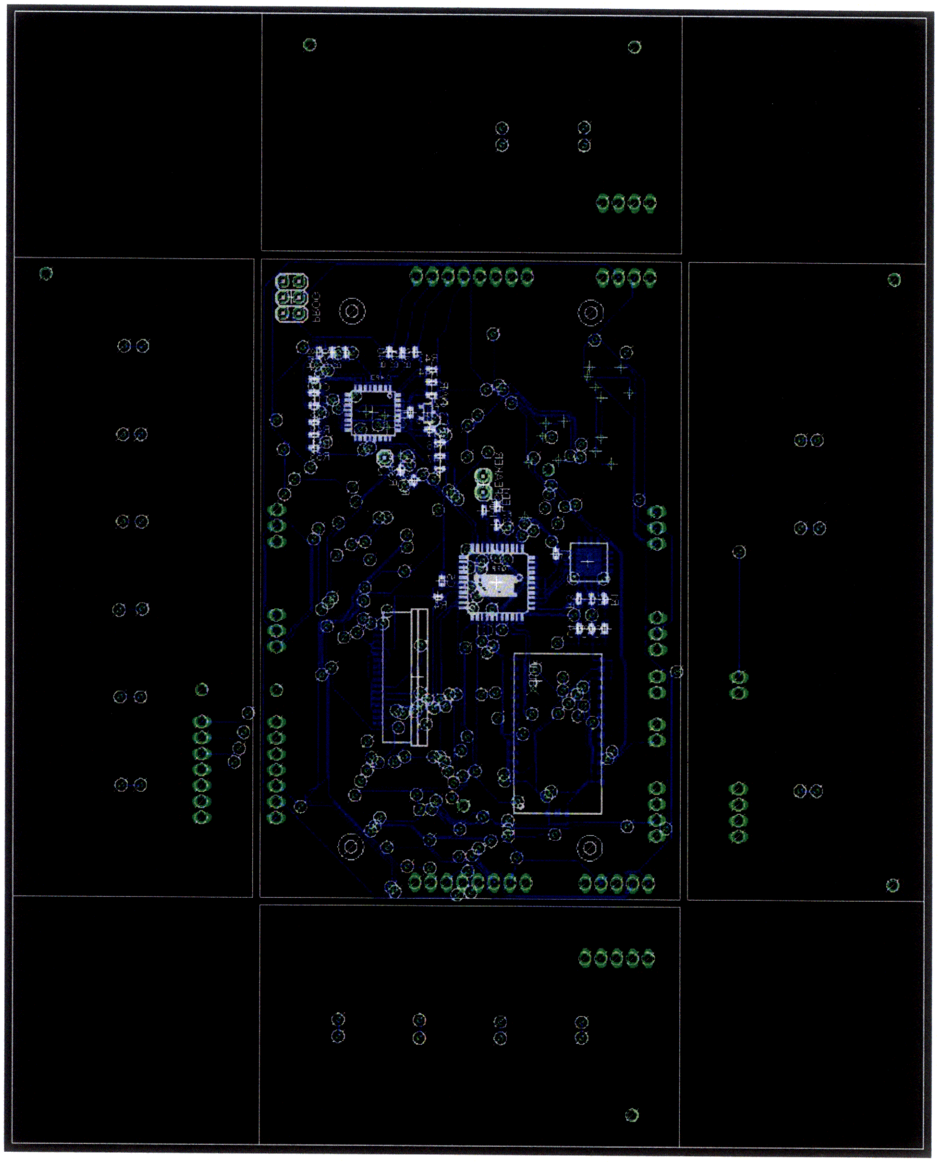

Fig. A-5 Bottom Layer of the Bar of Soap V3 PCB

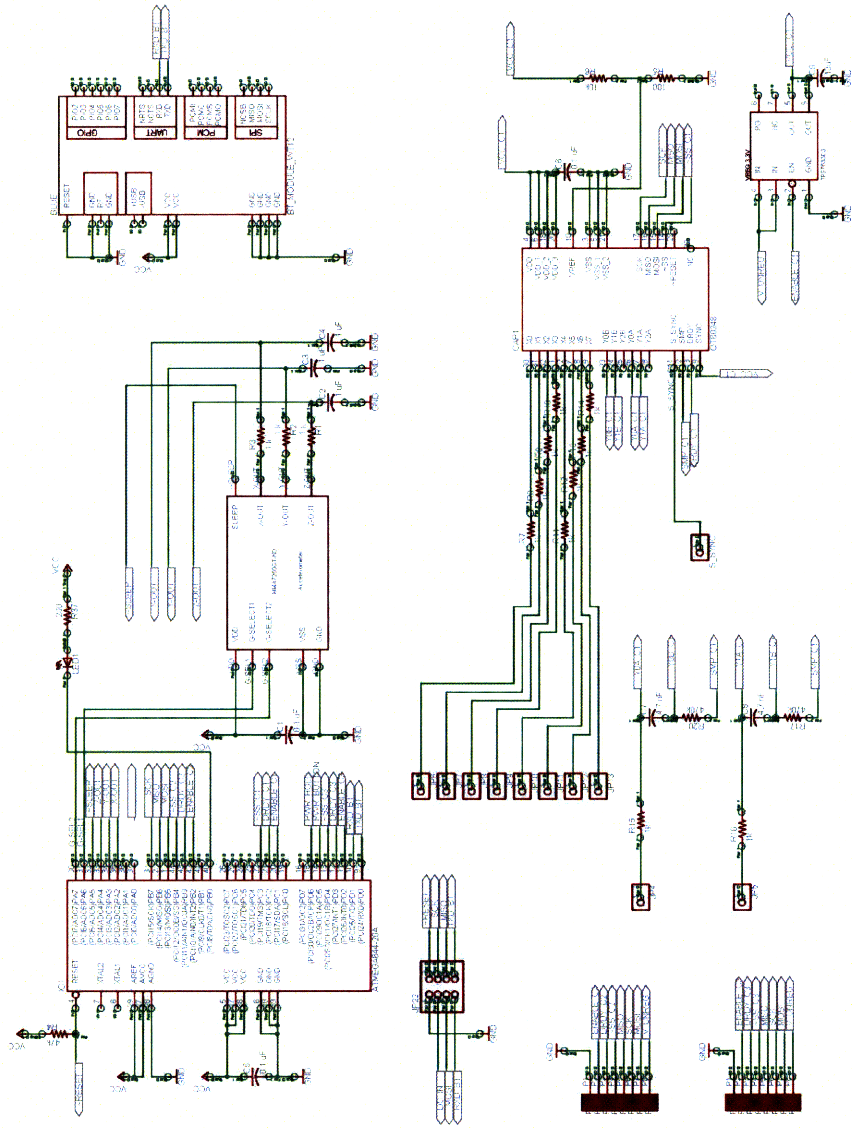

Fig. A-6 Schematics for the Ball of Soap V2 Main Board

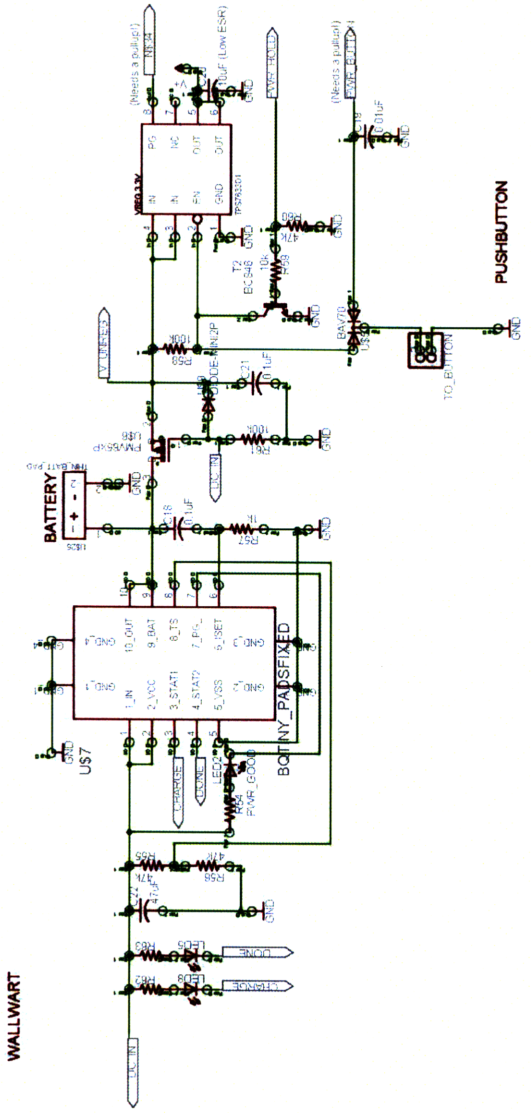

Fig. A-7 Schematics for the Ball of Soap V2 Power Circuit

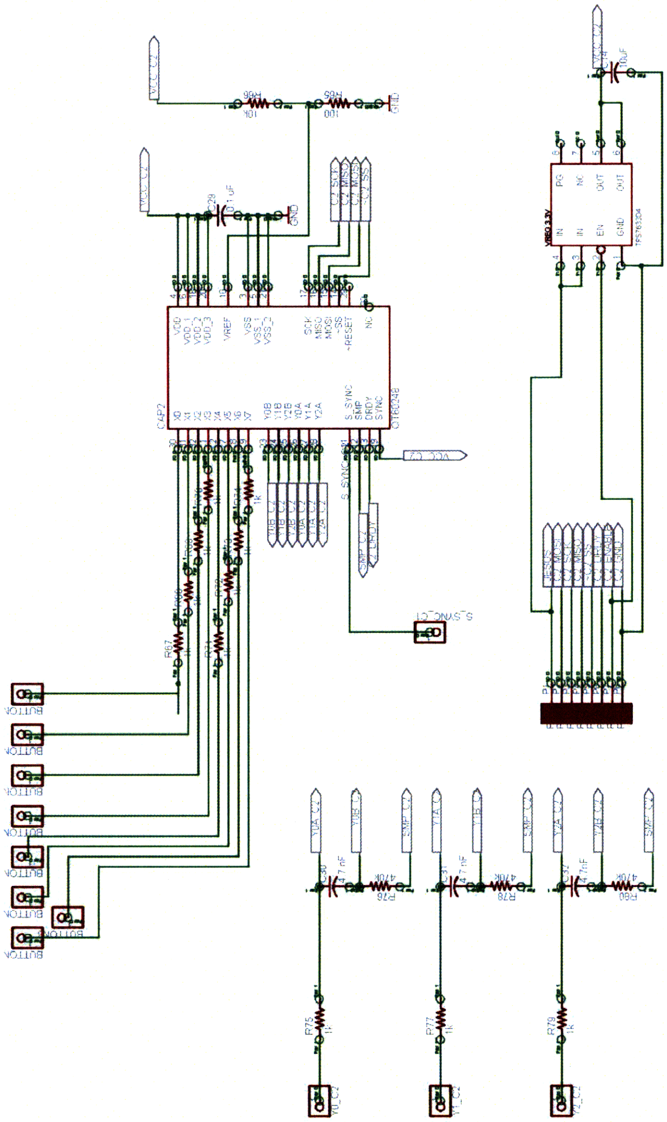

Fig. A-8 Schematics for the Ball of Soap V2 Top Board

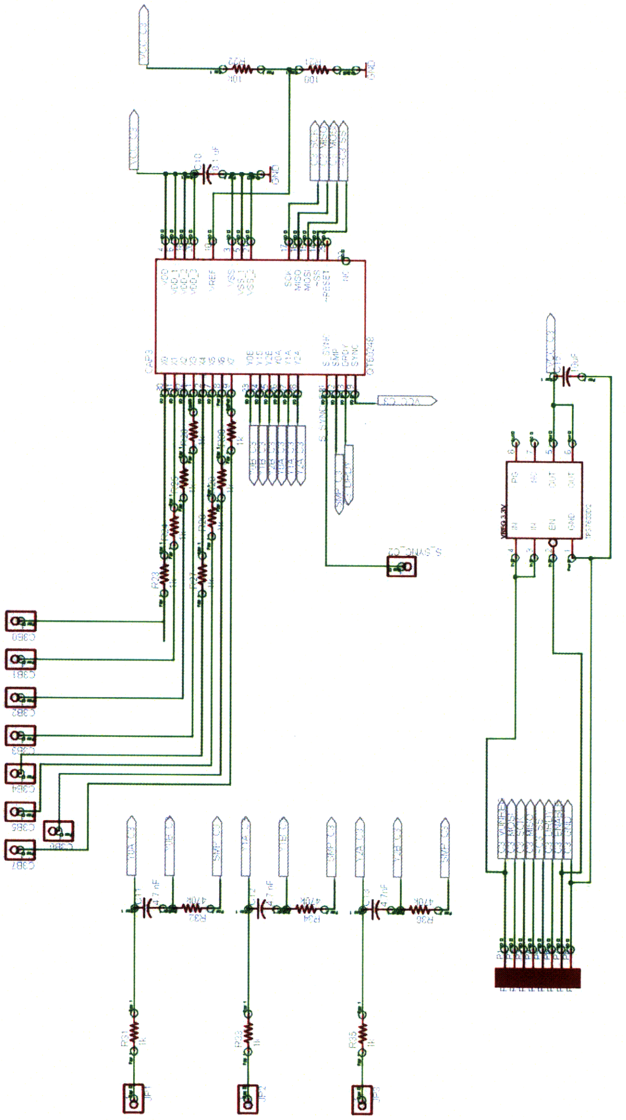

Fig. A-9 Schematics for the Ball of Soap V2 Bottom Board

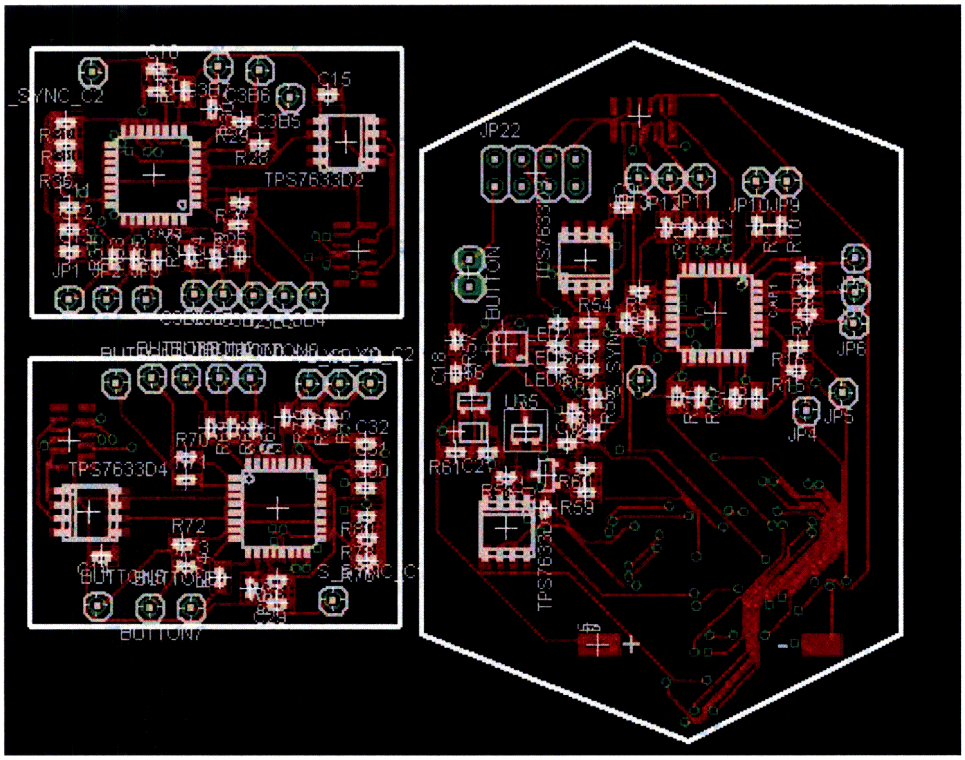

Fig. A-10 Top Layer of the Ball of Soap V2 PCB

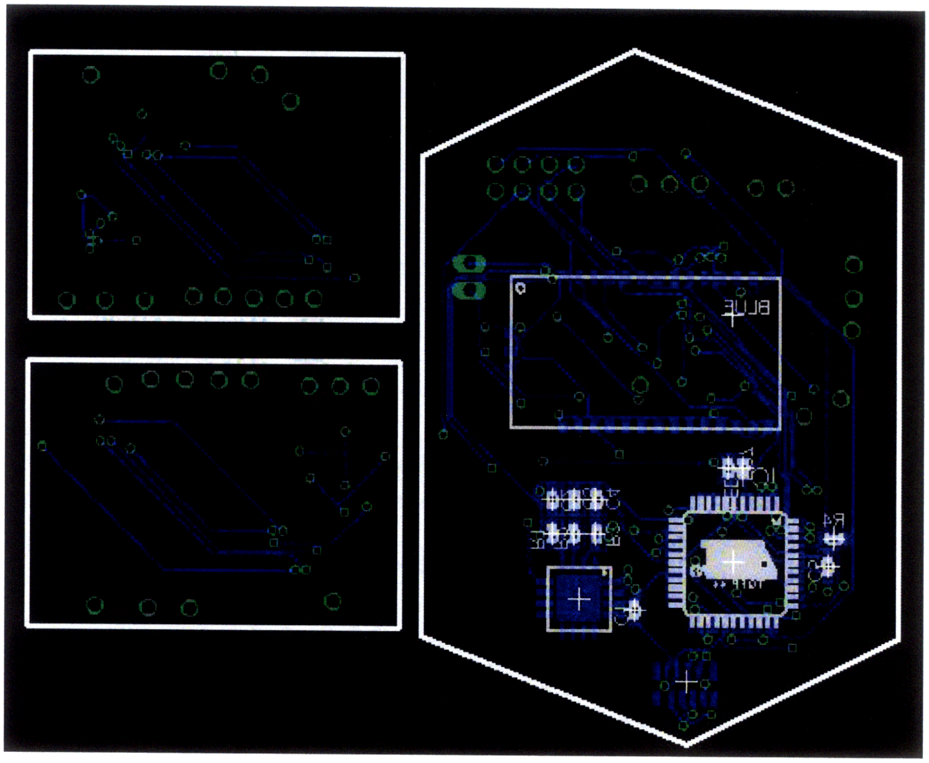

Fig. A-11 Bottom Layer of the Ball of Soap V2 PCB

Bibliography

[1] Apple iphone features http://www.apple.com/iphone/features.

[2] Emotive epoc http://emotiv.com/.

[3] Generating gollum: Anatomy of a cg creature http://express.howstuffworks.com/gollum3.htm.

[4] Hyung-Il Ahn, Alea Teeters, Andrew Wang, Cynthia Breazeal, and Rosalind Picard. Stoop to conquer: Posture and affect interact to influence computer users persistence. pages 582–593. 2007.

[5] Jussi Angelsleva, Ian Oakley, Stephen Hughes, and Sile O'Modhrain. Body mnemonics: Portable device interaction design concept.

[6] J. Baker. The dragon system–an overview. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(1):24–29, Feb 1975.

[7] W. W. Bledsoe and I. Browning. Pattern recognition and reading by machine. In *In Proceedings of the Eastern Joint Computer Conference*, 1959.

[8] Jeanette Borzo. Now you're talking. *Business 2.0 Magazine*, March 2007.

[9] Mourad Bouzit, George Popescu, Grigore Burdea, and Rares Boian. The rutgers master ii-nd force feedback glove. *Proceedings of IEEE VR 2002 Haptics Symposium*, 2002.

[10] Wook Chang, Kee Eung Kim, Hyunjeong Lee, Joon Kee Cho, Byung Seok Soh, Jung Hyun Shim, Gyunghye Yang, Sung-Jung Cho, and Joonah Park. Recognition of grip-pattterns by using capacitive touch sensors. *IEEE ISIE 2006*, 2006.

[11] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* John Wiley and Sons, Inc., 2001.

[12] George W. Fitzmaurice and William Buxton. An empirical evaluation of graspable user interfaces: towards specialized, space-multiplexed input. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 43–50, New York, NY, USA, 1997. ACM.

[13] George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. Bricks: laying the foundations for graspable user interfaces. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[14] Personal Robots Group. Huggable. http://robotic.media.mit.edu/projects/robots/huggable/overview/overview.html.

[15] Jari Hannuksela, Mark Barnard, Pekka Sangi, and Janne Heikkilä. Adaptive motion-based gesture recognition interface for mobile phones. pages 271–280. 2008.

[16] Beverly L. Harrison, Kenneth P. Fishkin, Anuj Gujar, Carlos Mochon, and Roy Want. Squeeze me, hold me, tilt me! an exploration of manipulative user interfaces. *CHI 98*, 1998.

[17] Kee-Eung Kim, Wook Chang, Sung-Jung Cho, Junghyun Shim, Hyunjeong Lee, Joonah Park, Youngbeom Lee, and Sangryoung Kim. Hand grip pattern recognition for mobile user interfaces. In *AAAI*, 2006.

[18] Paul Kry and Dinesh Pai. Grasp recognition and manipulation with the tango. In *International Symposium on Experimental Robotics*, July 2006.

[19] Joshua Lifton, Michael Broxton, and Joseph A. Paradiso. Distributed sensor networks as sensate skin.

[20] Vladimir J. Lumelsky, Michael S. Shur, and Sigurd Wagner. Sensitive skin. *IEEE Sensors Journal*, 1(1), June 2001.

[21] Vesa-Matti Mantyla, Jani Mantyjarvi, Tapio Seppanen, and Esa Tuulari. Hand gesture recognition of a mobile device user.

[22] David J. Merrill and Joseph A. Paradiso. Personalization, expressivity, and learning of an implicit mapping strategy for physical interfaces.

[23] Thomas A. Mysliwiec. Fingermouse: A freehand computer pointing interface. In *In Proc. of Int'l Conf. on Automatic Face and Gesture Recognition*, pages 372–377, 1995.

[24] Dinesh K. Pai, Eric W. VanDerLoo, Subarna Sadhukhan, and Paul G. Kry. The tango: A tangible tangoreceptive whole-hand human interface. *Proceedings of World Haptics*, 2005.

[25] Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human-computer interaction:a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, July 1997.

[26] Hal Philipp. Charge transfer sensing. *Sensor Review*, 19(2), 1999.

[27] Richard Quinnell. Touchscreen technology improves and extends its options. *EDN*, 40(23), Nov 1995.

[28] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[29] Jun Rekimoto. Tilting operations for small screen interfaces. Technical report, Sony Computer Science Laboratory Inc.

[30] Mariko Sanchanta. Nintendo's wii takes console lead. *Financial Times*, September 2007.

[31] Walter Dan Stiehl and Cynthia Breazeal. Affective touch for robotic companions.

[32] Walter Dan Stiehl, Jeff Lieberman, Cynthia Breazeal, Louis Basel, Levi Lalla, and Michael Wolf. Design of a therapeutic robotic companion for relational, affective touch. *2005 IEEE International Workshop on Robots and Human Interactive Communication*, 2005.

[33] Alea Teeters, Rana El Kaliouby, and Rosalind Picard. Self-cam: feedback from what would be your social partner. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Research posters*, page 138, New York, NY, USA, 2006. ACM.

[34] Mark Weiser. The computer for the 21st century. *Scientific American*, 1991.