Alps: Exploring the Intersection of Gesture Input Systems and Tangible Media

by

Ryan Jackson

S.B., C.S. Massachusetts Institute of Technology, 2010

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2011
[June 2011]

Author .....
Department of Electrical Engineering and Computer Science
May 18, 2011

Certified by .....
Hiroshi Ishii
Jerome B. Wiesner Professor of Media Arts and Sciences; Associate Director, MIT Media Lab
Thesis Supervisor

Accepted by .....
Dr. Chris J. Terman
Chairman, Masters of Engineering Thesis Committee

Alps: Exploring the Intersection of Gesture Input Systems and Tangible Media

by

Ryan Jackson

Submitted to the Department of Electrical Engineering and Computer Science
on May 18, 2011, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In this thesis, I designed and implemented a gesture-based system to manipulate tangible media. Namely, I created a gesture interface to Google Earth that sends data to a geo-spacial, three-dimensional display called Relief. This thesis is both a demonstration and an attempt to elaborate on the impact of free-hand gesture recognition and the use of tangible media. The project demonstrates the power and limitations of such a combination, with a strong focus on future extensions and directions for the Alps project.

Thesis Supervisor: Hiroshi Ishii
Title: Jerome B. Wiesner Professor of Media Arts and Sciences, Associate Director, MIT Media Lab

Acknowledgments

First and foremost, I would like to thank my entire family. I would have never had the opportunity to attend MIT without the constant support of my mother, father, and brother. I would also like to thank Daniel Ron, my fearless UROP and partner in developing Alps. Daniel Leithinger and all of the TMG group have been there through the thick and thin, providing amazing feedback and incredible knowledge. I would also like to thank Oblong for the opportunity to develop Alps on their G-Speak system. Finally, I thank Colin Fain for creating many of the figures in this document.

Contents

Chapter 1

Introduction

Long before humans could talk, text, or Skype, we relied on a form of communication that both came naturally and has survived the test of time—gestures [6]. With the advent of cheaper free-hand, gesture-tracking systems, we have begun to see a wide range of games and applications explode across the consumer and research space. This thesis is an exploration and proof-of-concept in how gesture-tracking systems can support and extend the current functionality of tangible media. By using a recent geo-spacial, tangible media project named Relief [7], we show how gestures can supplement the language of communication between a user and a tangible interface.

In the first chapter, the background and motivation is given, followed by a general overview of the functionality, set-up, and user interface of Alps, the system created for this thesis. The second chapter details all the technologies used to create the project. Chapter 3 describes the gesture vocabulary created for Alps. Chapter 4 and 5 will discuss existing issues and future work, respectively. Finally, the conclusion will follow in Chapter 6.

1.1 Background

1.1.1 Tangible Bits and the Tangible Media Group

With the advent of personal computing, an overarching paradigm of a Graphical User Interface (GUI) consumed and guided the development of personal computing user interfaces for over thirty years. It was in 1997 that Professor Hiroshi Ishii proposed a new form of ubiquitous computing [10] known as Tangible Media [5] [4]. Ishii encouraged researchers to explore the use of tangible objects in our physical world to interface with the digital world, which until 1997 had predominately been manipulated with only a keyboard and mouse. We train ourselves for decades to manipulate physical objects, whether it is by painting a wall or moving blocks. Would it not be truly ubiquitous to convert these natural actions into corresponding digital interactions?

Born from Ishii's research was the MIT Media Lab's Tangible Media Group (TMG), a research group dedicated to creating tangible user interfaces (TUIs). Many amazing projects have come out of this group, including Relief. And now, at what could easily be a turning point in human-computer interaction, it is important to consider the influence that gestural input systems will have on the phenomenon that is Tangible Media.

GUIs themselves are abstract. They are oftentimes digital representations of physical objects (i.e. documents, folders, desktops, mail, etc.). TUIs, on the other hand, use embodiment and metaphors in the opposite direction, connecting physical objects to logically appropriate digital functionality.

1.1.2 Gestural Input Systems

Gestural input systems are not considered "new" technology to the research community. There has been a significant amount of research surrounding free-hand gestures for many years. A commonly cited paper, which uses both gesture-tracking and speech recognition, is *Put That There*, and it was released in 1980 [1]. Certainly, in the pro-

ceeding thirty years, there has been much development within the gesture-tracking space.

Myron Kruger's Video Place is a great example of how an environment can respond to direct interaction using gestures. Using video capture and image processing, Kruger was able to recognize body poses all the way down to finger positions. His work set the stage for a future of gesture-manipulated environments.

1.1.3 Relief

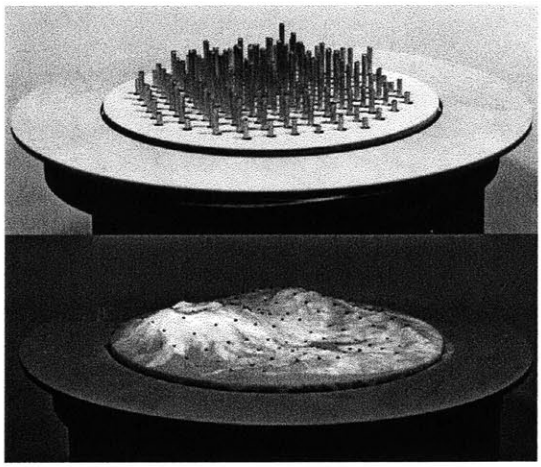

Figure 1-1: Relief (source: http://tangible.media.mit.edu/project.php?recid=132)

Relief, a project of the Tangible Media Group, allows one to have a three-dimensional, geo-spacial display. It achieves this display by elevating a 12x12 grid of actuated pins. A stretchy fabric is adhered to the tops of the pins, and a map is projected on top of the grid. Each pin can then display the scaled height of the terrain relative to the other pins. This scaled, relative display, while in very low resolution, is a great way to see the three-dimensionality of terrain as opposed to looking at a map on a screen with elevation lines.

Relief also includes a wheel around the grid that can serve as an input to either translate or rotate the projected map. Granted Alps does not currently take advantage of this wheel, it could also be mapped to other functionality as discussed in Chapter 5. Ultimately, Relief allows us to physically generate the actual shape of a

landscape.

1.1.4 G-Speak

The system we are using for gesture recognition and tracking is called G-Speak. G-Speak is developed by Oblong, a California-based company. Oblong is a company started by former Tangible Media Group student, Dr. John Underkoffler, along with Kwindla Kramer. Underkoffler is one of the inventors of the I/O bulb, an earlier TMG project [9]. Underkoffler and Kramer implemented and donated their G-Speak environment to the Media Lab, enabling TMG to do gesture tracking research.

The system uses a Vicon tracking system to accurately determine where the user's hands are in three-dimensional space. Though the Vicon system is expensive, the accuracy with which it can track (sub-millimeter precision) is desirable in a research environment.

G-Speak is a framework for gesture recognition, a way to build custom applications that integrate gestures. Oblong created many demos to demonstrate the power of their system, two of which are the following:

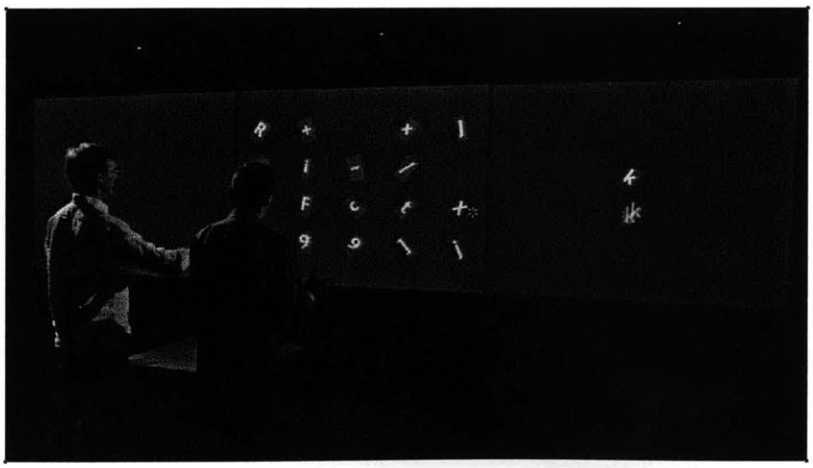

Figure 1-2: Grabby (source: www.oblong.com)

- Grabby

  Grabby, as shown in figure 1-2, is an application that places wiggling tiles

on the screen in a grid layout. Each tile has a letter in it. Using a set of specified gestures, one or two users can select, move, and reorganize any and all of the tiles. While this demonstration may at first seem trivial, one can imagine more sophisticated extensions. For example, place a single frame from different videos in each tile. Again, each tile can be moved, resized, and now each movie represented by the single frame in each tile can be played by adding a few more recognizable gestures to the application.

G-Speak is not only a gesture-input system but also a "spatial operating environment" (SOE). G-Speak knows the position, orientation, and size of screens in the environment enabling the user's gestures to "affect" the displays. Grabby is a good, simple example of the SOE. A user can manipulate and drag the tiles onto the table or screens. In fact, s/he can drag them into empty space, and if there were a screen in that direction, the tile would be displayed there.

Figure 1-3: Urf (source: www.oblong.com)

- Urf

  Urf, as shown in 1-3, is an example of G-Speak's ability to show large datasets in both a unique and useful way. The application projects a globe of the world in three-dimensional space. The globe is transparent and merely outlines the borders of countries and states. A user can fly around and through the globe to focus on any part. With a few gestures, one can see in slow-, real-, or fast-time

all of the flights in the United States on a given date. The planes' paths are showed by lines from one city to another. As the planes "fly", the user can also move in and around the globe to look at the motion of aviation in the United States.

G-Speak inspired the often-noted gestural system used by Tom Cruise in the movie Minority Report. The system has powerful applications that use gestures of a single or multiple user(s). G-Speak is an incredibly useful platform and tool in this field of research for its accuracy in tracking the position and orientation of the hand and body in three-dimensional space.

1.1.5 G-Stalt

Figure 1-4: G-Stalt (source: tangible.media.mit.edu)

One of the first projects using G-Speak in the Tangible Media Group was called G-Stalt, by Zigelbaum et. al [12]. G-stalt is a gesture interface that defines a gesture vocabulary to manipulate a three-dimensional space filled with video cartoons as shown in figure 1-4. Zigelbaum et. al noted the shift from manipulating the physical world to manipulating the digital. They developed a "pinch" gesture, among others, that allows one to grab a point in space to move or rotate it.

Our hands are powerful tools in the physical world that do not constrain us as much as the keyboard and mouse do in the digital world. Zigelbaum et. al sought

to demonstrate a software application that gave users to ability to control the digital space similar to how we control the physical world around us.

1.2 Motivation

There are many motivations for this particular project. With new technology being developed everyday, it is important to consider their effect on both new and old products and applications.

1.2.1 Why gestures?

There are a few reasons why Alps uses gestures:

1. Wide-Spread Adoption of Gesture Tracking Systems

   As previously mentioned, gesture tracking is not a new concept; however, it is only recently that the general public has had access to relatively inexpensive hardware. Namely, the Microsoft Kinect has provided gamers across the globe with a compact and affordable gesture-tracking system. Before the Kinect, a lot of research focused on image processing from cameras as a viable and cheap means of gesture recognition, but the Kinect provides distance data that a simple camera cannot, making it easier to track gestures.

   Though Microsoft did not provide a formal API for the Kinect with its release, many developers and researchers manipulated the hardware to create tracking applications that extend the original body position tracking. Notably, at the MIT Computer Science and Artificial Intelligence Lab (CSAIL), one group has successfully created a library enabling Kinect to track fingers [3].

   While nowhere near as accurate as the Vicon tracking system used in G-Speak, the Kinect provides sufficient hardware and software to extend the usability of an interface. And, if we consider the Kinect to be a proof-of-concept or one of many to come, we will begin to see cheap gesture tracking systems appearing in- and outside of the gaming community.

2. Gestures as a Natural Language

   Buxton says that "with respect to objects, we have a broad range of gestures that are almost universal, including pointing at objects, touching or moving objects, changing object shape, activating objects such as controls, or handing objects to others" [2]. Free-hand gestures are natural to humans. We can tell entire stories with just our hands or describe some of the most verbally complex concepts with a mere few gestures. To exemplify this claim, imagine describing to someone the shape of the magnetic field through a solenoid. Similar to Ishii's notion of the obviousness of using our physical surroundings as a means of manipulation of digital data, it follows that we should also be able to communicate with a computer in similar ways as we communicate thoughts to other people, using both speech and gestures.

   Buxton also surveys the work of other researchers who have defined the space of gesture recognition. The previously mentioned research *Put that There* [1] is described by Rime and Schiaratura as using *deictic gestures* or gestures used for pointing or drawing attention [11]. Alps uses both *deictic* and *iconic* gestures, where *iconic* gestures are those that specify the shape or orientation of an object. The gestures in Alps define the orientation and zoom level of a camera as well as allow you to select and center points on the display.

Another benefit to gestures is the ability to input commands in three-dimensional space. A standard mouse or keyboard confines one to two dimensions. Gestures give you the flexibility of moving one's hands through space to interact with a computer.

1.2.2 Why tangible media?

The Tangible Media Group has shown how unique interfaces can be developed and improved by using Tangible User Interfaces. It is important to consider how the advent of cheaper gesture-recognition systems will affect the notion of "interaction by touch." One can easily see two outcomes:

1. The Replacement of TUIs, Keyboards, and Mice

If we consider a world that is similar to that of *Minority Report*, we see systems that are "touchless" and driven solely by gestural input. While this paradigm shift is not prevalent, it is certainly being explored as a means to increase conversation bandwidth between users and computers using natural gestures.

2. The Extension of Tangible Media using Gestures

   A more likely case will be supplementing existing user interfaces with gestural input systems. As Buxton says, *deictic* gestures (pointing and drawing attention) dominate human-computer interaction using direct manipulation devices. We can easily replace direct manipulation devices and remotes with gesture recognition, but we will always have some attachment to the physical objects in our world. The question becomes: How can we increase data bandwidth between users and computers while decreasing time, given that we now have inexpensive access to gesture-tracking? And more generally, what are the possibilities and limitations of combining tangible media interfaces with gestural input? These questions are not answered by this thesis, but as stated, we hope to demonstrate the capabilities of uniting tangible media with gestures. By doing so, we hope to further the enthusiasm and research around gesture recognition and existing user interfaces.

1.2.3 Why Google Earth?

Google Earth is a compelling dataset that naturally lends itself to being explored using gestures. The notion of moving through three-dimensional space around the world and using terrain data to map relative elevations is only one of many use cases for Google Earth. The goal was to build an interface to Google Earth that can be ported to other projects, both tangible media and otherwise. Google Earth also provides a web-browser plugin that makes it easy to develop and share code for the project.

Aside from its portability, Google Earth also allows us to map three-dimensional input to three-dimensional space. Consider table 1.1:

-	Input	Google Earth	Relief
On Desktop	2D	2D	–
On Alps	3D	2D	3D

Table 1.1: Dimensions of inputs, data, and displays.

If we are using a mouse or other direct-manipulation device for input, we are restricted to two dimensions for input. This input is then fed to a three-dimensional engine that is then output on a two-dimensional screen. We have gone from two- to three- to two-dimensional space. Ultimately, the interaction suffers with direct-manipulation.

If we use Alps, as stated at the end of section 1.2.1, we input using gestures in three-dimensional space. The addition of Relief then allows us to supplement the two-dimensional output of Google Earth with the three-dimensional, geo-spacial mapping of Relief. And though the resolution of Relief is low, it is consistent with the dimensionality of the input and Google Earth.

1.3 Alps

Alps is an environment in which a user can use their hands to manipulate Google Earth and the geo-spatial display of Relief. By pointing and "clicking," a user can navigate Earth, selecting mountain ranges to analyze in either two- or three-dimensional space.

1.3.1 Program Functionality

Alps is a demonstration of how one can combine gestures and tangible media to improve user-experience by allowing a wider range of actions and manipulation of both objects and space. The purpose of Alps is to allow users to do the following:

1. Manipulate Google Earth using Gestures

   By using a simple gestural vocabulary, a user can control Google Earth. A user's control, while limited by the scope of this thesis, can be extended to

include panning, zooming, flying, going to a specific location, etc. When a user selects a particular spot on the surface of Earth, the view is centered around that point and elevation data is collected.

2. Show Real-Time Physical Output of Geo-Spacial Data

   As mentioned, Alps incorporates a previously developed, tangible media project called Relief [7]. Relief can show the relative elevations of mountain ranges at different zoom levels and locations. When a point is selected and centered on the map, that center point becomes the center point of the Relief table, and the surrounding, scaled elevations are reflected using the actuated pin display.

1.3.2 The Set-Up

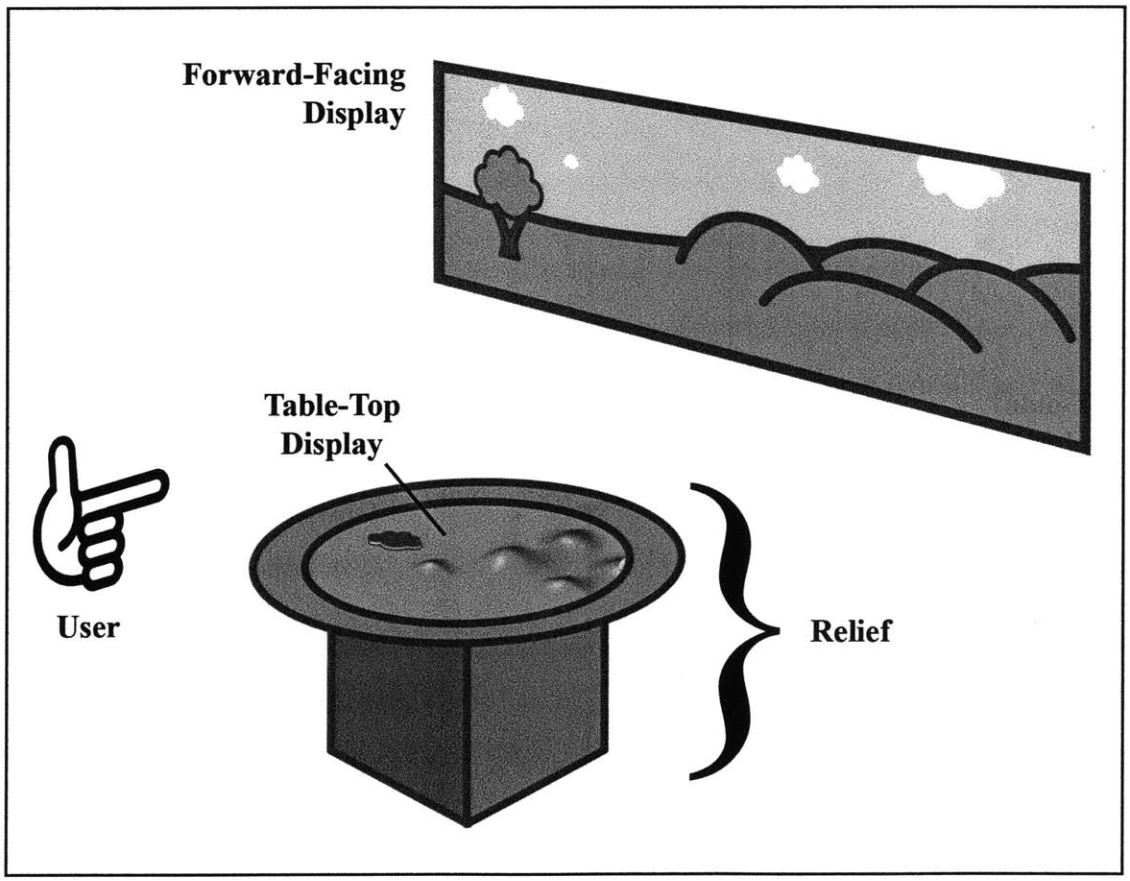

Figure 1-5: Diagram of G-Speak Space

The set-up of Alps is essential to understanding the concept and how it can be expanded or constrained. Four projectors and eleven Vicon tracking cameras are suspended from a metal frame. As shown in the figure 1-5, three of the projectors are used as the forward-facing display and the forth is used as a table-top display in the center of the area. This terminology will be used throughout the remainder of the thesis.

1.3.3 The User Interface and Experience

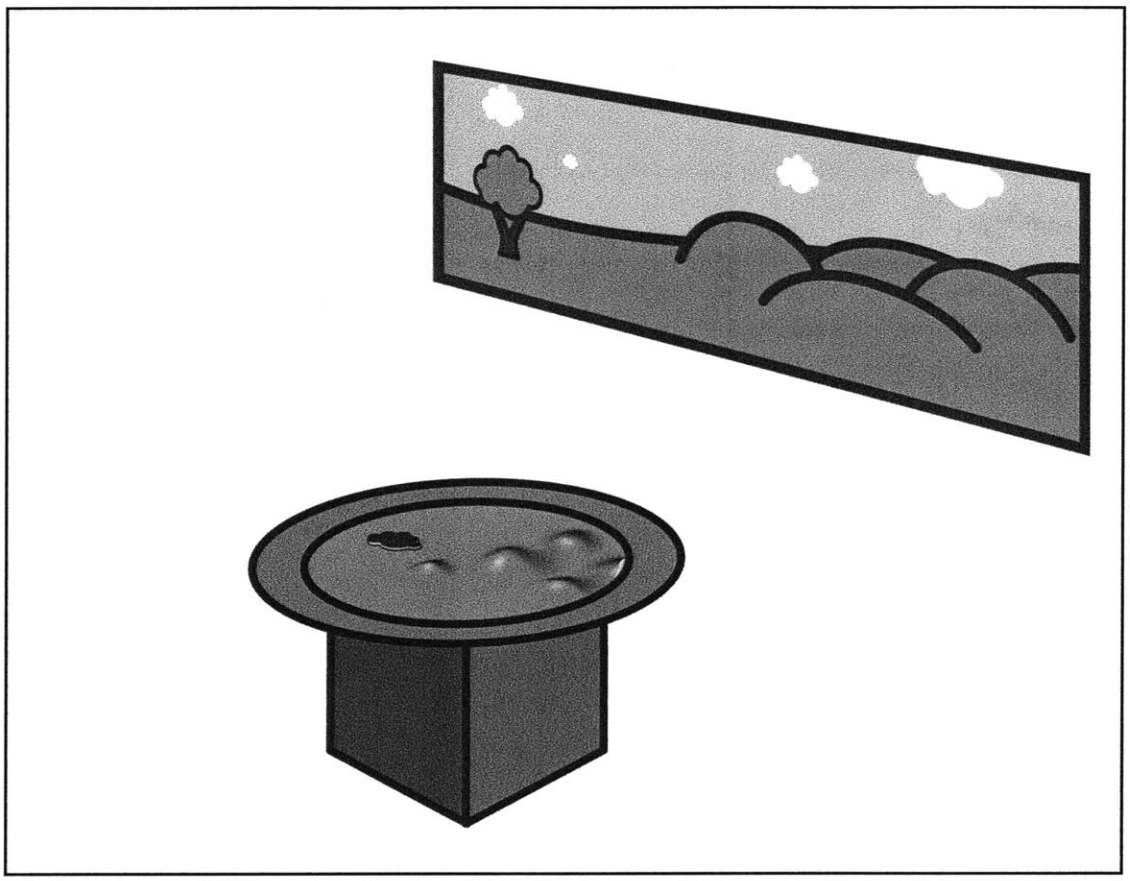

Figure 1-6: The idle state of Alps.

Before understanding the technology behind the Alps project, let us consider a single use case. A user walks up to Alps and gestures to center and analyze a mountain range currently projected on the forward-facing display. As shown in figures 1-6 through 1-10, this use case can be broken down into three parts.

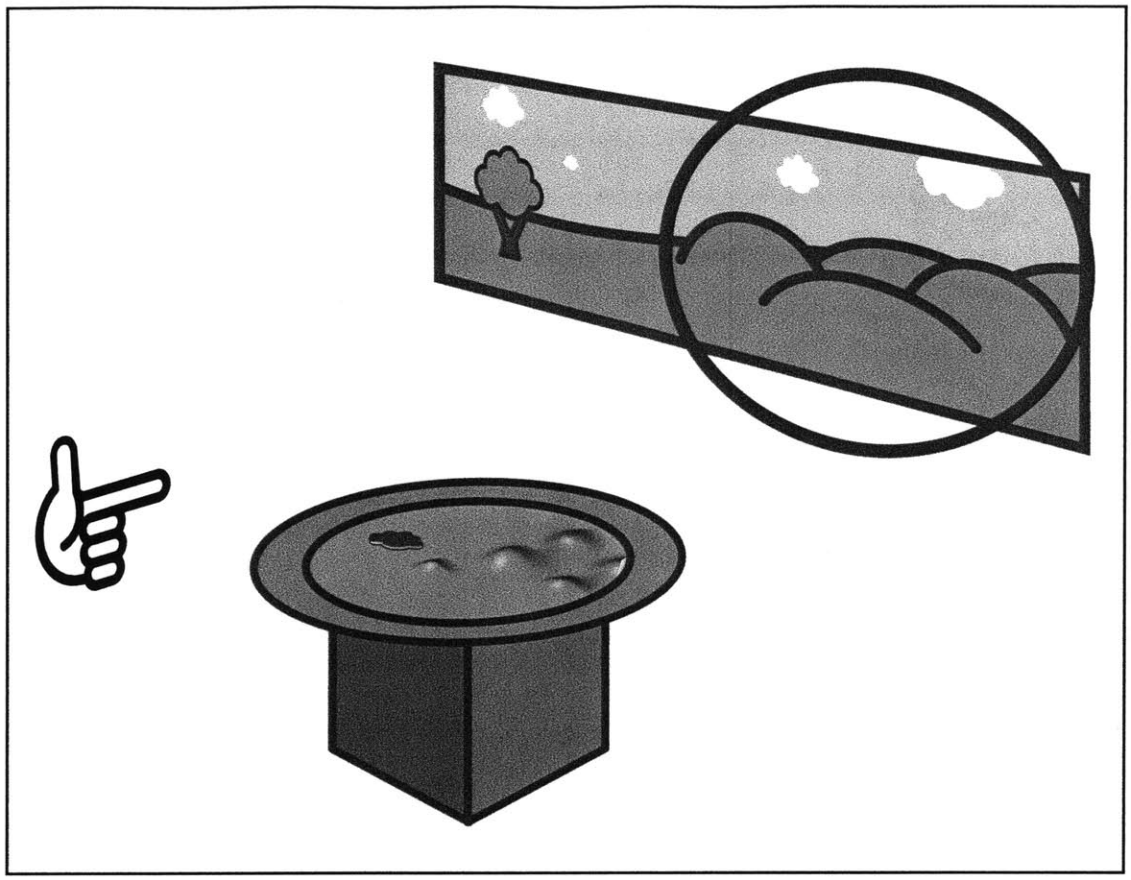

Figure 1-7: A user focuses on a location using the "point" gesture.

1. The state of Alps before a user interacts with the system.
2. The user pointing around before selecting a location.
3. The user selecting a section to go to on the map. This step actually causes three events to occur simultaneously:
   - The forward facing display centers the selected location.
   - The table top display also centers the selected location.
   - The locations elevation data is collected and displayed using the actuated pins of Relief.

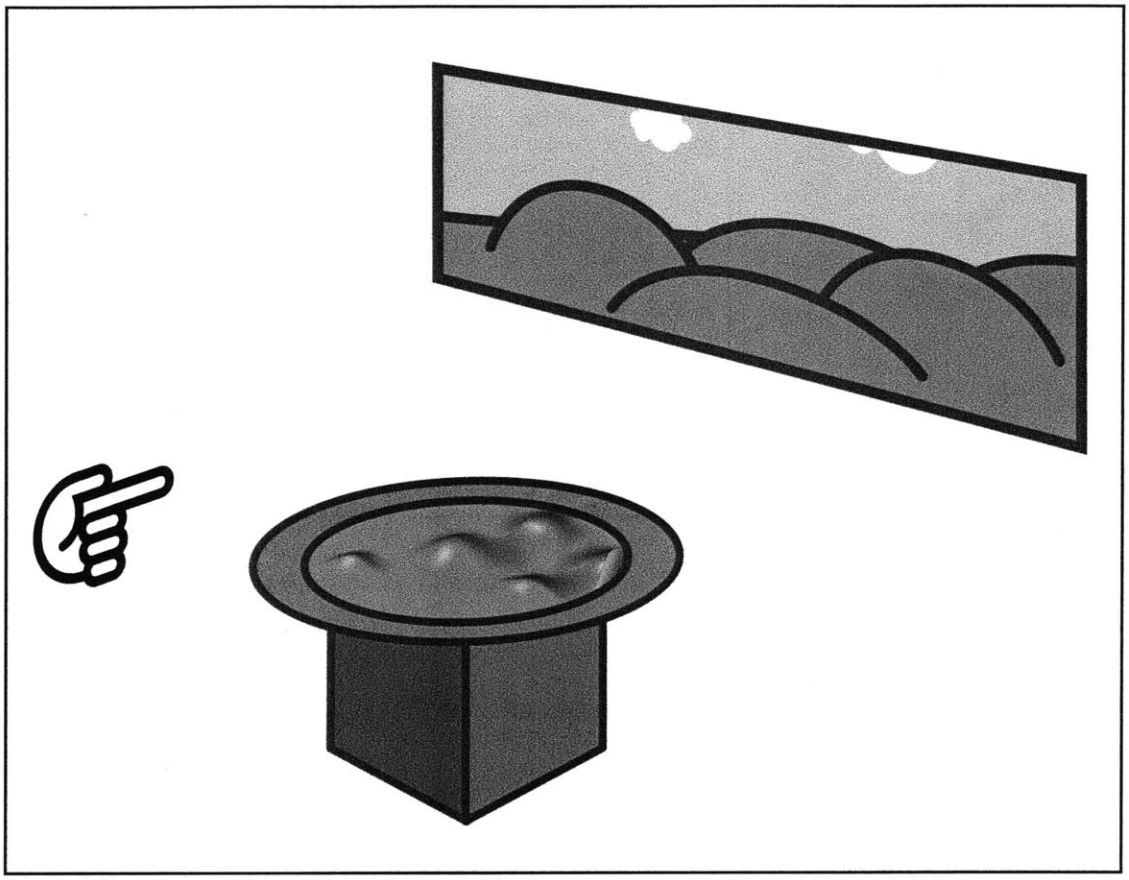

Figure 1-8: After finding an area of interest, a user performs the "go to" gesture by dropping their thumb which initiates figures 1-9 and 1-10 simultaneously.

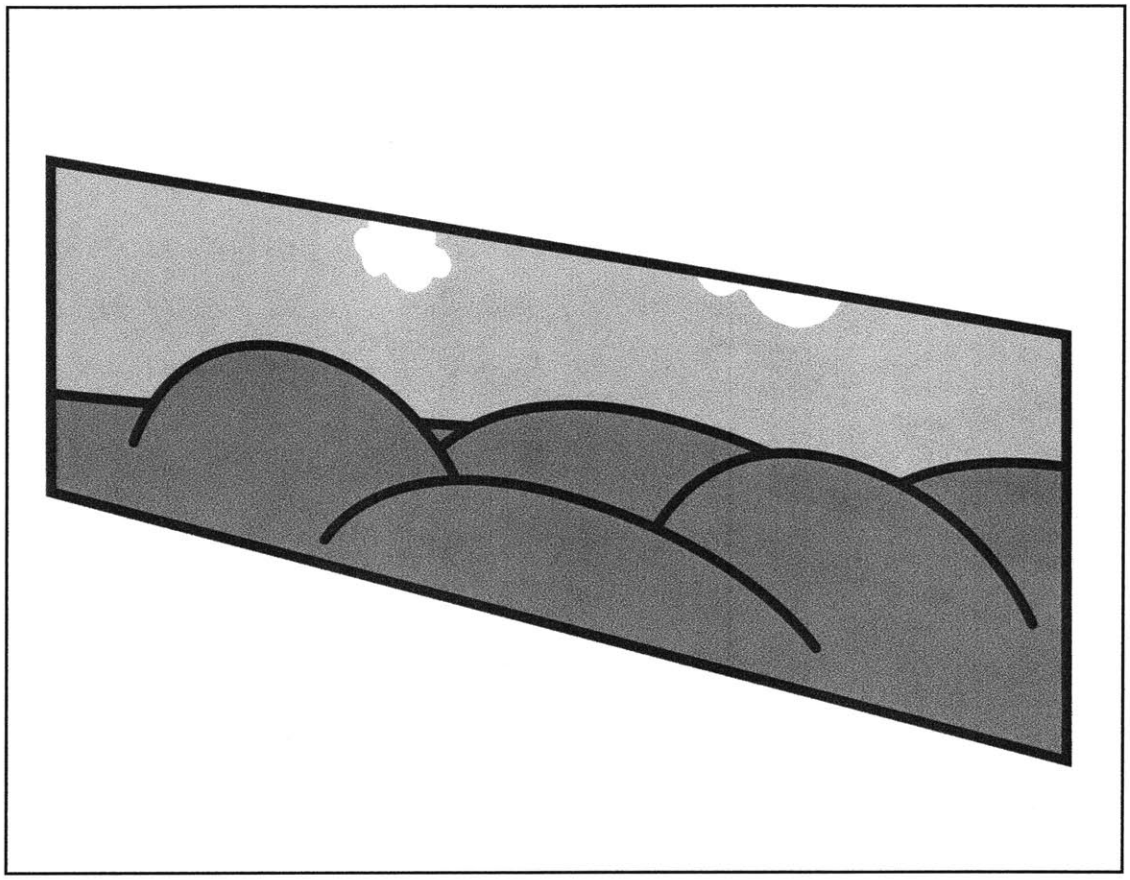

Figure 1-9: The forward-facing display centers the selected area of Google Earth.

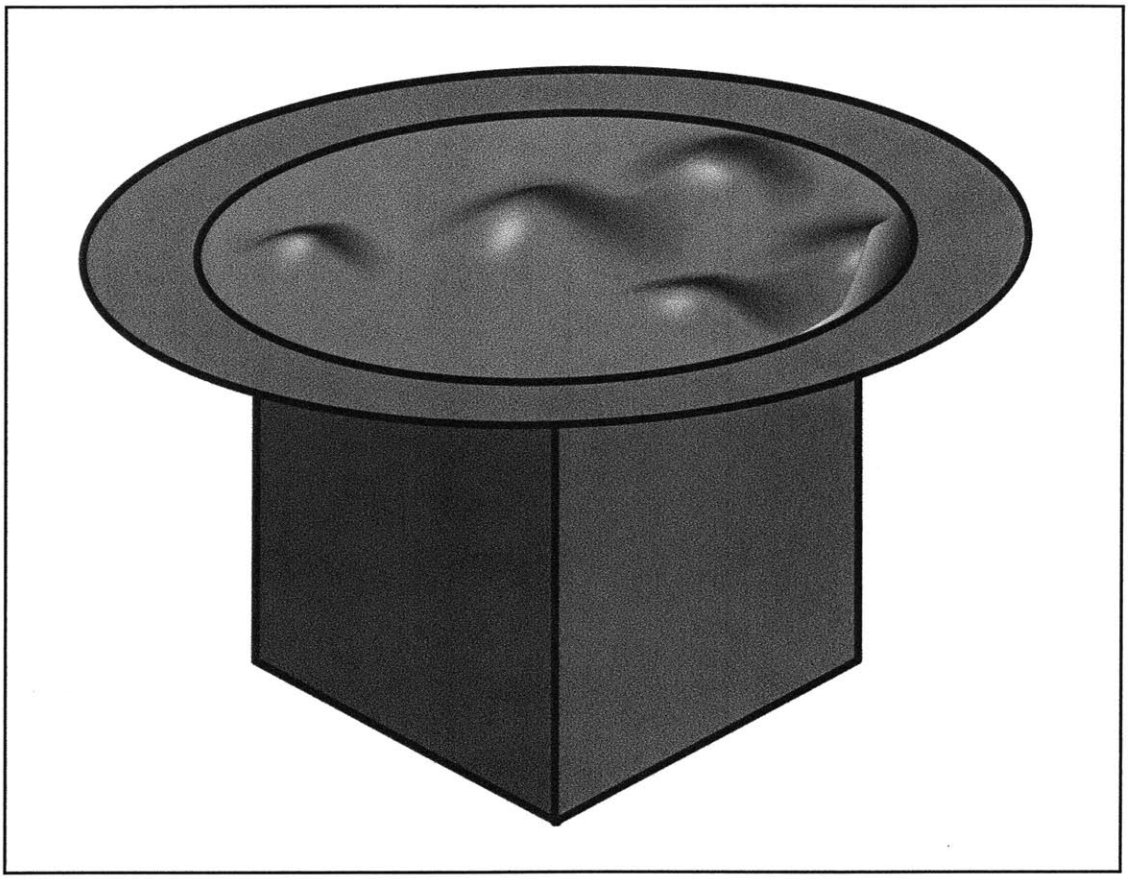

Figure 1-10: The table-top display centers the selected area of Google Earth and the pins update their relative elevations to match the projected terrain.

Chapter 2

The Technology

Alps is a combination of many different technologies. We use a sophisticated gesture-tracking system coupled with a publicly available API to Google Earth to send scaled elevation data to the aforementioned geo-spacial, tangible media project, Relief.

2.1 G-Speak

As previously mentioned, G-Speak, produced by Oblong Industries, is the system we use to track gestures. G-Speak provides extremely accurate gesture tracking by combining innovative hardware and software.

2.1.1 Hardware

Vicon Motion Capture System

G-Speak utilizes a Motion Capture System from Vicon to track the users' hands in three-dimensional space. The current setup includes eleven Vicon cameras. Each camera is equipped with an infrared strobe and a lens. The cameras produce infrared light which is reflected off retro-reflective markers and then caught in the lens of other cameras. The three-dimensional position of an infrared-reflective surface can be determined by combining the inputs from multiple cameras. Common uses of Vicon include placing retroreflective markers or tracking dots on objects to track

their location in space. The dots are simply reflective balls of varying sizes that can be easily tracked when in range of the cameras. The Vicon system can track the passive retroreflective dots at over 100 Hz yielding sub-millimeter precision.

Server

The server that powers G-Speak includes sixteen gigabytes of ram. The processing power allows G-Speak to process the massive amount of data from the Vicon tracking system described in the following section.

Gloves

Figure 2-1: The gloves with their unique tags (source: www.oblong.com)

A user of G-Speak wears a pair gloves that is tracked by the Vicon system. The gloves are covered in what Oblong calls "tags." Each tag contain multiple retroreflective dots, uniquely arranged on the same plastic backing. These tags are placed on the fingers and back of hand as shown in figure 2-1. The uniqueness is key to how G-Speak can locate a tag in space. If you consider only a single dot, it is indistinguishable from another dot in space. Tags use multiple dots in unique layouts so that each tag can be recognized and distinguished from other tags in space. With multi-tag recognition and tags on the back of the hand as well as the first and second fingers, G-Speak can track the location and, more importantly, the orientation

of one's hand and fingers in three-dimensional space. With gloves on both hands, G-Speak is equipped to track two handed-gestures.

2.1.2 Software

G-Speak's framework is written in C++. The libraries are used to convert the raw tracking reflective dot data from the Vicon system into (x, y, z) positions and movement vectors. On a higher level, the software then looks for tags within the multiple dots data and extrapolates to find the position and orientation of the hand. G-Speak's software can tell you the intersection of where you are pointing and the forward-facing display. This intersection is collected and exported to a database to be used by Google Earth.

2.1.3 Limitations

There are a couple limitations with G-Speak, most notably the issue of occlusions. Because G-Speak utilizes Vicon cameras, an object blocking the gloves or tracking tags from the cameras can cause a failure in hand tracking and gesture recognition. It is for this reason that G-Speak, in our current setup, is optimized for a maximum of two users. This restriction does not affect Alps in its current state, but, as will be discussed in Chapter 5, it is conceivable and possible to have multiple users using Alps at the same time. In fact, in its current state, two or more users could use Alps (save data loss due to occlusions), but a vocabulary specific to interactions by two users has not been considered or designed.

2.2 Google Earth

Google Earth provides a JavaScript API for manipulating Google Earth in a browser plugin (http://code.google.com/apis/earth/). There are two instances of Google Earth running in Alps, one projected on the forward-facing display and another projected on the table-top display. Both instances are run on a Chrome browser in an

effort to avoid compatibility issues with other browsers and the Google plugin.

2.3 The Computer for the Forward-Facing Display

The instance of Google Earth being projected onto three screens of the forward-facing display is driven by an iMac. The iMac loads a website running the Google Earth plugin on Google's Chrome browser and outputs it to three projectors using a Triple-Head To Go display hub.

2.4 The Computer Driving Relief and the Table-Top Display

Relief and the table-top display are driven by the same computer, a Mac Mini. The computer also loads a website on Google's Chrome browser and projects it over Relief. A program, written in C++, is then executed to receive and adjust the actuated pins when new elevation data is sent.

2.5 Communication between Servers and Clients

In order to communicate among G-Speak, Google Earth, and Relief, we use a MySQL database to store both tracking and elevation data. The schema for tables is as follows:

1. Communication between G-Speak and Google Earth

Name	Type
command	varchar(20)
x	int(11)
y	int(11)

Table 2.1: The schema for the *commands* table.

In order to communicate from G-Speak to Google Earth, we create a MySQL table with the schema shown in table 2.1. This table has a fixed number of rows

so as to support the possible gestures in the Alps project. The x and y here are the (x, y) coordinates of where the gesture intersects the screen. Although gestures will be explained in more detail in Chapter 3, we can consider the go-to gesture to illustrate.

If one is pointing at the forward-facing display and sees something s/he would like to inspect, s/he may use the go-to gesture while pointing. Alps captures the (x,y) pixel coordinates from the screen and stores them in the database. Alps is listening to changes in the database and, if one is seen, will execute queries and calculations to convert the pixel coordinates to latitude and longitude coordinates. Google Earth responds by centering on the new (latitude, longitude) coordinates. Alps then resets the values of x and y to 0 for the "goto" command.

2. Communication between Google Earth and Relief

Name	Type
command	varchar(20)
x	float(11)
y	float(11)

Table 2.2: The schema for the *relief_commands* table.

Relief must also have a commands table, though it will function slightly differently than the one described above. In the current version of Alps, the *relief_commands* table, shown in table 2.2, is only used for the "goto" gesture. The x and y hold latitude and longitude values, respectively, as opposed to (x, y) pixel coordinates.

Name	Type
x	int(11)
y	int(11)
elevation	int(3)

Table 2.3: The schema for the *relief* table.

The table that will hold the interim elevations for Relief is shown in table 2.3.

Relief has a 12x12 grid of actuated pins that can each be set to a value between 0—128. The table will contain a fixed number of rows, one for each pin on Relief's grid. The rows look like as follows, where [0–128] represents the possible integer values computed before insertion:

x	y	elevation
0	0	[0–128]
0	1	[0–128]
0	2	[0–128]
0	3	[0–128]
0	4	[0–128]
0	5	[0–128]
0	6	[0–128]
0	7	[0–128]
0	8	[0–128]
0	9	[0–128]
0	10	[0–128]
0	11	[0–128]
1	0	[0–128]
1	1	[0–128]
1	2	[0–128]
1	3	[0–128]
1	4	[0–128]
	...	
11	8	[0–128]
11	9	[0–128]
11	10	[0–128]
11	11	[0–128]

Table 2.4: The rows of the *relief* table.

After the table is populated, Alps will capture the changes and send the appropriate values to Relief.

Chapter 3

Gestures

The key to any gesture-enabled system is the vocabulary with which a user can communicate with the system. The current version of Alps only implements the gestures outlined in this chapter. Each gesture below contains the following sections:

- Use

  This section will explain the general purpose of each gesture and how it affects the interface.

- Gesture

  This section gives both a verbal and graphical description of how to perform each specific gesture.

- Schema—G-Speak to Google Earth

  Each gesture has its own row in the *commands* MySQL table as described in section 2.5, but each gesture may not use all of the columns. For this reason, this section will cover what columns this gesture requires, the possible values of each column, and for what each column is used.

- Schema—Google Earth to Relief

  Though not every gesture affects Relief, if so, the affected tables are discussed in these sections.

- Input Conversion

  Most, but not all, gestures require some type of conversion in order to be sent to Google Earth and/or Relief. We discuss in this section how the data given by G-Speak is converted and consumed by both Google Earth and Relief.

For consistency, hand positions will be described as fingers relative to each other. R1 is the thumb on the right hand. R2 is the index finger on the right hand. R3, R4, and R5 follow in the same fashion down to the fifth finger, opposite the right thumb. L1 is the thumb on the left hand, and L2 the index finger of the left hand. L3, L4, and L5 follow in the same fashion as the right hand. The fingers are labeled in the figure 3-1:

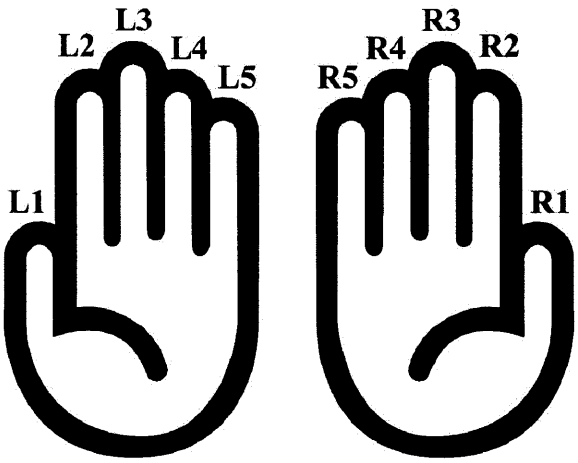

Figure 3-1: Labeled Fingers

3.1 Point

3.1.1 Use

The "point" gesture will place a cursor on the screen. Because the forward-facing display is driven by a different computer than the one tracking gestures, Alps renders a cursor on the display computer by converting the coordinates produced by the gesture-tracking server. This gesture is classified as "continuous", meaning that the gesture is constantly sampled from the time it appears until the time it disappears.

3.1.2 Gesture

The gesture is performed by holding up the left hand and curling fingers L3, L4, and L5 to the palm. L1 and L2 are extended to form a 90° angle. If one imagines a line extended from the finger in the direction pointed by L2, the cursor will appear where the line intersects the forward-facing display. You can also perform this gesture with the right hand, but the left hand is shown in figure 3-2.

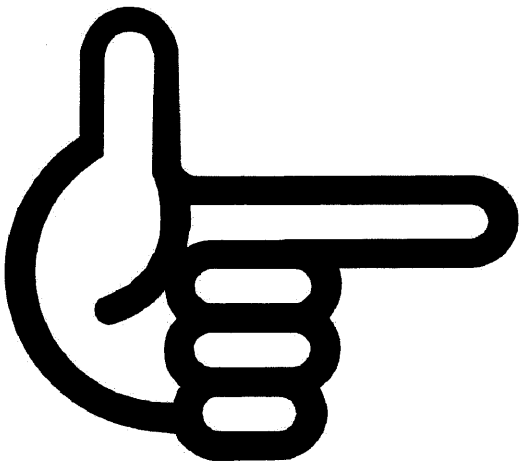

Figure 3-2: Point

3.1.3 Schema—G-Speak to Google Earth

Name	Value
command	point
x	$[0, 10^{10}]$
y	$[0, 10^{10}]$

Table 3.1: The name and possible values of the necessary columns in the *commands* table for the "point" gesture.

3.1.4 Schema—Google Earth to Relief

As there is no cursor displayed on the Google Earth projection over Relief, there are no calls to the database beyond placing the cursor on the forward-facing display.

3.1.5 Input Conversion

As alluded to earlier, G-Speak provides the intersection of where you are pointing and the forward facing display. The display is considered a two-dimensional plane, so G-Speak outputs an (x, y) coordinate. These coordinates are then converted into ratios by the following equations:

$$x_{ratio} = (x - x_{west})/(x_{east} - x_{west}) \tag{3.1}$$

Equation 3.1 converts the x pixel to a ratio, where x_{east} and x_{west} are the right- and left-most pixels of the forward-facing display, respectively.

$$y_{ratio} = (y - y_{south})/(y_{north} - y_{south}) \tag{3.2}$$

y_{ratio} is calculated in the same fashion. Using these new ratios, we can use the following equations to calculate the respective (x, y) pixel coordinate on which to overlay the cursor image on the forward-facing instance of Google Earth:

$$x_{new} = (x - x_{GEwest})/(x_{GEeast} - x_{GEwest}) \tag{3.3}$$

$$y_{new} = (y - y_{GEsouth})/(y_{GEnorth} - y_{GEsouth}) \tag{3.4}$$

In equations 3.3 and 3.4, $y_{GEnorth}$, $y_{GEsouth}$, x_{GEeast}, and x_{GEwest} refer to the top-, bottom, right-, and left-most pixel, respectively.

3.2 Go To

3.2.1 Use

The "go to" gesture is used to center a point on the screen and calculate the relative elevations to send to Relief. This gesture is a sister of the Point gesture, as it is merely a one-finger extension. The gesture is considered "singular". Upon forming

this gesture, it only triggers its associated code block once. Singularity means in order to execute this gesture twice, one must make the gesture, break the gesture, and make it a second time. Breaking the gesture can be going back to the aforementioned Point gesture, performing some other recognized gesture, or simply not making any recognizable gesture at all.

3.2.2 Gesture

To perform this gesture, simply form the Point gesture and from this position, lower L1 to be parallel to L2. As soon as L1 drops, the gesture will trigger the appropriate code. You can also perform this gesture with the right hand, but again figure 3-3 shows the left hand.

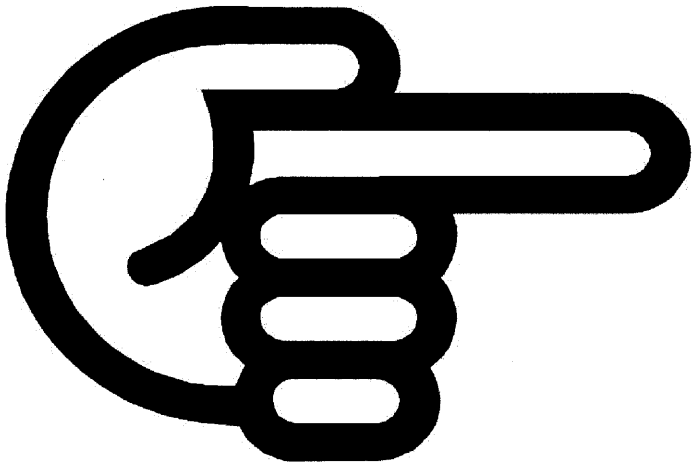

Figure 3-3: Go To

3.2.3 Schema—G-Speak to Google Earth

Name	Value
command	goto
x	$[0, 10^{10}]$
y	$[0, 10^{10}]$

Table 3.2: The name and possible values of the necessary columns in the *commands* table for the "go to" gesture.

3.2.4 Schema—Google Earth to Relief

Name	Value
command	goto
x	[0–128]
y	[0–128]

Table 3.3: The name and possible values of the necessary columns in the *relief_commands* table for the "go to" gesture.

3.2.5 Input Conversion

The (x, y) pixel coordinates given by G-Speak are converted into ratios by equations 3.1 and 3.2. These x and y ratios are then used to calculate the latitude and longitude on which to center using the following equations:

$$lat = x_{ratio} * (lat_{east} - lat_{west}) + lat_{west} \tag{3.5}$$

Using x_{ratio}, the latitude is calculated by equation 3.5, where lat_{east} and lat_{west} are the right- and left-most latitude, respectively, of the instance of Google Earth projected on the forward facing-display.

$$lon = y_{ratio} * (lon_{north} - lon_{south}) + lon_{south} \tag{3.6}$$

Similarly, y_{ratio} is used to calculate the longitude using Equation 3.6, where lon_{north} and lon_{south} are the top- and bottom-most longitude, respectively, of the instance of Google Earth projected on the forward-facing display.

The location specified by the calculated latitude and longitude coordinates is then centered on the forward-facing display and stored in the *relief_commands* table. The instance of Google Earth projected on the table-top display then also centers on the same coordinates. Once centered, the table-top instance is then divided into a 12x12 grid. An elevation is requested from Google Earth for each cell and is stored in a list, *elevations*. We then normalize and scale each elevation data point using equation 3.7.

$$elevation_{scaled} = (elevation - elevation_{min})/elevation_{max} * 128 \qquad (3.7)$$

The values elevation$_{min}$ and elevation$_{max}$ are the smallest and largest elevation, respectively. These elevations are then stored in the *relief* table and later retrieved as the heights of the actuated pins on Relief.

Chapter 4

Existing Issues

4.1 Latency

The use of a MySQL database is a double-edged sword. Using a remote database makes for some exciting and useful extensions which will be discussed in Chapter 5. The problem, however, is that Alps is currently dependent on data being sent and received over networks. Depending on the connection speed, this dependency can cause a serous lag in the user experience. For example, when using the point gesture to display a cursor on the screen, the lag time can vary between 200–800 milliseconds on average. There are instances where it is less than 200 or more than 800 milliseconds, but they rarely occur. The only piece that requires internet is the Google Earth JavaScript API. The rest, including the use of databases, can be accomplished on a local area network server (including any of the server or client computers used in the project). As discussed in section 5.7, it is believed possible to remove the databases completely. Removing the databases and driving the displays, Relief, and gesture-tracking from one computer would likely decrease latency. The reason this design was not chosen was the difficulty in communicating from the C++ output of G-Speak to the JavaScript API of Google Earth running in the Chrome browser.

4.2 Cost

The current set-up is both expensive and takes up a sizable amount of physical space. G-Speak is an incredibly accurate system, but this accuracy comes at a great cost. Having eleven Vicon cameras is ideal in the field of research, but the required number of cameras varies with the application. Alps does not require eleven cameras, which would help cut down the cost. It has not been assessed exactly how many cameras Alps requires, but the number of cameras will vary depending on the complexity of gestures in the interaction vocabulary.

Ultimately, though, we are beginning to see cheaper gesture-tracking devices such as the Microsoft Kinect. There has been a lot of research surrounding the Kinect, particularly at the MIT Media Lab as well as MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL). Kinect was optimized for tracking full-body poses, but research has show that it can also be used to track smaller gestures such as fingers and hands. It is, therefore, a reasonable extension to the Alps project to replace G-Speak with the Kinect, and depending on the complexity of the gesture vocabulary, a Kinect should provide ample tracking accuracy to drive Alps.

Chapter 5

Future Work

As mentioned, the current version of Alps is merely a proof of concept. It is an approach at demonstrating the combination of tangible media and gesture tracking. That is to say that there is much more that can be done to both improve or extend the current version of Alps. This list is certainly not exhaustive, but it has been tailored over the course of development as reasonable and beneficial work to be done in the future.

5.1 A Different View

Currently, the selected mountain range or geographical location is centered on both the forward-facing display and Relief's table-top display. Another approach is to consider the forward-facing display and the table-top display to be a single, contiguous plane. To realize this connected view, when a location is selected, the location would be centered on the table-top display, but the forward-facing would, instead of replicating the same geography, display the background of the selected location. This approach may be more desirable depending on the perspective the user wishes to have on the selected area.

5.2 Multimodal Input

> "Gesture is also intimately related to speech, both in its reliance on the speech channel for interpretation, and for its own speech-like qualities" [2].

Alps focuses on communication and interaction using the hands. In order to further increase the bandwidth between the user and Alps, one could envision, as Buxton does in his quotation, the addition of speech recognition. With speech, the interface could be improved by, and certainly not limited to, the following:

1. More Intelligent "go to"

   A user could say the location the map should focus on, stating a city and state or a zip code. The map would then automatically jump to that location. Currently, you would have to navigate from one destination to another by multiple "go to" gestures. If the locations are far apart, this action could become very tedious and tiresome. The addition of speech recognition would allow you to easily jump from one location to another.

2. Switch Terrain Views

   Google Earth provides many different views, including satellite and street views. You can also create your own overlays or load them in to show more information if desired. These overlays may be more topographical information or demographics to name a few. It would be convenient to switch among many different views by speaking the name of the desired overlay.

5.3 Two-Way Communication between Google Earth and Relief

Currently, Alps only allows Google Earth to communicate to Relief. Relief itself was originally enabled to scale and pan terrain maps by manipulating the table-top wheel around the actuated pins. Currently, the feedback from the wheel does not feed

into Alps, but it could be adapted to extend the current functionality, both on the table-top and forward-facing projections.

5.4 A More Compelling Demonstration

One of the beauties of Relief is its ability to track the manipulation of its actuated pins and wheel. One of the early dreams of the Alps project was to allow the manipulation of terrain through physical interaction with Relief. Imagine the following two scenarios:

1. Manipulation of Time

   Using the table-top wheel, one could go forward and backward through time, tracking the changes in terrain over days and nights or even years. With an accurate data set, one could see the effects of time on a landscape, mountain range, or glacier.

2. Manipulation of Elevations

   By using gestures or physically pushing and pulling the pins, one could change the relative elevations of the selected terrain. The changes in the pins would be reflected in the graphical display, both table-top and forward-facing.

Ultimately, these two compelling demonstrations were not possible due to the lack of data. There was no dataset to make these scenarios convincingly come to life. But, we know that the inputs are possible, and it really only depends on an accurate, time-organized data set for a large enough area.

5.5 Microsoft Kinect

As has been mentioned many times in this thesis, the Microsoft Kinect is hopefully a gateway tool for the future of gesture recognition. It would be an interesting, if not logical, extension of the Alps project to replace the current G-Speak and Vicon

tracking system with a single (or a couple) Kinect(s). This would greatly lower the cost of the entire system, making it even more portable.

5.6 Multiple Location-Independent Instances

The current version of Alps is supported by a remote database (which will also be discussed in the following section). One of the benefits to remote storage is the ability to drive multiple instances of Alps. A single user could control multiple Relief tables. And better yet, the tables do not need to be on the same local network. One could be in Hungary while another is in Japan. The database would simply need to be modified with a few additional fields and tables to allow the addition of more Relief tables.

It is even conceivable to have multiple users, collaborating over the internet. There would need to be special consideration for concurrency issues, but with enough checks and balances one could have two or more people driving, in theory, any number of remote instances of Alps.

5.7 Removing the Remote Database

Another consideration for an extension is the removal of the remote database. Granted the removal of the remote database will affect the ease with which to implement multiple location-independent instances of Alps, as discussed in Section 4.1, the databases cause perceivable latency, which can dampen the user experience quite a bit depending on the variable lag time. One solution to this problem is to remove the database all together and drive all of the components from one central computer or by simpler channels of communication.

Alps does not use a single computer because the computer driving G-Speak is running the Linux operating system. The Google Earth JavaScript Plugin is not currently available for Linux. We were then forced to run Google Earth from other computers (Macintoshes in our setup). This introduced the issue of having to com-

municate between the C++ of the Linux G-Speak server to the in-browser JavaScript of Google Earth running on the Macintoshes. It was for this reason that we chose to use a remote database, namely a MySQL database. MySQL was chosen because we found it compelling to use a well-known database so as to make it easier to duplicate the work done in Alps.

Should there be a distribution of the Google Earth Plugin for Linux, we could drive it all from one computer and consider how to communicate between G-Speak and Google Earth.

5.8 Screens and Augmentation

Finally, there have been discussions to use screens or iPads instead of the forward-facing projector display. Inspired by Sandscape [8], we could replace (or supplement) the forward-facing display with a monitor that could display other views and information. One can imagine having a cross section of the middle of the terrain currently being viewed on Relief.

The second idea of integrating iPads gives the user the ability to augment the space and move around Relief to get the proper perspective. The iPad's position and orientation could be determined using the Vicon tracking system. The iPad's screen would act as a window into the augmented reality of the terrain displayed on Relief. The augmentation could be a cross section of the terrain, satellite imagery, tectonic information, etc.

Chapter 6

Conclusion

Alps is an attempt to bridge the gap between tangible and intangible user interfaces. In fact, Alps has show how tangible media can be improved upon by coupling the existing interfaces with a gestural interface. Adding gesture-tracking to projects like Relief allow for a substantial increase in the bandwidth of the conversation between the user and the computer as well as a strengthened and more diverse vocabulary.

In section 1.2.2, we raised two important questions:

1. How can we increase the data bandwidth between users and computers while decreasing time, given that we are starting to see more inexpensive solutions for gesture recognition?

2. What are the possibilities and limitations of combining tangible media interfaces with gestural input?

To answer these questions is not something that a single thesis can do. It will likely take years to gain the experience and collect the data necessary to determine the outcome of gestural input on tangible media. This thesis demonstrates one of many applications when we marry tangible media and gestures. We created project Alps to serve as both a reference and a motivation for those seeking to answer the questions above and those like them. We simply hope that the Alps project brings us closer to finding those answers.

Bibliography

[1] Richard Bolt. "put-that-there": Voice and gesture at the graphics interface. *SIGGRAPH Computer Graphics*, 14, July 1980.

[2] Bill Buxton. *Haptic Input*, chapter 14.1. 2009.

[3] MIT CSAIL. Finger detection demo.

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

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

[6] Scott R. Klemmer, Björn Hartmann, and Leila Takayama. How bodies matter: five themes for interaction design. In *Proceedings of the 6th conference on Designing Interactive systems*, DIS '06, pages 140–149, New York, NY, USA, 2006. ACM.

[7] Daniel Leithinger. Design and implementation of a relief interface. Master's project, Massachusetts Institute of Technology, Media Lab, June 2010.

[8] Ben Piper, Carlo Ratti, and Hiroshi Ishii. Illuminating clay: a 3-d tangible interface for landscape analysis. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, CHI '02, pages 355–362, New York, NY, USA, 2002. ACM.

[9] John Underkoffler, Daniel Chak, Gustavo S. Santos, Jessica Laszlo, and Hiroshi Ishii. The luminous room: some of it, anyway. In *ACM SIGGRAPH 99 Conference abstracts and applications*, SIGGRAPH '99, pages 167–, New York, NY, USA, 1999. ACM.

[10] M. Weiser. the computer for the 21st century. *Scientific America*, 265:94–104, 1991.

[11] Alan Wexelblat. An approach to natural gesture in virtual environments. *ACM Trans. Comput.-Hum. Interact.*, 2:179–200, September 1995.

[12] Jamie Zigelbaum, Alan Browning, Daniel Leithinger, Olivier Bau, and Hiroshi Ishii. g-stalt: a chirocentric, spatiotemporal, and telekinetic gestural interface. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, TEI '10, pages 261–264, New York, NY, USA, 2010. ACM.