# Designing Gunslinger: An Intermodal Large Display Interaction

by

Mingyu Liu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2015

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Statement of Contributions

In this thesis, I describe the work that I carried out with my supervisor, Professor Daniel Vogel, and a collaborator, Doctor Mathieu Nancel, over the course of my Master's degree. Content of this thesis is included from conference submissions that I co-wrote with Daniel Vogel and Mathieu Nancel. The system described in Chapter 3 and 4, along with the study described in Chapter 5, are the subject of a submission to CHI 2015. Content from the submission was used in these chapters. All chapters of the thesis may include some content from the submission, though I made extensive revisions and wrote new content to make this thesis a comprehensive whole.

I made the major contribution to the research described in this thesis. I designed and implemented the system of the research, performed the literature search, designed and conducted the experiments, wrote the experimental code, performed the statistical analysis, shot the demo video for our system, and participated in the drafting of the CHI submission and further revisions. Daniel Vogel advised on all aspects of the study and edited the submission video. Both Daniel Vogel and Mathieu Nancel wrote some content, created original figures, and revised the CHI submission.

## Abstract

In this thesis we introduce Gunslinger, a mid-air barehand interaction technique using hand postures to trigger command modes and small finger and hand movements for events and parameter control. Unlike past work, Gunslinger uses an 'arms down' body stance where both sets of fingers are tracked in mid-air with thigh-mounted sensors. This stance not only makes input more subtle and less fatiguing, but two-handed input and the reduced physical space needed to perform gestures makes it more compatible with large display touch input.

The design of Gunslinger follows guidelines for relaxed barehand input that ensure that users can interact comfortably in mid-air without sacrificing the expressiveness of the interaction technique. We also provide continuous feedback about the hand sensing and posture recognition to ensure that the user never has to switch his visual attention to understand the system's responses. An implemented interaction vocabulary is described for map navigation which demonstrates how Gunslinger can be combined with touch input supported by a touch hand inference method leveraging the arms-down form factor. And we show how this can be achieved with an input vocabulary that is equivalent, coherent, and compatible across mid-air and touch input modalities.

We conducted a four-part study to evaluate Gunslinger for resilience to Midas Touch, posture recognition quality with hand cursor feedback, distant pointing and clicking performance, and general usability for Gunslinger alone and when mixed with touch input.

We then present the results of the study which show that Gunslinger has little Midas touch, reliable posture detection, good pointing throughput, and acceptable usability, even compared to faster touch input. In addition, we implemented and evaluated an rollback mechanism in order to address a stability issue arising from the study.

Finally, we summarize our findings and describe extended studies to work on in the future.

**Acknowledgements**

Without question I could not have completed this thesis without help, and I am grateful for this opportunity to express my thanks to those who have aided me on this journey.

First, I must thank my supervisor Daniel Vogel, who has guided me through the process of designing and conducting research, and more importantly, whose enlightenment and encouragement have guided me into the world of HCI research and throughout the course of my Master's degree. I would also like to give my special thanks to Mathieu Nancel, a colleague and collaborator of my research, for his continual support and help in realizing our research ideas.

I also want to thank Ed Lank and Edith Law for being my thesis readers, and the other faculty and students of the ever-growing HCI research group. Mike Terry, Filip Krynicki, Valerie Sugarman, Adam Fourney, Jeff Avery, Yuexing Luo, Jingjie Zheng, Qifan Li, Yunjia Sun, and everyone else in the lab, I learned many things from you and always found you interested in my research and eager to help. I am grateful to have had the chance to share the experiences of academia with you.

I would also like to everyone who came into the lab and bared with me for hours to complete seemingly odd barehand tasks. I literally could never have done my research without your help!

Last but not the least, I would like to thank my parents Wenbin and Xueru, for their incredible support and enduring love for me.

## Dedication

This is dedicated to the ones I love.

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

**Figure 1.1: Traditional mid-air interaction with a wall display**
A doctor is viewing medical images using gestures with Kinect sensor fixed on top of the display. Source: Microsoft's "Kinect Effect" promotion video on Youtube.

Mid-air barehand interaction techniques typically large hand and arm gestures performed in front of the body with sensing cameras mounted on the display [5, 30](Figure. 1.1). Arguably, this is partly due to limited tracking capabilities and/or occlusion: large and explicit arm motions in front of the body are easy for sensors to "see", thus increasing the accuracy of tracked gestures and usability of the techniques. Moreover, the reason why people suggest these kinds of large frontal gestures in elicitation studies may have more to do with performance and legacy biases than inherent naturalness [28].

However, there is some significant room for improvement for these existing mid-air barehand techniques. One major problem is that these front gestures are prone to fatigue and lead to a feeling of heaviness in the upper limbs, a condition termed as the gorilla-arm [16]. Meanwhile, the interactions are often conspicuous and require generous physical space, making them difficult to perform when standing near a display. Furthermore, explicit front-of-the-body interactions are often perceived to be socially awkward when performed in public settings.

To address these problems, barehand gestures need to be smaller, more comfortable, and more socially acceptable. In other words, the interaction should be made more "subtle", meaning "fine or delicate in meaning or intent". To enable subtle gesture interaction, the system requires precise finger tracking with minimal occlusion, most easily achieved by mounting sensors on the body such as fingers [9, 8], hands [21, 27, 20], arms [19], shoulder [15], chest [23], shoe [1], etc.

However, many of these tracking solutions require cumbersome or invasive hardware and the focus of past work has not been on interaction subtlety.

Another consideration is how to combine mid-air interaction with large touch-enabled displays. Previous work followed the Proxemics principles when designing an intermodal interaction system, such as changing completely from mid-air gestures to touch input when near the display [32], or assigning specific functionality to each input modality based on spatial relationships [2]. These are all traditional multimodel approaches [26] for combining different input modalities. And our focus in this thesis is to treat mid-air gestures and touch more equally to let user choose the most suitable input method regardless of his or her proximity to the display.

This thesis explores ways to address the above issues and answers the following research questions:

1. Can we design a subtle and relaxed mid-air barehand technique that is easy to use and achieves high precision?

2. Can this barehand technique integrate other input modalities to enable cohesive and compatible intermodal interaction capability?

In this thesis we introduce Gunslinger, a mid-air barehand interaction technique using hand postures to trigger command modes and small finger and hand movements for events and parameter control. Unlike past work, Gunslinger uses a ‘arms down’ body stance where both sets of fingers are tracked in mid-air with thigh-mounted sensors (Figure 1.2). This stance not only makes input more subtle and less fatiguing [16], but two-handed input and the reduced physical space needed to perform gestures makes it more compatible with large display touch input. For example, Gunslinger can be used exclusively from a distance or mixed with touch input when near a display. This enables people to choose a physical location based on the level of detail they wish to observe rather than what functionality is available at that location. We show how this can be achieved with an input vocabulary that is equivalent, coherent, and compatible across mid-air and touch input modalities, partly realized with a Gunslinger-enabled hand inference technique.

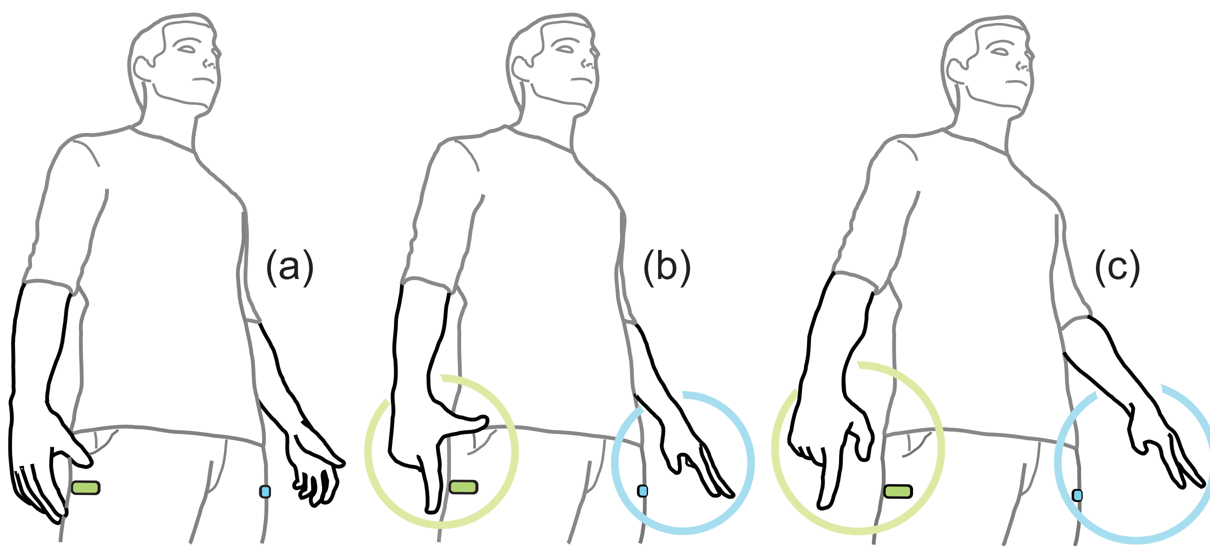



**Figure 1.2: Gunslinger metaphor**
From left to right: (a) both hands down in neutral posture; (b) command modes are entered by forming hand postures, such as a thumb and index finger on the right hand for pointing and two fingers on the left hand for zooming; (c) command events or parameters are provided with finger movements, like folding the thumb down on the right hand to click or moving the two fingers to zoom in or out.

To illustrate our system, we present a sample usage scenario for planning a trip itinerary using Google Map (different modes are illustrated in Figure 2.1):

*Jasper is making an traveling itinerary for his seven-day vacation. Firstly, to decide* *on which city to visit he takes a step back from the large display in order to comfortably browse the entire map area (Gunslinger Mode). Together using his left hand to pan and/or zoom the map and right hand to locate zooming center he successfully selects a city of interest. Then he moves closer to the screen to figure out traveling routes in detail (Touch Mode). He lifts his hands and uses touch gestures to pan the map and target all sorts of tourist attractions within the city. Occasionally he switches to Gunslinger pointing to select places on the map that is too far to reach on the screen (transition from Touch Mode to Mixed Mode). Having marked all the places of interest, Jasper takes a step back (return to Gunslinger Mode) and reviews his selections for the last time before generating and saving his itinerary with a Gunslinger command.*

This scenario showcases various Gunslinger functionalities: target acquisition via pointing and clicking, map navigation and menu selection via different command postures, close interactions via touch or a combination of both touch and Gunslinger, distant interactions via Gunslinger, and transitions between close and distant interaction based on proximity to the wall.

## 1.2 Contributions

We contribute:

- An arms-down barehand interaction space enabled by thigh-mounted commodity hand tracking sensors. Two sensors are mounted on both thighs and facing outward to track hand and finger gestures in a standing stance. This common yet novel arms-down controlling stance offers subtle and discrete tracking space and effectively avoids the problem of camera occlusion.

- A representative map navigation interaction vocabulary using Gunslinger with touch displays. The Gunslinger vocabulary follows a set of design principles focusing on subtlety, availability, eyes-free, and location independence without sacrificing precision and expressiveness. Additionally, the way we combine Gunslinger with touch interaction achieves high equivalency, coherency, and compatibility.

- A novel 'hand-cursor' to communicate recognized hand posture, command mode, and tracking quality. This real-time feedback is conveyed by the form of visual aids in varying shapes, sizes, and opacities surrounding the cursor.

- A touch hand inference technique made possible by the hand presence information reported from both sensors.

- A rollback mechanism for postures to make interaction more stable and reliable. This helps to minimize the occurrence of of high-speed “jerk” due to continuous sensing during intermediate state when user switches postures.

- An efficient, simple, and generic hand posture recognizer in the form of a nearest-neighbor classifier. The recognizer uses hand and finger features provided by the sensor and produces normalized similarity scores comparing the candidate posture against each vocabulary posture.

We conducted two studies to evaluate Gunslinger:

1. a four-part study to evaluate Gunslinger for resilience to Midas Touch, posture recognition quality with hand cursor feedback, distant pointing and clicking performance, and general usability for Gunslinger alone and when mixed with touch input;

2. a follow-up study to evaluate the general preference for the posture rollback mechanism.

Our results show Gunslinger: (1) is resistant to little Midas touch, which only occurs only 4.9% of the time when in Gunslinger stance; (2) enables reliable posture detection (100% detectable with visual feedback and 13.7% without feedback); (3) good pointing throughput ($497 + 483 \times ID,\ R^2 = .94$); and (4) acceptable usability with a median Likert scale for general impression above neutral, even compared to faster touch input.

## 1.3 Organization

The thesis is organized as follows:

- Chapter 2 describes the previous work on mid-air interaction techniques, including their design and application scenarios. It also introduces the Midas touch problem and summarizes previous research projects involving multiple input modalities.

- Chapter 3 describes the design and implementation of the Gunslinger system, including its setup, posture recognition algorithm and visual feedback.

- Chapter 4 describes the Gunslinger vocabulary in detail. It explains the design principles for combining Gunslinger with touch, an example of posture vocabulary for the map navigation system, a solution to mitigate the Midas Touch problem, and a hand inference technique.

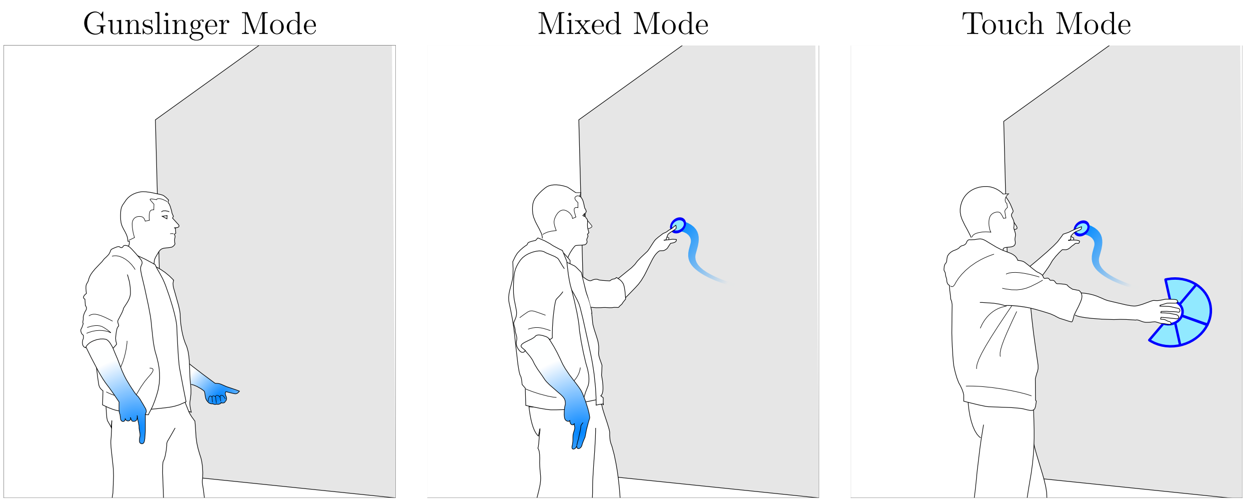



**Figure 1.3: Three Gunslinger Modes**
The interactions are governed by design principles (which we describe in Chapter 4) to minimize learning effort and enable free choice between input modalities. Left: Bi-manual Gunslinger mode where user interacts with pure arms-down gestures. Middle: Bi-manual mixed mode where user switch between left Gunslinger with right touch and right Gunslinger with left touch gestures. Right: Bi-manual Touch mode where user uses touch-only gestures for interaction.

- Chapter 5 describes the two studies we conducted to evaluate Gunslinger for resilience to Midas Touch, posture recognition quality, distant pointing and clicking performance, general usability in closed- and open-ended tasks, and the follow-up comparison study;
- Chapter 6 summarizes the contributions of the previous chapters and describes extended studies for future work;

# Chapter 2

# Related Work

Barehand, mid-air gestures – hand and finger movements performed in mid-air without holding any device – are well suited to interacting with large displays from a distance [33]. There is no device to retrieve or hold, so transitioning between distant and up close touch screen interaction can be fluid.

Interacting in mid-air has been studied in depth, especially with large displays. For related work we focus our survey on proposed and realized applications for mid-air barehand interaction techniques. Although some previous technologies could in theory be used arms down, there has been no exploration of a full interaction vocabulary performed from an arms-down stance explicitly focusing on subtlety.

## 2.1 Environment-fixed Mid-air Input

One common approach to track mid-air motion is to use environment-fixed camera sensors. Motion tracking systems using markers can reliably track fingers in a large space from a distance. Vogel *et al.* [33] used hand-mounted passive markers tracked by a VICON camera to detect hand orientation and postures in order to control mid-air pointing, clicking, clutching and mode switching on large, high resolution displays. However, this level of environment-fixed tracking without markers remains challenging. While sensor capabilities will improve, tracking issues when hands are occluded by other objects or people, or self-occlusion by other user's body parts, will not go away. The usual solution is to require people to perform large, explicit hand and arm gestures in front of their body to make tracking easier and reduce the chance of occlusion. Kinect(Figure. 1.1), a motion sensing device introduced by Microsoft in 2010, is a popular choice of environment-fixed sensor to track large body movement.

In summary, current environment-fixed cameras cannot reliably track bare fingers with high precision from a distance over a large area and they will always be susceptible to occlusion by other objects or people, or self-occlusion by the hand itself. In addition, when standing close to the display these large gestures might become impossible to perform as the tracking space is constrained to the camera's angle of view.

## 2.2 Hand-mounted Mid-air Input

To make barehand mid-air gestures "smaller" so they become more comfortable and socially acceptable, researchers have explored the idea of mounting sensors near the fingers for more accurate tracking to achieve high-precision hand tracking with minimal occlusion.

Interactive data gloves [21] have been used for decades to detect hand postures and gestures in mid-air for a variety of uses including virtual [20] and augmented reality [27]. Depend on the level of sophistication, these gloves could be based on contact using conductive patches or based on flexure using fiber-optic, mechanical or piezoresistive sensors to track finger and thumb movement information. The use of interactive gloves successfully resolves the problem of self-occlusion using fixed sensors at the expense of introducing a small inconvenience to the use. Kim *et al.* introduced Digits [19], a high-resolution sensing device mounted on the inside of the user's forearm that allows precise 3D reconstruction of hand postures. Unlike data gloves, Digits works without the need for full instrumentation of the hand. The authors claimed the system targets mobility and is designed to be low-power and easily reproducible using only off-the-shelf hardware.

More recently, smaller finger-mounted devices have been used to track subtle hand and finger motions with high mobility. FingerPad[8] uses a nail-mounted magnetic tracking device that turns the tip of the index finger into a touchpad, allowing private and subtle thumb interaction on the move, while uTrack[9] uses a pair of magnetometers to be worn on user's thumb and index finger to track fingertip movement in 3D.

While very accurate, these technologies can be cumbersome or prevent users from using touch-enabled surfaces. Moreover, while some technologies enable more subtle, arms-down input, this has not been an explicit goal of their work. A more related touch interaction example is PocketTouch [29], a technical proof-of-concept for a modified capacitive sensor placed in a pocket to enables arms-down touch interaction. However, despite the convenience of not having to remove handheld device from the pocket, the interaction is limited

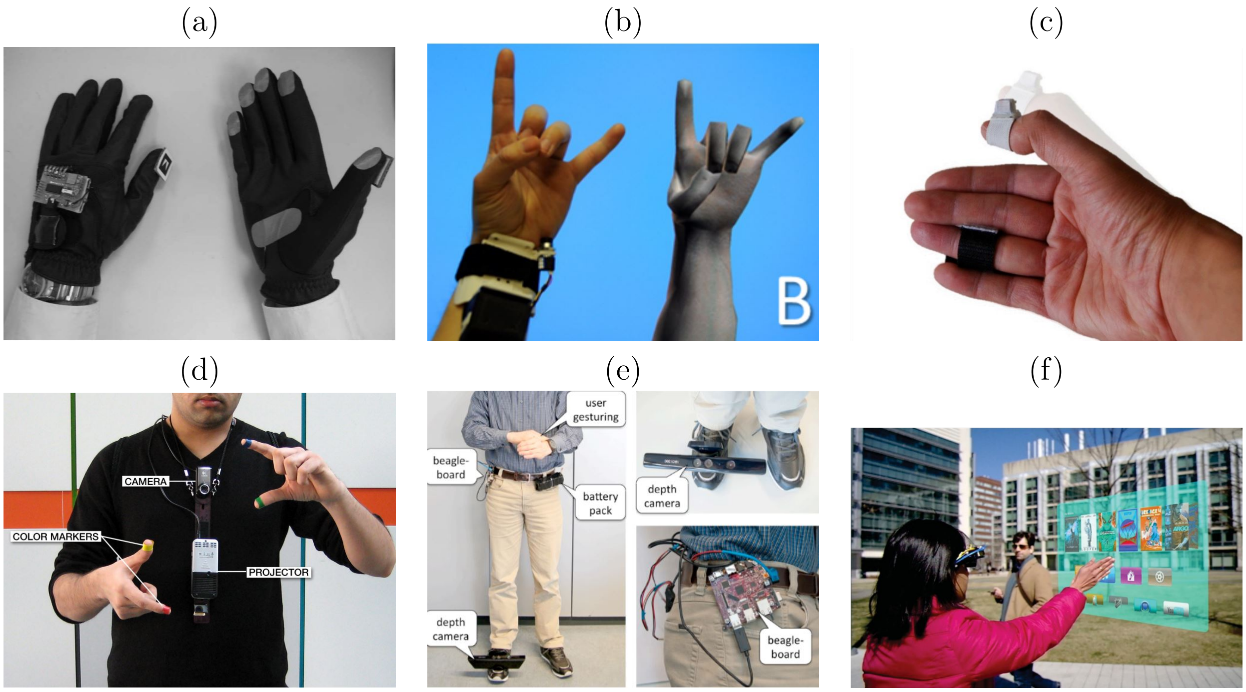



**Figure 2.1: Example of interactive systems from previous work**
From top left to bottom right: (a) Data gloves [27, Figure 1]; (b) Digits [19, Figure 1] ; (c) uTrack[9, Figure 1]; (d) SixthSense [23, Figure 1]; (e) ShoeSense [1, Figure 5]; (f) Mime [11, Figure 1]

to finger-strokes and cannot handle more sophisticated interaction such as pointing and clicking on a large display.

## 2.3 Body-mounted Barehand Input

Researchers have also been invesgating ways to mount sensors on other parts of the body to track finger motions with minimal occlusion. SixthSense [23] uses a chest-mounted projector to display content on any surface facing the user and detects hand gestures in front of the user with a camera thanks to finger-worn color markers. The authors suggested that this wearable gestural interface could provide augmented information to the physical world and allow the use of natural hand gestures for interaction. Similar to SixthSense, OmniTouch [15] combines a depth camera with a small projector to detect the users' hands and project content over any surface in its range (e.g., walls, held objects, and their own bodies). A depth-driven template matching and clustering algorithm is used for multitouch finger tracking to enable highly-available interactive capabilities. ShoeSense [1] uses shoe-mounted depth sensor to detect hand motions from below, and can recognize visual features like pinching or the number of fingers held horizontally in front of the user. This placement of a camera on the shoe enables discreet as well as large and demonstrative hand gestures in front of the body. Three subtle gesture sets were designed for users to perform quick gestures without drawing too much attention. Head-mounted devices have also been studied to achieve freehand interactions. Mime [11] comprises a battery operated active illumination three-pixel time-of-flight sensor, and a 2D RGB camera and supports unencumbered single-handed gestural interaction.

These technologies and techniques allow true barehand input with different levels of precision, but the sensors themselves can be heavy and quite conspicuous. The mounting point of sensors is an important consideration, placing sensors directly on the fingers is cumbersome for example, and solutions like chest, shoe, head, or shoulder mounted cameras still force most gestures to be in front of the body.

## 2.4 Combining Mid-air Gestures with Touch

Multimodal interaction was originally defined as different modes of communication [26], but a more encompassing definition is modalities as input devices [31]. Our interest is in input-based multimodal interaction for large wall displays, specifically using mid-air gestural and touch input.

The classic form of multimodal interaction can be traced to Bolt's seminal work "Put-that-there" [4]. Bolt combined raycast pointing with speech recognition to illustrate how each modality can amplify, modify, and disambiguate the other. This inter-dependence between multiple modalities as they work together has become the primary way multimodal interfaces are conceived. Oviatt's highly cited article states "Well-designed multimodal systems integrate complementary modalities to yield a highly synergistic blend in which the strengths of each mode are capitalized upon and used to overcome weaknesses in the other" [26].

As such, previous work combining mid-air gestures and touch have considered them two mutually exclusive input modalities. They applied the principles of Proxemics, where the input possibilities change based on spatial factors like distance. For example, Vogel and Balakrishnan [32] use mid-air gestures for mode selection or to turn tracking on and off when away from the display. As the user approaches, new functionalities are made available for touch and mid-air gesture functionalities are progressively removed. Another Proxemic mid-air and touch system by Ballendat *et al.* follows a similar pattern of assigning functionality to each modality based on spatial relationships [2].

Proxemic interactions help to ease privacy concerns, as user's accessibility to functionalities and datasets depends on his or her physical distance to the display and to other users, the user has little or no control over modality choice. However in contexts with little or no privacy issues, e.g. in work or gaming environments, this approach constrains the set of available commands to the physical location of the user; yet that same location could reflect only the level of detail that the user wishes to observe rather than the set of commands he or she plans to use. Consider entering text with speech or keyboard. Alternating between typing one word, then dictating the next word would be difficult. Even functionally equivalent modalities like these are usually assigned specific roles such as entry and error correction [31]. One reason why combining some modalities is difficult is because the way each input modality is performed is divergent (consider speaking vs. typing) and the command mapping incongruent (speaking 'copy' vs. typing 'Ctrl-C'). For this reason, we focus on intermodal interaction using mid-air gestures and touch since they have enough similarity that making their input consistent and mapping congruent is possible.

Bragdon *et al.* [5] is the closest example in the literature to our approach. They combine mid-air and touch interactions, public displays, and personal devices including mobile devices and laptops together to support co-located, small group developer meetings by democratizing access, control, and sharing of information across devices. Although some functionality is available across input modalities, such as distant pointing and touch, Bragdon *et al.* explicitly state a design principle that "Each modality should have a separate use." (Table 1).

## 2.5 Midas Touch Problem

The Midas Touch problem [17] occurs when the input channels and vocabulary for a particular interaction technique cannot be easily differentiated from normal, non-interactive user actions. Gesture-based interactive system, in particular, needs to find a way to diminish the problem as continual sensing would likely cause false positives and render the system dysfunctional. ShoeSense [1] (Figure 5.1) uses a specific pinch posture as a registration pose to active the system interaction. Pinch is a perfect delimiter posture as it is easy to perform, easy to recognize, and different from daily life gestures. However, this approach is less ideal for Gunslinger as one of our goals is to make the system *always available*

Alternatively, CHARADE [3] (Figure 5.1) limits the effective sensing range by only responding to user interactions within an "active zone" constrained by the projector's projection of the display on the screen. This, again, does not work for Gunslinger as it further limits the already restricted sensing space due to the mounting point and arms-down stance.

(a) (b)

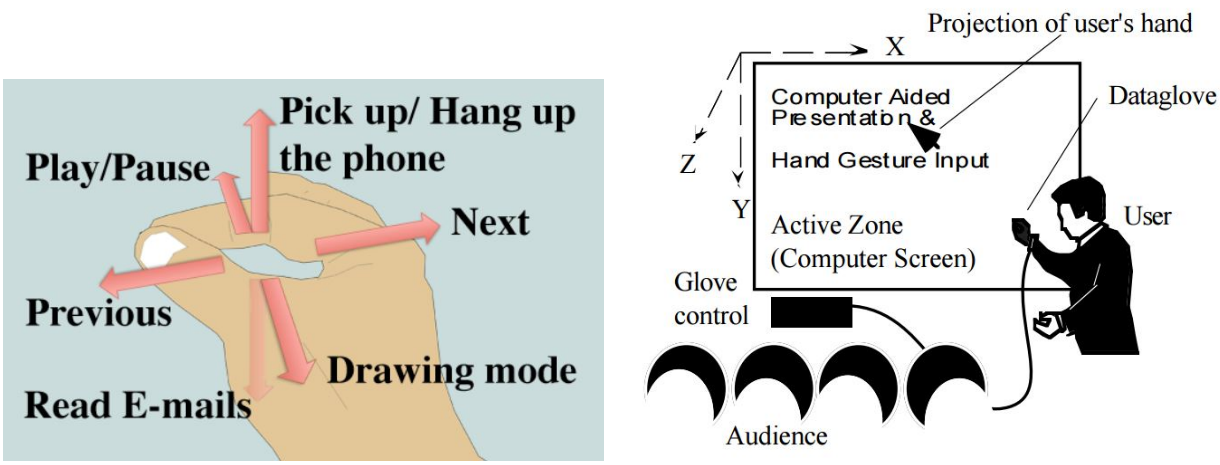



**Figure 2.2: Approaches to reduce Midas Touch**
(a) pinch registration pose as a delimiter by ShoeSense [1, Figure 3] (b) active zone by CHARADE [3, Figure 3]

# Chapter 3

# Gunslinger

Gunslinger is a mid-air, gesture-based interaction technique that is controlled arms-down using wearable 3D cameras on the user's thighs, thus avoiding problems of camera occlusion without cluttering the user's hands. The Gunslinger name refers to the holster-like placement of the 3D cameras and the quick transition to active command postures like index-and-thumb pointing (Figure 1.2). The finger and thumb pointing is reminiscent of how cowboy 'gunslingers' drew their guns in classic Hollywood films. As discussed in related work, many previous barehand interaction techniques use large, coarse gestures to achieve a high level of input expressiveness. Others enable smaller, more subtle gestures, but sacrifice expressiveness due to sensing capability or enable more subtle expression, but rely on intrusive sensing hardware. A general requirement for gesture-based interaction is that it should minimize fatigue and muscle strain for long periods of use, but not at the cost of lower expressiveness.

The goal of the Gunslinger concept is to enable subtle, but highly expressive interaction. Following are the five principles we have set when designing the Gunslinger technique:

- *Relaxed* – Mid-air input should keep large muscles as relaxed as possible to reduce fatigue.
- *Precise and expressive* – Mid-air input should support a broad range of precise and controllable input tasks.
- *Always available* – Providing input should possible without performing a universal input delimiter and without Midas Touch false-positives.

- *Eyes-free* – The user should not have to look at their hands to understand system state or recognized responses.
- *Location independence* – Gesture articulation and sensing should be feasible regardless of nearby obstructions.

An arms-down posture satisfies *relaxed input* in terms of arm fatigue. Mounting 3D cameras on both thighs enables tracking *precise and expressive* finger movements and hand postures performed with both hands. Relaxed, natural postures (e.g. relaxed fist, open hand) are reserved for the neutral system state to avoid Midas Touch and commands are *always available* with a short transition to specific command hand postures. Additional feedback indicates the recognized hand posture, current command mode, and tracking bounds for both hands for *eyesfree* input.

We now detail the design, technological and technical choices behind our implementation of Gunslinger relative to the design guidelines described above.

## 3.1 Hand Tacking from the Thighs

To enable hand tracking when arms are down on the sides, we attach a consumer Leap Motion (LM) device to each thigh just below the hips (Fig. 3.2). The LM device is a commercially available 3D camera with hand tracking software originally intended to be placed on a physical desktop, facing upward and for static use only. The effective range of the Leap Motion Controller extends from approximately 25mm to 600mm above the device. With a dimension of 13mm × 13mm × 76mm and weight of 45 grams, the LM device sits perfectly on the thighs without drawing too much attention. APIs in various programming languages are provided along with the development toolkit. The LM system employs a right-handed Cartesian coordinate system with the origin centered at the top of the device(Fig. 3.1). Hands and fingers are tracked with millimetre accuracy within a volume approximately .25 $m^3$ and software reports the size, orientation, and positions of the palm and fingers.

The 3D cameras face out to enable high resolution finger tracking in a comfortable area when the arms are down, making input gestures feasible even near walls, other users, or displays to achieve *location independence*. Since the cameras can only detect hands when arms are down, communicative gestures such as waving are outside of the sensing range and are ignored. This further reduces Midas Touch.

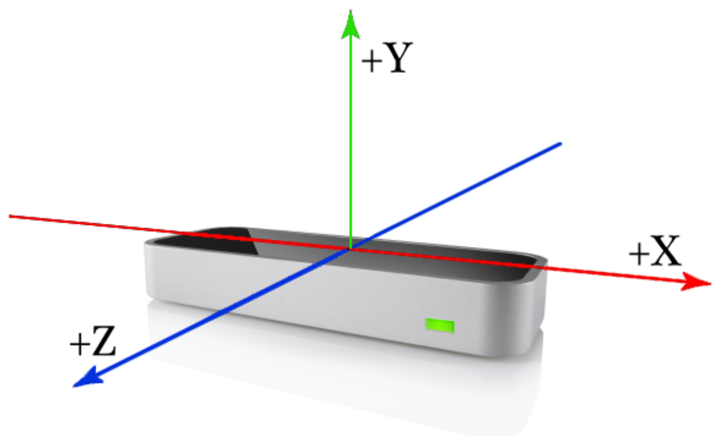



**Figure 3.1: The Leap Motion right-handed coordinate system**
Source: Leap Motion Development Document

### 3.1.1 Artificially Limited Control Area

The 3D cameras have a field of view of about 150 degrees with a an effective sensing range from 25mm to 600mm above the devices. Preliminary tests found that although the 3D cameras enable high-precision finger tip pointing with small wrist tilts, untrained users often used elbow and shoulder rotations to perform the same fingertip motions. Such movements would occasionally bring the user's hand outside the sensing range, thus losing track of an on-going interaction.

To encourage users to stay inside the actual sensing range and to keep a relaxed posture, we limit our own input to a smaller area that what the LM 3D cameras support. We use a 12-cm radius disc located 15 cm away from the sensor along the z-axis (xz in Fig. 3.2). Our system limits input to palm positions projected along the y-axis that fall inside this disc. The size and location was tuned for a comfortable interaction range without shoulder strain and we can use the remaining actual sensor input space to provide feedback of sensing range. This limited sensing range also helps filter out input interference, for example nearby surfaces and or other people's hands in collocated collaboration contexts where other people.

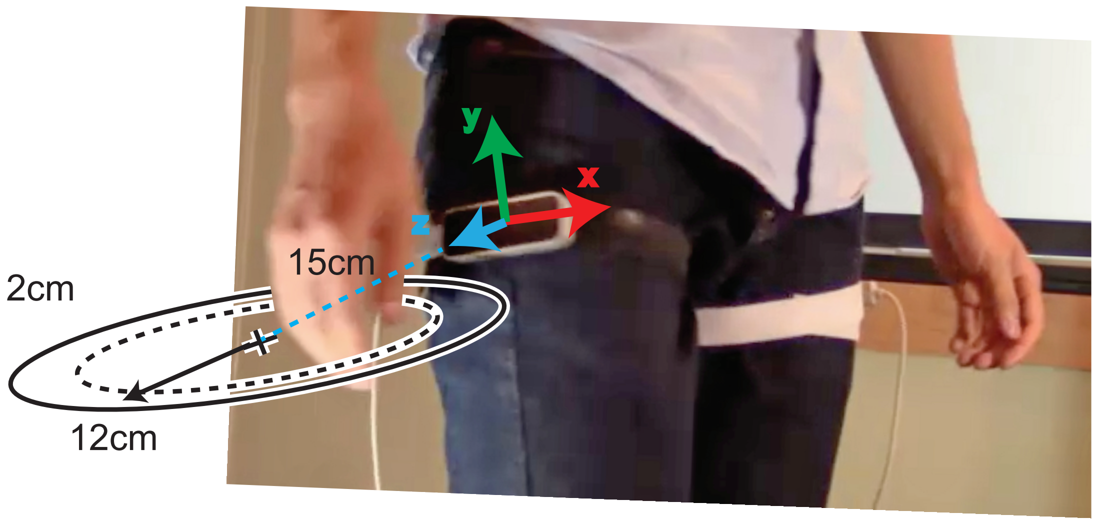



**Figure 3.2: Gunslinger Setup**
Leap Motion sensors are mounted on each thigh and x-y-z referential is defined for the input space. The circles represents the control range defined for Gunslinger.

## 3.2 Hand Posture Recognizer

Gunslinger uses discrete hand postures to activate interaction modes with subsequent hand or finger movement to issue commands or specific continuous or discrete 1D and 2D parameters. Our postures are defined by which fingers are raised or folded and whether the thumb is stretched out, aligned with the palm, or tucked into the palm. This combination of posture and movement creates a reasonably *precise and expressive* interaction language and forming hand postures can be done in a *relaxed* manner *eyes-free*.

We describe the two steps for the recognizer: filtering and recognition.

### 3.2.1 Filtering

Since the LM algorithms are tuned for desktop usage and hand tracking works best best when the controller has a clear, high-contrast view of an object's silhouette, we found we had to introduce additional heuristics to compensate for mis-reported measures:

- a hand continuously visible for less than 0.2s is filtered out to eliminate background noise and flickering;
- each LM device tracks one and only one hand object per frame;
- the left thigh-mounted camera can only see left hand thus hand object recognized as “right” is invalided from the frame (the rule is inverted for right camera);
- extended fingers with tips less than 10 mm apart are collapsed into a single finger to handle the frequent occurrence of the LM reporting one finger as two fingers stuck together.

We apply these filtering rules on data reported from LM devices before feeding them to the posture recognizer.

### 3.2.2 Recognition

Using these corrected features, we designed an efficient, simple, and generic finger posture recognizer in the form of a nearest-neighbour classifier. It uses a normalized similarity score $s_i$ between the features of the current finger posture ($C$) and each finger posture in a vocabulary ($V_i$). Given $C$, two values are computed for each $V_i$ by the two inner while loops in Algorithm 1: $n_i$ is the absolute difference in the number of raised fingers; $o_i$ is the distance, expressed in number of fingers, between the pattern of raised fingers in $C$ and $V_i$. For example, if the Pinky is raised instead of the Index finger, then $o_i = 3$ (fingers away). In more complex situations where more than one digit is mismatched, $o_i$ only considers the worst (i.e. most distant) mismatched digit. In practice, $o_i$ primarily detects when a digit is mistakenly raised instead of another and reinforces $n_i$ when the numbers of raised digits do not match.

The similarity between $C$ and $V_i$ is the weighted sum of these normalized values: $s_i = w_n \frac{n_i}{4} + w_o \frac{o_i}{4}$. $w_n$ and $w_o$ are constant weights set to .35 and .65. Lower values indicate better matches, so $C$ is recognized as $V_i$ if $S_i = \operatorname{argmin} S_i \mid S_i < \theta$. We found that a threshold of $\theta = .2$ provides accurate recognition without being overly restrictive.

**Algorithm 1** Calculate score $s_i$ between $C$ and $V_i$.

```
//first check if frame is empty
if all fingers are tucked in then
  //set the score of Fist Posture s to be 0 and reset everything else
else
  noiseFilter.apply(fingers) {data cleaning}

  //begin score calculation
  //iterate through the vocabulary postures
  i ← 1
  while i <number of vocabulary postures do
    //iterate through index to pinky finger
    d ← 1
    //calculate n_i for each of the mismatched finger
    while d <= 4 do
      //isMatch() compares the difference of this finger between candidate and vocab-
      ulary posture:
      n_i ← n_i+isMatch(i,d,fingers[d])
      //add each mismatches to diffArr[d]
      d ← d + 1
    end while
    //calculate o_i for each of the mismatched finger
    d ← 0
    while d < numMistatches do
      dist1 ← min(dist1, distance to nearest mismatched finger)
      dist2 ← min(dist1, distance to nearest correct finger)
      d ← d + 1
    end while
    o_i ← max(dist1, dist2)
    //final weighted score combining n_i and o_i
    s_i ← w_n (n_i/4) + w_o (o_i/4)
    i ← i + 1
  end while
end if
```

Once the finger posture is determined, we classify the thumb state using the normalized distance between the thumb tip and the index metacarpophalangeal joint (i.e. knuckle).

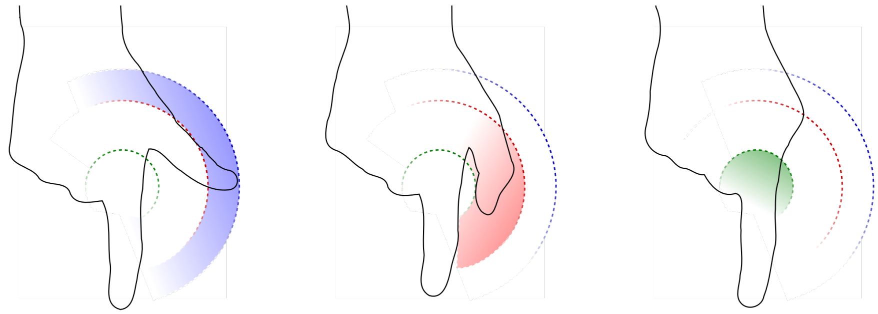


**Figure 3.3: Thumb states**

. Left in blue: thumb movement away from the hand, *thumb up* ; Middle in red: thumb movement near the hand, *thumb down*; Right in green: thumb tucked into the hand *thumb hidden*

The maximum thumb distance is calculated in a short calibration step recording the thumb tip position in an open hand and a clenched fist. We then classify the thumb as *up*, *down* or *hidden* using distance thresholds of 75%, 50%, and 25% ($w_{down}$, $w_{active}$, and $w_{up}$) with an additional transition hysteresis adjustment of 10% ($w_{adjust}$) (Figure 3.3). Following is the pseudocode for determining the thumb state.

**Algorithm 2** Determine the thumb state

//check current thumb state and apply different threshold accordingly
**switch** ($currentState$)
case "$thumbUp$":
**if** $thumbDis > (w_{up} + w_{adjust}) * maxDis$ **then**
  $currentState \leftarrow$ "$thumbDown''$
**end if**
case "$thumbDown$":
**if** $thumbDis > (w_{down} + w_{adjust}) * maxDis$ **then**
  $currentState \leftarrow$ "$thumbTucked''$
**else**
  **if** $thumbDis < (w_{active} - w_{adjust}) * maxDis$ **then**
    $currentState \leftarrow$ "$thumbActive''$
  **end if**
**end if**
case "$thumbTucked$":
**if** $thumbDis < (w_{down} + w_{adjust}) * maxDis$ **then**
  $currentState \leftarrow$ "$thumbDown''$
**end if**
**end switch**

## 3.3 Subtle Pointing

Direct manipulation through pointing and clicking remains by far the dominant interaction paradigm in conventional user interfaces [33]. In an effort to make pointing subtle and effortless for large display interactions, we designed a pointing algorithm that uses the index finger for pointer control and the thumb for clicking.

First, we noticed a significant amount of noise in the data reported by LM device, which causes cursor jitter. To minimize this visual disturbance, we used the 1 Euro Filter [7], a noise filtering algorithm to stabilize noisy signals. Besides its easy implementation, one advantage of 1 Euro Filter this is that is uses a first order low-pass filter with an adaptive cutoff frequency: a low cutoff stabilizes the signal by reducing jitter but is increased to reduce lag as speed increases. The filtered index fingertip position is then projected in the xz-plane before transforming to the coordinating system of the wall display as illustrated in Figure. 3.4.

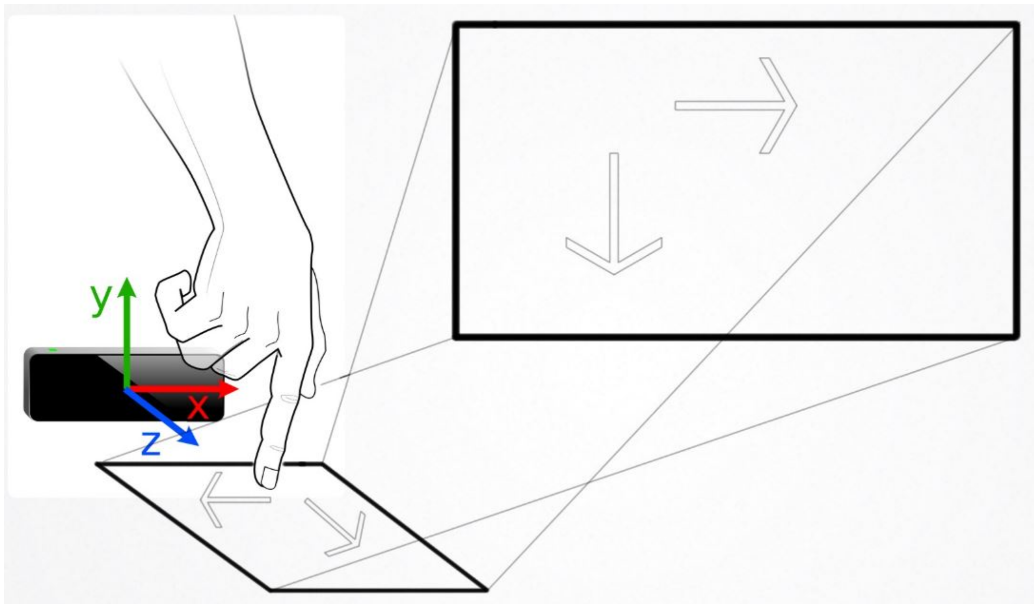



**Figure 3.4: The mapping between LM's and the wall's coordinating system**
The index fingertip position is projected in the xz-plane and transformed onto the wall display

We conducted a pilot study to compare the performance of our arms-down pointing with traditional front pointing. While there was no significant difference in terms of speed and accuracy, we observed that some participants had to take multiple clutches for long distance target acquisition. To minimize clutching and make the pointing more efficient, we adopted Nancel *et al.* [24]'s Control-Display (CD) transfer function. Their transfer function uses a sigmoid transfer function that can be characterized by a slope that smoothly gets steeper before decreasing again. To model such curves, we use the same logistic function of Nancel *et al.*'s:

$$CD(x) = \frac{CD_{max} - CD_{min}}{1 + e^{(-\lambda(x - V_{inf}))}} + CD_{min} \tag{3.1}$$

$$\text{where } V_{inf} = ratio_{inf}(V_{max} - V_{min}) + V_{min}$$

This function can be tuned with six parameters: $V_{min}$ and $V_{max}$ bound the lower and upper velocities of finger tip motion; $ratio_{inf}$ sets the inflection point within this range; $CD_m in$ and $CD_{max}$ defines the upper and lower CD gains; and lastly $\lambda$ defines the steepness of the curve. As a result, the lower slopes at each end of the curve enable high precision at low input velocities and bound cursor speed. After tuning, we found the following set of values for the parameters work best for Gunslinger: $V_{min}(10\ mm/s)$, $V_{max}(400\ mm/s)$, $CD_{min}(0.2)$, $CD_{max}(23)$, $ratio_{inf}(0.55)$, and $\lambda(0.01)$. With this transfer function a user is able to traverse the large screen with a quick wrist turn but still achieves high precision on small targets with subtle finger movement.

## 3.4 Visual Feedback

On-screen feedback provides information about how the system has classified the current hand postures, what commands (if any) are triggered, and notifying when either hand is nearly or completely out of sensing range. This follows our *eyes-free* guideline and also helps people learn the interaction vocabulary itself. Gunslinger accomplishes this by decorating two 'hand cursors' with associated feedback (Figures. 3.6, 3.7, 4.1).

The cursors already serve as direct manipulation feedback (e.g. pointing location, selected menu item) making them a natural focus of attention. The dominant hand cursor also functions as a positional pointer and the nondominant cursor is fixed near the bottom left of the display since it is associated with non-positional controls such as zoom and (rate-controlled) pan. For very large displays, the nondominant cursor could follow the dominant one like a trailing widget [12] to minimize visual distance. One exception would be bimanual tasks where both cursors point, such as scaling objects from two corners. The cursors and feedback are black and white with contrasting outlines to provide maximal contrast above any background image (Figure 3.5).

### 3.4.1 Posture and command feedback

#### Hand posture feedback

People know how they hold their own hand through proprioception, but the way the system sees these postures might differ. For example, aspects like sensor errors, posture recognition thresholds, or misaligned frames of reference could cause recognition errors that might be easily corrected with slight adjustments in posture. We provide real-time, discreet visual feedback about how the LM device perceives hand postures in the form of a hand proxy ring, a stylized graphic of a hand surrounding the cursor (Figure. 3.6). The ring has bumps representing flexed or raised fingers as perceived by the LM device. Depicting the user's hands also disambiguates which cursor corresponds to which hand in bimanual pointing configurations, from the orientations of the thumbs.

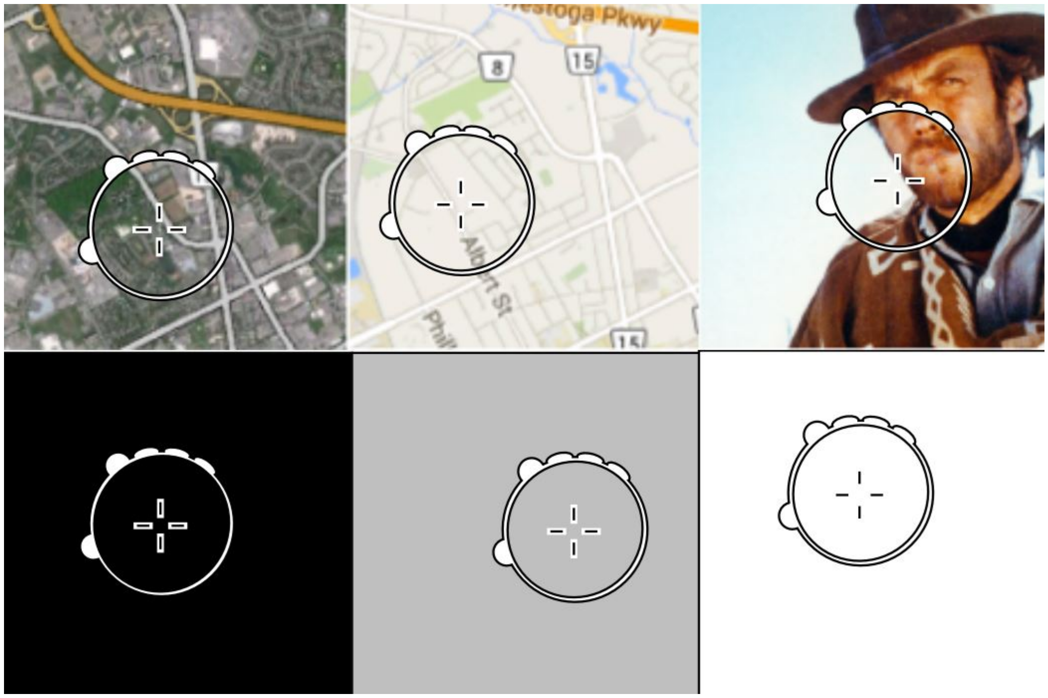


**Figure 3.5: Gunslinger cursor in different background**

#### Command mode feedback

A central icon represents the current pointing hotspot of the cursor and the currently recognized command mode.

For example in our demonstration system, a four-branched sight represents the pointing state, an eight-branched sight represents the pressed-down state for clicking and dragging, a circle represents the clutching state and a dot represents the neutral state (see Figure. 3.6).

### 3.4.2 Sensing limit feedback

Due to the physical limit of sensing, a user has to be constantly reminded of the hand location to avoid accidentally exiting sensing range and losing control of the system. To address this problem and achieve an eyes-free experience, we designed a feedback mechanism located inside the ring cursor to keep the user aware of their hand position. These visual aids convey when a hand is about to leave, or has left, the artificially limited ‘disc’

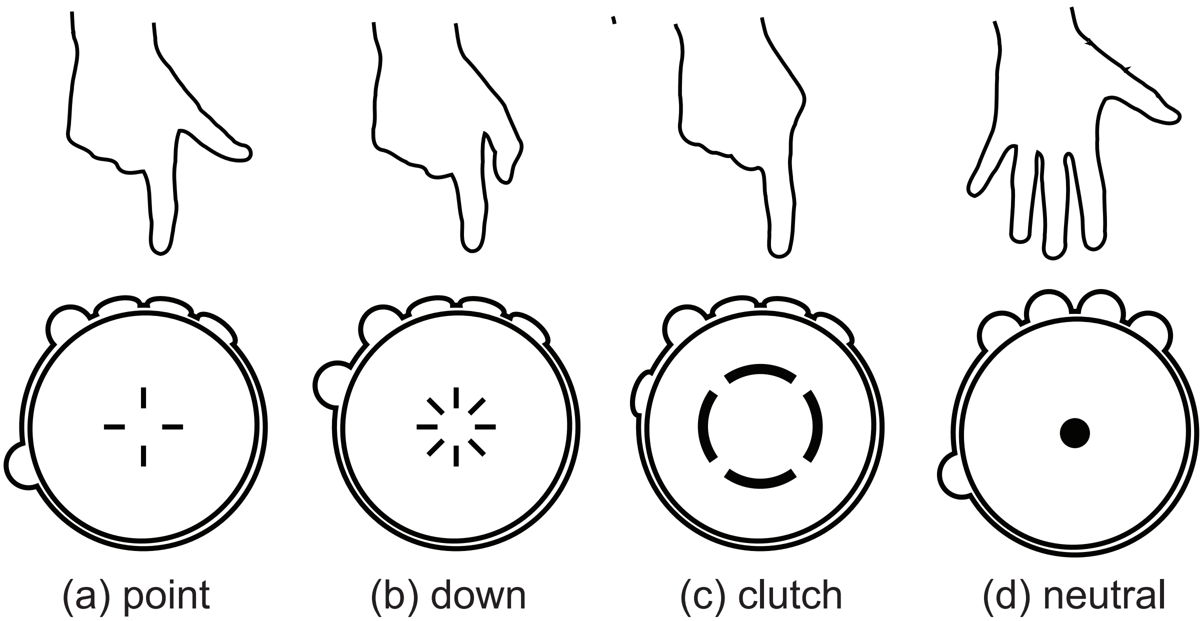



**Figure 3.6: Hand cursor states**
Hand cursor represents state of digits according to sensor as bumps on a ring, centre icon represents command state, e.g. (a) pointing; (b) thumb click down; (c) thumb tuck clutch; (d) open hand neutral. The thumb bump slides along the ring as thumb moves. All the bumps re-size when the corresponding digit is extended or folded. The cursor is drawn by black and white strokes so it would work in both dark and light backgrounds.

sensing range. When the projected hand position is more than 80% from the centre of the disc (i.e. outside the dashed circle in Fig. 3.2), two changes occur.

1. the cursor's opacity decreases linearly from 100% to 20% corresponding to the outward 20 % of the control range; the cursor never disappears for easy recovery.

2. in that same outward 20% range, the surrounding hand shape begins to shift from the centre icon in the opposite direction of the range bounds. This animated offset makes the cursor feel like it is approaching the bounds and indicates the direction where tracking will improve. The centre icon does not shift to maintain direct manipulation feedback.

These two visual aids, along with the changes of bump size on the cursor ring, serve as a simple way convey what the LM device actually "sees" and helps the user adjust their position or posture accordingly without having to look away from the display.

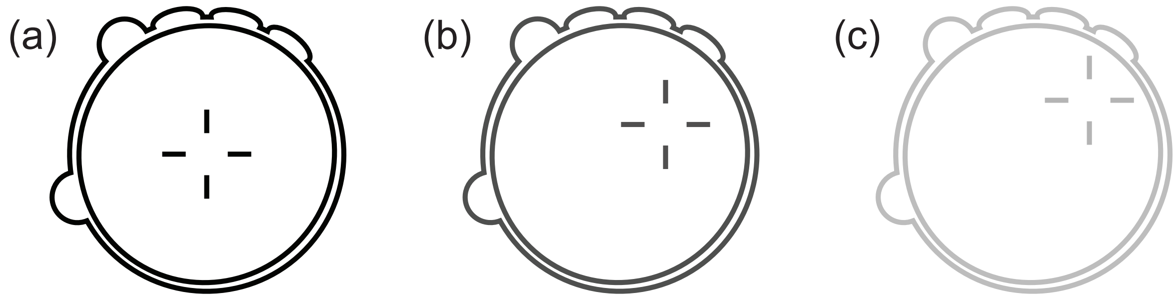



**Figure 3.7: The Change of Opacity and Shifting**
Hand cursor uses opacity and shifting of centre icon to convey movement towards sensing limit: (a) more than 20% from all sensing limits; (b) 15% from bottom-left limit; (c) at bottom-left limit.

# Chapter 4

# Interaction Vocabulary

As a proof of concept demonstration, we designed a Gunslinger interaction vocabulary for annotating and navigating a map on a large touchscreen display (Figure 4.3). The vocabulary investigates a variety of control types (absolute/relative, direct/indirect, position/rate control), and shows how Gunslinger can be effectively combined with touch input.

## 4.1 Enable intermodal interaction

As mentioned in Related Work, this thesis explores a way to enable intermodal interaction by designing an equivalent yet coherent set of functions for both Gunslinger and touch interaction. As an extension to the five general Gunslinger design principles, we add three principles for combining Gunslinger with large touch displays. The goal is to minimize learning effort and enable free choice between input modalities.

- *Equivalence* – a common set of functionality should be fully controllable with Gunslinger and touch (e.g. pointing using touch or using Gunslinger). This enables people to step back to get an overview and still accomplish the same tasks with Gunslinger.
- *Coherence* – Gunslinger and touch should share morphological or semantic aspects. This can be external coherence (e.g. Gunslinger uses similar established input conventions like two finger drag for scrolling) or articulation coherence such as mapping to the same hand (e.g. left hand navigates, right hand points with Gunslinger or touch)

or mapping to similar postures (e.g. two fingers opens a menu with Gunslinger or touch). This helps transfer learning between mid-air and touch modalities.

- *Compatibility* – The requirements for space and tracking should support the simultaneous usage of Gunslinger and touch when close to the display (e.g. right hand points with touch while left hand navigates with Gunslinger, or vice-versa). This allows the combination of mid-air and touch to accelerate tasks (e.g. left hand Gunslinger navigation with right hand touch) or complete difficult tasks (e.g. right-hand points with Gunslinger to reach distant targets while left hand navigates using touch).

## 4.2 Map Navigation Vocabulary

The Gunslinger and touch interaction vocabulary enables panning and zooming, selecting map style, defining landmarks through pointing, calculating an itinerary between these landmarks, and defining a zone within these landmarks (illustrated in Figure 4.1). Any hand posture not included is an inactive neutral state, the most common postures to form naturally, the open hand and closed fist, are neutral. This vocabulary is designed to provide *equivalent* functionality between Gunslinger and touch.

Hand and posture mappings are designed for *coherence* across modalities. The number of fingers for a Gunslinger posture and number of multi-touch contacts both map to the same functionality:

- One finger for pointing and panning by extending the index finger. The pointing is controlled by finger tip position and panning by the palm position. In order to reduce unwanted panning, the system only starts to register panning if the palm moves away from the initial starting position more than 3cm.

- Two fingers for invoking menus and zooming by extending index and middle fingers. Again, to prevent unnecessarily sensing, zooming will not be registered until the palm is 3cm away from the starting position.

- Four fingers to undo-redo by extending the index to pinky finger. The undo/redo action will be triggered once the palm passes a distance threshold of 8cm from the starting position.

For coherence, the dominant hand edits in context (pointing, contextual menu) while the non-dominant hand sets that context (pan-and-zoom, general menu). There is also

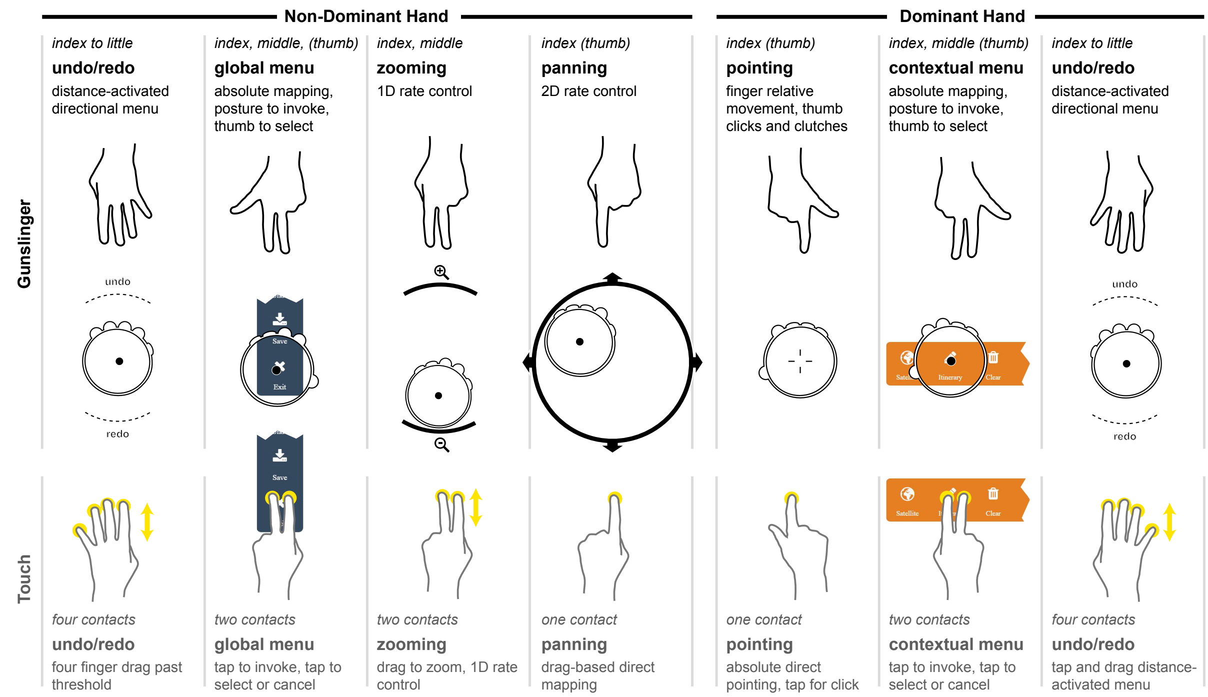



**Figure 4.1: Gunslinger and touch interaction vocabulary for the map navigation demonstration system.**

external coherence: two-finger postures and two contacts invoke menus like the established two finger tap gesture, and two-finger postures and two contacts trigger zooming reminiscent of two finger zooming in Google Maps.

We adapted the vocabulary to match inherent differences between Gunslinger and touch input. The arms-down stance for Gunslinger requires an indirect mapping in a small operating range so rate or relative control are more suitable for continuous input. While the touch vocabulary can use standard absolute-direct mappings for pointing, panning, and zooming, Gunslinger uses rate-based control for clutch-free panning and zooming and relative control with clutching for high precision pointing. While the touch vocabulary can use surface contact to 'click' on a location or menu item, Gunslinger requires an explicit delimiter: the thumb is used to click and clutch when pointing and selecting from menu items.

For *compatibility* when near the display, there are no bimanual-dependent mappings, so modalities can be mixed if desired: one hand can be used with Gunslinger and the other hand with touch. The frequent commands, undo and redo, are mapped to a four finger posture or four finger touch with either hand allowing so they may be triggered with the most convenient modality and hand.

## 4.3 Touch Hand Inference

The vocabulary relies on discriminating between right and left hands. This is trivial with Gunslinger given the arms down form factor, but current touch displays do not identify which hand is used. We created a simple state-machine that uses Gunslinger input history, touch proximity, front facing stance, and user handedness to infer which hand was used to touch (Figure 4.3). The state machine implements these high level behaviors:

- if a touch starts while one LM devices detects a hand, then the touch is credited to the other hand;
- if a touch starts while no LM device detects a hand, the handedness of the user and the distance to existing touch points are used to guess which hand is used;
- if a new touch is far left of a current left touch, the new touch is labeled as left and the current touch is relabeled as right (the rule is inverted for right touches);
- if touch points associated with one hand move too far apart, distant points are reassigned as different hands.

These rules do not provide perfect detection, but they work for common usage patterns. Fortunately, detection is easily correctable by swiping a hand past the thigh to reset touch-to-hand assignments. The full state machine (including support for multiple users) encoding these behaviors is included in Appendix.

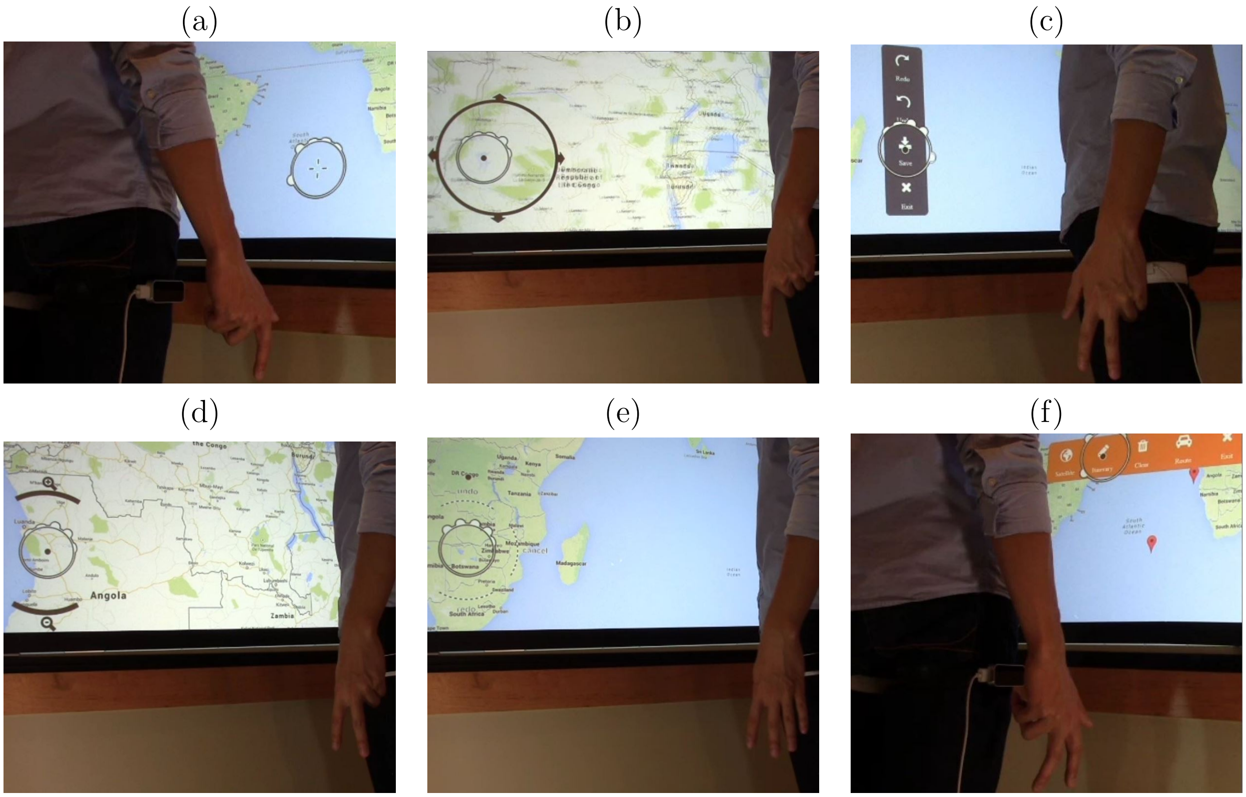



**Figure 4.2: Gunslinger interaction vocabulary for map navigation system**
From top left to bottom right: (a) pointing by right index finger; (b) panning by left index finger; (c) general menu selection by left index and middle finger (with thumb extending); (d) zooming by left index and middle finger; (e) undo/redo by index to pinky finger; (f) contextual menu selection by right index and middle finger (with thumb extending)

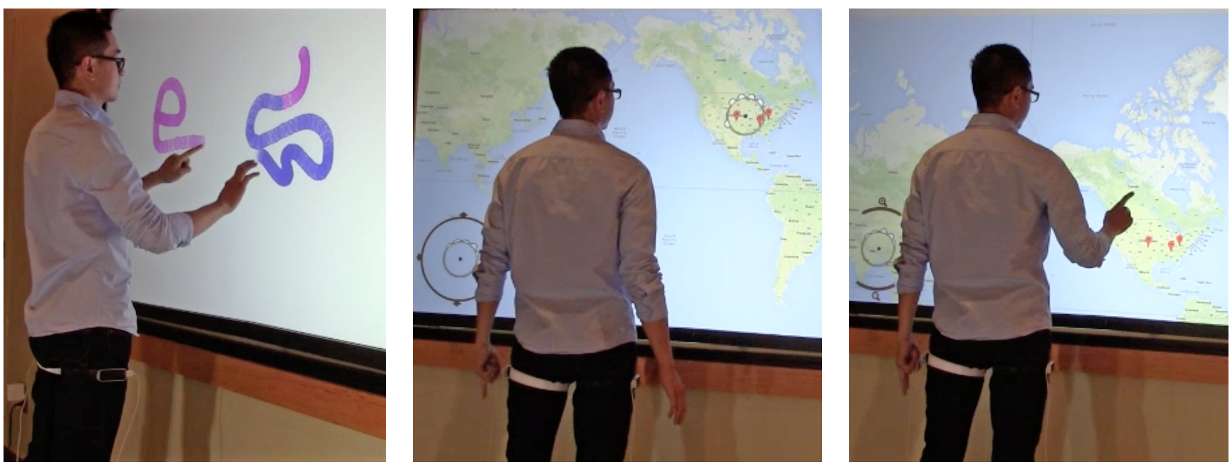


**Figure 4.3: Gunslinger Touch hand inference**
Left: Example of touch hand-discrimination. Middle: Bi-manual Gunslinger. Right: Bi-manual mixed touch+Gunslinger.

# Chapter 5

# Evaluation

We evaluated technical and usability aspects of Gunslinger system using a four part sequential study to test:

(1) Midas Touch robustness;

(2) effectiveness of posture recognition and hand cursor feedback;

(3) arms-down pointing performance;

(4) general usability with and without touch.

All participants completed these parts in sequence. This enabled them to incrementally learn Gunslinger and the intermodal interactive system by using more of its aspects at each stage – effectively training them for the final usability part. The combined technical and usability study approach is motivated by related studies by Bailly *et al.* [1] and Chen *et al.* [10].

## 5.1 Apparatus and Participants

We use the Gunslinger system described above with an 80 ”, 1280 × 720 px, back projected display with a PQ Labs multi-touch overlay. All software is written in JavaScript embedded in HTML5 web applications and run in the Chrome desktop browser. We use LM SDK version 2.20 and the JavaScript client library LeapJS to retrieve sensing data from the device. Since the SDK does not support multi-device sensing, we connect the second LM device to a virtual machine and communicate over a bridged network via WebSocket.

We recruited 11 participants (3 female, mean age 24.2), but we had to exclude the first due to technical problems. Of the remaining 10, 4 had experience with remote game controllers. All were right-handed to avoid confusion between Right-Left and Dominant-Non Dominant in the subsequent analyses. Since the Leap devices are wrapped around the upper thighs, we instructed participants to avoid wearing skirts. For Study Part 2 and Part 3 participants were asked to stand 2 meters away from the wall display when finishing the tasks. The task, design, and results for each part are described individually below.

## 5.2 Part 1: Midas Touch

The goal of this part was to elicit conversational gestures in order to investigate whether Gunslinger postures occur during "normal" standing conversation. Participants were equipped on each thigh with one Leap device whose location and orientation was adjusted to ensure hand postures were within tracking range. After setup, the experimenter diverted the participants' attention from the system and conducted a 5 minute interview to record demographics while the participant stood wearing Gunslinger (age, occupation, and experience with relevant input devices and techniques). During the interview, the system logged recognized postures and the scene was video recorded. The interview was extended with open questions such as "*List all the touch interfaces that you have ever used*" with additional follow-up questions as needed.

### 5.2.1 Results

Due to technical issue, left-hand postures were not recorded in this phase. To remedy our technical mistake, we later re-ran this part with 10 new participants (3 female, mean age 25.4) and results were similar. Following are the results reported from the follow-up study.

The interview took 4.79 min on average (SD 45.0 s). Overall the hand was outside the sensing range 91.4% of the time since conversational gesticulation often occurred in front of the body. Postures reserved for neutral states were recognized in most of the remaining time (fist 0.7%, open hand 3.0%). The remaining postures were detected on average 4.9 % of the time during the interview. Interestingly, some postures were almost never detected: Metal and Index+Pinky detected for only three participant; Thumb+Pinky for five participants. Detail statistics for each participants are shown in Figure 5.1. Overall we detected few false positives even though participants were not trained and made no compensating behavior. The lower quartile of the durations of detected postures is 68.8 ms, suggesting an activation

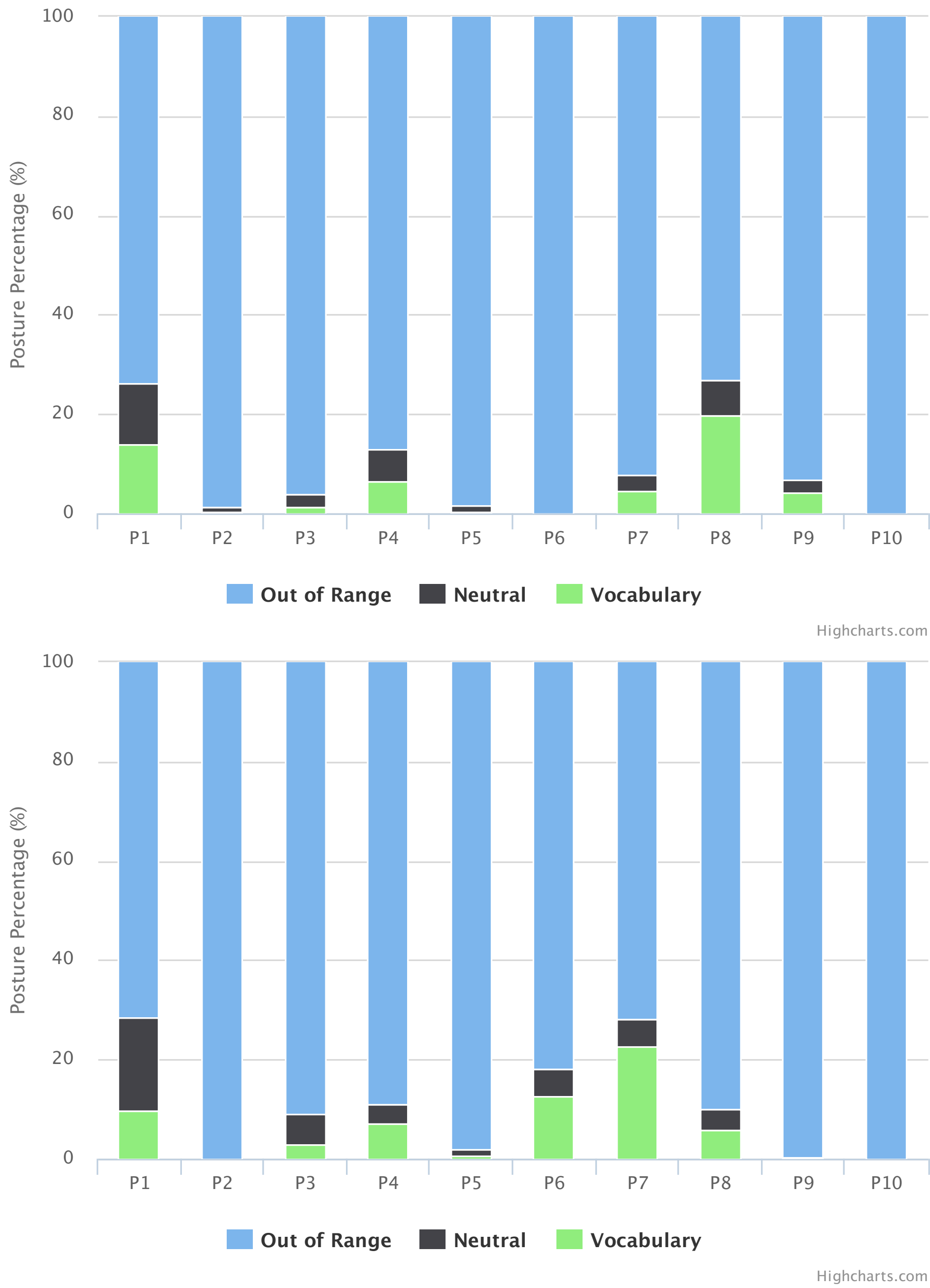



**Figure 5.1: Posture percentage for each participant**
Top: left hand postures; Bottom: right hand postures.

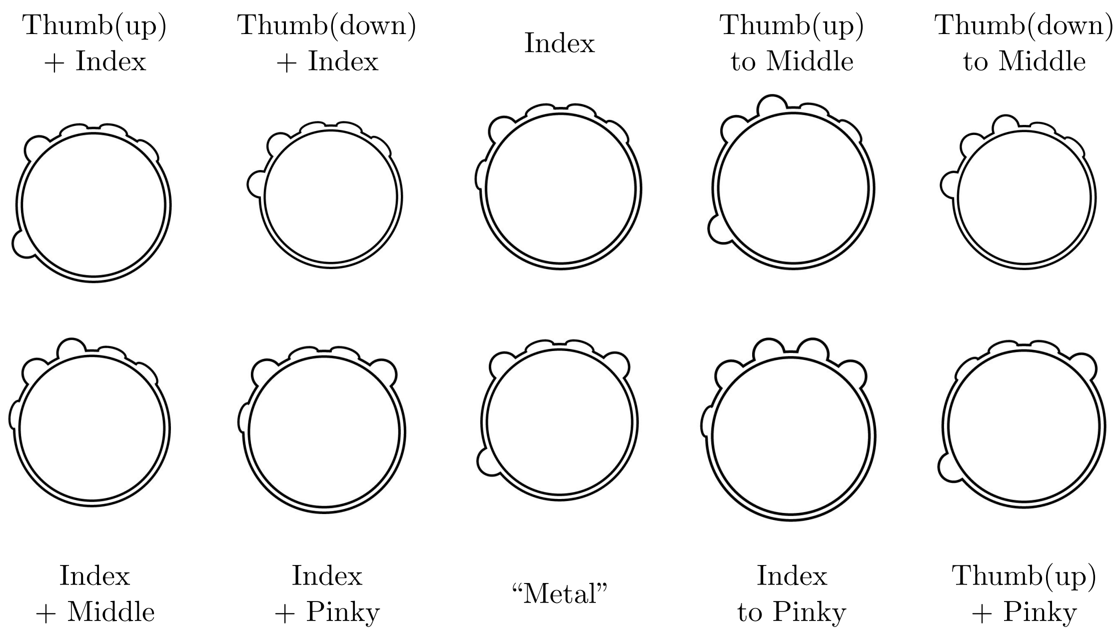



**Figure 5.2: The 10 postures used in the posture recognition part.**

duration filter could further reduce false positives. To test this, we re-processed the raw logs with a 50-ms duration threshold and a more conservative $\theta = .15$ in our posture recognizer. This lowered the false-positive vocabulary postures to only 0.5% and increased empty frames to 98.3% of the time. Overall, the Midas Touch is minimum. One limitation for this study is that it is for conversational gestures only and may be more applicable to Computer-Supported Cooperative Work and Social Computing (CSCW) applications.

## 5.3 Part 2: Posture Recognition and Feedback

The goal of this part was to investigate how users behave differently with or without the presence of visual feedback, and to assess how well the posture recognition metric functioned. After being briefed on hand cursor feedback, participants completed a sequence of trials with each hand. Each trial began when they formed an open-hand neutral posture, then a target hand posture was displayed using the hand cursor visualization, the participant replicated the posture with one hand and held it for a defined time after which the trial ended. All 10 Gunslinger postures were tested (Figure. 5.2).

There were two conditions: *Feedback* and *No-Feedback*. In the *Feedback* condition, as shown in Figure 5.3, participants had real-time feedback of their hand posture using a second hand cursor. They required time to hold the correct posture was 500 ms shown as a progress bar. The progress bar reset when an incorrect posture was detected. This part served as a posture training for the rest of the study and is a best-case scenario for posture recognition. Since the user actively monitors hand posture feedback, they can compensate for recognition errors.

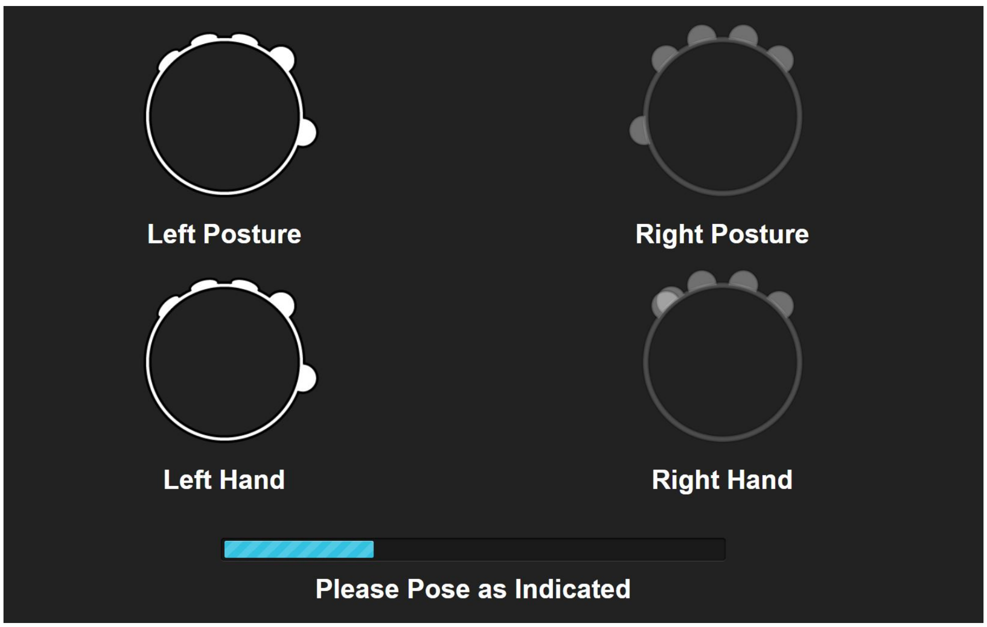



**Figure 5.3: Task in feedback condition of posture recognition study**
The left hand task is in session and user is asked to pose Thumb(up)+Index. The right hand visuals are grayed out to avoid distraction.

In the *No Feedback* condition, participants had no real-time feedback of their hand posture, they simply formed the posture to the best of their ability and held it for 4 seconds (again shown as a progress bar). The progress bar did not reset if the wrong posture was formed, each trial was exactly 4 seconds. This part served as a worst-case scenario where participants give their full attention to the task rather than posture feedback.

No Feedback always followed Feedback for each hand so participants had experience forming postures before feedback was removed. Hand order was counterbalanced and posture order was randomized. For each hand and condition, all postures were repeated 3 times. In summary:

2 feedback conditions $\times$

2 hands $\times$

10 postures $\times$

3 replications

= 120 data points per participant.

### 5.3.1 Results

Performance was measured as the completion time of the trial minus the 500 milliseconds trigger, as well as the number of wrong postures detected on the target hand.

In the No Feedback condition, 82 trials could not be completed in time for all participants (13.7 %). Posture had a significant effect on failure ($F_{9,81} = 6.31$, $p < 0.0001$). Only Index+Middle caused no such errors, and a Tukey post-hoc test revealed that it caused significantly less errors than postures involving three fingers or the pinky. Hand had no significant effect.

Using $\theta = .15$ in our posture recognizer and the 250-ms filter mentioned above would have increased the number of failed trials by only 4.

*Completion time* – After removing these errors from our data, we found significant effects of posture ($F_{9,81} = 5.88$, $p < 0.0001$), feedback ($F_{1,9} = 43.76$, $p < 0.0001$), and feedback $\times$ posture ($F_{9,81} = 3.19$, $p = .0033$) on completion time. Feedback was significantly slower than no feedback (2510.5 vs 1438.9 ms); Tukey tests revealed that Index (1304 ms) was significantly faster than "Metal" (2547), Thumb(up)-to-Middle (2825) and Index+Pinky (3072). The details of completion time for both Feedback and Non-feedback condition can be seen in Figure 5.4 These effects are increased when combining Posture and Feedback. Hand had no significant effect.

## 5.4 Part 3: Arms-down Pointing

The goal of this part was to evaluate pointing and clicking performance. The task is similar to previous studies [33]. Participants stood 2 metres from the display and selected a sequence of circular targets using the point, clutch, and click gestures. A trial was successful if the first click-up and click-down events occurred inside the target bounds. Each target had to be successfully selected to continue. We combined two Amplitudes (1400 and 350 mm) and three target Widths (40, 80 and 160 mm) creating an Index of

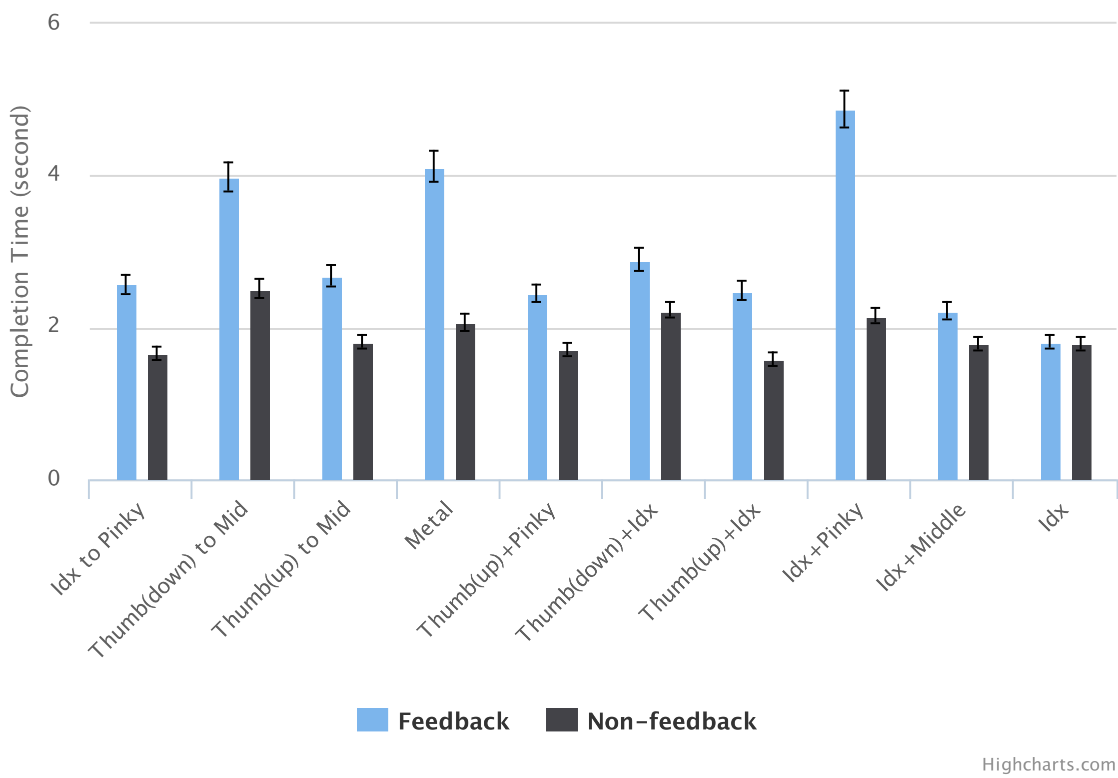



**Figure 5.4: Posture completion time for Feedback and Non-feedback tasks**
(all error bars in figures are 95% CI)

Difficulty (ID) range of 1.7 to 5.2 bits. We used the gain transfer function and calibration process described in [25]. Participants completed 1 block of practice trials and 3 blocks of measured trials. Each block contained 3 sections and each for one Width condition. Each section had two sets and each set contained all combinations of Amplitudes and Widths.

3 Blocks ×

3 Sections ×

2 Sets ×

2 Amplitudes ×

3 Widths ×

= 108 data points per participant.

### 5.4.1 Results

We calculated error rate and median target acquisition time (the median accounts for skewed distributions). A multi-way Anova found a significant effect of Width on Error ($F_{2,18} = 23.53$, $p < 0.0001$): 40 mm (18.1 %) caused significantly more errors than 80 mm (8.3 %) and 160 mm (6.4 %). The error rate of 40 mm target is slightly higher compared to Myopoint [14]'s 15% for 48 mm targets, and 5% error rate for 144 mm targets.

Errors were removed from time analyses. Times ranged from 1.62 s for ID 1.7 up to 3.17 s for ID 5.2. A Fitts' law regression has good fitness and Gunslinger has a similar slope to Myopoint's:

Gunslinger $MT = 497 + 483 \times ID,\ R^2 = .94$
Myopoint $MT = 171.59 + 609.36 \times ID,\ R^2 = .97$

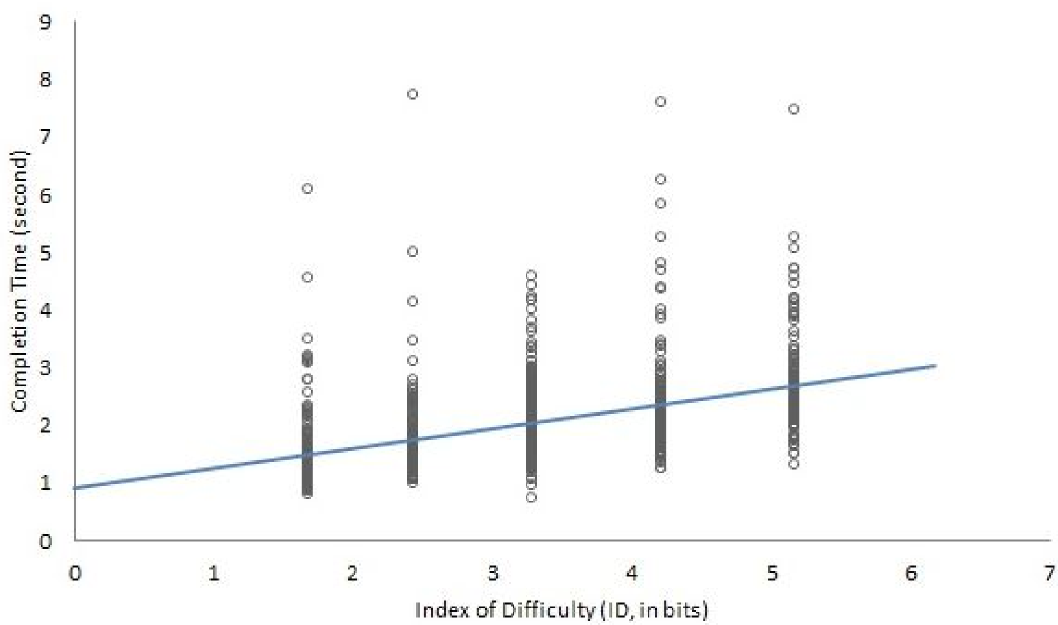



**Figure 5.5: Scatter plot and regression line for pointing task**

## 5.5 Part 4: Usability

The goal of this part is to evaluate the usability of Gunslinger with realistic tasks in two ordered phases: *Controlled* and *Open-Ended*.

After the full vocabulary was described (Figure 4.1), participants performed a sequence of three *Controlled* Tasks:

(1) T1 required locating and pinning two cities, each city must roughly fill the whole screen when pinning;

(2) T2 required undoing, then redoing the last pin-drop;

(3) T3 required changing to satellite view, generating an itinerary using the contextual menu, then saving the itinerary with the global menu.

Each task sequence was completed under three input conditions: Gunslinger-only, touch-only, and mixed in which the dominant hand uses touch and the non-dominant hand uses Gunslinger. The symmetric mixed configuration (dominant Gunslinger and non-dominant touch) was not included since it is really only advantageous for reaching far targets on displays much larger than our 85".

The task order was fixed and input condition was counterbalanced. For each input condition, all three tasks were performed once as practice and a second time for observation. We logged task completion time and participants were asked to 'think-aloud.' After all conditions were completed, participants rated input condition for easiness, fatigue, speed, precision, and general opinion on a 7-point numeric scale (higher is better).

In the second phase, participants were given an *Open-Ended* task: "*You have a 3-month vacation with unlimited budget: plan your ideal trip, in the order that suits you best; generate an itinerary, and save it. You can use any combination of modalities that you like and take as much time as you need.*" The task requires the global and contextual menus and is designed to elicit map navigation and exploration. There was no minimum or maximum time limit, it was up to the participant to determine when they completed the task. We logged all input and participants were asked to 'think-aloud', especially regarding input choice. This provided unconstrained subjective and observational feedback of general usability. At the end of this phase, participants were asked for additional comments.

### 5.5.1 Results

#### ***Controlled*** Tasks

The time to complete each of the tasks for each input condition can be shown in Figure 5.6. We found a significant effect of input on completion time for T1 ($F_{2,18} = 4.09$, $p = .0344$), T2 ($F_{2,18} = 14.57$, $p = .0002$), and T3 ($F_{2,18} = 4.82$, $p = .0211$). Post-hoc tests showed Gunslinger (92.4 s) was significantly slower than Touch (60.6 s) for T1; Gunslinger (16.3 s) significantly slower than Mixed (11.7 s) and Touch (9.3 s) for T2; and Gunslinger (37.3 s) significantly slower than Mixed (16.5 s) and Touch (15 s) for T3.

Participants said Gunslinger and Mixed were not as easy to use as Touch (medians 4 and 4.5 vs 6); 2 ratings were below neutral for Gunslinger, 1 for Touch, and 1 for Mixed. Overall, fatigue was not an issue (medians 5, 6, 5 for Gunslinger, Mixed, Touch) though 2 participants were below neutral for Gunslinger and 1 for Mixed. There may be some bias towards touch given experience and familiarity. Overall, perceived speed was comparable (medians 4.5, 5 and 5 for Gunslinger, Mixed, Touch), though 2 participants were below neutral for Gunslinger, 1 for Mixed, and 1 for Touch. Overall, precision was good (medians 5, 5, 6 for Gunslinger, Mixed, Touch), though 1 rated Gunslinger below neutral, and 1 for Touch. The general impression was good overall (medians 5, 5.5, 6 for Gunslinger, Mixed, Touch), though 2 rated Gunslinger below neutral, and 1 for Mixed.

#### ***Open-ended*** Task

Gunslinger alone was used mostly for saving (4 participants) and undo/redo (3 participants, 4 did not undo), marginally for navigation (1 for pan and zoom), and never for adding markers and computing itineraries. Mixed was used more often: for adding markers (3), for panning and zooming (5 and 6), and for computing itinerary (1). Details are illustrated in Figure 5.7.

Participant comments provided interesting insights. The novelty of Gunslinger was noted ("refreshing", "Touch is boring. I like GS more", P5 and P10), but also that Gunslinger may have hindered performance (P10) especially compared to Touch (P4, P7). Comments about fatigue favoured Gunslinger, with comments stating is was more relaxing than touch (P10, P9). Feelings were mixed about Gunslinger pointing, some had comments like "intuitive and subtle" (P9) others found it impractical (P5). Mixed was appreciated as a sensible (P1, P9) and "more natural" (P6) combination, but requiring more practice (P4). Gunslinger was considered advantageous at a distance with larger displays (P2), and up close with high targets (P9). Some said Gunslinger had adequate feedback (P3) and

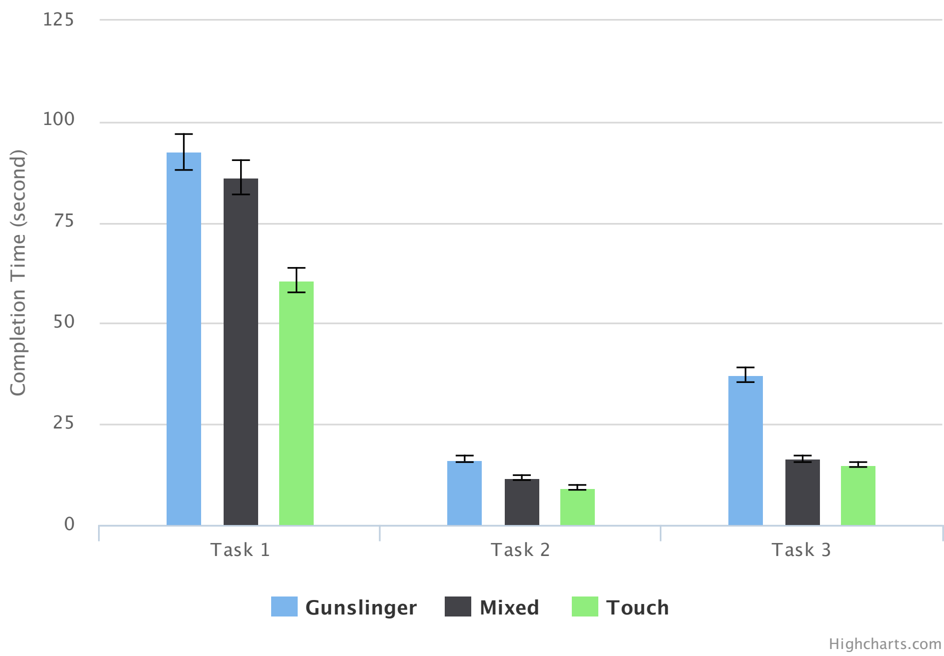



**Figure 5.6: Completion time for the three tasks in different input conditions**

was "quite responsive" (P9), but some also said Gunslinger is "too sensitive" (P3) and the sensing range is too small (P6, P9).

## 5.6 Discussion

Overall, Gunslinger is usable with acceptable performance. Arms-down postures are promising: Midas touch is minimal and 7 to 10 postures of various complexities can be performed and recognized reliably, even without visual feedback. To eliminate Midas Touch even more, an explicit yet simple delimiter could be added to enter interaction mode. We discussed tweaking a threshold and adding a 50 ms detection window to further reduce false positives. When designing future vocabularies, postures that take longer to form should be reserved for infrequent commands. Arms-down pointing and clicking is achieved with reasonable time and error rate, despite the novelty of the technique and of its unusual stance.

All map tasks were feasible with Gunslinger and although most participants did not perceive a pronounced speed difference, task completion times with Gunslinger are slower

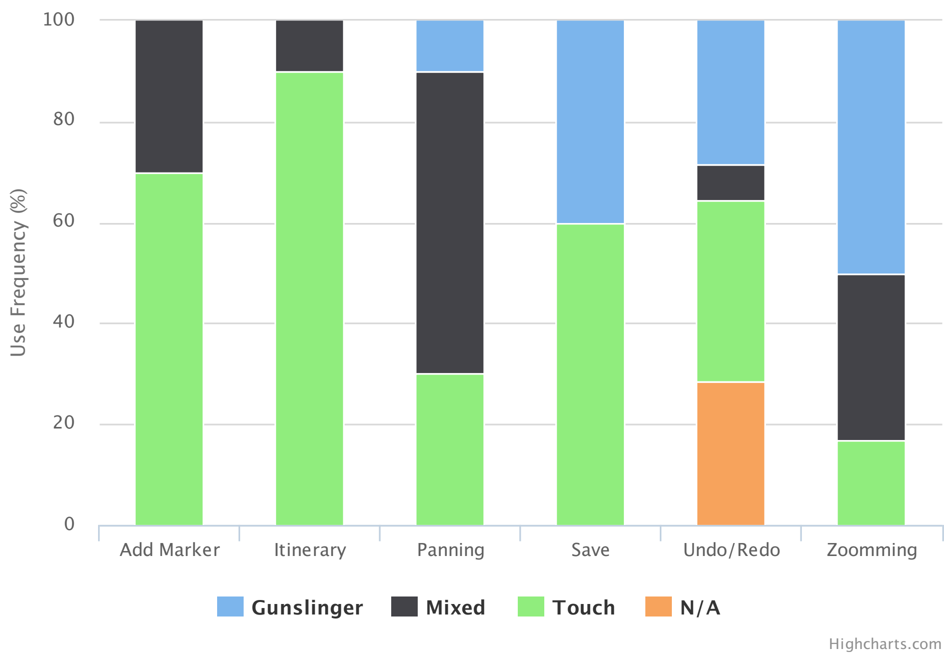



**Figure 5.7: Usage by different modalities in *open-ended* task**

than touch. Touch is more familiar, has very high quality tracking, and tactile feedback – this is hard to beat. However, touch is not viable at a distance and the performance and enthusiasm for mixing Gunslinger and touch is encouraging. The majority of participants also said Gunslinger was less tiring, perhaps speed alone is not the definitive measure.

Moreover, making Gunslinger adaptive to different users could also improve the performance. For example, some participants found it difficult to register "click" action as a result of inappropriate thumb state thresholds. Therefore pre-configuring the Gunslinger system to best adapt to individual user will substantially enhance the user experience.

# Chapter 6

# Post-evaluation Improvement

The four-part evaluation offers some valuable insights and opportunities for improvement. Based on the results we took steps to address the issue of Gunslinger being overly sensitive ("too sensitive") perceived by some of the participants.

We believe this is partly due to occasional erroneous input when a finger providing continuous control (e.g. pointing) sends a short, high-speed "jerk" as it curls in to form a neutral fist posture. Moreover, when a user switches postures or enters/exits the Gunslinger Mode, the Gunslinger system behaves unexpectedly as a result that the system is still responding to user input even during intermediary phrase. This is especially noticeable when a user switches posture between pointing to anything else, which often causes the cursor to drift away from the desired position. User often needs to pay special attention when switching postures or entering/exiting Gunslinger sensing range.

## 6.1 Rollback Mechanism

To address this, we introduced a rollback mechanism to make posture switching less troubling. A *Recorder* is implemented to maintains records of the system in the last 60 frames, which approximately counts for the last two seconds of Gunslinger interaction. These records are the "snapshots" of system states which also include the motion data reported from the device. Whenever a change of state happens, the recorder gets notified and iterates backwards through the records until it finds the first *stable* state and rolls back the system to that state. We define a stable hand as a hand object moving at a speed lower than 20mm/s. Following is the pseudo code when rollback action is triggered:

**Algorithm 3** Roll back to the most stable state.

```
Require: The Recorder is initialized.
  i ←number of records
  while i >= 0 do
    if records[i].hand.velocity > 20 then
      rolls back to records[i]
    else
      //clear the records and break the iteration
      recorder.clear();
      break;
    end if
    i ← i − 1
  end while
```

## 6.2 Evaluation

To evaluate whether this approach effectively resolves this problem, we conducted an additional comparison study to elicit user's general preference over this change.

### 6.2.1 Apparatus and Participants

The setup was the same with the second part of the study described in Chapter 5. 10 participants (3 female, mean age 25.4) completed this study. Participants were instructed to complete three tasks in five repetitions in two Techniques: *Rollback* and *No-rollback*:

(1) Switch from Pointing to Neutral state;

(2) Switch from Neutral to Pointing state;

(3) Raise the hand to exit the Gunslinger mode from the front.

Those tasks were selected to represent common posture switches made during normal interactions. Participants were asked to focus on the changes to the cursor while performing the tasks (Figure 6.1). 'Think-aloud' was also encouraged similar to the usability study. After task completion, we asked participants for their general preference and reason.

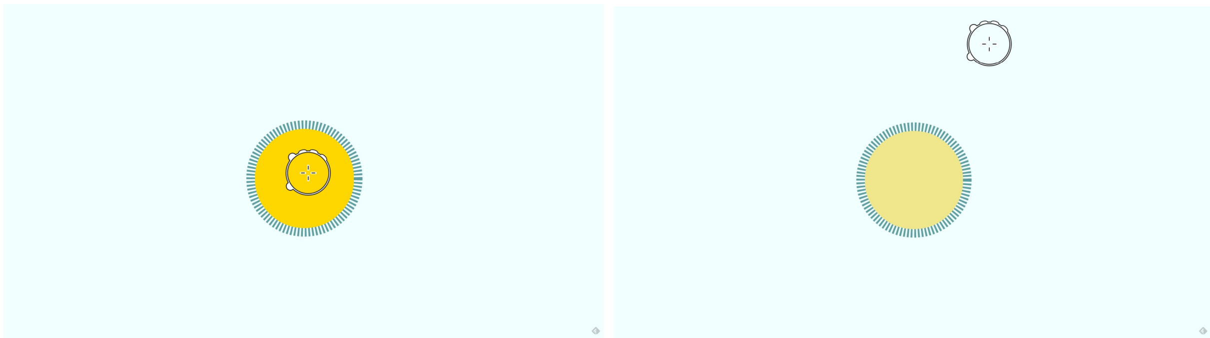


**Figure 6.1: Comparison study task**
The left is the initial state where participants are asked to perform the tasks. A subtle change of color indicates whether cursor ends up outside of the center circle.

### 6.2.2 Results

The 10 participants unanimously preferred the Rollback technique on various levels. Some comments are worth mentioning. Overall participants were able to discern different cursor behaviors in the two techniques. Some mentioned Rollback is easier to use (P1, P2, and P6), and they feel more in control of the pointer (P6). Precision improvement was appreciated (P5, P8) when Rollback was provided , whereas No-rollback caused instability ("the cursor position is unpredictable"). More specifically, P5 mentioned some scenario where having Rollback is essential: "If I am in the middle of drawing (using Gunslinger) I wouldn't want my pointer to move away."

While the comments were mostly positive, some improvements could be made. P10 noticed sometimes the Rollback did not work properly, which suggests that the fixed threshold that triggers the Rollback action should be adaptive to each user to achieve better performance.

Overall, the comparison study showed some positive results regarding the Rollback Mechanism. We believe this change will improve the easiness and precision of Gunslinger interaction, especially for the pointer control.

# Chapter 7

# Conclusion and Future Work

In this chapter, we present a summary of the design and implementation process described in this thesis, compromising the motivation, the survey of previous research into mid-air interaction, the design and evaluation of Gunslinger system. We also describe areas that could be worth exploring for additional study, either by optimizing or extending the current Gunslinger system or performing additional experiments to better understand how users interact with it.

## 7.1 Conclusion

We introduced Gunslinger, a barehand interaction technique that uses thigh-mounted Leap Motion devices to enable arms-down and subtle input gestures to reduce input performance space, fatigue, and social awkwardness. With a goal of designing a subtle mid-air interaction in mind, we surveyed a wide range of previous technologies in related area. Although some of them could in theory be used arms down, there has been no exploration of a full interaction vocabulary performed from an arms-down stance explicitly focusing on subtlety. The design of Gunslinger follows guidelines for relaxed barehand input that ensure that users can interact comfortably in mid-air without sacrificing the expressiveness of the interaction technique. We also provide continuous feedback about the hand sensing and posture recognition to ensure that the user never has to switch his visual attention to understand the system's responses. An implemented interaction vocabulary is described for map navigation which demonstrates how Gunslinger can be combined with touch input supported by a touch hand inference method leveraging the arms-down form factor.

The results of a comprehensive evaluation show the Gunslinger approach and related techniques work. Some insightful feedback from participants during the study shows potential areas for improvements.

## 7.2 Future Work

Following are several questions arising from the work that are worth investigating:

1. Is using index fingertip the best way to control large wall display? Based on the observation from pointing study, some participants often had to make large hand movement to traverse a longer distance. An interesting follow-up study is to investigate if using palm or an combination of both finger and palm motion is more suited for wall interaction.

2. Can existing mid-air interaction techniques be adapted to arms-down, subtle gestures? We envision that the benefits of previous mid-air and barehand techniques could be applied to Gunslinger without significant loss of expressiveness. The set of subtle gestures designed in ShoeSense [1] is a good start in this direction.

3. How well can Gunslinger fit in multi-user configurations, with various levels of proximity between users, and can it support collaborative interaction? One such interaction could be waving from one user's LM device to another's, to send data without using intermediary GUI elements. We have already taken multiuser scenario into consideration when we designed the state machine (see appendix ??), but it remains to be seen whether this multi-user setting will be a fruitful area to explore.

4. Can Gunslinger be used for mid-air text entry? Previous work like Vulture [22] opens opportunities for Gunslinger to incorporate mid-air finger sliding as a way for text entry.

5. What about using Gunslinger while seated? Can this stance achieve a similar or even better performance with the standing stance? In addition, it is also worth investigating the performance and applicability of using Gunslinger while walking.

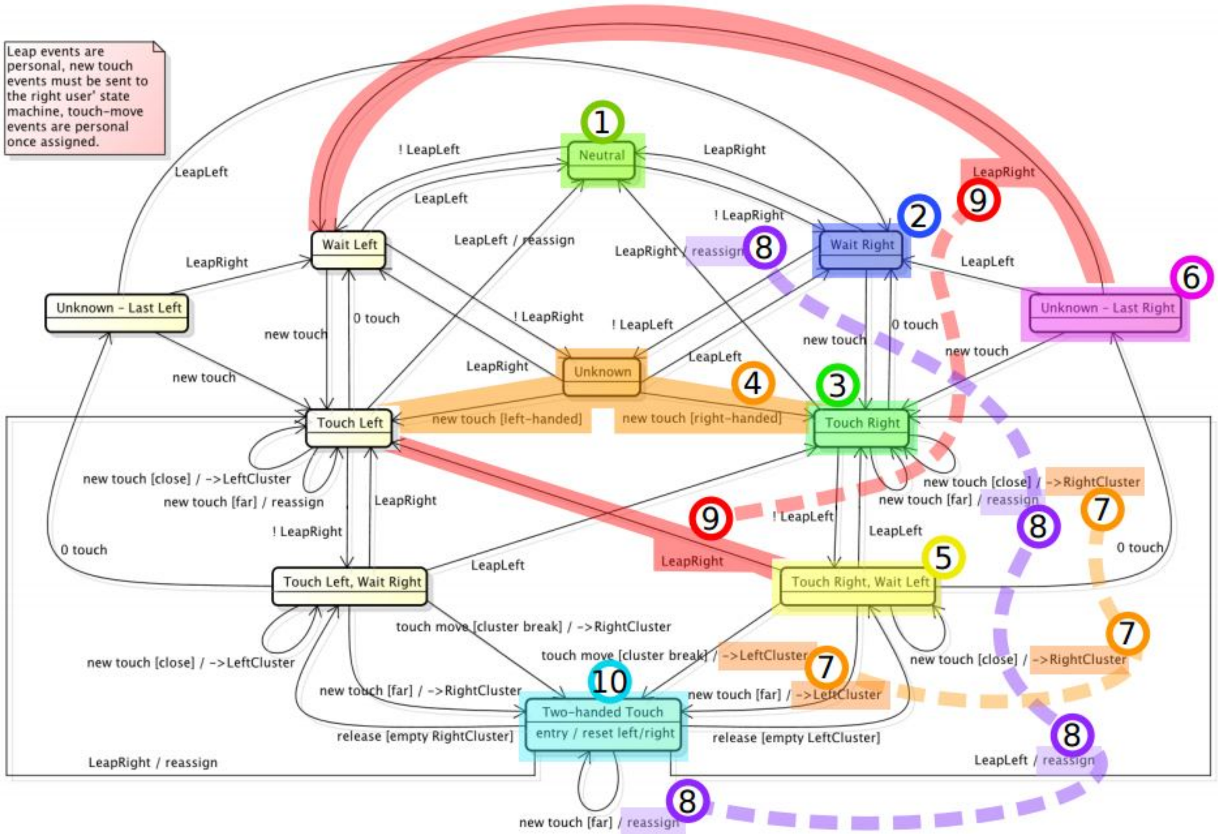
6. How can we design and deployable Gunslinger-specific device (focusing on size, portability, resilience to outside light sources, and additional sensors such as sonar to further reduce inference)?

7. Can Gunslinger work in tandem with head-mounted displays for virtual and augmented reality? Recently, Oculus Rift has partnered with Leap Motion to explore ways to enrich the immersive virtual reality experience through front interaction. We image Gunslinger along with Oculus Rift could yield some interesting findings. Moreover, controlling a smartphone when in a pocket or bag via Gunslinger is another direction to extend our study.

# APPENDICES

# Appendix A

# State Machine for touch hand inference




1. Neutral state: both LeapMotions detect a hand and no touch is detected on the wall

2. A hand has disappeard from one of the LeapMotions, but no touch detected yet

3. Simple touch case: a touch is detected while only one of the LeapMotions is "empty"

4. Unknown state: both LeapMotions are "empty" and there is no touch. Next touch is considered dominant hand

5. One hand is touching the screen, the other one disappeard from the other LeapMotion

6. Unknown state with knowledge of the hand that performed the last touch: the next touch will be on the same side unless LeapMotion events induce otherwise

7. Close touches are grouped in clusters corresponding to the fingers of a same hand.

8. In cases where a touch event occurs that is incompatible with the current state (e.g. both hands are supposed to be on the screen and a remote touch event occurs), the touch event is sent back to be reassigned* to another user.

9. In cases where a Leap event occurs that is incompatible with the current state (e.g. the right hand is on the screen and the right Leap detects a hand), the state is corrected accordingly.

10. Both hands are on the screen. When this state is entered, the clusters are reset (left-right) depending on their relative locations (in case there had been a mistake before, a bit like 8)

* A separate system runs on top of all the users' statemachines and assigns them touch events. This system keeps track of who each touch is assigned to and also receives the `reassign' calls.

# Appendix B

# Questionnaire for the first study

PARTICIPANT ID # ___________ DATE ______________________

## Pre-experiment Questionnaire

1. Gender: *Male* *Female*
2. Age: _______
3. Which hand do you write with? *Left* *Right*
4. How many hours per week on average do you use a computer:
   _____ *hours per week*
5. Have you used a Kinect, Nintendo Wii, or similar gestural game controller?
   *No* *Yes* If Yes, how many hours per week on average _____

## Post-experiment Questionnaire

Note: 1 is strongly disagree and 7 strongly agree, with 4 being neutral.

1. Gunslinger:

   Easiness:

   1 2 3 4 5 6 7

   Fatigue:

   1 2 3 4 5 6 7

   Perceived speed:

   1 2 3 4 5 6 7

   Perceived precision:

   1 2 3 4 5 6 7

PARTICIPANT ID # ____________ DATE ________________________

General preference:

1 2 3 4 5 6 7

2. Touch:

Easiness:

1 2 3 4 5 6 7

Fatigue:

1 2 3 4 5 6 7

Perceived speed:

1 2 3 4 5 6 7

Perceived precision:

1 2 3 4 5 6 7

General preference:

1 2 3 4 5 6 7

3. Mixed:

Easiness:

1 2 3 4 5 6 7

Fatigue:

PARTICIPANT ID # ___________ DATE ______________________

1 2 3 4 5 6 7

Perceived speed:

1 2 3 4 5 6 7

Perceived precision:

1 2 3 4 5 6 7

General preference:

1 2 3 4 5 6 7

## Semi-structured Interview

1. Any additional comments?

# Appendix C

# Questionnaire for the second study

PARTICIPANT ID # __________ DATE ____________________

## Pre-experiment Questionnaire

1. Gender: *Male* *Female*
2. Age: ______
3. Which hand do you write with? *Left* *Right*
4. How many hours per week on average do you use a computer:
   _____ *hours per week*
5. Have you used a Kinect, Nintendo Wii, or similar gestural game controller?
   *No* *Yes* If Yes, how many hours per week on average _____

## Post-experiment Questionnaire

1. Do you prefer Rollback over No-rollback?
   Yes / No

   Reasons:

## Semi-structured Interview

1. Any additional comments?

# References


[1] Gilles Bailly, Jörg Müller, Michael Rohs, Daniel Wigdor, and Sven Kratz. Shoesense: A new perspective on gestural interaction and wearable applications. In *Proc. CHI '12*, pages 1239–1248. ACM, 2012.

[2] T. Ballendat, N. Marquardt, and Saul Greenberg. Proxemic interaction: Designing for a proximity and orientation-aware environment. In *Proc. ITS '10*. ACM Press, 2010.

[3] Thomas Baudel and Michel Beaudouin-Lafon. Charade: Remote control of objects using free-hand gestures.

[4] Richard A. Bolt. Put-that-there: Voice and gesture at the graphics interface. In *Proc. SIGGRAPH '80*, pages 262–270. ACM, 1980.

[5] Andrew Bragdon, Rob DeLine, Ken Hinckley, and Meredith Ringel Morris. Code space: Touch + air gesture hybrid interactions for supporting developer meetings. In *Proc. ITS '11*, pages 212–221. ACM, 2011.

[6] Michelle A. Brown, Wolfgang Stuerzlinger, and E. J. Mendonca Filho. The performance of un-instrumented in-air pointing. In *Proc. GI '14*, pages 59–66. Canadian Information Processing Society, 2014.

[7] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1 filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2527–2530. ACM, 2012.

[8] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng, and Bing-Yu Chen. Fingerpad: Private and subtle interaction using fingertips. In *Proc. UIST '13*, pages 255–260. ACM, 2013.

[9] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. utrack: 3d input using two magnetic sensors. In *Proc. UIST '13*, pages 237–244. ACM, 2013.

[10] Xiang 'Anthony' Chen, Tovi Grossman, Daniel J. Wigdor, and George Fitzmaurice. Duet: Exploring joint interactions on a smart phone and a smart watch. In *Proc. CHI '14*, pages 159–168. ACM, 2014.

[11] Andrea Colaço, Ahmed Kirmani, Hye Soo Yang, Nan-Wei Gong, Chris Schmandt, and Vivek K Goyal. Mime: compact, low power 3d gesture sensing for interaction with head mounted displays. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 227–236. ACM, 2013.

[12] Clifton Forlines, Daniel Vogel, and Ravin Balakrishnan. HybridPointing: fluid switching between absolute and relative pointing with a direct input device. In *Proc. UIST'06*, pages 211–220. ACM, 2006.

[13] Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. Imaginary interfaces: Spatial interaction with empty hands and without visual feedback. In *Proc. UIST '10*, pages 3–12. ACM, 2010.

[14] Faizan Haque, Mathieu Nancel, and Daniel Vogel. Myopoint: Pointing and clicking using forearm mounted electromyography and inertial motion sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3653–3656. ACM, 2015.

[15] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. Omnitouch: Wearable multitouch interaction everywhere. In *Proc. UIST '11*, pages 441–450. ACM, 2011.

[16] Juan David Hincapie-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proc. CHI '14*, pages 1063–1072. ACM, 2014.

[17] Robert J. K. Jacob. What you look at is what you get: Eye movement-based interaction techniques. In *Proc. CHI '90*, pages 11–18. ACM, 1990.

[18] Todd Johnsgard. Fitts' law with a virtual reality glove and a mouse: Effects of gain. In *Proc. GI '94*, pages 8–15. CANADIAN INFORMATION PROCESSING SOCIETY, 1994.

[19] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: Freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proc. UIST '12*, pages 167–176. ACM, 2012.

[20] J.-C. Levesque, D. Laurendeau, and M. Mokhtari. Bimanual gestural interface for virtual environments. In *Proc. VR '11*, pages 223–224, March 2011.

[21] Gan Lu, Lik-Kwan Shark, Geoff Hall, and Ulrike Zeshan. Immersive manipulation of virtual objects through glove-based hand gesture interaction. *Virtual Real.*, 16(3):243–252, September 2012.

[22] Anders Markussen, Mikkel Rnne Jakobsen, and Kasper Hornb\a ek. Vulture: A Mid-air Word-gesture Keyboard. In *Proc. CHI '14*, pages 1073–1082, New York, NY, USA, 2014. ACM.

[23] Pranav Mistry and Pattie Maes. Sixthsense: A wearable gestural interface. In *Proc. SIGGRAPH ASIA '09*, pages 85–85. ACM, 2009.

[24] Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Pourang P. Irani, and Michel Beaudouin-Lafon. High-precision pointing on large wall displays using small handheld devices. In *Proc. CHI '13*, pages 831–840. ACM, 2013.

[25] Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Pourang P. Irani, and Michel Beaudouin-Lafon. High-precision pointing on large wall displays using small handheld devices. In *Proc. CHI '13*, pages 831–840, New York, NY, USA, 2013. ACM.

[26] Sharon Oviatt. Ten myths of multimodal interaction. *Commun. ACM*, 42(11):74–81, November 1999.

[27] Wayne Piekarski and Ross Smith. Robust gloves for 3d interaction in mobile outdoor ar environments. In *Proc. ISMAR '06*, pages 251–252. IEEE Computer Society, 2006.

[28] Jaime Ruiz and Daniel Vogel. Soft-constraints to reduce legacy and performance bias to elicit whole-body gestures with low arm fatigue. In *Proc. CHI '15*, page 4p. ACM, 2015 in press.

[29] T. Scott Saponas, Chris Harrison, and Hrvoje Benko. PocketTouch: Through-fabric Capacitive Touch Input. In *Proc. UIST '11*, pages 303–308, New York, NY, USA, 2011. ACM.

[30] Garth Shoemaker, Leah Findlater, Jessica Q. Dawson, and Kellogg S. Booth. Mid-air text input techniques for very large wall displays. In *Proc. GI '09*, pages 231–238. Canadian Information Processing Society, 2009.

[31] Bernhard Suhm, Brad Myers, and Alex Waibel. Multimodal error correction for speech user interfaces. *ACM Trans. Comput.-Hum. Interact.*, 8(1):60–98, March 2001.

[32] Daniel Vogel and Ravin Balakrishnan. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proc. UIST '04*, pages 137–146. ACM, 2004.

[33] Daniel Vogel and Ravin Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. UIST '05*, pages 33–42. ACM, 2005.