Design and Implementation of a Relief Interface

by

Daniel Leithinger

M.Sc, Upper Austria University of Applied Sciences (2007)
B.Sc., Upper Austria University of Applied Sciences (2005)

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

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010

Author ______________________________
Program in Media Arts and Sciences
June, 2010

Certified by ______________________________
Hiroshi Ishii
Muriel R. Cooper Professor of Media Arts and Sciences
Program in Media Arts and Sciences, MIT
Thesis Supervisor

Accepted by ______________________________
Prof. Pattie Maes
Associate Academic Head
Program in Media Arts and Sciences

Design and Implementation of a Relief Interface

by

Daniel Leithinger

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on June, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Tangible representations of virtual content allow for sensory-rich interactions with computers through real-world objects. However, these objects are commonly static handles, with limited capabilities to present changing information through their form. This thesis presents a human-computer interface able to generate physical 3D shapes and sense user input through them. It consists of an array of sensors and actuators to deform a malleable surface. We define this type of system as a relief interface. This thesis documents the hardware and software design and proposes a set of pressure-based gestures, which are evaluated through applications for geospatial exploration, surface modeling and multilayer viewing.

Thesis Supervisor: Hiroshi Ishii
Title: Muriel R. Cooper Professor of Media Arts and Sciences, Program in Media Arts and Sciences, MIT

Design and Implementation of a Relief Interface

by

Daniel Leithinger

The following people served as readers for this thesis:

Thesis Reader____

Pattie Maes
Associate Professor of Media Arts and Sciences
Program in Media Arts and Sciences, MIT

Thesis Reader____

Chia Shen
Director, SDR Lab
School of Engineering and Applied Sciences (SEAS), Harvard University

Acknowledgements

I would like to thank my advisor, Hiroshi Ishii, for his guidance and support throughout the research process. I would also like to thank my readers, Pattie Maes and Chia Shen, for their time and feedback regarding this thesis.

Special thanks to all the members in the Tangible Media Group (Adam Kumpf, Keywon Chung, Sean Follmer, Xiao Xiao, Jinha Lee, Cati Vaucelle, Leonardo Bonanni, Amanda Parkes, and Jamie Zigelbaum) for their suggestions, constructive criticism, and help with prototyping over the past two years.

To my family, I am thankful and highly in dept for their loving support of my studies and travels.

Contents

Chapter 1

Introduction

To date, human computer interfaces are dominated by visual and auditory displays, which are able to communicate large amounts of information, but do not engage all human senses. The mapping between user input and output is often based on the paradigm of "window, icon, menu, pointing device" (WIMP), which does not take full advantage of the users ability to perceive and manipulate real-world objects. Therefore, alternative approaches like tangible user interfaces (TUI) have been proposed [22]. TUI represent digital information and computation in physical embodiments. Through colocated input and output, phycial objects and user manipulation of these objects are directly mapped to digital computation and feedback. The physical affordances of objects allow users to interact by grasping and manipulating them, while receiving passive haptic feedback through their shape, size and weight. Active haptic feedback can be added by actuating objects to control their shape and position in space. An optimal actuated TUI would couple physical and digital inseperably, with the ability to control physical matter computationally like pixels on a computer screen. Such an interface would be able to engage all user senses by creating and transforming physical representations of digital information and computation.

This thesis describes a system inspired by this vision of computatinally controlled creation and transformation of physical objects. We define *relief interfaces* as computer interfaces with the ability to physically change shape and sense direct user input. A relief interface consists of an array of actuators, which deform a malleable surface into a transformable shape. Users can view, touch and manipulate this surface to interact with virtual content. Previous related work has proposed various systems with such shape output capabilities, but interactions and applications for relief interfaces have not yet been explored extensively [26], [43]. One reason is the complexity of the required hardware and software.

Therefore, one goal of this thesis is the documentation of a simple and low-cost relief interface, consisting of commercially available components and open source frameworks. The second goal is to propose appropriate gestures for direct user input. We present three applications utilizing the system with the proposed gestures. These are a tool for geospatial exploration, a 3D modeling application and a multilayer data browsing tool. The thesis also reports observations and feedback gathered from users interacting with these applications using the proposed gestures.

1.1 Thesis Outline

Chapter 2 contains an analysis of related work in actuated user interfaces. Chapter 3 describes a low-cost sensing and actuation platform based on commercially available components. Chapter 4 contains a documentation of the software software framework for rendering and input detection. Chapter 5 investigates appropriate user interactions with set of pressure-based gestures. Chapter 6 demonstrates applications that utilize the shape output and the proposed gestures. Chapter 7 contains our initial observations of users interacting with the system and feedback gathered. Chapter 8 summarizes the lessions learned and Chapter 9 provides an outlook of future work.

Chapter 2

Related Work

> "The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked."
>
> Ivan Sutherland (1965) [52]

Although Sutherland is being fantastical in his example of how computers can control and modulate physical matter, his very material ambition is that physical matter be shifted, modulated, and controlled via computational means, in the same way that pixels and sound are. Since this vision was proposed, various ways of augmenting human computer interfaces through actuation have been developed.

Poupyrev et al. identify the application areas for actuated user interfaces as *aesthetics*, *information communication*, *mechanical work*, *data consistency for tangible controls*, and *people-to-people communication* [44]. However, as Poupyrev notes, applications often overlap in a single interface. An example is a humanoid personal robot [9]. Such a robot might be actuated to communicate to a user by gesturing, which places it into the category of information communication. At the same time, the robots could move for aesthetic reasons, or perform mechanical work. If the robot mimics the interaction of remote users, its application area is people-to-people communication.

A different approach to categorize actuated user interfaces is by identifying the sensory channels through which they communicate with the user. The sense of touch is the most evident channel, as computers are not able to provide active tactile feedback through non-actuated interfaces like keyboard, mouse and monitor. However, communication through visual or auditory channels is often just as important or even the sole reason for actuation. To provide a better understanding of prior work in actuated interfaces, the following sections investigate them by their primary channel of communication.

2.1 Touch

Haptic perception or the sense of touch can be divided into cutaneous and kinesthetic perception. Receptors in the skin provide cutaneous (tactile) information, while kinesthetic information is perceived through muscles, tendons and joints [35].

Benali-Khoudja et al. provide a survey of tactile interfaces by application area [4]. Their proposed applications for tactile interfaces are: *Teleoperation and telepresence*, *Laboratory prototypes to study the different tactile parameters*, *Sensory substitution*, *3D surface generation*, *Braille systems*, and *Games.* Some tactile interfaces are not actuated, such as electrical stimulation interfaces, as proposed by Kajimoto et al. [29] and thermal sense interfaces as proposed by Kushiyama et al. [34].

Examples of actuated haptic interfaces include research projects and commercial interfaces for surgical planning like the Immersion TouchSense surgical simulators and the P02 Tactile Feedback System for Robotic Surgery. The most common commercial interface to explore three-dimensional shapes with a haptic pen is the SensAble PHANTOM Omni Haptic Device [37], with larger form factors explored by Brooks et al [5]. Examples of commercial haptic gaming interfaces include the Novint Falcon, Logitech WingMan Force Feedback

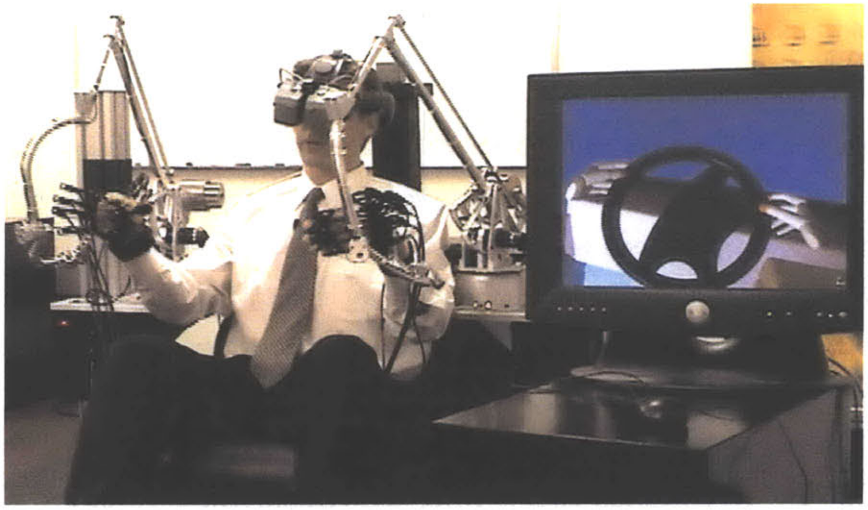


Figure 2-1: The Immerstion Haptic Workstation 2003 (Image courtesy of Immersion Corporation)

Mouse, and Orbit 3D Trackball; while interfaces for immersive virtual reality are VRLogic CyberGrasp and CyberTouch. These products are commonly utilized in conjuncion with visual interfaces like monitors or head-mounted displays, the two channels are not directly colocated. One of the most elaborate examples is the Immersion Haptic Workstation, which allows users to grasp virtual objects with all fingers of both hands [20]. As depticted in Figure 2-1, the shape of the haptic interface is dictated by the functionality and not intended to be visually perceived by the user.

While most haptic interfaces are used in conjunction with a visual display, some communicate only through touch, such as the vibrating alert of cell phones. Proverbial Wallet by Kestner et al. follows a similar approach to communicate financial information through the sense of touch for privacy reasons [30].

2.2 Sight

While computer screens are the most common form of communication, changing the physical shape of an object is an intriguing way of creating calm, persistent interfaces or 3D visualizations that might be hard to achieve with current display technology. Shutters by Coelho and Maes is an architectural surface, which communicates information through the subtle opening and closing of fabric shutters [8]. Kinetische Skulptur by art+com consists of an array of suspended metal spheres, whose position is actuated by electromotors [2]. While it allows multiple viewers to perceive 3D shapes, due to the size and placement of the interface, users are prevented from touching it (see Figure 2-4). The Source by Greyworld follows a similar approach [15]. Kodama's sculptures like Protrude, Flow and Morpho Tower utilize electromagnets to create 3D artwork with ferrofluids, which are silent and not meant to be touched [32]. Wooden Mirror by Daniel Rozin utilizes 900 electromotors to create an analog depiction of a video feed from a camera pointing at the observer [54, p. 82].

Actuation is also utilized for volumetric displays, where a motor spins an illuminated plate to create a persistent three-dimensional image [11]. Due to the speed of the rotation, the plate can not be touched and is not perceived as an actuated object.

2.3 Sight and Hearing

An early example of combining visual and aural information is Dangling String by Natalie Jeremijenko, a piece of string attached to a motor, which rotates to indicate network traffic. Mark Weiser uses this project as an example to describe his vision of calm technology [57]. Pinwheels by Ishii, Ren and Fry follows a similar approach to communicate abstract information through an array of spinning wheels [23]. Cloud by Troika is a digital sculpture with thousands of small electromechanical indicators, that can change their state audibly from black to silver [54, p. 82].

2.4 Combination of Touch and Sight

Since the introduction of graspable [12] and tangible user interfaces [24], researchers have tried to couple the digital state of a program with the physical state of the interface through actuation. Such an interface communicates both on a visual and on a haptic level. Examples of actuated interfaces for educational purposes are Curlybot by Frei et al. [13] and Topobo [46], which can be programmed by example. InTouch by Brave and Dahley is an interface for remote communication, consisting of wooden bars connected by virtual springs to transmit touch [6].

Examples of tangible tabletop interfaces which allow the computer to move objects physically include Actuated Workbench by Pangaro et al. [24], Augmented Coliseum by Kojima et al. [33], and Planar Manipulator Display by Rosenfeld et al. [50].

Patten investigates actuation for tangible user interfaces through a tabletop interface, which is able to move objects placed on it through an array of electromagnets [41]. This way, the program state is represented by the position of the tangible objects and the user can intervene physically by moving objects or by restricting their movement on the surface.

Super cilia skin by Raffle et al. is a computationally enhanced membrane coupling tactile input with tactile and visual output [45].

Relief interfaces utilize an array of pins to deform a shape. This shape is a visual communication channel, but can also be touched. The following section describes previous approaches in creating relief interfaces.

2.4.1 Relief Interfaces

Relief interfaces have been developed in various sizes; from small scale to provide tactile stimuli to the users fingertips [55] to wall-sized [14]. The scale this thesis investigates allows the user to touch and interact with the interface using both hands and is therefore closely related to the projects described in the following paragraph.

The first publicly demostrated relief interface is FEELEX by Iwata et al. [26], where 36 motorized pins actuate a malleable surface, onto which graphics are projected (see Figure 2-2). Iwata states the FEELEX was designed to adress shortcomings of previous haptic interfaces created by his group. These shortcomings include a lack of spatially continuous

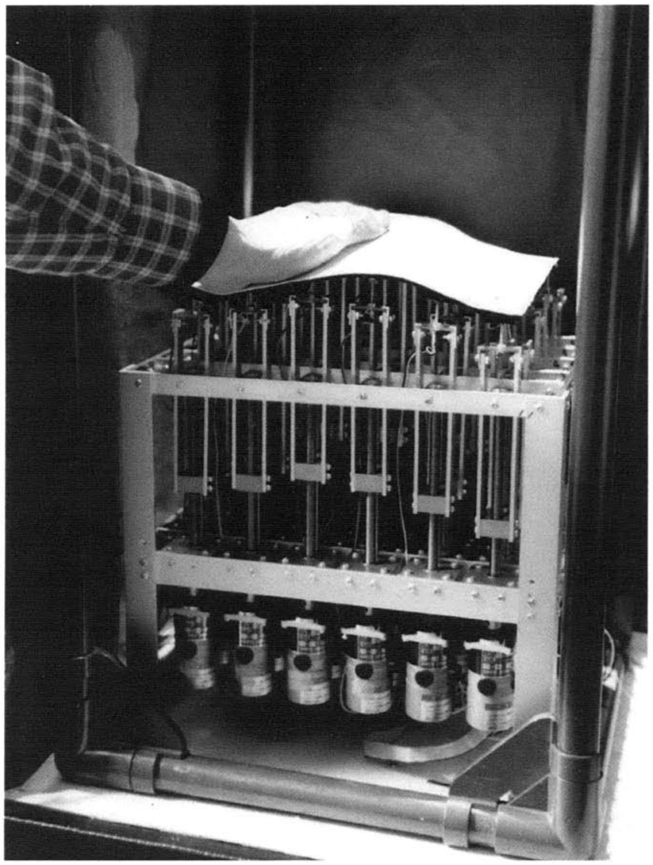


Figure 2-2: Feelex 1 1997 (Image courtesy by VR Lab, Univ. of Tsukuba)

contact points between interface and user and the discontinuity of the haptic channel and the visual output, which is presented on a separate display [25].

Lumen by Poupyrev et al. [43] utilizes shape memory alloy to individually actuate 169 rods on a tabletop display (see Figure 2-3). Poupyrev refers to this interface as an RGBH display, with RGB refering to the color component and H to the height component of each pixel. Capacitive sensing detects touch input on the display. However, input strength is not detected.

The Terrain Table by Northrop Grumman is a large high-resolution tabletop display utilized for geospatial military applications [17], [40]. 7000 rods render terrain through an attached silicon surface, which is augmented with projected graphics.

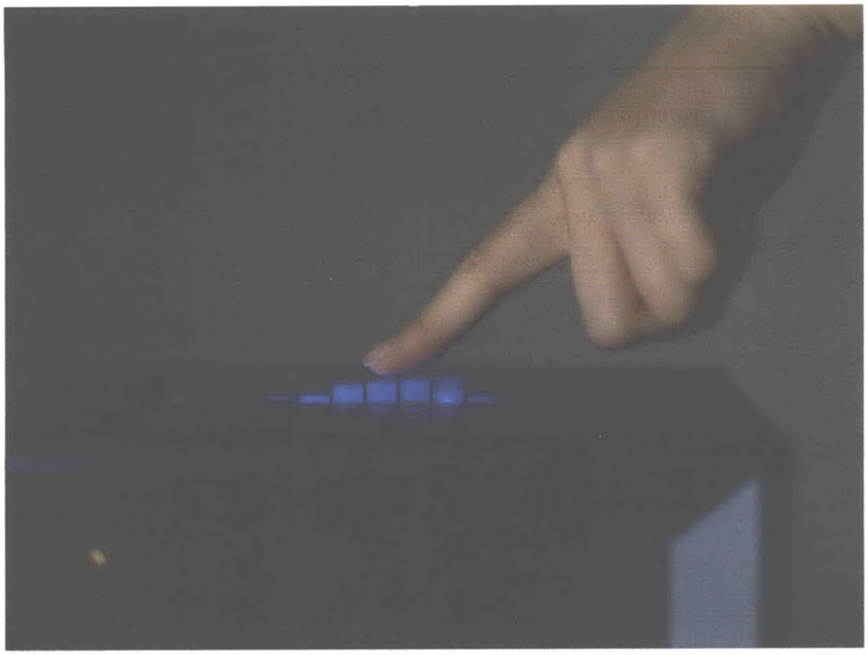


Figure 2-3: Lumen 2004 (Photograph by Makoto Fujii, courtesy AXIS magazine)

Surflex by Coelho et al. is an approach to deform a surface through spring actuators embedded inside the fabric instead of an external array [7].

Most proposed systems use custom hardware with a high mechanical complexity, which limits the scalability for research projects, as Table 2.1 shows. Some research projects like Digital Clay by Haihong [58] address this issue for possible future commercial production. However, low-cost solutions are still unavailable to HCI researchers today.

Project	Actuation Technology	Component availability
Terrain Table	Pneumatic	custom made
Digital Clay	Hydraulic	custom made
Lumen	Shape memory alloy	custom made
FEELEX 1	Screw driven electromotors	custom made
BubbleWrap	Solenoids	custom made
FEELEX 2	Servos	commercially available, expensive
Glowbits	Motorized Potentiometers	commercially available

Table 2.1: Comparison of actuation technologies for relief interfaces.

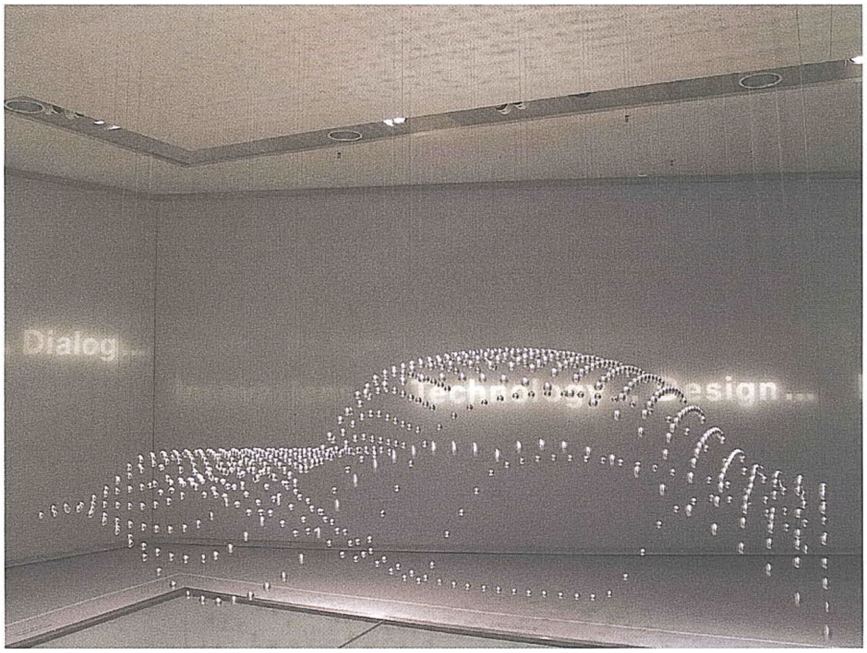


Figure 2-4: Kinetische Skulptur (2009) by art+com at the BMW museum (Image courtesy of Ole Pophal)

Electric slide potentiometers have the advantage of providing linear actuation and sensing in a relatively compact, commercially available package. For this reason, they have previously been utilized to create actuated arrays for sensing user input and rendering shapes, such as in AR-Jig by Anabuki and Ishii [1]. Another example of a 2D array using electric slide potentiometers is Glowbits by Hirschmann (see Figure 2-5) [19]. Both projects feature unique form factors for specific application domains.

In contrast to the related work, this thesis proposes a generic actuation apparatus, which can be extended with minimal effort. As the main focus is on exploring user interactions with the interface, we utilize the sensing of the slide potentiometers for direct input.

Some systems have hardware similarities to relief interfaces, as they use an array of actuators to deform a malleable surface. However, they are not able to render continuous 3D shapes and therefore are different in their applications and interactions. These projects include Shade Pixel [31] and Bubblewrap [3].

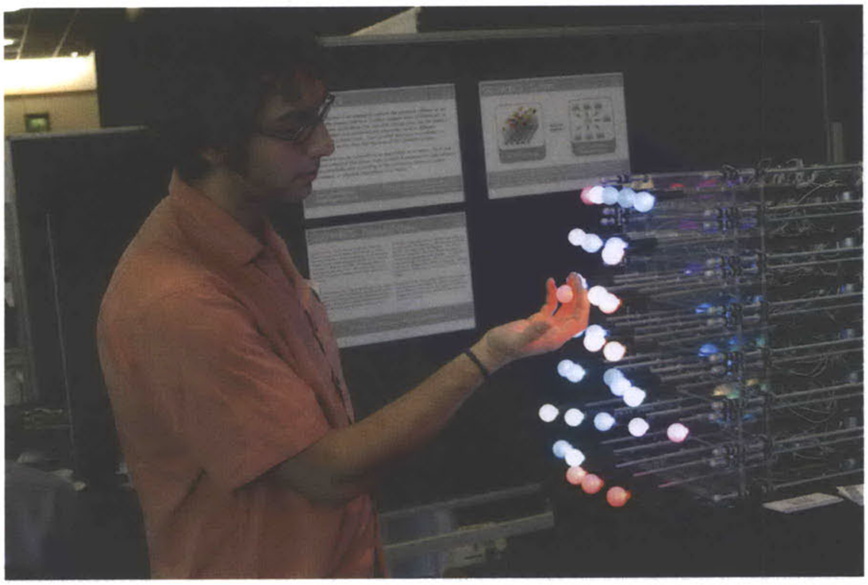


Figure 2-5: Daniel Hirschmann with Glowbits (2004) demo at Ubicomp 2004 (Image courtesy of Tom Igoe)

Chapter 3

Hardware Implementation

While the software and interactions described in this thesis can be applied to a range of relief interfaces, no such interface is commercially available at this time. This section describes the hardware setup built for the relief project and the considerations influencing the individual aspects of the hardware.

Figure 3-2 depicts the hardware setup, consisting of a table with a built-in actuation and sensing apparatus (a), which is connected to a malleable fabric surface through aluminum rods (b). A revolving ring senses additional user input (c). The table is controlled by a single computer (d), which also outputs graphics onto the surface through a ceiling-mounted projector (e).

3.1 Sensing and Actuation Apparatus

3.1.1 Integrated Actuators

The actuation and sensing apparatus is a crucial part of any relief interface. Proposed technologies to create an array of actuators include electromotors [28], solenoids [3], hydraulic [58] and pneumatic elements [14] and shape memory alloy actuators [43].

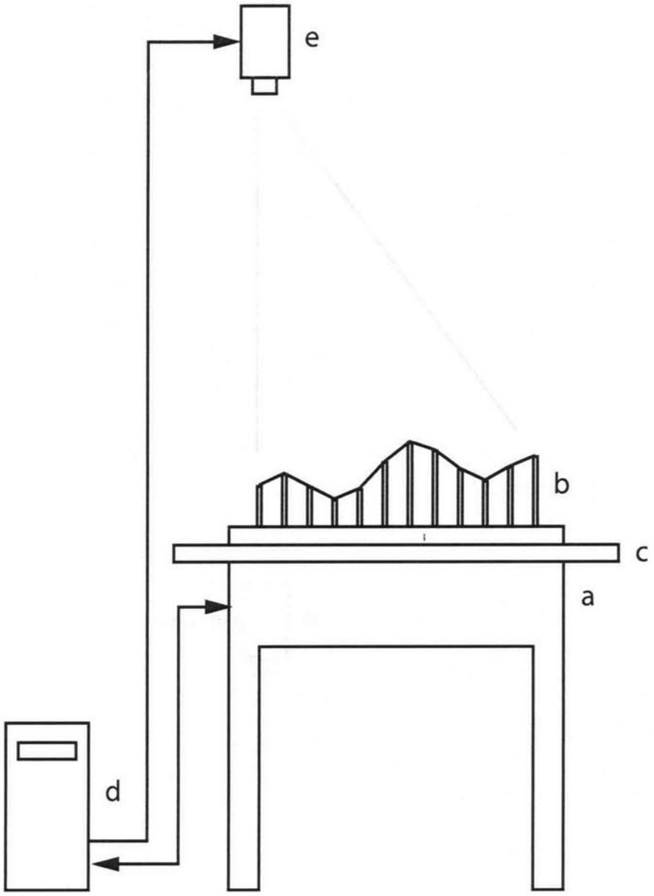


Figure 3-1: Schematic of Relief hardware setup.

Figure 3-2: Schematic of Relief hardware setup. Actuation and sensing apparatus (a), fabric surface with aluminum rods (b), revolving ring (c), computer (d), projector (e)

The main considerations when designing the actuators and sensors were latency, strength, range, size and cost. **Latency** refers to the delay between the application sending a shape to the actuation apparatus and the point at which the shape is completely rendered. It also refers to the delay between a user touching the hardware interface until the application registers this input. Haptic devices usually require update rates of 1 kHz or higher to convey the sense of touch [16]. However, as the Relief actuation apparatus is transforming an actual physical shape instead of rendering virtual haptics, the user always experiences haptics, independent of the actuator latency. Therefore, the latency can be higher, as it is is only dependent on the desired speed of the shape change and the input lag. Brooks et al. report a minimum required update rate to convey haptics with their GROPE system as 15 Hz [5]. Based on this information, further related work by MacKenzie and Ware [36] and informal tests, we therefore settled on a maximum latency of 66 milliseconds. **Strength** refers to the strength of the individual actuators. It has to be sufficient to transform the physical interface shape and resist user presses up to a certain extent to convey the sense of touching an object. However, excess strength poses a risk to the user in case of a software or hardware malfunction. The **Range** is the maximum length of the extended actuator. **Size** refers to the actuator dimensions and determines how tight the individual actuators can be stacked next to each other. Smaller actuators allow for a higher resolution. **Cost** is a combination of the cost of materials and assembly time required to build the interface. It is critical, as it limits the feasibility of the project.

To satisfy the constraints of low cost and scalability while fulfilling the other specifications, our actuation apparatus is built around commercially available motorized linear potentiometers from ALPS. The Alps RSA0N11M consists of an analog slide potentiometer with a travel of 5 inches. A DC motor drives the slider of the potentiometer with a gear belt. The typical application domain for these components are audio faders in mixing consoles (see Figure 3-3). To create actuators with these slide potentiometers, we attached aluminum rods to the sliders, which protrude when the slider is actuated. The size of the interface and the spacing of the actuators was determined through a series of mockups. Figure 3-4 shows a mockup utilizing glue sticks. At this point, different fabric surfaces to cover the actuation apparatus were tested. Figure 3-5 depicts a mockup covered with a Spandex fabric and

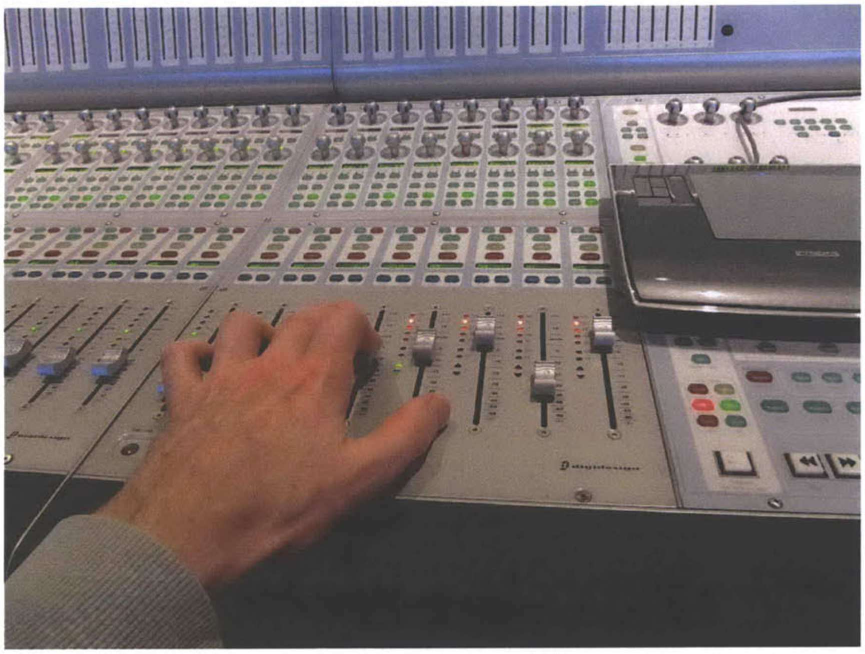


Figure 3-3: Motorized faders on a Digidesign audio mixing console. (Image courtesy of Hens Zimmerman)

augmented with a projected landscape. After testing the mockup to see if the resolution and range would be sufficient, we decided to utilize an array of 120 actuators, spaced 1.5 inches from each other and built into a circular tabletop. The arrangement of the individual actuators is depicted in Figure 3-6.

3.1.2 Microcontrollers

To interface between the application software (see Chapter 4) and the actuation modules, we utilize microcontrollers. The software sends the desired position of an actuator to the microcontroller. The microcontroller measures the input from the potentiometer and drives the motor to the desired position. As the microcontroller does not provide sufficient current to drive the motor directly, Texas Instruments L293DNE quadruple high-current half-H drivers are utilized. To avoid overshoot and oscillation when the position of the actuator reaches the desired position, a PID term is applied, which slows down the motor as the

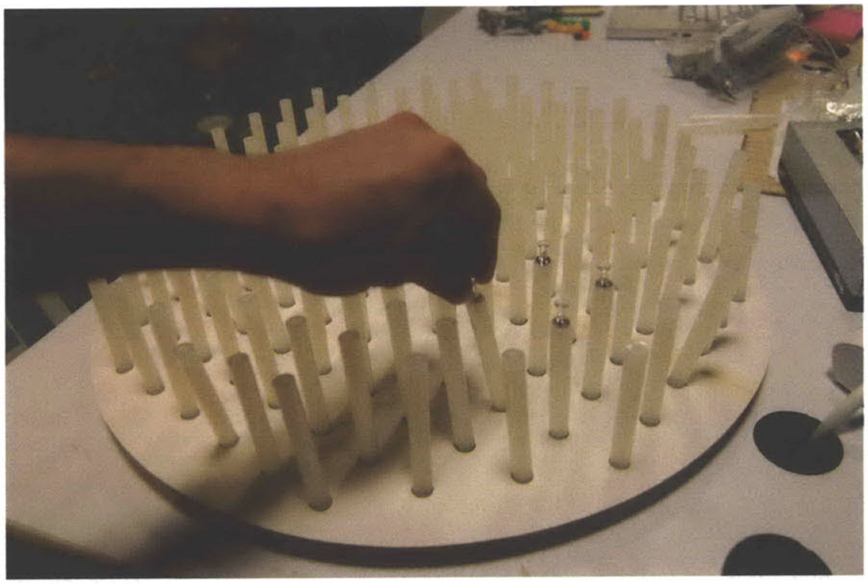


Figure 3-4: Mockup with glue sticks.

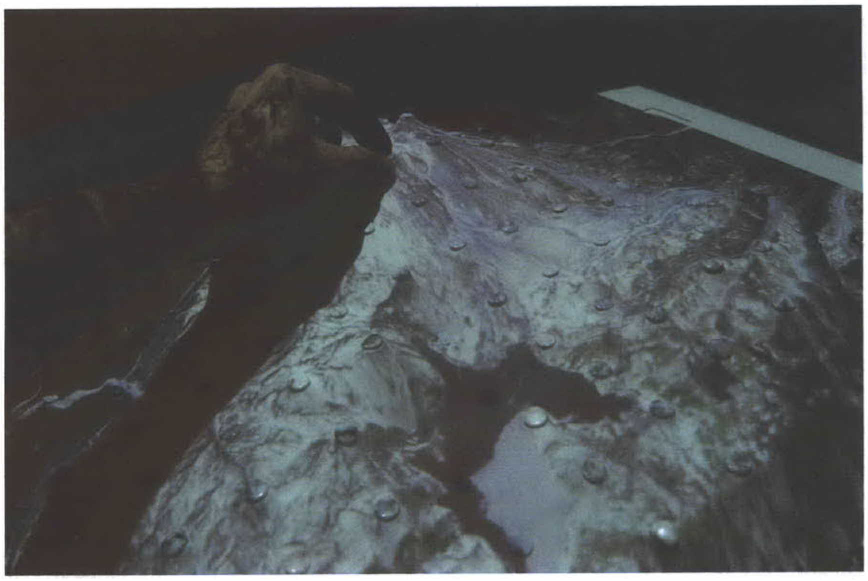


Figure 3-5: Mockup with fabric cover and projected landscape.

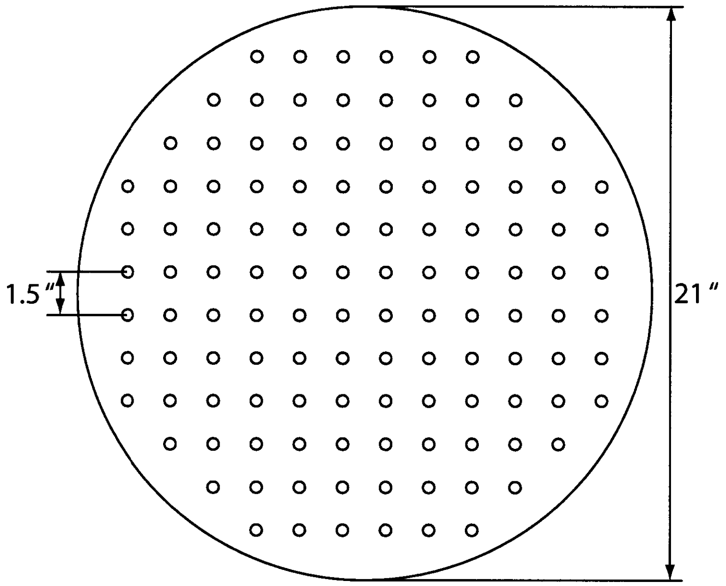


Figure 3-6: Arrangement of actuators in the Relief tabletop.

slider reaches the final position. To provide feeback to the software about user input, the microcontroller also sends the current actuator position to the computer.

We decided to utilize Arduino Pro boards with ATMega 328 Microcontrollers running at 16 Hz. These microcontrollers provide 6 analog inputs. 2 of these inputs are used for communication with the computer, therefore 4 motorized slide potentiometers can be controlled from each Arduino board. Each board is able to control more potentiometers if the analog inputs are multiplexed, but we decided not to use a muliplexer for a tight feedback loop between position sensing and motor control. The Arduino boards drive the motors through motor shields by Ladyada. While the Ladyada motor shields provide more functionality than required for the project, their low cost, ease of use and reusability were key arguments for utilizing them.

Figure 3-7 depicts the schematic of the actuation and control unit consisting of an Arduino board (a), motor shield (b) and motorized potentiometer (c). Figure 3-8 depicts a photograph of an assembled unit with Arduino, motor shield and 4 potentiometers build into a plywood box.

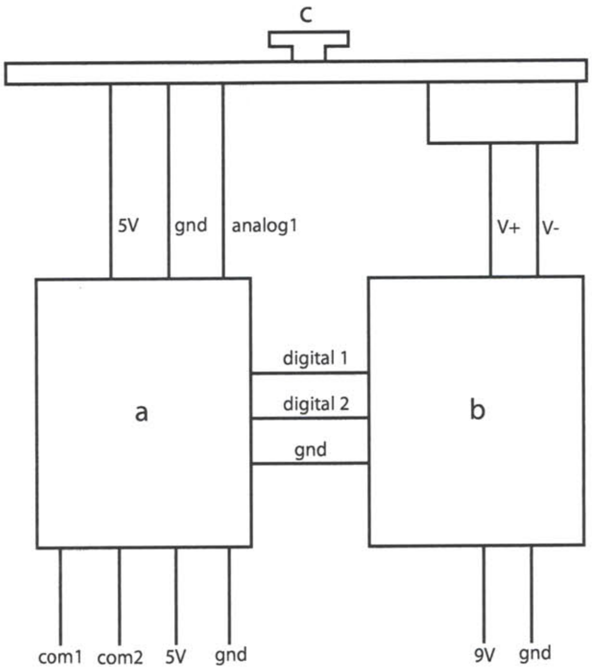


Figure 3-7: Schematic of acutation unit consisting of Arduino board (a), motor shield (b), and motorized potentiometer (c)

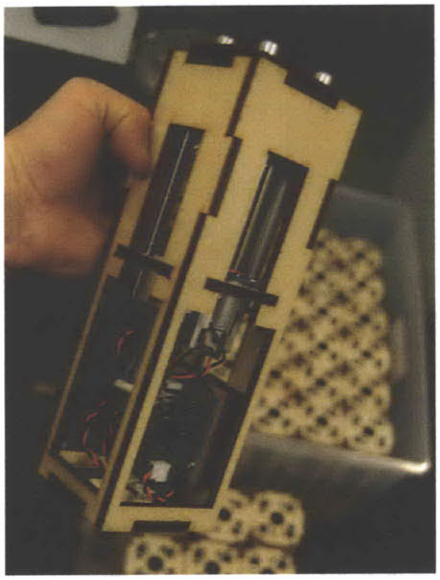


Figure 3-8: Photo of assembled actuation unit.

3.1.3 Communication

As each Arduino Pro controls up to 4 actuators, the application needs to communicate with a total of 32 Arduinos. The built-in modes of communication provided by Arduino are serial and I2C. To communicate directly with a computer over USB, an FTDI chip connecting to the serial interface of the board is commonly utilized. However, 32 individual USB connections were deemed too complex, therefore two different ways of communication were implemented. The first one is a chain of serial connections, which worked but was eventually abandoned due to high latency and unstable connection. The second one is utilizing I2C buses, which provide a sufficient speed and connection quality without the need for additional hardware.

Serial Connection

The serial connection is a daisy chain of modules, which pass messages to each other. A schematic of this implementation is depicted in Figure 3-9. All units (a, b, c) are connected in a serial chain to forward messages to and from the computer (e) through an FTDI adapter(d). The units receive unique addresses with a running counter message when the system starts up to enable addressing each pin individually. After startup, each module analyzes messages it receives from the computer and either interpret it or forwards them to the next module. If the module interprets the message, it replaces the desired actutator position with the current actuator position. While the serial chain is easy to implement, the length of the chain adds a noticeable delay to the communication between the computer and the Arduinos. Therefore, we chose to switch the communication to I2C.

I2C (Two Wire Interface)

The ATMega 238 provides a hardware interface for I2C (called two-wire interface by ATMEL). This interface is available through the Wire library on Arduino and therefore easy to implement. I2C is a master/slave bus system with a connection speed of 100 kbit/sec.

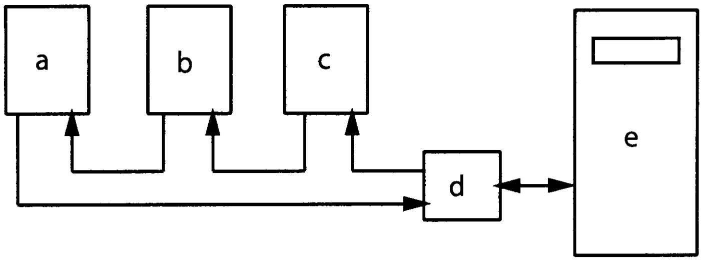


Figure 3-9: Arduino serial chain schematic: All units (a, b, c) are connected in a serial chain to forward messages to and from the computer (e) through an FTDI adapter(d).

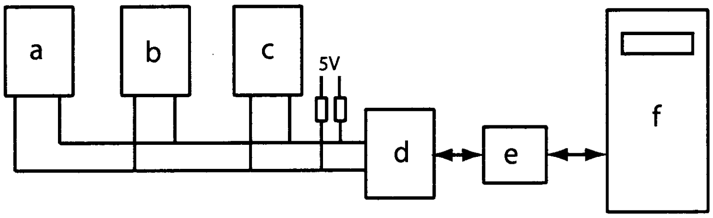


Figure 3-10: Schematic of I2C bus connecting Arduino slave modules (a, b, c) and computer (f) through master module (d) with FTDI chip (e)

One of the main limitations of I2C is the bus length, limited to 9 feet. Therefore, we chose to implement three independent buses with one Arduino master board and 10 - 12 slaves modules for each bus. Figure 3-10 depticts the connection schematic three slave units. The Relief application running on the computer (f) communicates with the master board (d) over USB through an FTDI chip (e). After receiving a command from the application, the master board then initializes communication with each slave (a, b, c), sends the desired actuator positions to them and receives their current actuator positions, which it forwards to the computer.

3.2 Tabletop Setup

The actuation hardware is built into a table with a circular top. The table contains all actuation and control modules (a - g), an optical mouse (i), usb hub (h), and two power supplies(j, k), as depicted in Figure 3-11. Due the heat generated by the motor and their driver chips, sufficient cooling of all modules is necessary. A combination of fans (l) with an air channel design allow for constant flow of air.

The table is constructed entirely from plywood. Using a Shopbot and laser cutter to create all table parts allowed for rapid prototyping of the individual parts (see Figure 3-12).

Around the circular tabletop, an articulating ring provides an additional input parameter for the user, which can be mapped to functions explained in the Interaction Technique Section. An optical mouse connected to the computer, as depicted in Figure 3-13, measures the relative ring rotation.

3.3 Fabric Surface and Projection Setup

As the individual aluminum rods protruding from the table surface are spaced at a distance from each other, the actuation apparatus does not generate a connected shape, as depicted in Figure 3-14. Therefore, the array of rods is covered with a white 4-way stretch spandex fabric, which forms a tensile structure to give the illusion of a continuous shape to the user. The fabric is augmented with graphics by a ceiling mounted projector (see Figure 3-2). Figure 3-15 depicts the pins covered with the fabric and augmented by projection to depict a landscape.

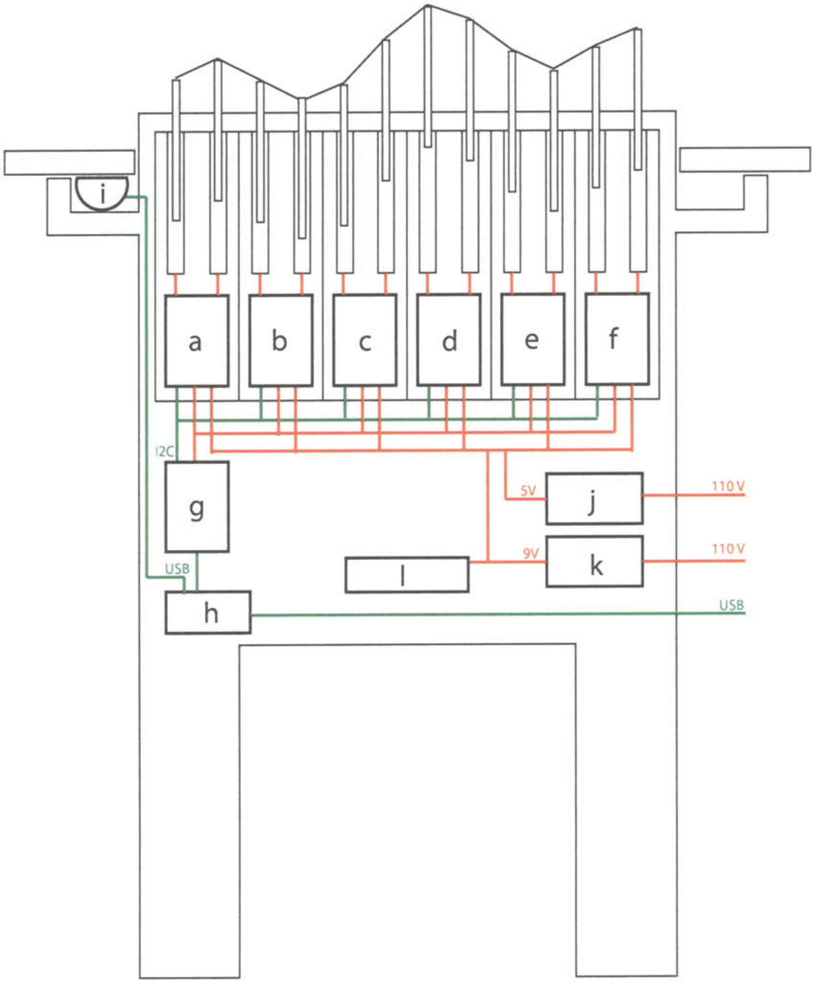


Figure 3-11: Schematic of the relief table with actuation and control unit, consisting of actuation and control modules (a - g), an optical mouse (i), usb hub (h), and two power supplies(j, k).

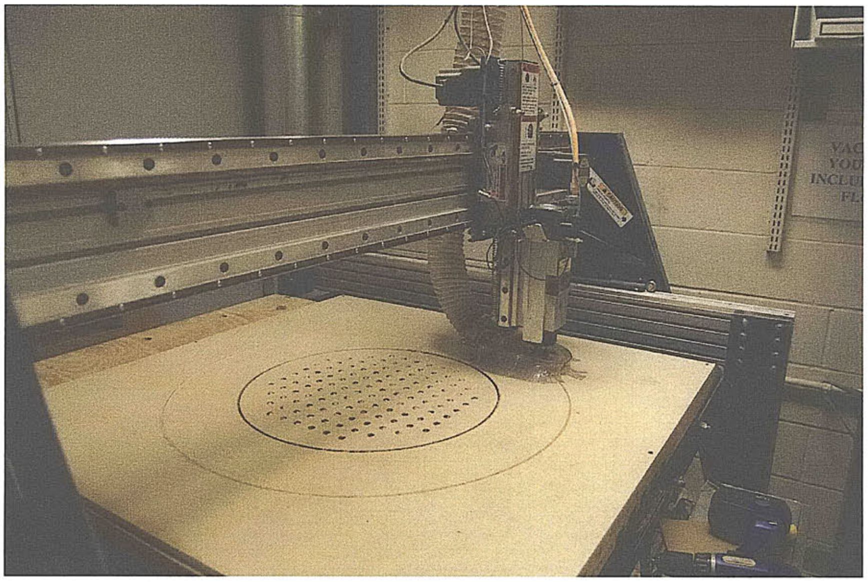


Figure 3-12: Shopbot cutting tabletop from plywood.

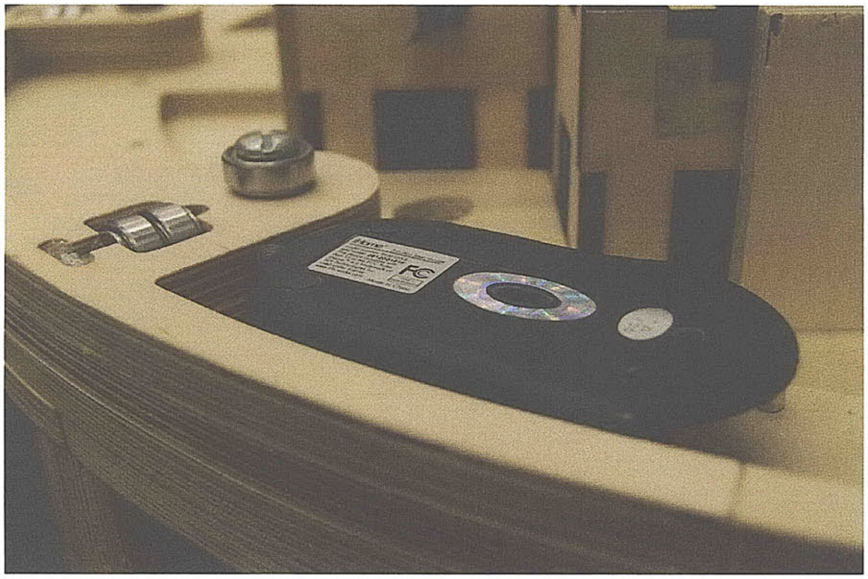


Figure 3-13: Optical mouse for ring input and ring bearings.

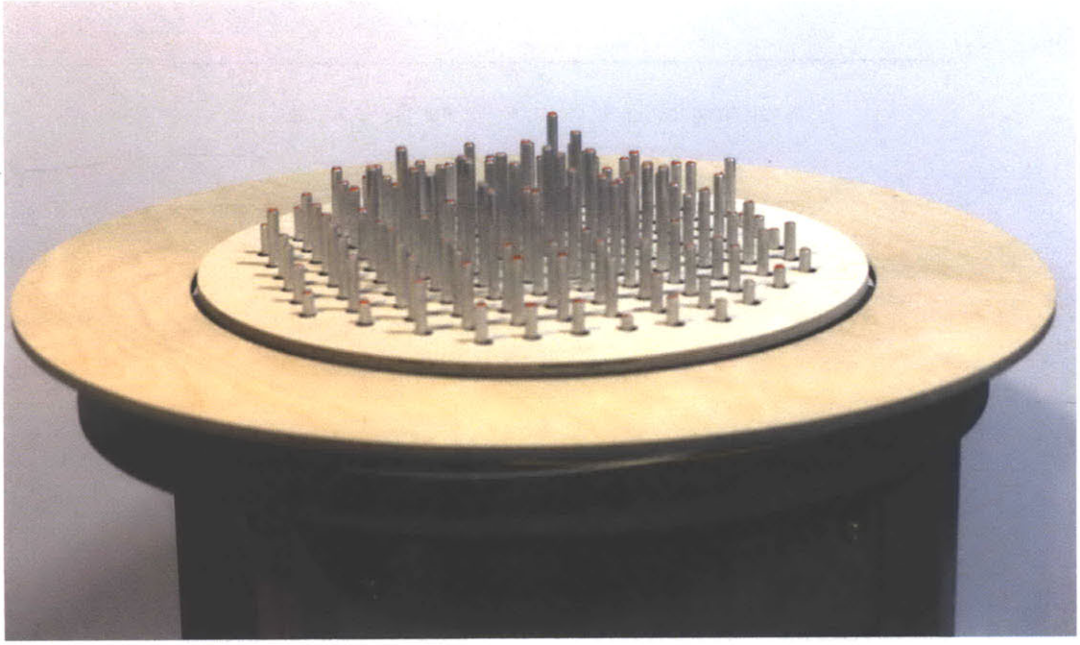


Figure 3-14: Table with bare aluminum rods.

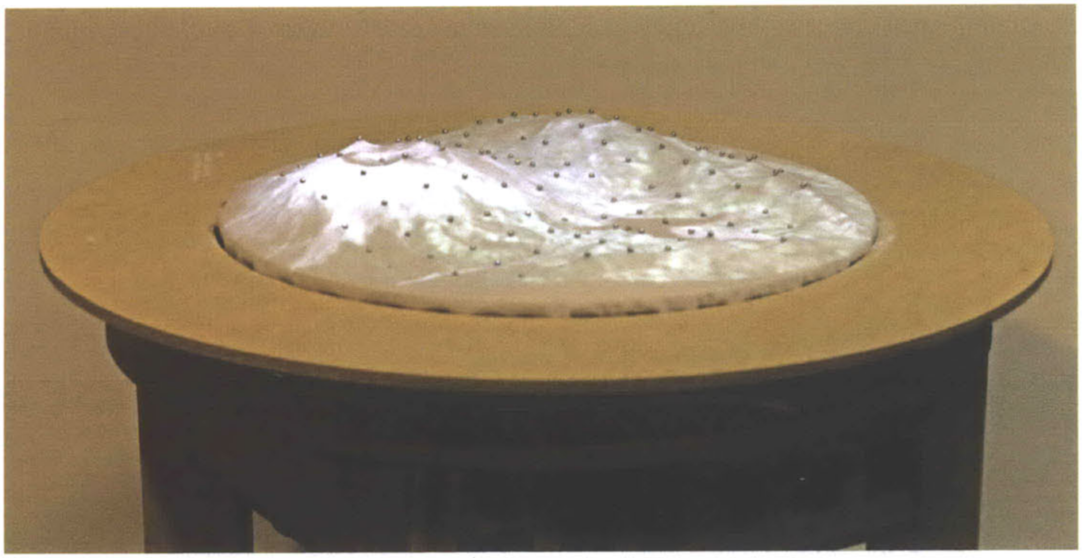


Figure 3-15: Rods covered with spandex fabric and augmented with projected graphics.

Chapter 4

Software Implementation

The actuation apparatus of Relief is able to render shapes similar to the 2.5D shapes a 3-axis milling machine can produce. While the generated shapes are three-dimensional, no overhangs can be generated. Therefore, a relief interface can only render a limited set of geometry accurately, a problem demonstrated by the example in Figure 4-1. In addition, the range of the actuated rods has to be considered. If the dimensions of a 3D model exceed the actuator range, parts of the rendered shape will be cropped and appear flat, as depicted in Figure 4-2. The rendering pipeline has to account for these particular hardware characteristics. The next section introduces our approach to create such a pipeline.

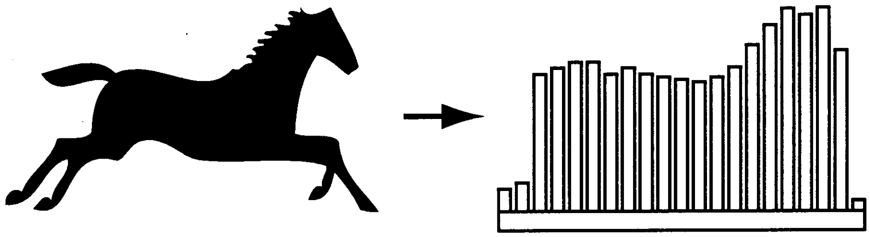


Figure 4-1: Relief interface rendering 2.5D approximation of 3D geometry.

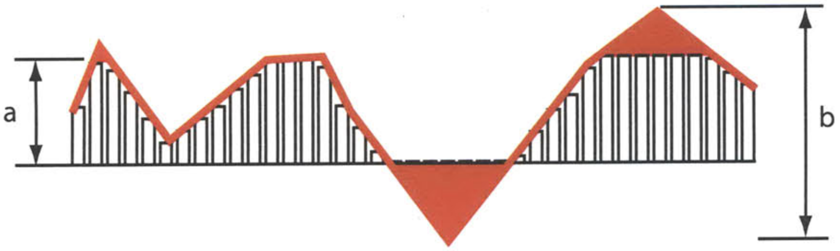


Figure 4-2: Rendering of terrain: The actuator range (a) is clipping terrain geometry (b). Clipped regions are highlighted red.

4.1 Rendering and Sensing Pipeline

The current implementation of the renderer loads either 3D models in 3DS format or surface meshes stored as height maps. Figure 4-3 depicts the software pipeline, which consists of the following steps:

(a) To render shapes with the Relief Interface, a height map of the shape is stored in a PNG file and loaded by the application. The height of each actuated rod of the hardware corresponds to a pixel in the height map image, with the height of the rod encoded in the pixel brightness. In addition to the height map, texture image files are loaded by the software and projected onto the hardware interface. By loading a series of corresponding height maps, animated shapes can be rendered. The application can also load surfaces meshes as PNG images, with the height value stored as greyscale brightness values. For this mode, the references to the height image, additional textures, and the height ratio of the stored values is loaded from an XML file.

(b) While rendering height maps is the most straightforward way to interface with the actuation hardware, it limits the content a programmer can utilize on the display. Even with the previously described hardware limitations, loading real 3D geometry and rendering a 2.5D approximation of it provides additional flexibility for application creation. This functionality is part of the Relief rendering pipeline. For 3DS models, the model loader provided by OpenFrameworks is utilized.

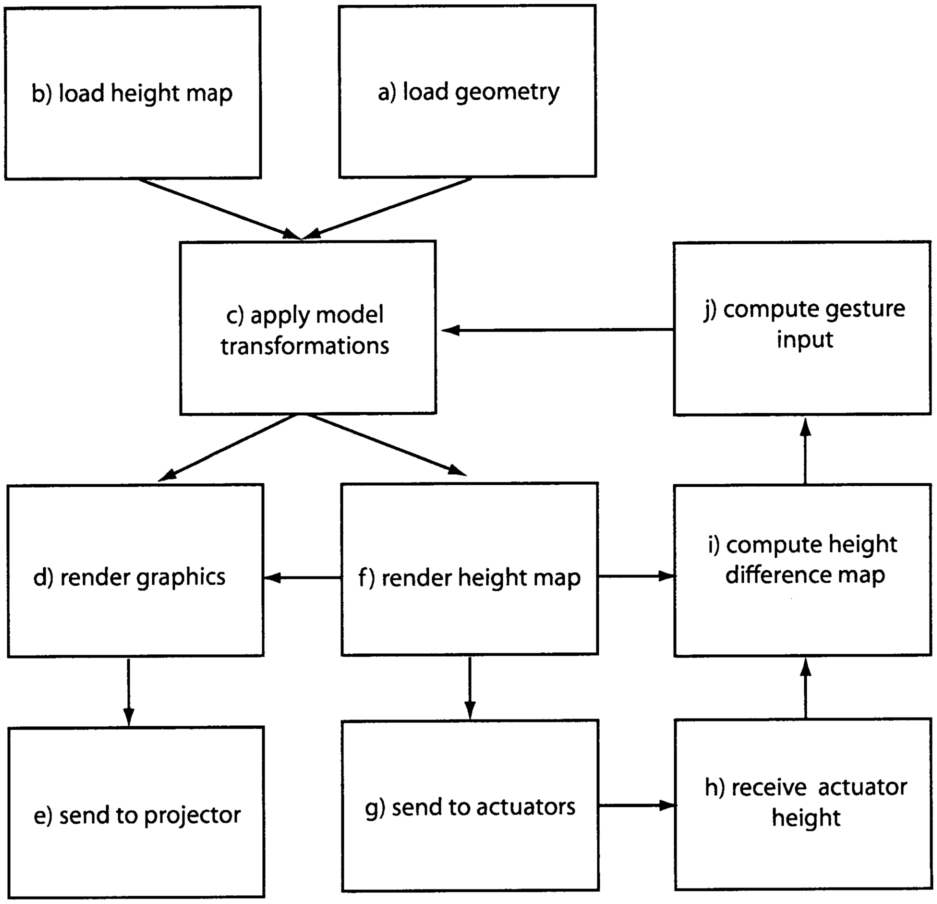


Figure 4-3: Relief software pipeline. The individual modules are described in detail in Section 4.1.

(c) Apply model transformations like moving, scaling and rotating a whole model or individual vertices of geometry.

(d) Rendering the graphical output of the 3D model. The 3D model is rendered from the approximate viewport of the projector. In a GLSL shader, the rendered output is combined with the previously generated depth map (f). Pixels exceeding the range of the actuators are highlighted in red to provide feedback about the systems inability to render their shape correctly.This pipeline can also be utilized when loading height meshes as described in the previous section. In this case, the height map is rendered as a 3D model and all subsequent steps applied. The advantage of this approach compared to sending the height map values to the actuation apparatus directly is a simplified way for programmers to apply transformations like scaling, translating and rotating.

(e) Send the graphical output to the projector. By projecting it onto the surface of the relief, it augments the physical shape output with texturing and shading. In addition, the graphical output provides visual feedback of exceeded actuator range to the user.

(f) Render a height map of the 3D model with all transformations applied from an orthogonal projection. The output is rendered into a frame buffer object. A GLSL shader renders the linearized depth buffer values into a depth map. Each depth value is also compared with a near and far plane, which are parallel to the projection plane and represent the maximum range the actuators of the relief interface can cover. The depth values within the range of the interface are stored in the red component of the frame buffer object, the values exceeding the range are stored in the blue component for values too far away and in the green component for values too close.

(g) Sending the depth values of the frame buffer object to the hardware interface to update the height of the actuators. For all values exceeding the actuator range, the pins are set to their respective maximum or minimum values.

(h) An individual thread receives the current actuator height from the hardware interface.

(i) Compare the two-dimensional array of the height map sent to the hardware interface (f) and actual actuator positions received from the interface (h). From these two arrays, a height difference map is computed.

(j) Simple computer vision algorithms can be applied to the height difference map to localize user input. This input is then applied to the model transformations (c).

4.2 Software Frameworks in Different Programming Languages

Two software frameworks were implemented to provide greater flexibility for application developers. Version 1 utilizes Processing and allows for simple implementation of quick prototypes, while version 2 in OpenFrameworks has a larger set of features with a full rendering and gesture input pipeline and is intended for more mature applications.

4.2.1 Version 1 with Processing

Processing is an open source framework for visual designers and artists [48]. It is built on top of Java and provides simple routines for rendering, window handling and networking. The first software version to control Relief was built around the Processing framework (version 1.0.9). This version does not use individual threads to communicate with three i2c buses of the hardware interface and is therefore slower. Due to the simple coding environment, it is still in use for a quick implementation of applications and to prototype interactions where performance is not critical.

4.2.2 Version 2 with OpenFrameworks

OpenFrameworks is an open source C++ toolkit with a similar target audience as Processing [38]. While the framework requires more programming effort than Processing, it features a greater number of available libraries and plugins, as well as faster rendering. The OpenFrameworks version of our software has all features described in this thesis implemented and is the primary development environment. The applications described in the following section are created with this version.

Chapter 5

User Interaction

To create a compelling user experience, relief interfaces not only need to render physical shapes, but also enable the user to interact with these shapes in a natural manner. This chapter describes the motivation for direct input on the interface surface, lists previous systems with similar interactions and proposes a set of gestures with their initial user reactions.

5.1 Disadvantages of External Controls

The need for direct controls on the interface became apparent when testing the first prototype of our system with external controls, which were mapped to physical shape transformations. By pressing buttons on an attached keypad, users could move, rotate and scale a shape rendered on the interface. As the mapping between the key press and the operations was abstract, users complained about the experience being unintuitive. An additional problem is the interface orientation. The tabletop is acessible from all sides, therefore the cognitive load increases for the user when moving around the table while trying to control object transformations with the keypad. When multiple users interacted with the system simultaniously, users touching an object would get confused if another user transformed the object with the keypad at the same time. A set of rotation invariant gestures directly on the interface solves these problems, as the input is collocated with the output and interactions are legible to other users.

5.2 Previous Systems with Gesture Interactions

Feelex by Iwata et al. is equipped with pressure sensors in the malleable surface of the interface [26]. These sensors allow users to experience reactive behaviour. In an artistic application, a moving creature is rendered on the interface as a physical shape with overlayed graphics. Users can influence the motion by touching the interface. No further gestural input is explored with Feelex. Similar reactive behaviour is implemented with capacitive touch sensors in Lumen by Poupyrev et al. [44]. These sensors are also utilized in applications to render dynamic physical buttons and enable remote haptic communication. While capacitive sensing is capable of detecting the location of touch, it does not measure pressure. Therefore, the applications proposed for Lumen do not allow haptic exploration of a shape without simultaneous input on the system.

A pressure-based sensing technology is able to make such a distincion by introducing an additional sensing dimension. Users can experience a shape by touching the surface and interact with the system by pressing onto it.

3D input through pressure-based gestures is explored by Watanabe et al. with the deformable workspace [56]. While this system enables users to transform and create content through a set of gestures, the interface is not actuated and the output is rendered in 2D. Gestural input on a passive interface is also explored by Overholt's Matrix (Multipurpose Array of Tactile Rods for Interactive eXpression) for music creation [39].

5.3 Pressure-based Gestures

We propose two sets of gestures, which are based on pressing onto the surface of the relief interface. The first set allows users to modify the geometry of an object's surface, similar to a sculptor deforming clay. These gestures are called *sculpting gestures.* Gestures to change the view transformation are called *view changing gestures.* These two gesture sets can not be used at the same time, as the sculpting gestures are inspired by physical interaction with a clay-like material. For this interaction, the design goal was allowing input on the entire interface surface, which conflicts with other gestures. The current applications on the interface only use either one of these gesture sets.

These proposed gestures were developed for the particular sensing hardware of our prototype system, but we believe the discussion of underlying principles of pressure-based gestures are of use for any similar relief interface. We also describe the particular limitations of pressure-based gestures and our proposed solution of providing additional input through a revolving ring controller.

5.4 Sculpting Gestures

By deforming the shape of fabric surface on the relief interface, a user can sculpt arbitrary surfaces in 3D space. Table 5.1 lists all sculpting gestures with graphic descriptions and user feedback, which is explained in more detail in the following paragraphs.

We tested the interface with users informally by extending all actuators to their highest position and observing users approaching and trying it without prior instuctions. As the projected graphics provided feedback about the height of the virtual model, pushing down on the surface did not require an explanation of the mapping between interface shape and virtual shape. Users immediately started to adopt strategies like pushing down large areas with their whole hands or objects like books to form rough shapes and pushing down individual areas with their fingertips to define more detailled shapes. However, at some point every user wanted to be able to rise indivual areas of the surface. Some users also wanted to reset all actuator positions to their highest point to start over again.

To provide this functionality, we experimented with spherical handles to pull up individual control points on the surface. Users did not feel comfortable with this interaction, as pulling

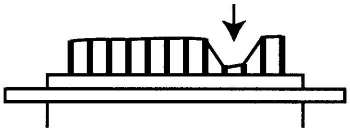
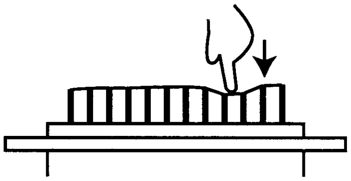
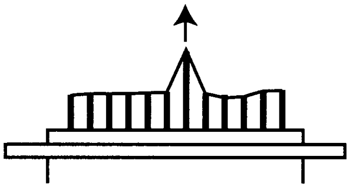
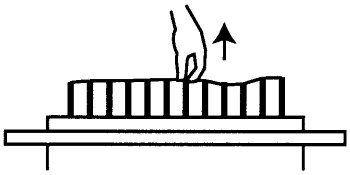
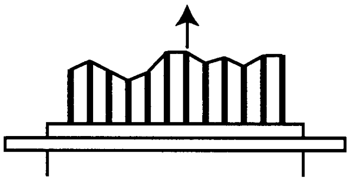
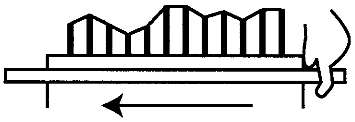
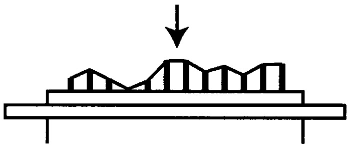
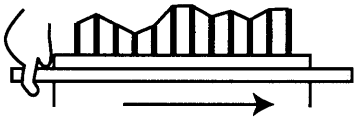
Operation	Gesture	User Feedback
 Deform surface	 Push down on surface	Self-explanitory Perceived intuitive
 Rise parts of the surface	 Pull on surface handles	Tried by some users Perceived cumbersome
 Rise whole surface	 Rotate revolving ring left	Multiple tries Intuitive
 Lower whole surface	 Rotate revolving ring right	Multiple tries Intuitive

Table 5.1: Sculpting Gestures on Relief.

up 120 pins was perceived tedious. In addition, the handles distracted from the surface shape and the projected graphical feedback.

Another option tested was a conservation of mass mode, which starts wtih all actuators extended half way. The combined actuator height always stays the same. When one actuator is pushed down, all other actuators rise in height accordingly. However, this approach limited the shapes, which could be generated by the interface and was therefore dismissed as well.

Therefore, we resorted to use an additional input with a revolving ring around the tabletop surface. The current option to raise the surface is a *balloon mode*, which is controlled by rotating the ring. In connection to the ring rotation, the whole surface inflates (rises) or deflates (lower) similar to a balloon. This interaction has the disadvantage of being a global operation, which affects the whole surface. Therefore, no single position can be modified as with the pulling handles. However, having to push down on the surface repeatingly was tolerated and actually enjoyed by test users, as they commented that the mapping of ring rotation to global surface height was fun and engaging. This feedback is supported by the observation that users often started playing around with the revolving ring before starting with a model. While most users had to rotate if a few times before perceiving the mapping, all of them understood and actively used the global height change feature. None of the users sculpted models longer than 10 minutes and most of them were novices to 3D CAD software. Therefore, a further evaluation is required before being able to state how useful the proposed sculpting gestures are for expert users.

5.5 View Changing Gestures

As the actuation apparatus of our relief interface is limited in size and resolution, the ability to change the view of rendered 3D geometry is crucial. This need is apparent in the landscape application described in Chapter 6, where the topography appears flat when displayed at a large scale. To be able to view both an overview and details of the landscpape, zooming is required. In a detailed view, the ability to pan is necessary. When

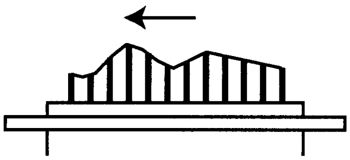
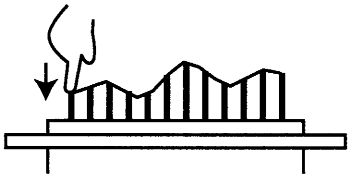
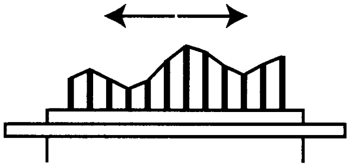
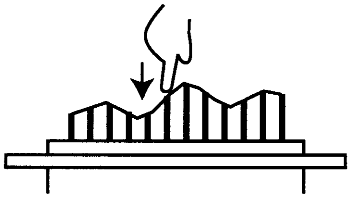
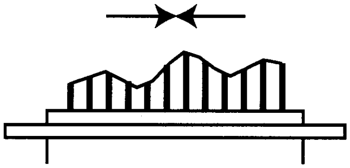
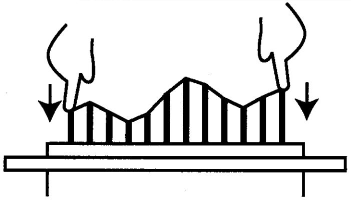
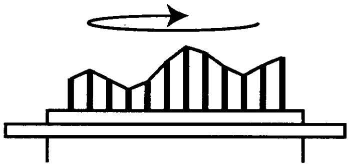
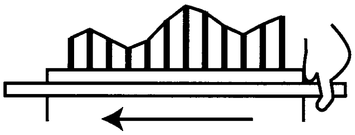
Gesture	Operation	User Feedback
 Pan	 Push on surface edge	Self-exploration Perceived intuitive
 Zoom in	 Push on surface center	Self-exploration Perceived intuitive
 Zoom out	 Push down on two edge regions	Required explanation Perceived intuitive
 Rotate	 Rotate revolving ring	Self-explanitory Perceived intuitive

Table 5.2: View Changing Gestures on Relief.

multiple users stand around the interface, rotating the landscape enables all of them to view it from different sides without having to physically move around it.

To support these interactions, we implemented a set of view changing gestures, which are depicted in Table 5.2. To move the rendered shape on the interface in a direction, the user pushes on the edge of the interface surface, which consists of a ring of 32 actuators. The shape moves in the direction of a vector defined by the pressing location and the surface center.

When pushing on any pin which is not part of the interface edge, the shape with grow, analog to zooming in the view. To shrink the shape, and zoom out, the user pushed on two oposing edges of the surface. We stuggled to find an intuitive pressure-based gesture for rotation, and therefore mapped this action to the revolving ring around the tabletop. Rotating the ring also rotates the shape in the direction of the ring movement.

Initial Feedback of these view changing gestures was gathered during various presentations of the system to collaborators and visitors to our laboratory. When not provided with any explanation of the gestures, users would first start to rotate the ring around the surface in both direction. After that, most user would touch the surface while rotating to feel the change in shape. Some users would start to randomly press down on the interface and find out how to zoom in. Pushing on an edge pin to pan was another commonly discovered interaction. However, pushing down two opposing edges simultaniously to zoom out was seldomly discovered without explanation. Once the gesture was described to them, users immediately adopted it and started to use all four gestures with ease. Comments on this interaction was very positive, with the proposed gestures being described as intuitive and effective. However, a formal long term study with the interface is required to see how effective the proposed pressure-based gestures are.

Chapter 6

Applications

The following applications were developed to explore how users perceive and interact with transforming shapes. The *geospatial exploration* application allows users to explore landscapes by zooming, panning and rotating a physical terrain on the table surface. With the *surface modeling* application, users can sculpt 3D shapes by deforming the interface surface. The *multilayer viewing* application blends multiple 2D image layers with each other as defined by a non-planar cut-section. Modifying the shape of the interface surface allows users to view independent layers or blend multiple layers with each other.

6.1 Geospatial Exploration

As noted by Imhof, geospatial representations of terrain correspond to an imagined model formed in the users head [21]. Maps aid the user's perception of a three-dimensional shape through a range of visual effects like shading and coloring and contour lines. However, as Faulkner suggests, physical terrain models are easier to comprehend by non-experts and aid in group discussions [10]. This is supported by a study conducted by Tory et al. to compare visualization task performance with various two-dimesional and three-dimensional display types [53]. The disadvantages of physical terrain models are bulkines and production effort.

To explore a dynamic physical terrain model, we built the geospatial exploration application, which is able to present physical representations of different terrains at multiple scales. The rendered shape is augmented with top-mounted graphics like contour lines and satellite imagery. This approach is similar the tangible geospatial modeling environment Illuminating Clay presented by Piper et at. and further investigated by Shamonsky [42], [51].

The actuation hardware creates the shape of the landscape, which is overlayed with projected satellite imagery or contour maps. Users can move and scale the landscape by pressing onto the shape and rotate it with the articulating ring.

6.1.1 Content

Terrain data is stored in a height map and loaded at the start of the application, together with a corresponding satellite image and a contour map. The satellite and contour data-sets where downloaded from Google Maps and stitched in Adobe Photoshop. Using google maps allows a comparison between the relief interface and standard interface hardware combined with commercional desktop and web software.

6.1.2 User Interaction

This application utilized the view changing gestures described in the previous Chapter to move, scale and rotate the rendered landscapes. Users press down on an edge of the screen to move the terrain, as depicted in Figure 6-1. Pressing any center pin of the interface, zooms in at the terrain, as depicted in Figure 6-2. To zoom back out, the user presses two pins at the edge of the interface, as depicted in Figure 6-3.

6.2 Surface Modeling

The surface modeling application converts a shape a user models on the interface into a 3D model and projects feedback about it onto the surface. This setup was inspired by

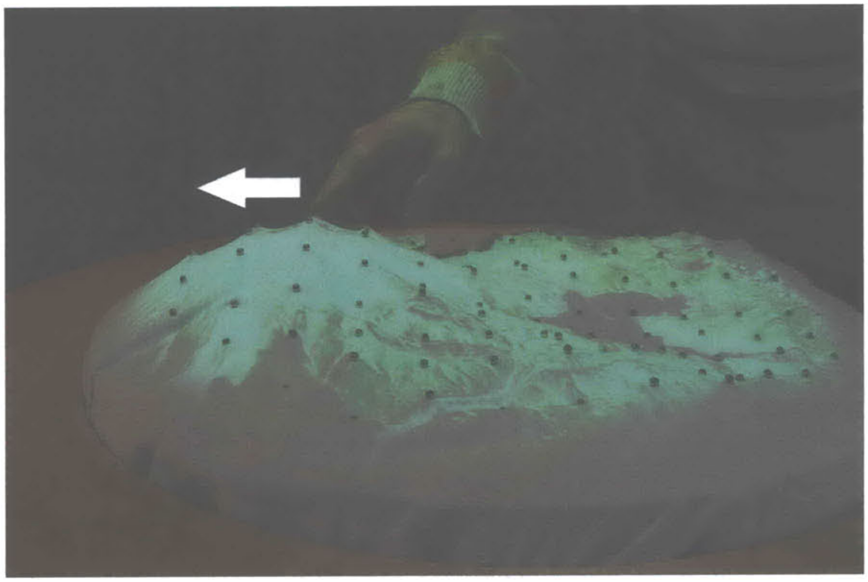


Figure 6-1: Moving terrain by pressing on an edge pin.

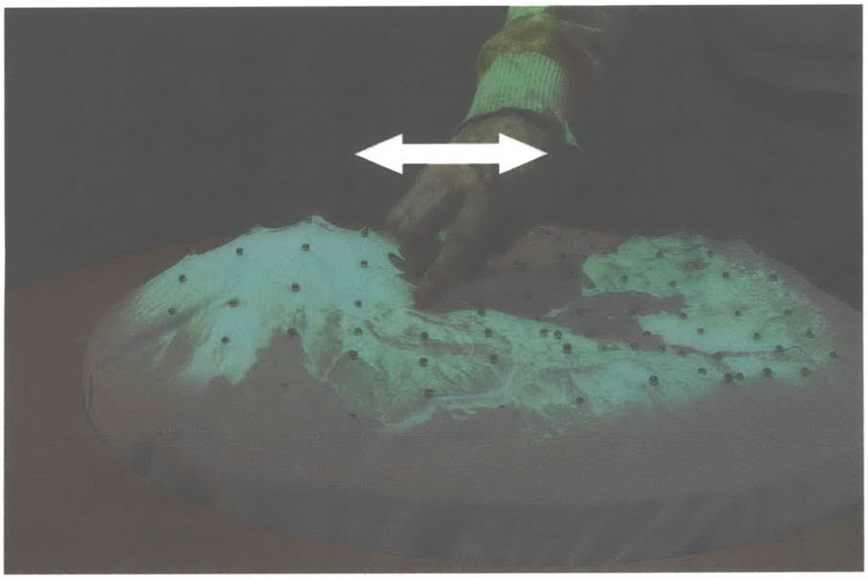


Figure 6-2: Scaling terrain by pressing onto a center pin.

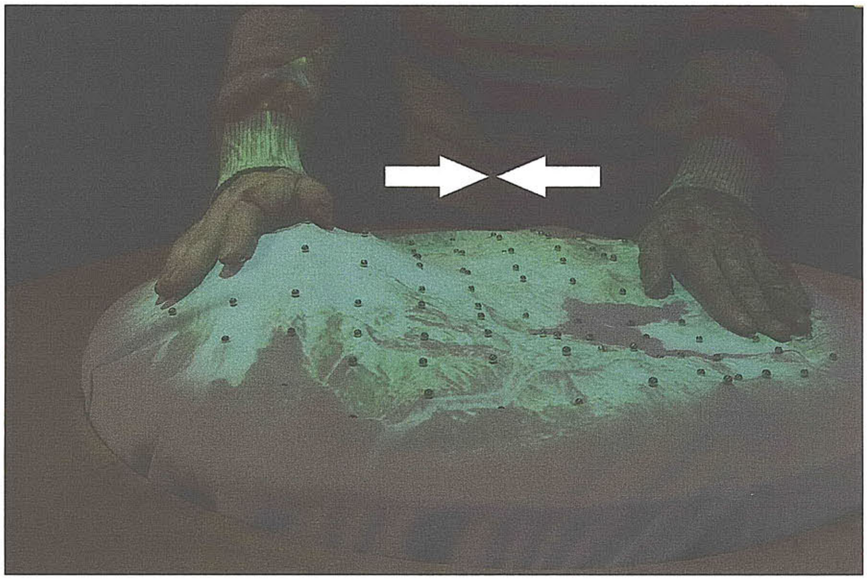


Figure 6-3: Scaling out terrain by pressing onto edge pins.

Illuminating Clay by Piper et al. [42]. However, the interaction is different due to the materials utilized. While Illuminating Clay allows users to remove and apply clay or sand to the surface, the only interaction on the Relief interface is pushing down on the surface to deform it (see Figure 6-4). Therefore, we implemented a set of gestures as discussed in Section 5.4.

6.3 Multilayer viewing

The multilayer application allows users to view correlating images arranged in virtual layers (see Figure 6-5). The graphics projected onto the interface are defined by a non-planar cut section, defined by the deformable surface. To shape the surface, the user interacts in a way similar to the surface modeling application, described in Section 6-4. Figure 6-6 depicts a user defining the cut section by pressing on the surface to view multiple layers. An application for the multilayer viewing is browsing MRI data, as proposed by Hinckley et al. [18]. While the interface proposed by Hinckley is limited to planes, the cut sections created with the relief surface can be arbitrary, similar to Phoxel-Space proposed by Ratti

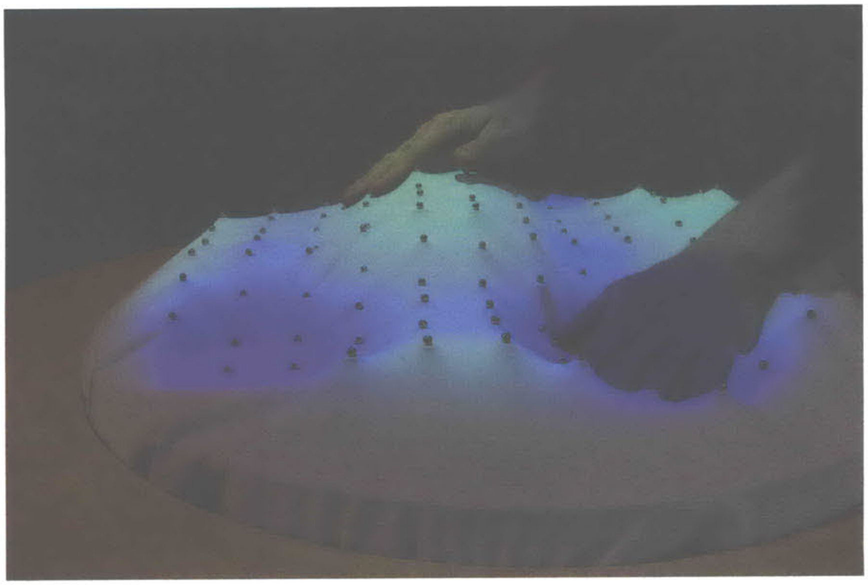


Figure 6-4: Forming a three-dimensional shape by pressing onto the interface.

et al. [47]. As the surface is actuated, the physical state can be linked to a digital state like a saved surface or allow for global operations such as raising the whole cut section uniformely.

The images arranged in layers can also time-correlated to blend video frames into a single output image, similar to the interaction proposed by Castellini et al. in Khronos projector [6].

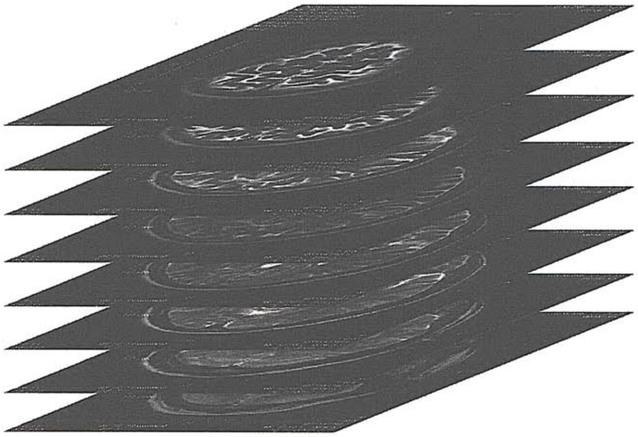


Figure 6-5: Virtual layers stacked on top of each other.

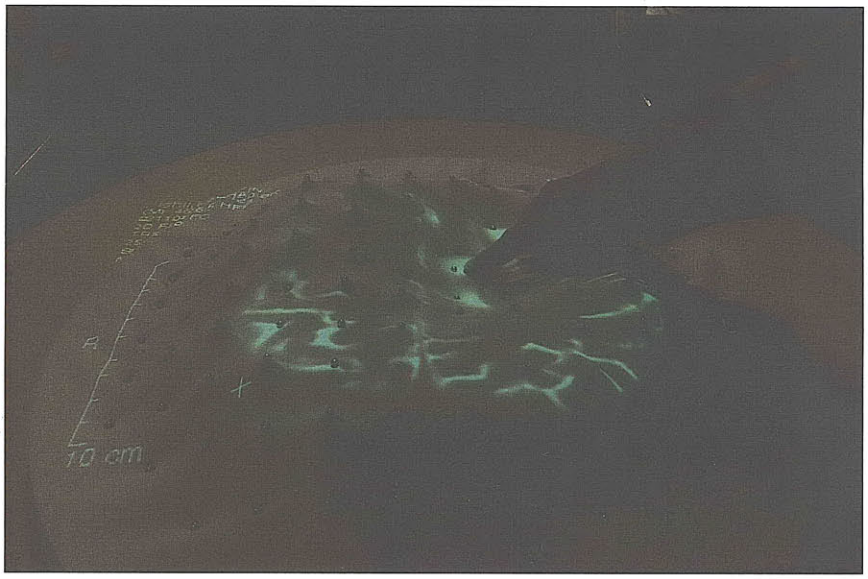


Figure 6-6: Defining a cut-surface to view multiple layers simultaniously.

Chapter 7

Evaluation

While a formal user study of Relief has not been completed yet, the system has been tried by public audiences on several occasions. In this section, we describe the observations made and feedback gathered during these events.

7.1 Ars Electronica Festival 2009

The first public demonstration of our relief interface was during the Ars Electronica Festival from September 3 to 8 2009. The interface was showing an abstract wave animation, which could be controlled by rotating the wooden ring around the interface. Technical problems with the actuation modules led to breaking of individual microcontrollers when pressing down on the interface too forceful, therefore the relief interface had to be covered with plexiglass after the first day of the exhibit (see Figure 7-1). However, even though they could perceive the shape change only visually, visitors where intrigued by the aesthetics of the virtual waves deforming the surface. The interface was intentionally left unattended during extended periods to observe user reactions. A common behaviour found with most visitors of the exhibit was a frequent change in viewing position like walking around the moving shape and bending to see it at eye level.

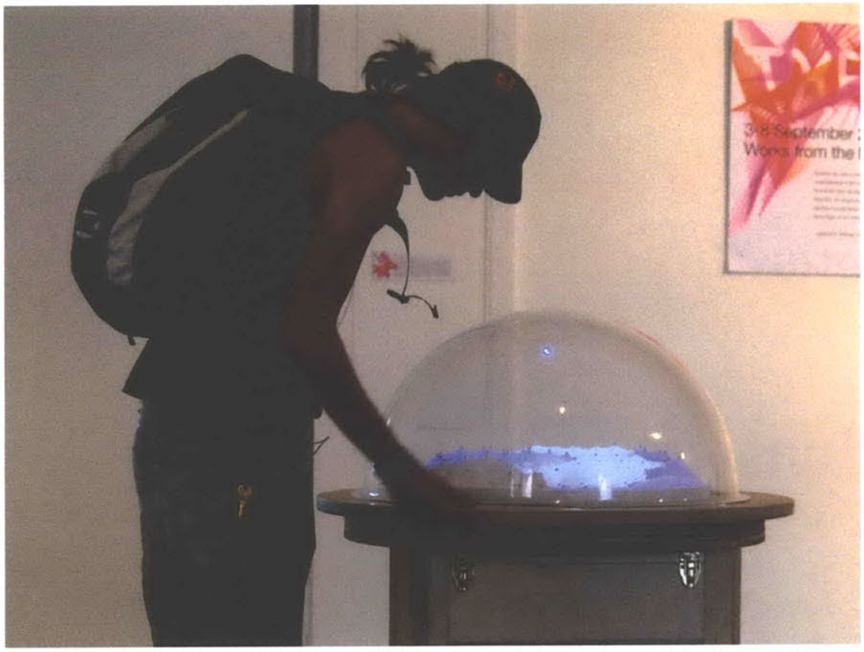


Figure 7-1: Plexiglass protection during Ars Electronica 2009.

7.2 Media Lab Sponsor Week Fall 2009

During the Media Lab Sponsor week from October 13-15 2009, Relief was explained to and tried by approximately 100 visitors. As this version of the hardware allowed direct touch interaction, visitors were able to view landscapes and resculpt them using their hands. A landscape could be resculpted by pushing down on the surface or by pulling on magnetic handles on top of the actuated surface. As pulling the handles was deemed too cumbersome for moving multiple actuators during informal testing, the interface also featured a *balloon mode*, which pushed up all actuators by rotating the wooden ring around the tabletop. The disadvantage of this mode is that it is a global operation, which affects the whole surface. Therefore, no single pin can be modified as with the pulling handle. However, it was prefered by the majority of users, even when only wanting to modify the position of a single actuator. As pushing down pins proved easier than pulling them up, users would change the elevation of all pins until the pin they wanted to modify was at the desired location. As a next step, they would push down all the other pins to their initial position from before the operation.

7.3 TEI 2010 Demo

A demonstration of Relief was given to approximately 50 visitors on January 25 2010. 20 of them briefly tried out the interface themselves. For this version, the geospatial exploration mode was demonstrated with zooming and moving implemented. To zoom into the geometry, users could rotate the revolving ring, pressing buttons on an external keypad would move the terrain. While user reactions to the interface were very positive, the cropping of terrain due to the limited actuator range proved problematic. Also, users suggested that a feature to rotate geometry would be useful, as multiple users around the table would prohibit individuals from moving around to see the terrain from all sides. These suggestions were implemented in the current version of the geospatial application, which is currently being evaluated.

7.4 TEI 2010 Tabletop Studio

A one day workshop with 5 participants at a studio session on interactive tabletops on January 26 2010 gave us an opportunity to demonstrate the hardware and software framework of Relief to practicioners in the HCI community [27]. The feedback about the technical implementation of Relief was very positive, with attendants noting that they would feel comfortable implementing a similar system themselves after the session. The possibility to use widely available hardware components and open-source software was especially appreciated by studio participants.

7.5 Geospatial Evaluation

We are currently working on an evaluation of Relief with the geospatial application to demonstrate the value of direct input on the surface and on the benefits of transforming a phsical shape to render landscapes.

Chapter 8

Conclusion

This thesis has presented a novel relief interface for fast rendering of physical shapes through a malleable surface. The shape output is complemented by precise sensing of pressure input through the surface. We have also presented a set of pressure gestures appropriate for direct interaction with relief interfaces. The decision to utilize open-source hardware instead of making custom circuit boards proved very helpful in the creating of our setup, as it allowed for greater flexibility in modifications and valuable community support. We hope to have a chance to conduct workshops with HCI practicioners to explore the community aspect in actuated interfaces further. While the formal evaluation of the interface has not been completed at the time of writing, hundreds of users tried our prototype during public demonstrations and were very enthusiastic about the future potential of relief interfaces. The hardware resolution and size of the display are currently limiting factors in generating a large range in representations of virtual shapes, but proved sufficient for certain application domains like geospatial exploration and form-giving.

Chapter 9

Future Work

As the current prototype received very positive feedback from users, we are currently working on an in-depth user study to investigate advantages and limitations of the interface when representing three-dimensional shapes.

Due to the spacing of the actuators, the interface can only render rough approximations of shapes. We are currently evaluating the cost and technical feasibility of increasing the interface resolution. This would permit experimenting with a larger variety of different surfaces on top of the actuators or allow to omit such a surface altogether. This thesis focuses entirely on interaction through the relief interface surface. However, the actuation apparatus could be utilized to move external objects like spheres rolling on the interface or liquids forming pools and streams. These external object could facilitate novel interactions with digital content.

We are currently developing a greater set of gestures to accomodate richer interaction with the interface. This thesis has not investigated touch gestures on the surface, which does not require pressing down on it. Such gestures depend on aditional sensing hardware, such as capacitive sensing [49]. Interacting through touch gestures could be benefitial to define parametric shapes or draw textures. Another highly interesting input modality is the use of mid-air gestures, with the relief interface acting as a physical reference. Gestures could include defining a camera viewport of a virtual view rendered on a secondary display. The G-Speak framework could be utilized to track user hand gestures for such input [59].

Finally, multiple users of the interface have suggested to create an interface for blind users. While a number of tactile braille displays has been developed for better accessibility, rendering three-dimensional shapes specifically for blind users has not been explored extensively. The actuation hardware of relief might not be appropriate for such an application, but it is worth investigating.

The current applications have a very direct mapping between the interface and content. Digital geometry linked to the physical surface of the interface. In future applications, we want to investigate more abstract links, like physical shapes correlating with music or financial data. Such connections could allow users to make more meaning of information hard to understand otherwise.

Bibliography

[1] M. Anabuki and H. Ishii. Ar-jig: A handheld tangible user interface for modification of 3d digital form via 2d physical curve. *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, 0:1–10, 2007.

[2] art+com. Kinetische skulptur - bmw museum, mnchen. http://www.artcom.de/kinetik/, 2010.

[3] O. Bau, U. Petrevski, and W. Mackay. Bubblewrap: a textile-based electromagnetic haptic display. In *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 3607–3612, New York, NY, USA, 2009. ACM.

[4] M. Benali-Khoudja, M. Hafez, J.-M. Alexandre, and A. Kheddar. Tactile interfaces: a state-of-the-art survey. In *ISR 2004, 35th International Symposium on Robotics*, pages 721–726, March 2004.

[5] F. P. Brooks, Jr., M. Ouh-Young, J. J. Batter, and P. Jerome Kilpatrick. Project grope - haptic displays for scientific visualization. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 177–185, New York, NY, USA, 1990. ACM.

[6] A. Cassinelli and M. Ishikawa. Khronos projector. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Emerging technologies*, page 10, New York, NY, USA, 2005. ACM.

[7] M. Coelho, H. Ishii, and P. Maes. Surflex: a programmable surface for the design of tangible interfaces. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3429–3434, New York, NY, USA, 2008. ACM.

[8] M. Coelho and P. Maes. Shutters: a permeable surface for environmental control and communication. In *TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 13–18, New York, NY, USA, 2009. ACM.

[9] S. Coradeschi, H. Ishiguro, M. Asada, S. C. Shapiro, M. Thielscher, C. Breazeal, M. J. Mataric, and H. Ishida. Human-inspired robots. *IEEE Intelligent Systems*, 21:74–85, 2006.

[10] L. Faulkner. Physical terrain modeling in a digital age. In *Proceedings of Military Modeling and Simulation 2006 (MMS 2006)*, 2006.

[11] G. E. Favalora. Volumetric 3d displays and application infrastructure. *Computer*, 38(8):37–44, 2005.

[12] G. W. Fitzmaurice, H. Ishii, and W. 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.

[13] P. Frei, V. Su, B. Mikhak, and H. Ishii. curlybot: designing a new class of computational toys. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 129–136, New York, NY, USA, 2000. ACM.

[14] M. Goulthorpe. Hyposurface. http://www.hyposurface.org, April 2010.

[15] Greyworld. Source. http://www.greyworld.org/artwork/source, 2004.

[16] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, and G. Robles De La Torre. Haptic interfaces and devices. *Sensor Review*, 24(1):16–29, 2004.

[17] W. Hillis and F. B. Daniel. Raised display apparatus. US Patent, October 2008. US 7436388.

[18] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell. Passive real-world interface props for neurosurgical visualization. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 452–458, New York, NY, USA, 1994. ACM.

[19] D. Hirschmann. Glowbits. In *Ubicomp 2004, Adjunct Proceedings.* ACM, 2004.

[20] IEEE. Products. *Computer*, 36:89, 2003.

[21] E. Imhof. *Cartographic relief presentation.* ESRI Press, Redlands, Calif., 1st ed. edition, 2007.

[22] H. Ishii. Tangible bits: beyond pixels. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages xv–xxv, New York, NY, USA, 2008. ACM.

[23] H. Ishii, S. Ren, and P. Frei. Pinwheels: visualizing information flow in an architectural space. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 111–112, New York, NY, USA, 2001. ACM.

[24] H. Ishii and B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, New York, NY, USA, 1997. ACM.

[25] H. Iwata. Art and technology in interface devices. In *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 1–7, New York, NY, USA, 2005. ACM.

[26] H. Iwata, H. Yano, F. Nakaizumi, and R. Kawamura. Project feelex: adding haptic surface to graphics. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 469–476, New York, NY, USA, 2001. ACM.

[27] S. Jordà, S. E. Hunter, P. Pla i Conesa, D. Gallardo, D. Leithinger, H. Kaufman, C. F. Julià, and M. Kaltenbrunner. Development strategies for tangible interaction on

horizontal surfaces. In *TEI '10: Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 369–372, New York, NY, USA, 2010. ACM.

[28] B. C. Kaanta. Pins : a haptic computer interface system. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2004.

[29] H. Kajimoto, Y. Kanno, and S. Tachi. Forehead retina system. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Emerging technologies*, page 11, New York, NY, USA, 2006. ACM.

[30] J. Kestner, D. Leithinger, J. Jung, and M. Petersen. Proverbial wallet: tangible interface for financial awareness. In *TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 55–56, New York, NY, USA, 2009. ACM.

[31] H. Kim and W. Lee. Shade pixel. In *SIGGRAPH '08: ACM SIGGRAPH 2008 posters*, pages 1–1, New York, NY, USA, 2008. ACM.

[32] S. Kodama. Dynamic ferrofluid sculpture: organic shape-changing art forms. *Commun. ACM*, 51(6):79–81, 2008.

[33] M. KOJIMA, M. SUGIMOTO, A. NAKAMURA, M. TOMITA, M. INAMI, and H. NII. Augmented coliseum: An augmented game environment with small vehicles. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 3–8, Washington, DC, USA, 2006. IEEE Computer Society.

[34] K. Kushiyama, M. Inose, R. Yokomatsu, K. Fujita, T. Kitazawa, M. Tamura, and S. Sasada. Thermoesthesia: about collaboration of an artist and a scientist. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Educators program*, page 23, New York, NY, USA, 2006. ACM.

[35] M. C. Lin and M. A. Otaduy. Sensation-preserving haptic rendering. *IEEE Comput. Graph. Appl.*, 25(4):8–11, 2005.

[36] I. S. MacKenzie and C. Ware. Lag as a determinant of human performance in interactive systems. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 488–493, New York, NY, USA, 1993. ACM.

[37] T. H. Massie and J. K. Salisbury. The phantom interface: a device for probing virtual objects. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Nov 1994.

[38] J. Noble. *Processing: A Programming Handbook for Visual Designers and Artists.* O'Reilly Media, 2009.

[39] D. Overholt. The matrix: a novel controller for musical expression. In *NIME '01: Proceedings of the 2001 conference on New interfaces for musical expression*, pages 1–4, Singapore, Singapore, 2001. National University of Singapore.

[40] D. J. Page. Reconfigurable surface. US Patent, June 2005. US 6903871.

[41] J. Patten and H. Ishii. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 809–818, New York, NY, USA, 2007. ACM.

[42] B. Piper, C. Ratti, and H. Ishii. Illuminating clay: a 3-d tangible interface for landscape analysis. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 355–362, New York, NY, USA, 2002. ACM.

[43] I. Poupyrev, T. Nashida, S. Maruyama, J. Rekimoto, and Y. Yamaji. Lumen: interactive visual and shape display for calm computing. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Emerging technologies*, page 17, New York, NY, USA, 2004. ACM.

[44] I. Poupyrev, T. Nashida, and M. Okabe. Actuation and tangible user interfaces: the vaucanson duck, robots, and shape displays. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 205–212, New York, NY, USA, 2007. ACM.

[45] H. Raffle, M. W. Joachim, and J. Tichenor. Super cilia skin: an interactive membrane. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 808–809, New York, NY, USA, 2003. ACM.

[46] H. S. Raffle, A. J. Parkes, and H. Ishii. Topobo: a constructive assembly system with kinetic memory. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 647–654, New York, NY, USA, 2004. ACM.

[47] C. Ratti, Y. Wang, B. Piper, H. Ishii, and A. Biderman. Phoxel-space: an interface for exploring volumetric data with physical voxels. In *DIS '04: Proceedings of the 5th conference on Designing interactive systems*, pages 289–296, New York, NY, USA, 2004. ACM.

[48] C. Reas, B. Fry, and J. Maeda. *Processing: A Programming Handbook for Visual Designers and Artists.* The MIT Press, 2007.

[49] J. Rekimoto. Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120, New York, NY, USA, 2002. ACM.

[50] D. Rosenfeld, M. Zawadzki, J. Sudol, and K. Perlin. Physical objects as bidirectional user interface elements. *IEEE Comput. Graph. Appl.*, 24(1):44–49, 2004.

[51] D. J. Shamonsky. *Tactile, Spatial Interfaces for Computer-Aided Design - Superimposing Physical Media and Computation.* PhD dissertation, Massachusetts Institute of Technology, Department of Architecture, September 2003.

[52] I. E. Sutherland. The ultimate display. In *Proceedings of the IFIP Congress*, pages 506–508, 1965.

[53] M. Tory, A. E. Kirkpatrick, M. S. Atkins, and T. Moller. Visualization task performance with 2d, 3d, and combination displays. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):2–13, 2006.

[54] Troika, C. Freyer, S. Noel, and E. Rucki. *Digital by Design: Crafting Technology for Products and Environments.* Thames and Hudson, 2008.

[55] C. R. Wagner, R. D. Howe, and S. J. Lederman. A tactile shape display using rc servomotors. *Haptic Interfaces for Virtual Environment and Teleoperator Systems, International Symposium on*, 0:354, 2002.

[56] Y. Watanabe, A. Cassinelli, T. Komuro, and M. Ishikawa. The deformable workspace: A membrane between real and virtual space. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, pages 145 –152, 1-3 2008.

[57] M. Weiser and J. S. Brown. Designing calm technology. *PowerGrid Journal*, 1, 1996.

[58] H. Zhu. *Practical Structural Design and Control for Digital Clay.* PhD dissertation, Georgia Institute of Technology, School of Mechanical Engineering, July 2005.

[59] J. Zigelbaum, A. Browning, D. Leithinger, O. Bau, and H. Ishii. g-stalt: a chirocentric, spatiotemporal, and telekinetic gestural interface. In *TEI '10: Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 261–264, New York, NY, USA, 2010. ACM.