

**Body-Centric and Shadow-Based Interaction for Large  
Wall Displays**

by

Garth Shoemaker

B.Sc. Physics and Computing and Information Science, Queen's University, 1998

M.Sc. Computer Science, Simon Fraser University, 2000

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF

**Doctor of Philosophy**

in

THE FACULTY OF GRADUATE STUDIES

(Computer Science)

The University Of British Columbia

(Vancouver)

December 2010

© Garth Shoemaker, 2010

# Abstract

We are entering an era in human-computer interaction where new display form factors, including large displays, promise to efficiently support an entire class of tasks that were not properly supported by traditional desktop computing interfaces. We develop a “body-centric” model of interaction appropriate for use with very large wall displays. We draw on knowledge of how the brain perceives and operates in the physical world, including the concepts of proprioception, interaction spaces, and social conventions, to drive the development of novel interaction techniques. The techniques we develop include an approach for embodying the user as a virtual shadow on the display, which is motivated by physical shadows and their affordances. Other techniques include methods for selecting and manipulating virtual tools, data, and numerical values by enlisting different parts of the user’s body, methods for easing multi-user collaboration by exploiting social norms, and methods for mid-air text input. We then present a body-centric architecture for supporting the implementation of interaction techniques such as the ones we designed. The architecture maintains a computational geometric model of the entire scene, including users, displays, and any relevant physical objects, which a developer can query in order to develop novel interaction techniques or applications. Finally, we investigate aspects of low-level human performance relevant to a body-centric model. We conclude that traditional models of performance, particularly Fitts’ law, are inadequate when applied to physical pointing on large displays where control-display gain can vary widely, and we show that an approach due to Welford is more suitable. Our investigations provide a foundation for a comprehensive body-centric model of interaction with large wall displays that will enable a number of future research directions.

# Preface

All research in this dissertation was conducted under the supervision of Dr. Kellogg S. Booth. Dr. Yoshifumi Kitamura co-supervised some of the design of the techniques described in Chapter 4.

Ethics approval for experimentation with human subjects was provided by the Behavioural Research Ethics Board, UBC BREB Number: H08-00040.

I am the primary contributor of all work described in the thesis. I collaborated with Tony Tang on the prototypes described in Chapter 3. He implemented the back-lit IR source capture technique. I collaborated with Takayuki Tsukitani on the design of techniques described in Chapter 4, although I performed nearly all of the implementation. I collaborated with Leah Findlater and Jessica Q. Dawson on the work described in Chapter 5. For the second experiment, Leah helped with the evaluation and Jessica ran participants. Both helped write the resulting paper.

Elements of chapters 3, 4, and 5, have all been published at peer reviewed conferences:

- Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. Shadow Reaching: A New Perspective on Interaction for Large Wall Displays. In Proceedings of UIST 2007. Pages 53–56.
- Garth Shoemaker, Takayuki Tsukitani, Yoshifumi Kitamura, and Kellogg S. Booth. Body-Centric Interaction Techniques for Very Large Wall Displays. In Proceedings of NordiCHI 2010. Pages 463-472.
- Garth Shoemaker, Leah Findlater, Jessica Q. Dawson, and Kellogg S. Booth. Mid-Air Text Input Techniques for Very Large Wall Displays. In Proceedings of Graphics Interface 2009. Pages 231–239.

# Table of Contents

<b>Abstract</b> . . . . .	<b>ii</b>
<b>Preface</b> . . . . .	<b>iii</b>
<b>Table of Contents</b> . . . . .	<b>iv</b>
<b>List of Tables</b> . . . . .	<b>ix</b>
<b>List of Figures</b> . . . . .	<b>xii</b>
<b>Acknowledgements</b> . . . . .	<b>xx</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 The Evolution of Interactive Computing . . . . .	3
1.2 The Promise of New Technologies and Form Factors . . . . .	4
1.3 Body-Centric Interaction . . . . .	6
1.4 Problem Statement . . . . .	6
1.5 Research Questions . . . . .	7
1.6 Research Methodology and Overview . . . . .	8
1.7 Summary of Research Contributions . . . . .	9
1.8 Outline of the Dissertation . . . . .	10
<b>2 Related Work</b> . . . . .	<b>12</b>
2.1 Large Physical Surfaces . . . . .	12
2.2 Interactive Wall Display Systems . . . . .	14
2.3 Design Guidelines for Large Display Systems . . . . .	16
2.4 Interaction Techniques for Large Wall Displays . . . . .	18



2.4.1	Direct Touch Techniques . . . . .	18
2.4.2	Distance Techniques . . . . .	20
2.5	Reality-Based and Whole Body Interaction . . . . .	22
2.6	Summary . . . . .	24
<b>3</b>	<b>Shadow Reaching . . . . .</b>	<b>27</b>
3.1	Related Work . . . . .	27
3.2	Body-Centric Use Case Scenarios . . . . .	30
3.2.1	Scenario 1: University lecture halls . . . . .	31
3.2.2	Scenario 2: Construction planning meetings . . . . .	31
3.2.3	Scenario 3: Collaborative web design “war rooms” . . . . .	32
3.2.4	Scenario 4: Public events and installations . . . . .	32
3.2.5	Design requirements derived from the four use case scenarios	33
3.3	The Design of Shadow Reaching . . . . .	34
3.3.1	Supporting Distance Interaction . . . . .	35
3.3.2	Supporting Interpretable Actions . . . . .	38
3.4	Implementations . . . . .	39
3.4.1	Prototype 1: Real Shadows and Virtual Cursors . . . . .	39
3.4.2	Prototype 2: Virtual Shadows and Physical Interaction . . . . .	41
3.4.3	Prototype 3: Magic Lens Shadows . . . . .	43
3.5	Conclusions . . . . .	45
3.5.1	A Note About Shadow Geometries . . . . .	45
3.5.2	Limitations of Our Initial Prototypes . . . . .	46
<b>4</b>	<b>Body-Centered Interaction . . . . .</b>	<b>48</b>
4.1	Inspiration from Other Fields . . . . .	49
4.1.1	Interaction Spaces . . . . .	49
4.1.2	Binding Spaces and Shadow Interaction . . . . .	51
4.1.3	Proprioception . . . . .	52
4.1.4	Social Conventions . . . . .	53
4.2	Supporting the Development of Rich Whole-Body Interaction Tech- niques . . . . .	54
4.2.1	Scene Model . . . . .	54

4.2.2	Application Context . . . . .	55
4.3	Single User Interaction Techniques . . . . .	56
4.3.1	Virtual Shadow Embodiment . . . . .	57
4.3.2	Body-Based Tools . . . . .	57
4.3.3	Body-Based Data Storage . . . . .	59
4.3.4	Dynamic Light-Source Positioning . . . . .	60
4.4	Collaborative Interaction Techniques . . . . .	62
4.4.1	Synchronized Shadow Projections . . . . .	63
4.4.2	Access Control and Conflict Management . . . . .	63
4.5	Preliminary Evaluation . . . . .	65
4.6	Design Iteration . . . . .	67
4.6.1	Performance Improvements . . . . .	67
4.6.2	Shadow Visualizations . . . . .	68
4.6.3	Body-Based Control Surfaces . . . . .	69
4.7	A System for Supporting Universal Body-Centric Interaction in the Windows Operating System . . . . .	71
4.7.1	Event Management . . . . .	72
4.7.2	Rendering . . . . .	74
4.8	Conclusions . . . . .	75
4.8.1	Limitations . . . . .	76
<b>5</b>	<b>Text Input . . . . .</b>	<b>77</b>
5.1	Related Work . . . . .	79
5.1.1	Techniques for Large Display Text Input . . . . .	79
5.1.2	Techniques for Small Display Text Input . . . . .	80
5.2	Limiting the Design Space . . . . .	81
5.3	Candidate Interaction Techniques . . . . .	83
5.3.1	QWERTY Keyboard . . . . .	84
5.3.2	Circle Keyboard . . . . .	85
5.3.3	Cube Keyboard . . . . .	86
5.4	Implementation . . . . .	87
5.5	Experiment 1: Exploring Mid-Air Techniques . . . . .	89
5.5.1	Methodology . . . . .	89

5.5.2	Results	91
5.5.3	Discussion	97
5.6	Experiment 2: Investigating Distance Independence	100
5.6.1	Methodology	100
5.6.2	Results	102
5.6.3	Discussion	105
5.7	Conclusions	107
<b>6</b>	<b>Body-Centered API</b>	<b>110</b>
6.1	Drawing from Related Work	112
6.1.1	Lessons from Databases	112
6.1.2	Lessons from Sensor Fusion	113
6.1.3	Lessons from Augmented Reality and Virtual Reality	114
6.2	The Body-Centric Application Programming Interface	115
6.2.1	Uncertainty	117
6.2.2	BAPI Query Language	119
6.2.3	Limitations of the Design	121
6.3	Usage Scenario	122
6.4	A BAPI Implementation	125
6.5	Conclusions	130
6.5.1	Limitations	131
<b>7</b>	<b>Revisiting Fitts' Law When Gain Varies</b>	<b>132</b>
7.1	Related Work	135
7.1.1	Fitts' Law for Computer Pointing	135
7.2	One-Part and Two-Part Models of Pointing Performance	136
7.2.1	Pointing at a Distance and the Kopper $k$ Exponent	138
7.2.2	Unit Dependence in Two-Part Models of Pointing	139
7.2.3	Alternate Models	141
7.3	A Re-Analysis of Pointing Experiments Selected from the Literature	141
7.3.1	Physical Pointing on a Small Display	142
7.3.2	Mouse Pointing on a Large Display	146
7.3.3	Mid-Air Pointing on a Large Display	151

7.3.4	Speculation About Limitations in Previous Work . . . . .	153
7.3.5	Re-Analysis Conclusions . . . . .	154
7.4	Evaluation of Pointing Performance on Large Displays . . . . .	156
7.4.1	Apparatus . . . . .	156
7.4.2	Task and Stimuli . . . . .	157
7.4.3	Participants . . . . .	158
7.4.4	Design . . . . .	158
7.4.5	Results . . . . .	161
7.4.6	Discussion . . . . .	166
7.5	Conclusions . . . . .	169
7.5.1	Rethinking Fitts' Law for Modelling Pointing on Interac- tive Displays . . . . .	169
7.5.2	Developing a Model for Interaction with Very Large Wall Displays . . . . .	170
7.5.3	Future Work . . . . .	171
<b>8</b>	<b>Conclusions . . . . .</b>	<b>173</b>
8.1	Interaction Techniques . . . . .	174
8.2	Interaction Architecture . . . . .	174
8.3	Theoretical Models of Performance . . . . .	175
8.4	Final Words . . . . .	176
	<b>Bibliography . . . . .</b>	<b>177</b>
	<b>Appendices . . . . .</b>	<b>197</b>
<b>A</b>	<b>Text Experiment 1 Questionnaires . . . . .</b>	<b>198</b>
<b>B</b>	<b>Text Experiment 2 Questionnaires . . . . .</b>	<b>204</b>
<b>C</b>	<b>Fitts' Law Experiment Questionnaires . . . . .</b>	<b>208</b>
<b>D</b>	<b>Large Display Luminance Properties . . . . .</b>	<b>211</b>
D.1	Luminance as RGB Varies . . . . .	212
D.2	Luminance as Angle Varies . . . . .	213

# List of Tables

Table 2.1	A modified groupware matrix emphasizing modes of activity. Adapted from Tang et al. [193]. . . . .	14
Table 2.2	Comparison of systems (as of 1993) and potential next generation systems. Adapted from Nielsen [158]. . . . .	25
Table 3.1	Typical values for properties of each use case. It can be seen that these use cases are strongly associated with collaboration with large displays, and involve interaction by users at a distance from the display. . . . .	33
Table 3.2	Some design requirements for large display systems supporting several use cases. Distance input is a universal requirement, and other input types, including text input and touch input, are also expected to present. . . . .	34
Table 3.3	The Shadow Reaching design space as sampled by the three prototype systems. . . . .	45
Table 5.1	Design space matrix of distance- and visibility-dependence, with some representative techniques. Emphasized techniques are evaluated in this chapter. One cell in the matrix is empty due to a lack of reasonable representative techniques. . . . .	84
Table 5.2	Summary of significant pairwise comparisons for movement time to sub-cubes with different number of hard sides. . . . .	93
Table 6.1	Sensors mapped to the requirements of the application. Some requirements demand a fusion of data collected from multiple sensors. . . . .	123

Table 6.2	Major classes implemented as part of the scene model. Important properties of the classes are named and described. . . . .	128
Table 6.3	Major classes implemented as part of the sensing component. Important properties of the classes are named and described. . .	130
Table 7.1	Modelling of movement time from Graham data using the Fitts formulation, the Shannon formulation, and the Welford two-part formulation. For the Fitts and Shannon formulations $R^2$ decreases with lower gain. For the Welford formulation $R^2$ is consistently good. Fitts formulation data is from Graham [69].	145
Table 7.2	Modelling of movement time from Casiez et al. data using the Fitts, Shannon, and Welford formulations. $R^2$ values for the Fitts and Shannon formulations decrease with lower gain, but the Welford formulation is consistently good. The data was provided by Casiez et al. . . . .	149
Table 7.3	Results of a statistical $F$ -test comparing regressions using the Welford formulation to those using the Fitts formulation for the Casiez data. For gain levels with significant results, the Welford formulation models the data significantly better than the Fitts formulation. . . . .	151
Table 7.4	Modelling of movement time from Tsukitani data using the Fitts formulation, the Shannon formulation, and the Welford two-part formulation. . . . .	152
Table 7.5	Summary of Fitts' law experiments analyzed, and conclusions drawn. . . . .	155
Table 7.6	Significant ANOVA results for movement time in the mid-air large display pointing experiment. . . . .	161
Table 7.7	Linear regression constants determined when using the Fitts formulation, the Shannon formulation, and the Welford two-part formulation. Movement times were averaged over all participants. Actual movement amplitude $A$ and actual target width $W$ were used. . . . .	162

Table 7.8	Linear regression constants determined when using the Fitts formulation, the Shannon formulation, and the Welford two-part formulation. Movement times were averaged over all participants. Actual movement amplitude $A$ and effective target width $W_e$ values were used. . . . .	162
Table 7.9	Results of a statistical $F$ -test comparing regressions using the Welford formulation to those using the Fitts formulation for our experimental data. Data analyzed was actual width data from the mid-air pointing experiment. For gain levels with significant results, the Welford formulation models the data significantly better than the Fitts formulation. . . . .	164
Table 7.10	Results of a statistical $F$ -test comparing regressions using the Welford formulation to those using the Fitts formulation for our experimental data. Data analyzed was effective width data from the mid-air pointing experiment. For gain levels with significant results, the Welford formulation models the data significantly better than the Fitts formulation. . . . .	164
Table 7.11	Significant ANOVA results for error rate in the mid-air large display pointing experiment. . . . .	166
Table D.1	Luminance as it varies based on RGB and distance of projector from display. . . . .	212
Table D.2	Luminance as it varies based on angle of meter to display. . . . .	213

# List of Figures

Figure 1.1	The dependency of major components in our development of our body-centric interaction approach. Our approach combines the design of new interaction techniques, the development of an interaction architecture for supporting new applications, and the development of a theoretical understanding of low-level performance in body-centric contexts. . . . .	9
Figure 3.1	Users interacting with a large wall display system employing a shadow input metaphor. The installation was located in a science center (developer of system is unknown), and users explored the system without any prompting or instruction. Photos taken by the author. . . . .	28
Figure 3.2	A mock-up of two users interacting on a large display using laser pointer style input. A third party (collaborating directly or simply observing) may want to know which user is performing which operation, in order to either understand the logic behind the sequence of actions, or in order to integrate their own actions in a reasonable manner. Without richer feedback, unfortunately, it is difficult to determine which of the two users is controlling which of the two cursor. . . . .	36
Figure 3.3	The user can control the reach of her shadow by moving closer to and farther away from the display. . . . .	37



Figure 3.4	The size of a user’s shadow, and related control-display gain, depends directly on the distance of the user from the display, and the distance of the light from the display. The user can smoothly adjust gain by moving in the room. . . . .	38
Figure 3.5	Shadow Reaching prototype 1. A physical shadow is cast with a bright lamp. The user’s hand locations are measured using trackers. Cursors are rendered on the display near the shadow. . . . .	40
Figure 3.6	Placing a real lamp in the room creates physical shadows on the display. Users near the middle of the display generate clear shadows, but users near the edge of the display generate indistinct shadows due to the intensity falloff of the directional lamp. . . . .	41
Figure 3.7	Shadow Reaching prototype 2. A user bounces balls using his virtual shadow. . . . .	42
Figure 3.8	Placing a camera and infrared light source in the locations shown allows the camera to see users as a backlit shape. Vision algorithms can then extract the contour of the user. . . . .	43
Figure 3.9	A mockup of users interacting using Magic Shadows. Their shadows display satellite photos, while the context display maps. . . . .	44
Figure 4.1	Peripersonal space (red volume) is that which is in reach of the arms, and often does not intersect with a large display during use. . . . .	49
Figure 4.2	The pipeline used to generate the scene model. Hand and shoulder locations of the user are measured with magnetic location trackers, then a skeleton estimation of the user’s pose is generated, then a human mesh is mapped to the pose of the skeleton. . . . .	55
Figure 4.3	Screenshot of the map exploration and editing application. Users can sketch geo-referenced annotations, type text, and insert documents. . . . .	56

Figure 4.4	A user accesses a tool stored on her right hip by placing her hand at that location. A variety of tools can be stored at different body locations. . . . .	58
Figure 4.5	A user accesses her personal data store. The data store is centered on the user's personal shadow embodiment. She can browse through the file hierarchy and move documents to the shared space. . . . .	59
Figure 4.6	Three possible light source behaviours, coded by colour. Green: user following. Red: orthographic. Yellow: manually positioned. This visualization was created in a modified version of the experimental system using real data. . . . .	61
Figure 4.7	As a first user approaches a second user and enters that user's private space, both users' light sources transition to behaviours that are conducive to collaboration. . . . .	64
Figure 4.8	A user passes a private document to a collaborator. The sharing protocol requires close physical proximity, and encourages direct eye contact. Feedback on the screen is a green circle surrounding the projection of the two users hands, to indicate a successful pass. . . . .	65
Figure 4.9	Four different styles of shadow rendering developed to explore shadow embodiments. From left to right: real geometry, sharp shadows, soft shadows, and body contour. . . . .	69
Figure 4.10	Two examples of body-base control surfaces. Left: a 1 dimensional slider mounted on the user's arm. He moves his hand up and down his arm to adjust a numeric value. Right: a 2 dimensional colour selector. The user selects a colour with one hand and draws with the other. . . . .	70
Figure 4.11	A user interacting with Microsoft Visio. The interaction techniques supported by our system allow the user to select and drag objects in the application as he would normally do with a mouse. . . . .	72

Figure 4.12	A user interacting with Microsoft Visio. A virtual keyboard integrated into the system allows the user to input text without using a physical keyboard, and without being within touch distance of the display. . . . .	73
Figure 4.13	Flow of click events in the application. Clicks with the Wii Remote signal the event coalescer. The coalescer determines the Hand that currently has control of the cursor based on priority: first click grabs control until the click is released. The coalescer then determines the cursor location based on the location of the hand as projected to the display. The coalescer then sends mouse control events to the operating system. . . .	74
Figure 5.1	The three text input techniques as used in Experiment 1. From left to right: Circle, QWERTY, and Cube. Dimensions refer to the size of the feedback during the experiment. . . . .	83
Figure 5.2	Selection of a character in the Circle technique is based on the point of intersection of a ray cast from the input device. The angle of the intersection point relative to the origin determines the selected character. . . . .	85
Figure 5.3	Hypothesized relative performance of selecting sub-cubes using the Cube technique, based on number of impenetrable sub-cube sides. . . . .	87
Figure 5.4	Triangulating position of hand-held Wii Remote using 2 fixed Wii Remotes on stands. Red lines indicate vectors from detected IR light source to fixed Wii Remotes. . . . .	88
Figure 5.5	Mean input speed in words-per-minute for the three text input techniques. Error bars represent standard error. $N = 12$ . . . .	92
Figure 5.6	Mean time for users to point to sub-cubes with three, two, one, and zero hard faces. Error bars represent standard error. $N = 12$ .	93
Figure 5.7	Mean time for users to point to sub-cubes in the front, middle and back layers. Error bars represent standard error. $N = 12$ . .	94
Figure 5.8	Mean error rates for the three text input techniques. Error bars represent standard error. $N = 12$ . . . . .	95

Figure 5.9	Mean scores for the three text input techniques from a NASA TLX based questionnaire. Ratings are on a scale of one to five (longer bars are better). Error bars represent standard error. $N = 12$ . . . . .	96
Figure 5.10	Mean scores from user rankings of the three techniques from best to worst (1=best, 3=worst, shorter bars are better). Error bars represent standard error. $N = 12$ . . . . .	97
Figure 5.11	The Circle keyboard (left) and QWERTY keyboard (right) interfaces as used in Experiment 2. . . . .	101
Figure 5.12	Performance for the four text input conditions in words per minute. Error bars represent standard error. $N = 16$ . . . . .	103
Figure 5.13	Mean error rates (percentage) for the four text input conditions. Error bars represent standard error. $N = 16$ . . . . .	104
Figure 5.14	Mean scores for the four text input conditions using a NASA TLX based questionnaire. Ratings are on a scale of one to five (longer bars are better). $N = 16$ . . . . .	105
Figure 5.15	Mean user rankings for the four text input conditions from best to worst (1 = <i>best</i> , 4 = <i>worst</i> , shorter bars are better). $N = 16$ . . . . .	106
Figure 6.1	The design of our Body-centric application programming interface puts the user at the center of the model. Development is performed by primarily querying user states, rather than accessing devices such as mice and sensors and querying their states. . . . .	116
Figure 6.2	Data fusion pipeline for producing a single consistent scene model. Adapted from Hall and Llinas [77]. . . . .	117
Figure 6.3	Hybrid data fusion pipeline for producing a single consistent scene model. A hybrid model allows optimized early processing of raw data from similar sensors. Adapted from Hall and Llinas [77]. . . . .	118

Figure 6.4	Uncertainty is captured in one of three ways in the response to the query for position of an object. <i>Point uncertainty</i> assumes that the position is exact and correct. <i>Interval uncertainty</i> provides a range of possible values. <i>Probabilistic uncertainty</i> provides a function describing the likelihood of the position being of a certain value. . . . .	120
Figure 6.5	Processing pipeline of data from a variety of sensors to produce a coherent scene model for use by an application. In the first step raw data is captured. In the second step raw data is processed in isolation. In the third step state vectors are transformed into a consistent coordinate system. In the fourth step state vectors are associated with entities in the scene. Retained state information may be reused to refine further iterations of model construction. . . . .	124
Figure 6.6	Representation of users in a scene. Red circles represent some of the pertinent details that must be captured from different sensing systems. Examples in the image include touch locations and gaze fixations. Other data could include body contours and user identities. . . . .	125
Figure 6.7	UML class diagram of major classes in the modelling component of the implemented architecture. . . . .	127
Figure 6.8	UML class diagram of major classes in the sensing component of the implemented architecture. These classes are generally responsible for updating the scene model to properly represent the physical scene. The LightModel classes are special purpose, and set the locations of virtual light sources according to specific behaviours. . . . .	129
Figure 7.1	Lines connect points representing tasks with the same amplitude. With $A$ held constant movement time varies roughly linearly with $ID$ . Adapted from Graham [69, page 47]. . . . .	142

Figure 7.2	Lines connect points representing tasks with the same target width. With $W$ held constant movement time varies roughly linearly with $ID$ . The data points are exactly the same as in Fig 7.1, only the lines have been drawn differently. Adapted from Graham [69, page 47]. . . . .	143
Figure 7.3	A regression analysis of the Graham data using average MT results for every $ID$ value at $gain = 1$ . Poor fit is concealed due to averaging of data points. . . . .	144
Figure 7.4	A regression analysis of the Graham data using MT results of every combination of $A$ and $W$ at $gain = 1$ . Poor fit is evident as a result of analyzing all data points. Adapted from Graham [69, page 45]. . . . .	145
Figure 7.5	The $k$ values for the Graham data. . . . .	146
Figure 7.6	Movement time results for all $A/W$ combinations for the Casiez data at $gain = 2$ . Lines connect points representing tasks with the same amplitude. . . . .	147
Figure 7.7	Movement time results for all $A/W$ combinations for the Casiez data at $gain = 2$ . Lines connect points representing tasks with the same target width. The data points are exactly the same as in Fig 7.6, only the lines have been drawn differently. . . . .	148
Figure 7.8	The $k$ values for the Casiez data. . . . .	150
Figure 7.9	Regression analysis of the Tsukitani et al. results using a Fitts' model. . . . .	152
Figure 7.10	The $k$ values for the Tsukitani data. . . . .	153
Figure 7.11	Layout of experimental apparatus. Labelled components are: (1) center target, currently not active. (2) cursor. (3) right target, currently active. . . . .	157
Figure 7.12	A user interacting with the experimental system. The view from the back of the user shows the elements on the screen: the center target in grey, the right target in blue, and the cursor in between the two targets. . . . .	158
Figure 7.13	The Wii Remote mounted with reflective Vicon tracking markers.	159
Figure 7.14	The $k$ values relative to gain computed using actual $A$ and $W$ . . . . .	163

Figure 7.15	The $k$ values relative to gain computed using actual $A$ and effective $W_e$ . . . . .	163
Figure 7.16	Dependence of $b_1$ on gain. . . . .	165
Figure 7.17	Dependence of $b_2$ on gain. . . . .	165
Figure 7.18	Mean scores of task difficulty overall, at low gain (2, 5), medium gain (8, 12) and high gain (16, 20) levels, with standard error. Ratings on a scale of one (impossible) to five (easy). $N=19$ . . .	167
Figure D.1	Luminance as it varies based on RGB and distance of projector from display. . . . .	213
Figure D.2	Luminance as it varies based on angle. . . . .	214

# Acknowledgements

Many people contributed to this dissertation. Primary thanks goes to my supervisor, Kellogg Booth. The fact that I am no longer surprised when he answers an email at 1:00 AM, or 5:00 AM, or 9:00 AM, or 2:00 PM, or 10:00 PM, or on a weekend or Christmas Eve, speaks to his dedication. My supervisory committee of Sidney Fels, Jocelyn Keillor, and Michiel van de Panne, provided continuous encouragement and invaluable feedback over the years, and only occasionally gave me a hard time. The examining committee of Peter Graf and Alan Mackworth and the examining chair, Paul Gustafson, also made much appreciated contributions. The external examiner, Patrick Baudisch, who's work I've admired over many years, simultaneously provided me with support and set me straight on some important points. I am grateful on both counts.

The communities I've worked in have helped me in innumerable ways. Specific UBC students, including but not limited to Tony Tang, Joel Lanir, Karen Parker, Leah Findlater, and Karyn Moffat have all contributed either directly or indirectly to making my work better. At the University of Osaka, Yoshifumi Kitamura provided a welcome home. I wish all of his students good luck in the future. Gambate.

Many thanks to Radiohead, Weezer, Sigur Rós, Explosions in the Sky, Godspeed You! Black Emperor, Portishead, Pink Floyd, and Arcade Fire. I couldn't write without you.

I am grateful to my parents, who made me who I am. I am most grateful to Iliana, for everything.

The research reported in this dissertation was funded by the Networks of Centres of Excellence of Canada through GRAND, the Graphics, Animation and New Media NCE, and by the Natural Sciences and Engineering Council of Canada under



the postgraduate scholarship program, the discovery grants program, the strategic project grants program, and under the strategic networks program through NECTAR, the Network for Effective Collaboration Technology through Advanced Research. Microsoft Corporation and SMART Technologies were major sponsors of NECTAR. Defense Research and Development Canada supplemented the NSERC postgraduate scholarship. Additional equipment and other infrastructure necessary for the research was provided by funding from the Canadian Foundation for Innovation and the British Columbia Knowledge Development Fund. The Department of Computer Science at UBC provided technical and administrative support that made the research possible. All of this support is gratefully acknowledged.

Under no circumstances should you ever do a PhD. It is a complete waste of time.

– Dr. Edward M. Shoemaker

For all the darkest nights that passed me by  
Unlit by spark of novel thought I sat  
Crazed dreams of input's promise seen then die  
Kept firm, held fast, my thesis to combat

The days drew bright and warm and chapters grew  
High hopes had I that final draft was close  
If thought in written form can fit I knew  
Somehow my thesis finished would I boast

Set bells to chime, defence is done and past  
Hard work remains, to make revisions clear  
I work more nights, we promise you our last  
Til acrid end, so close yet still I fear

Year starts anew, plus three did thesis rob  
Onwards I move, and have to get a job

# Chapter 1

## Introduction

In this dissertation we introduce and investigate a body-centric interaction approach designed specifically for large wall displays. England et al. [53] define Whole Body Interaction as “the integrated capture and processing of human signals from physical, physiological, cognitive and emotional sources to generate feedback to those sources for interaction in a digital environment.” We choose such an approach for several reasons. First, large displays (and surfaces) are frequently used collaboratively, and the geometric inter-relationship of bodies is a critical component of the collaboration process. Second, users of large displays tend to move their bodies in space. They move to look at different regions of the display, to access items in the environment, or to collaborate with another person. Third, the scale of a large display is similar to that of a human body. While most of the body of a user of a traditional display remains fairly stationary, and indeed may be very constrained by a sitting posture, almost the whole body of a user of a large display is likely to be in constant motion and could potentially be an integral part of the interactive process.

Body-centric interaction is both a philosophical approach for designing interactions, and a practical means of supporting system design that is well suited to emerging computing form factors, especially large display form factors. From a user’s viewpoint body-centered interaction is exemplified by techniques that leverage natural capabilities and properties that have evolved over millennia. From a designer’s viewpoint, body-centered interaction suggests natural design choices,

and helps keep the user central to the design process. We will investigate several areas of body-centric interaction that will help set a foundation for the field of research.

A theme that carries through our work is that of a user shadow as an embodiment of user action. We will find that user shadows enable expressive input, support awareness in collaborative settings, and serve as a unifying foundation for the development of a whole suite of body-centric interaction techniques.

Two terms will be used a lot in this dissertation: “large display” and “model.” We consider a large display to be one that is considerably larger than a person’s arm span and thus cannot be viewed in its entirety while a user is standing close enough to touch the surface, and for which the user must move around a lot to directly touch everything if, indeed, that is even possible. We employ the term “model” in a number of ways. We develop a high level model of interaction that is a description of how people use large displays and how interaction techniques can be designed to support this use. The synthesis of this model is the topic of this dissertation. The high level model is supported by a much more concrete lower level model. The low level model is a software architecture that can serve as the basis for designing techniques that are referred to in the literature as “reality-based” or “whole body” interaction. In this case a model is an internal geometric representation of a user’s location, body posture, gaze direction, intention, and relationships to other users and the objects and artifacts that are physically or virtually present in the room or other setting in which interactions take place. This is of course a tall order. In practice a body-centric model will often be an approximation to this ideal. Another notion of model that we use is much narrower. We will be interested in models of human performance, such as Fitts’ law. A fully developed high level body-centric model would no doubt incorporate one or more performance models in addition to everything else, but performance models are also used on their own to provide a theoretical framework in which to evaluate interaction techniques. These terms and many others will be described in more detail at the appropriate points in the narrative.

## 1.1 The Evolution of Interactive Computing

The currently dominant model of computing interaction was introduced by Engelbart et al. with NLS [52]. This so-called WIMP (windows, icons, menus, pointer) model of interaction stresses a single user working independently while seated at a desk. The user employs a mouse and a keyboard for input, and receives feedback on a relatively small vertically oriented display at roughly arm's length. Engelbart's model was further refined and commercialized with the twin introductions of the Xerox Star [14] and the Apple Macintosh [7]. Several aspects of the traditional desktop computing system and the model of interaction that it supports pose limits on how users work. First, the user generally works in isolation. The system is not designed to support face-to-face collaboration. Second, interaction with the system generally only involves a very small portion of the user's body, and a fraction of the body's highly evolved capabilities. Only the fine motor skills of the user's hands are used for input. The ability of a human to navigate and manipulate in three-dimensions, to sense their own body in space, and to relate to other humans, are not fully employed.

The recognized limitations of traditional computing systems have prompted researchers to call for new interaction models and new hardware form factors to support them. A consensus is yet to emerge regarding how these devices will function and how users will interact with them, but researchers have provided some useful guidelines. Myers et al. [153] argued that "new interfaces will break out of the desktop box where they have been stuck." Nielsen [158] believes that "it may be one of the defining characteristics of next-generation UIs [user interfaces] that they abandon the principle of conforming to a canonical interface style and instead become radically tailored to the requirements of individual tasks." The predictions of van Dam include systems that exploit all human senses, include natural language processing, and support multiple users [203]. Some researchers have proposed that employing the whole human body may be a key element of some Post-WIMP interfaces. Jacob et al. have argued that Reality-Based Interfaces (RBIs) are an emerging theme in many emerging interface designs [101]. The concept of RBIs includes a major emphasis on natural use of the human body. Klemmer et al. have also argued for an emphasis on whole body interaction in

next-generation computing interfaces [118].

We are currently in an era of rapid evolution in computing interfaces. Advances in sensing technologies, new display form factors, and an enthusiasm by researchers to explore options outside of the traditional WIMP model of interaction, are leading us towards the acceptance of new kinds of computing. There remains a great deal of uncertainty in the evolution of interaction. We may end up with many interface styles that are tailored to specific tasks. We may end up with models that are dramatically different from the traditional WIMP model. We may also end up with interaction styles that emphasize the richness of human whole body input. Helping define the next stage in the evolution of the computer interface is the driving force behind this dissertation.

## **1.2 The Promise of New Technologies and Form Factors**

We are on the cusp of a new era in computing. We were locked into a single model of interaction for approximately 30 years, but momentum is building to a point where new computing form factors and approaches are becoming widespread and accepted by the general public. Instead of being constrained by the common interaction layout of mouse and keyboard on a desk in front of a static display, we are more and more able to take advantage of computing capabilities in different contexts.

Handheld computing is a prime example of a new computing form factor and approach gaining in popularity. While handheld computing saw early commercialization in the 1990's, it met limited success, with 170,000 personal digital assistants (PDAs) sold in 1995, and 11.7 million sold worldwide in 2000 [56]. Sales have exploded in recent years, and smart-phones are now ubiquitous, mostly having replaced old-style PDAs. Gartner reports that 54.3 million smartphones were sold in the first quarter of 2010 alone [67]. Interaction with smart-phones is markedly different from traditional interactive computing. Users do not use a traditional mouse and keyboard. Instead, small keyboards and touch-screens are the standard input devices, and output has eschewed the traditional windowed presentation in favour of feedback models appropriate to small screens. Handheld devices are powerful, in that they allow a user to move freely while interacting. This is a promise that is

relevant to users of other devices, such as large displays.

Large displays also hold significant promise in changing how people interact with computers, and in supporting new models of interaction. The design of large display systems draws significant inspiration from interaction in the physical world. In the physical world, people are not so constrained by input and output devices as they are in the virtual world. A person can sit or stand at a small or large table, can stand and work at a whiteboard, and can move about a room interacting with collaborators. The physical real estate available is almost always much greater than the real estate available on standard desktop monitors. Furthermore, the physical world offers a much richer set of input devices than does the virtual world. A person can use pens and pencils, scissors, paintbrushes, straight-edges, miter saws, pool cues, and violin bows. Regardless of the task at hand, there is almost always a workspace configuration and set of tools, sometimes highly specialized, that are appropriate. Work and interaction in the physical world is much more flexible, fluid, and dynamic than work that is constrained by the limitations of traditional desktop computing. One goal of developing large display systems is to maintain the advantages of physical world interaction, while simultaneously benefitting from the computational abilities of modern technology.

Despite the potential of large display systems, there remains much to be accomplished in defining how these systems should be designed. Although large interactive displays share many of the properties of large physical surfaces, they are more than just surfaces. If we are to exploit the additional capabilities made possible by interactive surfaces we must look beyond the limiting history of traditional computing interfaces, and also beyond interaction with physical surfaces overly constrained by physical reality, while still respecting the lessons that have been learned in relation to both. This means defining a new model of interaction and understanding the implications of that model for both users and developers. We must do so carefully. There is a risk we will fall into the same trap that early human-computer interaction designers fell into, where decisions become de facto standards based not on merit, but on a short-term lack of better alternatives.



### **1.3 Body-Centric Interaction**

In considering interaction models for any new form factor, especially that of large wall displays, we must recognize the clear trend in human-computer interaction towards reality-based interfaces [101] and interfaces that involve the whole body [118]. These interfaces leverage the natural abilities of human beings that have evolved over millenia. Rather than depend on arbitrary mappings between icons and pointers and other abstract representations present in WIMP interfaces, they place more emphasis on the role of the human body, physical context, and direct manipulation. Referring back to England's definition of whole body interaction [53], we conclude that making use of the full spectrum of input available from the body, what he calls the "integrated capture and processing of human signals," provides a basis for the design of a new form of computing interaction. This is a holistic approach that can potentially capture much of what is important to the interactive experience.

While the trend towards such interfaces is clear, research in the area has tended to be piece-meal. Isolated research systems have been developed, and sometimes evaluated. Interaction techniques involving one or perhaps a few body parts have been introduced, their designs often driven by intuition. The potential benefits of whole-body interfaces are clear, but there has been no concerted effort to investigate the practicalities of how a whole-body approach can be realized, and what the benefits to users might be. This dissertation is a step in that direction.

### **1.4 Problem Statement**

Interaction techniques for computing systems were developed in an era where technical limitations put huge restrictions on what was practical to develop. It is also the case that researchers and designers didn't have the broad experience of past work we have today to draw on for inspiration. These two factors contributed to the development of a dominant model of interaction (WIMP) that assumes the use of certain input devices (a mouse or mouse-like device, and a keyboard) used in a certain configuration (while seated at a desk). This model supports certain kinds of tasks very well, but others not so well and some not at all.

The main research problem addressed in this dissertation is how to design interaction techniques suitable for display form factors and use cases other than what

are supported by traditional systems. In particular, we focus on large wall displays as a form factor. In our design context, large displays are those that are considerably larger than a person's arm span, such that a person can't physically reach across the display without moving their body, and can't have detailed view of all regions of the display at once.

## 1.5 Research Questions

The high-level goal of this dissertation is to investigate interaction approaches that are suitable for use with interactive very large wall display systems. More specifically, our goal is to answer the following research questions:

1. What is an appropriate model of interaction for use with very large wall display systems?
  - (a) What metaphors can be leveraged that aid in increasing user comprehension and performance with these systems?
  - (b) What can we learn from other fields that might inform our adoption of a model of interaction?
2. What novel interaction techniques are made possible by the adoption of a new interaction model?
  - (a) What techniques best address interaction challenges specific to large wall displays, such as the problem of reaching over a distance, and supporting group awareness?
  - (b) How can techniques build upon physical metaphors in order to make interaction more natural and powerful?
  - (c) Certain atomic interaction tasks, such as text input, must be supported regardless of the model of interaction. How can we support these fundamental tasks with the new interaction model?
3. How can we ease the development of new interaction techniques and applications using this new model of interaction?

- (a) A new model of interaction can demand the use of sensors and devices in ways unintended when they were designed. How can we allow developers flexible use of sensors and devices?
  - (b) How can we support the extension of our model in directions that even we do not anticipate? What can we do to “future-proof” our approach?
4. What implications on low-level performance are there to adopting a new model of interaction, and supporting interaction in a way that was previously not possible or convenient?
- (a) Are established theoretical models of performance adequate to describe performance with these new form factors?

## **1.6 Research Methodology and Overview**

Our approach is to employ perspectives from technology, psychology, and design, in order to pursue answers to our research questions.

From a technology perspective, we must recognize the limitations of existing hardware and software, but simultaneously anticipate future advances in both. In each of our investigations we embody our ideas in prototype implementations that can be evaluated to at least some extent. In some cases these evaluations are preliminary (e.g. Chapter 4), and in some cases they are more formal (e.g. Chapter 7). In most cases the prototype systems are some distance from being refined enough to be widely deployed. It is our belief, however, that our approach of developing techniques that anticipate future technologies will serve the human-computer interaction community well as technology evolves.

From a psychology perspective, we draw on some of the vast collection of knowledge available in that field. Psychology and related fields such as kinesiology have had a much longer time to develop models of human behaviour, compared to the field of human-computer interaction, and there is much of value that can be drawn from their models. It is especially pertinent to the subject of study in this dissertation. We both build techniques that use psychological insights as a foundation, and we perform controlled experiments that expand on well-researched psychological theories.

<b>Body-Centric Interaction</b>		
<b>Novel Interaction Techniques</b>	<b>Interaction Architecture</b>	<b>Theoretical Models of Performance</b>
<i>chapter 3, 4, 5</i>	<i>chapter 6</i>	<i>chapter 5, 7</i>

**Figure 1.1:** The dependency of major components in our development of our body-centric interaction approach. Our approach combines the design of new interaction techniques, the development of an interaction architecture for supporting new applications, and the development of a theoretical understanding of low-level performance in body-centric contexts.

From a design perspective, we attempt to always couch our interaction techniques in what is familiar and intuitive to everyday people. We wish to make our techniques inviting, while also leveraging the additional power that computing makes available to support more powerful ways of accessing manipulating information.

The components of the research reported in this dissertation are summarized visually by the diagram in Figure 1.1. Our theory of body-centric interaction emerges as the combination of a variety of elements. The novel interaction techniques that we describe demonstrate possible specific means of interacting that stress the role of the whole human body. The interaction architecture that we propose responds to the practicalities of realizing our approach. Finally, the theoretical models of performance that we develop provide insight into low-level human performance when operating in contexts that we define as being body-centric.

## **1.7 Summary of Research Contributions**

We summarize the primary contributions of the research described in this dissertation. These are revisited in the final chapter (Chapter 8) of the dissertation.

1. We implement an interaction technique involving a shadow-based metaphor for manipulation of information on very large wall displays. The technique

is designed with the goal of allowing users to reach across large distances and interact outside of arms reach of the display and the information represented on it. The technique is also designed to support mutual awareness of multiple users' interactions in collaborative scenarios. We describe a number of possible implementations.

2. We review a portion of the psychology literature to motivate a “body-centric” approach to interaction with large wall displays. We design a number of interaction techniques that employ the user’s body in different ways. We implement these techniques in a map exploration application and a generic application that runs under and augments the interaction capabilities of the Windows operating system.
3. We explore text input on very large wall displays in a body-centric context. We design, implement, and evaluate a number of prototype techniques in two controlled experiments. We highlight the two properties of distance-dependence and visibility-dependence as they apply to large display interaction techniques in general.
4. We present an extensible software and hardware architecture for supporting the development of novel interaction techniques and applications based on a body-centric model of interaction. We describe an implementation of a subset of the features described in the architecture.
5. We study the low-level performance properties of pointing on large displays, which are applicable in particular to the techniques we have developed. We show that accepted models of pointing performance do not adequately model large display use, and we present a theoretical model that better predicts performance.

## **1.8 Outline of the Dissertation**

In Chapter 2 we discuss related work. In Chapter 3 we describe our initial investigation into body-centric interactions that focussed on employing the user’s shadow as a virtual embodiment. In Chapter 4 we explore a broadening of that investigation

into more general body-centric interactions, with additional interaction techniques motivated by previous research in psychology and sociology. In Chapter 5 we describe interaction techniques and related evaluations specific to text input on large wall displays. In Chapter 6 we present an architecture for supporting the development of novel body-centric interaction techniques, and a related implementation. In Chapter 7 we explore the applicability of Fitts' law to large display pointing, find it to be lacking, and offer an alternate model that extends a formulation introduced by Welford. In Chapter 8 we summarize our results and reflect on the significance of our overall contributions and the potential for future work in this area.

## Chapter 2

# Related Work

In this chapter we provide a summary of previous work that is relevant to all of the other chapters in this dissertation. The two general categories of prior research include that specific to very large wall displays, and that related to the development of a “body-centric” model of interaction. Later chapters will include sections dedicated to additional related work specific to those chapters.

### 2.1 Large Physical Surfaces

In the physical world we complete tasks aided by a variety of different surfaces. Desks serve as a horizontal support for a multitude of items such as pens, paper, and other tools and artifacts. Specialized desks such as drafting tables hold materials at a certain angle to optimize work. Larger tables serve to support a number of people gathering in a group to collaborate. These large surfaces all play important roles in the work environment. As a specific example of the power of large surfaces, Buxton states that the introduction of the blackboard into the classroom in the 19<sup>th</sup> century was a huge advancement in educational technology, as compared to earlier individual slates [23]. He argues that the change in *scale*, *location*, and *usage* of the large surface is what was important in the blackboard. These properties supported a social and organizational change that reshaped how people learn.

The wide variety of large physical surfaces deployed in the real world serve in a sense as workspace “toolboxes” [164]. There is a need for a variety of form

factors to support a variety of tasks, and users of physical surfaces have the option of selecting the appropriate “tool” for any individual task. In the computer world, if we wish to broaden the set of tasks that computational systems can efficiently support, we need to explore new form factors for computational interfaces that are as flexible as the set of physical surfaces at our disposal.

A great deal is known about how people interact with physical surfaces. Cherubini et al. [36] investigated how software developers employ whiteboards to support the creative process. Results of their interviews showed that developers use whiteboards to draw diagrams in order to *understand*, *design*, and *communicate*. It was also found that whiteboards are useful at supporting face-to-face communication and allowed developers to externalize their mental models of code. Myntatt [156] performed an evaluation examining physical whiteboard use in a real-world context. She found that the simplicity of using a whiteboard, and the impossibility of permanent storage and flexible editing, are defining characteristics of the whiteboard experience, on both the positive and negative side. Tang et al. [193] performed their own evaluation of physical whiteboard use, with the goal of informing the design of large display software. They constructed a  $2 \times 2$  matrix (Table 2.1) with dimensions of *independent* vs. *collaborative* and *synchronous* vs. *asynchronous* and found that whiteboards are commonly used for all four combinations, with independent asynchronous being most common (appearing on 61% of whiteboards surveyed). It was found, however, that this matrix was insufficient to capture all aspects of use. They identified “ongoing reference on a semi-public whiteboard,” “lo-fi ideation, deferral and storage of personal activity,” and “persistent team scheduler” as activities supported by whiteboards that are not easily classified.

A lot is also known about how people interact with physical information artifacts, particularly paper. For example, researchers have examined how people organize paper on their desks [138]. Key lessons include that people organize their desks in such a way to not only help them *find* what they are looking for, but also to help them *remember* what they need to do. It was also found that categorizing information is a challenging task that is not possible to do perfectly. Whittaker and Hirschberg investigated the management of physical paper in an office environment [210]. They found that duplicates of information are often kept, despite



	Independent	Collaborative
Sync	<i>Worker</i> Word processor Spreadsheet CAD software	<i>Real-time interaction</i> telephone Video conferencing Instant messaging
Asyn	<i>Personal management</i> PIM, schedule, agenda, task list Reminders, post-it notes	<i>Ongoing tasks</i> Team rooms Bulletin boards Email

**Table 2.1:** A modified groupware matrix emphasizing modes of activity. Adapted from Tang et al. [193].

the need for only a single copy, and that people tend to be primarily either *filers* or *pilers*, with differences in storage behaviour for both.

## 2.2 Interactive Wall Display Systems

A number of systems have been developed that make use of large interactive displays to support different tasks. Examinations of these systems help clarify the landscape of large interactive display systems, and can provide hints as to where large interactive displays can find a home.

ClearBoard [99] is a prime example of a large display system that leverages physical properties of the surface, while providing enhanced functionality. The ClearBoard system builds on an elegant metaphor: that two users are working at a glass wall, and can see each other through the surface while they draw on it. This is something that can be prototyped trivially in the physical world, using nothing more than a window and marking pens, but the ClearBoard system makes the same thing possible over large distances, with the support of cameras and projectors. The cameras capture each user, and project the remote user on top of the local user's work surface. ClearBoard is interesting both for its mimicking of physical windows, and also for its exploration of user embodiments in the workspace.

Tivoli is another early example of a large display system [163]. It is an electronic whiteboard application that was designed to support workgroup meetings. The design goals of the system emphasized ease of learning and mimicking of

traditional physical whiteboards, while also introducing additional capabilities. Interaction with Tivoli is accomplished primarily with a stylus that writes strokes on the interactive surface. The strokes are stored by the system as atomic objects, and can be manipulated (e.g. selected, moved). While the description of Tivoli does not include an evaluation, it is important to note the emphasis designers put on both leveraging physical metaphors, and adding additional capabilities made possible by the computational abilities of the computer. This is an early example of what could be described as a “Reality-Based Interface,” as described by Jacob et al. [101], which we will examine later.

Flatland is another example of a large interactive display system developed for deployment [157]. It is also a whiteboard-like system developed for the purpose of supporting informal office work. Like Tivoli, Flatland also employs a pen as the primary input device, and also groups strokes into groups for easy manipulation. Flatland, however, incorporates broader editing abilities accessed through both pie menus and through gestures. Flatland also includes what are called “behaviours,” which is post-processing applied to strokes, in order to alter the presentation of the strokes. For example, a map behaviour will change single line pen strokes to double lines, in order to represent roads. A particularly intriguing feature of Flatland is the ability to move forwards and backwards in time, in order to explore the history of what has been drawn on the whiteboard. This solves the problem that physical whiteboards have, identified by Mynatt [156], with respect to storing sketches. The automatic recording of sketches, without the requirement to explicitly name files, is an elegant feature.

Notification Collage [72] is a more recent large display collaborative system. It was designed to support ad-hoc sharing of media by both co-located and remote collaborators. The Notification Collage system provides clients that allow users to upload videos, sticky notes, web pages, and other elements to the system from their own personal computers. Elements are added somewhat randomly to the surface, and cover up older elements as the surface becomes full. During deployment, it was found that the notification collage was useful at sparking collaboration and communication in unexpected ways. It was also found that many users preferred to have a private version of the collage on their personal display. It was thought that this was because the large display was not visible to many inhabitants of the

laboratory.

MessyBoard and MessyDesk [57] are two systems for displaying information on large and small displays, respectively. The design of the systems was inspired by literature examining knowledge workers. Of particular focus was the nature of knowledge workers as “pilers,” rather than “filers,” and the observation that “... the defining characteristics of knowledge workers is that they are themselves changed by the information they process.” Towards the goal of supporting knowledge workers in their natural piling behaviours, the system was designed to aid memory and allow users to build their own personalized contexts through decoration. The system was deployed in a number of groups. It was found that it was used in very different ways, depending on the nature of the projects that the groups were undertaking.

The above broad overview of some important large display systems reveals some interesting trends. First of all, collaboration (i.e. groupware) is one of the universal themes being explored. The collaborative nature of physical whiteboards has been adequately expressed in the design of the systems. Another theme is the support for casual or ad-hoc work. The systems are generally designed as very general tools that can support any sort of discussion or collaboration. Flexibility and generality is a strength, as identified by Huang et al. [93]. In terms of interactions, the systems tend to support either pen input for local interaction with the large display, or traditional computer interaction for input performed remotely. It is interesting that there has been little investigation into distance techniques that support interaction with the large display from outside of arms reach. This oversight is likely due to the display sizes (large, but not very large), and relatively small group sizes supported. It is possible that a focus on distance techniques would emerge if display sizes were on the order of multiple meters, or if group sizes were in the dozens.

### **2.3 Design Guidelines for Large Display Systems**

We have described research that explores the space of large display systems, but the research describing this work typically does not delve deeply into the aspects of evaluation or constructing design guidelines. Evaluation is important in order to

identify concerns specific to large interactive displays. Evidence of this is provided by Czerwinski et al. [44], who demonstrated that, although large displays provide significant performance benefits, there are design challenges specific to these form factors. It is therefore useful to consider other work that focusses more on the design challenges of large displays.

Huang et al. [93] performed a review of large display systems with the goal of determining key features and potential flaws of such systems. In examining patterns that emerged from their review, they were able to make a number of recommendations for designers of large display groupware systems in order to ease adoption by users. First, they suggest that a new system should provide clear and easily understood advantages over existing systems. Moreover, the system should be able to be integrated into existing workflows. Second, flexibility was found to be valuable, as we observed in section 2.2. While users often wished to initially integrate a new system into a specific workflow, flexibility allowed them to later use the system for broader uses. Third, visibility of others' interactions was found to be key. Viewing other people using a large display encouraged others to get involved in the process. Fourth, a low barrier to use was found to be important. A small amount of effort configuring the system, for example, encouraged use. Finally, it was argued that a core group of dedicated users made adoption by a broader audience much more likely. Otherwise, it was found that the system was likely to fall into disuse.

Rogers and Lindley [174] examined the relative strengths and weaknesses of large horizontal (table) and vertical (wall) displays. They note that wall displays support collaboration among dynamically changing groups, and are ideal for supporting presentation to an audience. They also conclude that table displays have some benefits over wall displays; in particular table displays allow group members to switch between roles more often, and explore more ideas. It should be concluded that wall and table displays are suited to different kinds of tasks, and it would be best for spaces to provide both kinds of displays to support the widest variety of tasks. As Buxton has argued, every interaction technique is best for one thing, and worst for something else. Part of the design challenge is determining the suitability of different display form factors for different tasks.

Rogers and Rodden [175] investigated the importance of spaces and display

configurations in order to support collaboration. They classify existing systems into three categories: embedded displays, stand-alone displays, and integrated multiple displays. Embedded displays are integrated members of a physical space, while stand-alone displays are inserted into spaces after the fact. Integrated multiple display environments, on the other hand, are environments with multiple displays that are used together. One important conclusion of Rogers and Rodden relates to the relatively slow adoption of large display systems, in comparison to novel small display systems. They note that the lack of adoption is not due to any inherent faults of large displays, but is instead due to limitations in current hardware. In anticipation of better hardware, we should strive to establish effective interaction approaches, so that when these systems are widely deployed users can immediately gain full benefit.

## **2.4 Interaction Techniques for Large Wall Displays**

Many individual interaction techniques have been developed that are particularly applicable to large displays. Most of these techniques have been developed independently of whole systems, but they can be potentially integrated into a wide variety of systems. We will discuss the space of such techniques here, considering touch techniques separately from distance techniques.

### **2.4.1 Direct Touch Techniques**

Many touch techniques have been developed that support interaction with large displays. Some of these techniques focus on solving the reaching problem, that of interacting across large display surfaces, while other techniques deal with other aspects of interaction.

Many techniques have been developed that allow users to either reach objects that are on the other end of a large display, or move an object that is close by to the other end. An early example is the throwing metaphor introduced by Geißler in the DynaWall project [68]. The metaphor was further explored by Hascoet [85]. They found that the two throwing models they developed outperformed traditional drag and drop, but the techniques may have been limited by the fact that the outcome of a throwing action is defined at the point of a release. There is no correction

possible as the object reaches the destination. This limitation was addressed in an exploration by Reetz et al. [171]. Their SuperFlick technique allows the user to adjust the trajectory of a thrown object as it travels.

Several techniques are based on the metaphor of bending space, making large distances shorter. The Vacuum technique [15] employs a directional virtual vacuum that can suck distant objects to be closer to the user. An advantage of the technique is that it can be used to access multiple objects simultaneously, as long as they are located physically close to each other. Other similar techniques include drag-and-pop, push-and-throw, and drag-and-throw [41]. This suite of techniques allows users to move objects to locations that are out of reach. The metaphor is one of throwing, but a live preview of the result of the action is shown, and the object moves instantly once the action is triggered, so there is no actual physics-like throwing action. These techniques are powerful, however, they are somewhat limited, because the system must have knowledge of interactable objects. The techniques do not easily support, for example, placing an icon in an empty region of the display.

Other techniques have been developed that do not fit into a convenient metaphorical category. Frisbee [114] is a widget-based technique that allows a user to see into a distant location on the display. The widget can be dragged in order to adjust either the viewer's location or the location being viewed. It was found that the widget was preferred over physical walking for travel distances of greater than 4.5 feet (about 1.4 meters). This approach is similar to WinCuts [189], with the addition that WinCuts allow regions to be shared over multiple devices. Not all techniques necessarily involve direct interaction. Spotlight [115] is a technique that aids in directing user attention to a certain region of the display. This is performed using highlighting, and is relevant to large displays because it is easy for events to occur unnoticed on a very large display.

Touch interaction raises the issue of occlusion. If a user is physically touching the display at a point of interest, the finger will necessarily occlude some of the content. This is a problem that is particularly evident for small displays, but also exists for large displays. An example of a technique designed to address this issue is "Shift" [206], which creates an inset window showing the touched region offset from where the finger is actually touching. Occlusion was explored in a

more general sense in Vogel and Balakrishnan's work [205]. They developed both a geometric model of where occlusion might impact interaction, and a suite of techniques that can be used to minimize potential problems.

### **2.4.2 Distance Techniques**

Distance techniques for large displays are techniques that support interaction with on-screen content for users outside of physical reach of the display. Such users are often standing, rather than seated at a desk, and make use of the techniques without the aid of a supporting surface, so many of these techniques can also be described as "mid-air" techniques.

Put-that-there [17] by Bolt is a very early example of a distance interaction technique for large wall displays. The put-that-there technique allows users to manipulate onscreen content by simultaneously pointing at a location and speaking instructions. The user can say commands such as "create" or "move" followed by specifiers such as "that" and "there." A full sentence such as "put that there" while indicating first a virtual object and then a location with a pointed finger results in the virtual object being moved. This combination of voice and physical pointing, in combination with the casual language supported, is a powerful approach. More recent work has explored similar combinations of voice and pointing. Speech-filtered bubble rays [198] explores voice disambiguation techniques combined with physical pointing. This extension of the Bolt work helps validate the approach using more modern and reliable sensing platforms.

Charade [10] is a system that built on previous work, and introduced more complex hand gestures. Charade allows users to trigger commands such as "next page," "mark page," and "hilite area" by performing pre-defined hand motions in a specified area. The strength of this approach is a relatively large vocabulary of commands that are more powerful than those available in, say, put-that-there. The downside of this approach is that the hand gestures can be hard to remember. The creators of Charade recognize the difficulty of learning and remembering arbitrary gestures, and argue that gestures should be self-revealing, however, the command set for Charade is clearly not self-revealing. The problem of learning gesture command sets has been further investigated by a number of researchers. Wobbrock et

al. [218] explored the approach of allowing typical users to define gesture sets with the hopes that the resulting sets would be easily discoverable, whereas Gesture-Bar [20] coaches new users in pre-defined gesture sets.

Laser pointers, and laser-pointer-like devices, are an obvious approach for interacting with large displays. These devices offer intuitive interaction without the requirement to learn the complex command sets associated with gestural input. Kirstein and Muller [117] offered an exploration of this interaction approach. Jiang et al. [103] explored a different implementation, where the sensing camera is integrated into the pointing device. Sceptre [211] is a third implementation that makes use of a pattern of infrared dots, allowing the system to recognize orientation of the pointing device. Laser pointing has a direct analog to mouse pointing, however, it possesses a significant drawback, namely the difficulty in pointing accurately at targets, especially as user distance to the display increases. Myers et al. [154] compared the performance of laser pointers to other devices, and found that directly tapping objects was faster than laser pointer input. Their work is limited, however, by the fact that their display was relatively small. As a display grows much larger, direct touch input becomes less practical.

Baudisch et al. [11] explored the design of alternative pointing devices for use in mid-air pointing. Their device, “Soap,” is a hard object wrapped in a flexible wrapping, that the user is able to roll in their hand like a bar of soap. The device is described as a mouse alternative, offering the precision of a mouse without the requirement of a hard surface on which to use it. One advantage of the device is that operation does not change depending on user distance to the display. Precision does not decrease as user distance increases, as with laser pointers. A potential downside is the tradeoff in precision and speed that occurs when gain is adjusted, potentially causing difficulties in traversing large distances on the display or hitting small targets.

It is also possible to interact with large displays through manipulation of small personal handheld displays. The Pebbles project by Myers et al. [155] is an early example of this approach. They used multiple PDAs (personal digital assistants) networked to a single PC in order to support collaboration. A more recent investigation into this approach by Finke et al. [59] investigated the relative strengths and weaknesses of placing widgets on a large display, or splitting widgets across a



large shared display and small personal display. They found no significant difference in performance between the two conditions. There likely remains a substantial amount of work to understand the differences between the approaches. Touch Projector is a more recent variation on this approach, which attempts to cross the boundary between distance interaction and direct touch [18].

As ambient sensing becomes more powerful, the need for dedicated input devices such as mice and laser pointers decreases. Vogel and Balakrishnan [204] explored freehand pointing and clicking, meaning techniques for indicating click events unsupported by input devices. Their techniques involve moving fingers relative to one another, the movements of which were captured by a Vicon tracking system. While their approach relies on somewhat invasive optical markers, Stødle et al. [187] have developed an alternate approach that relies on cameras and microphones. Their technique instead relies on finger snapping to trigger events. Eventually, it is expected that sensing hardware and algorithms will be developed that allow for even more subtle triggering of events. Recent work investigating the Microsoft Kinect hardware [215] indicates that this may be forthcoming sooner than expected.

## 2.5 Reality-Based and Whole Body Interaction

Reality-Based Interaction (RBI) is a unifying concept in interaction design that serves to describe a whole class of emerging interaction techniques. As defined by Jacob et al. [101], RBI is an example of a post-WIMP [203] interaction style that draws from numerous themes related to interaction in the real world. The themes include *naïve physics*, *body awareness & skills*, *environment awareness & skills*, and *social awareness & skills*.

Naïve physics relates to our ability to intuitively navigate and function in the real world, without having to necessarily analyze or fully understand details. For example, it is possible to catch a thrown ball without having detailed knowledge of the effects of gravity and wind turbulence and the curve the ball traces through the air.

Body awareness & skills relates to our sense of physical self in space. It is a relatively simple task to touch your own nose, even with eyes closed. Environment

awareness & skills relates to our understanding of environmental cues. The horizon can serve as an indicator of orientation (so powerful that aircraft contain an artificial horizon [42]), and there are a number of different cues that indicate the distance of objects in the environment [123].

Social awareness & skills relates to how we interact with other people, including both verbal and non-verbal cues. Social behaviour is so important that it has been indicated as being a primary driving force in evolution [216].

Jacob et al. claim that while Reality-Based Interaction has not been previously codified, there are many examples of systems and research that fall into the category of RBI. It is pointed out that tangible interaction [200] draws on naïve physics, ubiquitous computing [2] draws on both social awareness & skills and environment awareness & skills, and virtual reality systems [100] draw on body awareness & skills and environment awareness & skills.

Whole body interaction is another theme in interaction that is closely related to Reality-Based Interaction. As defined by Klemmer et al. [118] in their investigation of “how bodies matter,” there are five themes related to bodies that are particularly relevant to the design of interactive computing systems. These themes include *thinking through doing*, *performance*, *visibility*, *risk*, and *thickness of practice*. In another investigation of whole body interaction, England et al. [53] note the overlap of the topic with other areas of specialization, but differentiate based on the observation that whole body interaction gives us a more integrated view. They state: “...ubiquitous computing is more concerned with the notion of *Place* rather than capturing the full range of actions. Physical computing is more concerned with artifacts than the physical nature of humans.”

The lessons learned from RBIs and “how bodies matter” are consistent. They both suggest that we should leverage real world experience and capabilities when designing virtual interactions, while simultaneously leveraging the “more-than-real” power that is possible when we are not constrained by physicality. As Jacob et al. point out, computing systems should not perfectly mimic the real world. They must offer something beyond what the physical world offers, otherwise no benefit is derived.

The themes explored in RBIs and whole-body interfaces are consistent with the transition away from the WIMP model of interaction, which has been a topic

of interest for several decades. As Nielsen noted [158] in his discussion on the future of noncommand interfaces, next-generation interfaces will likely differ largely from traditional interfaces in that they will stress direct interaction rather than explicit commands, and that possibly a variety of interaction models will emerge to support different tasks. This is consistent with what has been noted about RBIs and whole body interaction, and these may in fact be more recent examples of the next-generation advances Nielsen was predicting.

It is useful to examine some of the specifics of the next-generation systems that Nielsen was predicting. Table 2.2 contrasts Nielsen's predicted properties of next-generation interactions to then-current (as of 1993, yet still relevant) interactions. Many of these properties are familiar to the fields of RBIs and whole body interactions, and will be critical elements of the techniques we explore in later chapters. For example, Nielsen predicted a lack of interaction syntax, the potential for hidden objects, a lack of turn taking, programming-by-demonstration, and an interface locus embedded in the user's environment. These are all aspects of body-centric interaction as we will explore it in this dissertation.

## **2.6 Summary**

Our survey of related work shows, first, that there is significant potential for the development and deployment of interactive computing systems based on very large wall displays. Systems based on these form factors can support a host of tasks that are ill-suited to traditional computing form factors. Furthermore, while a large set of interaction techniques have been developed for use with large displays, a coherent model of interaction has yet to emerge. There are several design approaches, however, that hold promise to guide the development of a coherent interaction model. Reality-based interfaces and whole body interfaces, as examples of post-WIMP interfaces, make use of the natural capabilities of users as human beings, and promise to leverage both the inherent physical properties of large surfaces, and the more-than-physical interactive properties of computing systems.

Our approach to developing a body-centric model of interaction with large wall displays will build from what has been learned in this chapter. Chapters 3 and 4 will build from the largely abstract foundations of RBIs and whole body interaction,

	<b>Current interface generation</b>	<b>Next-generation interface</b>
User focus	Controlling computer	Controlling task domain
Computer's role	Obedying orders literally	Interpreting user actions and doing what it deems appropriate
Interface control	By user (i.e. interface is explicitly made visible)	By computer (since user does not worry about the interface as such)
Syntax	Object-action composites	None (no composites since single user token constitutes an interaction unit)
Object visibility	Essential for the use of direct manipulation	Some objects may be implicit and hidden
Interaction stream	Single device at a time	Parallel streams from multiple devices
Bandwidth	Low (keyboard) to fairly low (mouse)	High to very high (virtual realities)
Tracking feedback	Possible on lexical levels	Needs deep knowledge of object semantics
Turn-taking	Yes; user and computer wait for each other	No; user and computer both keep going
Interface locus	Workstation screen, mouse, and keyboard	Embedded in user's environment, including entire room and building
User programming	Imperative and poorly structured macro languages	Programming-by-demonstration and nonimperative, graphical languages
Software packaging	Monolithic applications	Plug-and-play modules

**Table 2.2:** Comparison of systems (as of 1993) and potential next generation systems. Adapted from Nielsen [158].

and will provide a set of design guidelines rooted in the psychology literature. Chapters 5 and 7 will describe controlled experiments that shed light on the low-level performance characteristics of systems built using our particular post-WIMP approach. Chapter 6 will delve into the practical concerns of developing systems in a research environment where there are many competing approaches making use of heterogenous sensing systems in different ways.

## Chapter 3

# Shadow Reaching

In this chapter we discuss Shadow Reaching, an interaction technique that employs a shadow metaphor to support human interaction with large interactive wall display systems. Shadow Reaching is our first example of an interaction technique that builds from the body-centered design approach. It draws on a body representation to overcome some of the interaction difficulties that are particular to large display environments. The strength of making use of a body representation is that the technique remains grounded in reality, in a form that is immediately understandable to the user. We will see in Chapter 4 that not only does this technique deal with many of the issues identified earlier in this dissertation, but it also turns out to be easily extendable to make other interaction techniques possible. Thus, Shadow Reaching serves both as a concrete example of an interaction technique, and as motivation for the research reported in the chapters that follow. It illustrates some of the elements of a body-centered design approach. Later chapters will add to this.

Elements of this chapter have previously been published [183]. This was the preliminary work that led us to the idea of body-centric interaction.

### 3.1 Related Work

We summarize the related literature specific to the use of interactive shadows deployed in public spaces, in art projects, and in research systems.

Human body shadows provide powerful cues that we naturally perceive and



**Figure 3.1:** Users interacting with a large wall display system employing a shadow input metaphor. The installation was located in a science center (developer of system is unknown), and users explored the system without any prompting or instruction. Photos taken by the author.

understand. This is evident in Figure 3.1, which shows guests at a science center reception exploring a large wall display installation employing a shadow input metaphor. What is important about this example is that the installation showed evidence of being inviting, explorable, and intuitive. It is apparently inviting because people chose to spend time with it instead of with many other available distractions (including free beer). It appears explorable because users were seen trying different motions and eliciting different responses from the system. These were motions that were not explained to the users as being possible commands to the system, yet the users presumably guessed that they might have meaning. It is also evidently intuitive because the users were observed to be successfully interacting with the system. What is intriguing about this example is that all of this was possible with a system that likely few had ever seen before, and which didn't have any instructions or a manual.

The power of shadows, as made evident in the above science center experi-

ence, has been explored in many ways, both by artists and researchers. Krueger et al. developed a very early example of a system employing user shadows to enable interaction [122]. Their system, VIDEOPLACE, used a live video image of the user on the display as a contour that could directly impact the elements shown on the screen. This early work was followed by many other examples that expanded on the theme. Recently, Camille Utterback has been at the forefront of explorations into user embodiments in virtual space, with a stated goal of “bridging the conceptual and the corporeal.” Text Rain, an interactive installation by Utterback and Achituv [201], demonstrated the use of body shadows to interact with text elements that dynamically change on the display. A more recent example is “Untitled 6,” which generates an interactive abstract painting based on human motion. Lozano-Hemmer is another artist whose works are in a similar vein. His “Shadow Box” and “UnderScan” installations share the goal of breaking down the barrier between one’s personal space and the shared space upon which the shadow is cast. This is a form of expressive embodiment, as described by Gutwin and Greenberg [76].

There are also many research projects that have made use of shadows. The “Shadow Communication” system by Miwa and Ishibiki used shadows to facilitate remote collaboration [149]. Their design was based on the following observed properties of shadows:

1. A person and their shadow are absolutely inseparable.
2. The existence of a person can be evoked by their shadow.
3. A shadow has the function of expanding a body image.
4. A shadow changes its movement or location proportionally in response to the bodily movement or position of the shadow’s owner.

These observations are very insightful, and mesh well with our theme of body-centric interaction. In particular, they speak to Klemmer’s themes of *visibility* and *thinking through doing*. Despite relatively little hard evidence to back up these claims, they seem intuitively correct, and they serve to inspire the development of shadow-metaphor interfaces. Later in this dissertation we will delve into other



literature that will further substantiate the claims made by Miwa and Ishibiki, and extend them.

VideoWhiteboard [195] is another example of a similar system. It allows remote collaborators to work together through the use of shadow representations. VideoWhiteboard is a collaborative drawing application that was based on observations of natural drawing collaborations. VideoArms [194] is a system that supports mixed-presence collaboration. It emphasizes the display of local and remote users' arms differently, in order to help overcome the known problem of users focussing their attention primarily on other local users [191].

Other systems have explored the use of shadows in yet other ways. Remote Impact [151] is an exertion interface that allows spatially distributed users to compete in a boxing match. Remote users' shadows are projected onto a large soft surface that the local user can punch in order to score points. This highlights the use of shadows as a natural and immediately understandable representation of a remote user. Taking a different approach, Meisner et al. [145] explored the use of shadow interaction and a hand-puppet metaphor to ease human-robot interaction.

What most of these approaches share is an assumption that shadows provide a representation of the user that is powerful in a way that a more abstract representation, such as a cursor or avatar, cannot be. There is something about a body shadow that designers know holds meaning, and that helps connect users both to the virtual space they are exploring and to other users of the system. For this and other reasons, human body shadows are worth exploring as part of our body-centric design approach.

## **3.2 Body-Centric Use Case Scenarios**

Before exploring the details of Shadow Reaching, we consider some use case scenarios that we consider appropriate for body-centric techniques applied to large wall displays. The discussion of scenarios is useful not only for supporting our exploration of Shadow Reaching, but also as a point of reference for our later exploration of interaction techniques in Chapter 4, and our discussion of text input in Chapter 5.

We consider four scenarios for large display use: university classroom lectures,

architecture/construction planning, web design collaboration, and public space display installations. In analyzing these scenarios we arrive at an initial taxonomy of use cases for large wall displays, which we summarize in Table 3.1.

### **3.2.1 Scenario 1: University lecture halls**

University classrooms are ideal candidates for the integration of large wall displays. Such displays add interactive capabilities to the capabilities already provided by traditional whiteboards and blackboards and they bring access to digital information into the classroom.

In examining how lecturers use large surfaces to teach, Lanir et al. [125] discovered that the ability to gesture to content, and the ability to perform in-depth exploration of content, are both important elements in classroom presentation. The physical nature of a lecturer's interaction with the information being presented suggests that a body-centric approach for interaction might be suitable. There is also the fact that at times some or many of the students in the classroom may be simultaneously interacting with the display [150], which suggests the need for approaches that support collaboration.

### **3.2.2 Scenario 2: Construction planning meetings**

The design and planning of buildings is an inherently collaborative activity that can involve physical as well as virtual artifacts shown on either small or large displays. It is also an activity that demands the exploration of very complex alternatives, and involves the use of large and detailed visualizations. Team meetings involving a variety of stakeholders (architects, construction managers, contractors, clients, and possibly others) are a primary means for coordinating the many activities that must be undertaken. The exchange of information and much of the decision making is driven by discussions that focus on visual artifacts. More and more, these artifacts exist in digital form. But some of the artifacts are physical, such as scale models of the site or the proposed building(s) and blueprints or other hardcopy representations of the building.

Tory et al. [197] discovered that physical gestures, navigation, annotation, and viewing were four critical tasks in collaborative building design and construction

management. In their analysis of interaction bottlenecks, they concluded that interaction techniques should provide more tangible and direct means of interaction, aid in navigation (i.e. panning and zooming), and aid in exchanging data through physical tokens. They also concluded that virtual pointing should better embody the user, by capturing and expressing different kinds of pointing such as *five-fingered pointing*, indication of areas, as well as *anchored* pointing using two hands, and other subtle forms of physical pointing that are not captured by a simple cursor. This, plus their emphasis of using large vertical displays to provide shared visualizations to stakeholders suggests that our body-centric approach coupled with large wall displays is appropriate.

### **3.2.3 Scenario 3: Collaborative web design “war rooms”**

Web design, especially with the maturing of the web development world, is a largely collaborative effort, involving developers, artists, product managers, and others. It is a form of software development, but one that has a richly visual component and for which the temporal sequence of actions and visual presentations is important. Often a room or other area will be dedicated to a persistent display of design artifacts, ranging from rough sketches to finished web pages, as well as representations of current or planned workflow.

Klemmer et al. investigated the processes that web developers follow to ply their craft [119]. They concluded that tools to support web design should emphasize aspects of fluidity and physicality, should be both tangible and virtual, and should involve large vertical displays. From these conclusions a body-centric interaction approach coupled with large wall displays is a clear candidate for adoption in these “war rooms”.

### **3.2.4 Scenario 4: Public events and installations**

The widespread availability of large display screens in public places coupled with wireless networking and seemingly ubiquitous hand-held devices such as smart phones offers tantalizing opportunities for public events in which the boundaries between audience and performer begins to blur, and public art installations where the distinction between artist and viewer is similarly vague. Mechanisms for ad hoc

Context	Display Size	# Users	# Simult. Input	User Dist. from Display
Classroom	3–10m	10–200	1–many	0–20m
Construction	2–5m	5–20	1–3	0–4m
Web Design	2–5m	2–10	1–10	0–4m
Public Spaces	2–20m	1–1000	1–many	0–50m

**Table 3.1:** Typical values for properties of each use case. It can be seen that these use cases are strongly associated with collaboration with large displays, and involve interaction by users at a distance from the display.

participation and collaborative activity may require new approaches to interaction.

Müller et al. [152] developed a taxonomy of interaction with publicly situated displays. They identified a number of elements of such displays that indicate they might benefit from a body-centric interaction approach. First, they liken public displays to stages. They argue that in interacting with a public display, a user is putting forward a “presentation of self” to viewers of the display. We argue that this would suggest an interaction style emphasizing the whole self of the user (i.e. body-centric representation), rather than an abstract iconization or cursor. They also suggest a number of possible interaction modalities, including body position, body posture, and facial expression, which clearly suggests a body-centric interaction approach.

### 3.2.5 Design requirements derived from the four use case scenarios

In considering the four use case scenarios described we arrived at a taxonomy of use. We can extend this to include a summary of requirements for these use cases, shown in Table 3.2. Classroom lectures don’t require touch input, as the board is frequently out of reach of all users, but requires distance input and text input, as most lectures involve text of some sort. Construction planning is generally a smaller scale activity as compared to classroom lectures. The display is frequently more intimately located, and therefore touch input is an additional requirement. Construction planning also requires that interaction techniques incorporate physical artifacts, such as models. Web Design is highly collaborative, and similar in

Context	Touch Input	Distance Input	Text Input	Integration w/ Artifacts
Classroom	no	yes	yes	no
Construction	yes	yes	yes	yes
Web Design	yes	yes	yes	no
Public Spaces	?	yes	?	?

**Table 3.2:** Some design requirements for large display systems supporting several use cases. Distance input is a universal requirement, and other input types, including text input and touch input, are also expected to present.

some ways to construction planning, but involves physical artifacts to a lesser degree. The class of public space installations encompasses many possible systems, and at the extreme can involve interaction by many thousands of people, as in the case where displays are on the sides of buildings [167], or in sports stadiums. In general, these requirements show that a variety of uses cases demand a variety of input modalities, especially distance input. Note that the conclusions are for the general case of each use case. It is certainly possible to construct special cases that fall outside of these parameters.

### 3.3 The Design of Shadow Reaching

In developing our interaction technique we looked at two major problems that are specific to very large wall displays.

First, the larger a display is, the less likely it is that a user will be within physical reach of any particular location on the display. For a small display, such as a desktop display, a user can always have the entire display within reach. For a very large display of, for example,  $5\text{m} \times 3\text{m}$  in size, it is impossible for the user to have even half of the display within reach. For targets out of reach it might be possible to move to the desired location of interaction, but this can be inconvenient and may become tiring over time. It may even be impossible if, for example, the display is taller than the user can reach. Thus, as displays grow larger, interaction techniques based on direct touch become less practical, and it becomes more desirable

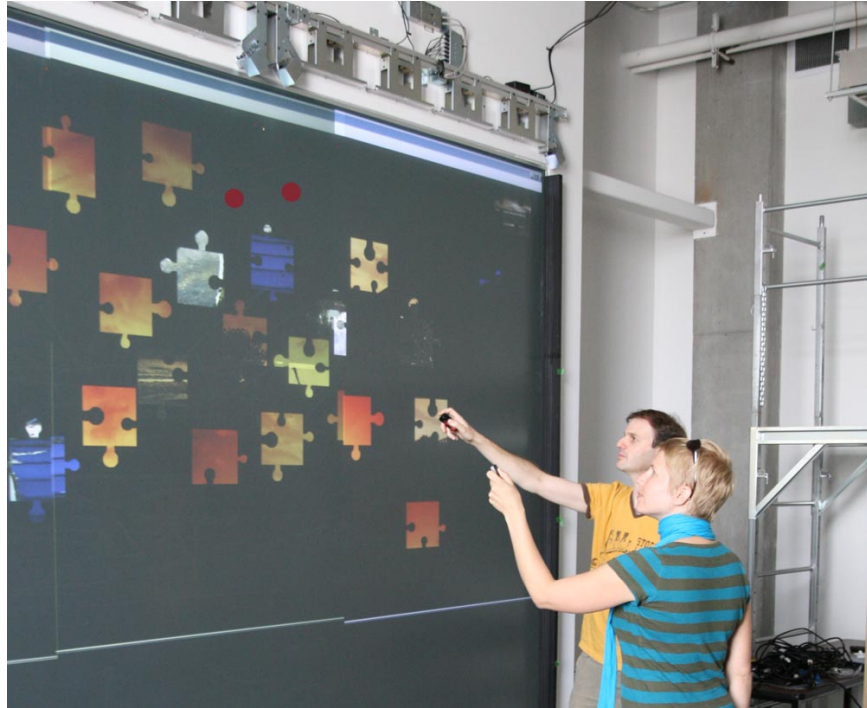
to develop interaction techniques that can be used regardless of where the user is standing relative to the desired point of interaction.

The second problem is one of awareness. It can be difficult, especially in collaborative large-display environments, to figure out who is doing what in the virtual space. For a single user, the user always knows who is causing the action. For two users, each user knows that any action not caused by himself has been caused by the other user. With three or more users, however, ambiguity can be a problem. As an example, Figure 3.2 shows a mock-up of two users interacting with a large display using laser pointers. The points of interaction are shown on the display as red circles. It is very difficult, as a third user, to figure out which user is controlling which cursor. Gutwin and Greenberg argue that collaboration can be clumsy and inefficient when user's have insufficient awareness of *who* is doing *what*, *where* [76]. If a user is unable to comprehend the state of interactive elements and predict the future state of those elements, as in Figure 3.2, collaboration and the accomplishment of the task objectives is compromised. As a simple example, if a user observes a disembodied cursor dragging a file from a file folder into the workspace, the observing user will not necessarily know who the owner of the file is. It is this kind shortcoming of awareness that Shadow Reaching is meant to address. The association of a specific shadow with a specific user will in turn associate the dragged file with the user, making the operation more meaningful.

We thus set two design goals in developing the system. First, the system must allow users to interact with a display at a variety of distances. We wanted to allow users to be either within reach of the display, or at a significant distance from the display. It was also important that users be able to transition fluidly and seamlessly between different distances. Second, the technique was meant to support awareness in multi-user scenarios. We wanted to provide interpretable actions that would be useful regardless of where the initiating user was physically located, relative to the action caused or relative to other users who might be observing the actions.

### **3.3.1 Supporting Distance Interaction**

Perhaps the simplest form of a shadow is one generated by a point light source that projects light in all directions. When the light strikes an object it illuminates that



**Figure 3.2:** A mock-up of two users interacting on a large display using laser pointer style input. A third party (collaborating directly or simply observing) may want to know which user is performing which operation, in order to either understand the logic behind the sequence of actions, or in order to integrate their own actions in a reasonable manner. Without richer feedback, unfortunately, it is difficult to determine which of the two users is controlling which of the two cursor.

object, but does not continue onward. An object or surface behind the illuminated object is not illuminated by the point light source, but it may still be lit due to other sources of light, such as global ambient lighting.

The shadow of a person cast on a flat wall possesses interesting properties. As observed in Figure 3.3, the size of the shadow changes as the user changes her position relative to both the wall and the light source. As she approaches the wall, her shadow shrinks until it becomes the same size as her body. As she approaches the light source her shadow grows until it covers the entire wall. This property of point-light-source shadows has important implications if the shadow is to be used

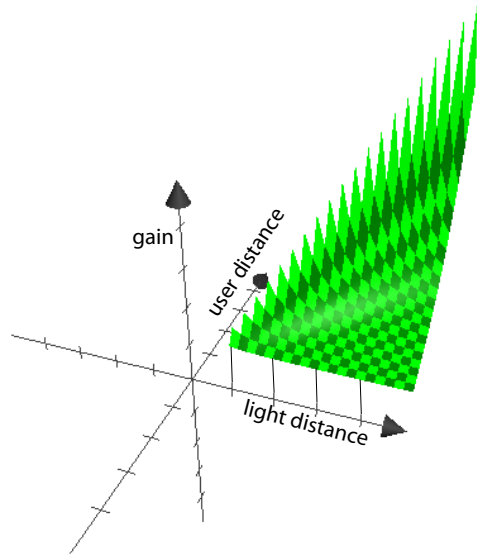


**Figure 3.3:** The user can control the reach of her shadow by moving closer to and farther away from the display.

for interactive purposes. If the control of on-screen elements is directly mapped to shadow-projected body parts, the control-display gain is governed by the position of the user in the room. By moving her body, the user can directly control CD gain (Figure 3.4), because gain varies directly with the distance of the user ( $U$ ) from the display as well as the distance of the light ( $L$ ) from the display ( $gain = \frac{L}{L-U}$ ). Gain is an important component influencing the speed-accuracy tradeoff in pointing-based interaction. Another strength of this approach is that, as the user moves further away from the display to get the “big picture,” the CD gain increases, and her motions are magnified, which is appropriate for “big picture” manipulation. In contrast, as she approaches the display to focus on a small region, her manipulations becomes appropriately more specific and localized.

A further benefit of point-light-source shadows relates to how distance interaction can transition fluidly to direct touch interaction. While we argue that distance interaction is a critical feature, direct touch interaction is also very powerful [116]. With the natural change in shadow geometry, shadow reaching will transition automatically to touch interaction as the user moves her hand and touches the surface of





**Figure 3.4:** The size of a user’s shadow, and related control-display gain, depends directly on the distance of the user from the display, and the distance of the light from the display. The user can smoothly adjust gain by moving in the room.

the display. Her hand and the hand’s shadow converge to a single location, which becomes the location of interaction. This is an approach similar to that described by Parker et al. [160], with the added benefit of a shadow embodiment.

### 3.3.2 Supporting Interpretable Actions

Our second design goal was to provide interpretable interactions, regardless of where a user is located relative to the display. Our hypothesis in designing Shadow Reaching was that a shadow can serve as a powerful embodiment that will provide consequential communication of activity in the form of continuous feedback, so others can interpret and understand those interactions [76]. Our choice of a shadow was made based on many of the same observations made by other researchers such as Tang et al. [194], Mueller [151], and particularly the observations of Miwa et al. [149], that were previously discussed. We felt that, based on our evolving design framework for body-centered interaction, the choice of a body shadow as an

awareness representation would pay further dividends not immediately obvious in an initial exploration. This decision was validated in later work, which is described in Chapter 4.

## **3.4 Implementations**

We implemented three different prototypes in our exploration of Shadow Reaching. The purpose of developing several prototypes was to test different methods for generating the shadow, and to then explore different means of interacting using the shadow. The prototypes employed a mix of real and virtual shadows, and interaction approaches ranging from single point clicking to whole body interaction. The prototypes were developed in parallel and represent an exploration of different options, rather than iterative improvement.

### **3.4.1 Prototype 1: Real Shadows and Virtual Cursors**

The first Shadow Reaching prototype (Figure 3.5) used real-world shadows generated by a bright lamp, and used magnetic tracking (Polhemus Liberty Latus) to track the user's hands in 3D. We used a real world lamp both for convenience, and to evaluate the quality of a real shadow (Figure 3.6). To support interaction with onscreen content the position of the hands, as sensed by the Polhemus, was geometrically projected onto the display and then graphically rendered as cursors so that the cursor locations corresponded with the physically cast shadows of the hands. The user held a button in each hand which was wired to the computer using a Phidgets interface board [71], and was able to send click events to the software.

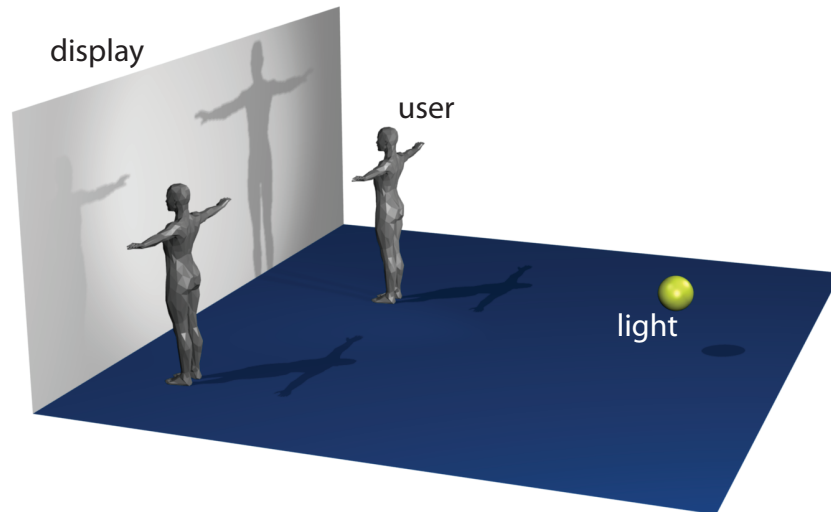
We developed a puzzle-building application to allow exploration of the technique in a realistic task involving manipulation of virtual objects. It was determined through informal evaluations that the mapping of virtual cursors to physical shadows was perceived as being natural by users. Users were able to reach to puzzle pieces scattered at all locations on the display, and had fine enough control to lock the pieces together. They took naturally to the bi-manual interaction made possible by the system.

There were, however, significant disadvantages to the approach. It was found that although the shadow was easily visible directly in front of the light and pos-



**Figure 3.5:** Shadow Reaching prototype 1. A physical shadow is cast with a bright lamp. The user's hand locations are measured using trackers. Cursors are rendered on the display near the shadow.

sessed aesthetically pleasing soft edges, and the shadow appeared useful for conveying awareness information in this region, the light was unable to cover the entirety of the  $5\text{m} \times 3\text{m}$  display. Because of this limitation, as a user moved towards the side of the display the shadow became indistinct. The light also proved to be very hot. It was uncomfortable to stand for any great length of time in front of it. A final drawback was that the light was fixed in space. There was no possibility of having a dynamically changing light source location, which limited interaction possibilities. A user standing directly in front of the light could only interact effectively in the region at the center of the display. As the user moved to one side, the shadow moved further and further to that side, limiting the scope of interaction.



**Figure 3.6:** Placing a real lamp in the room creates physical shadows on the display. Users near the middle of the display generate clear shadows, but users near the edge of the display generate indistinct shadows due to the intensity falloff of the directional lamp.

We concluded that although a physical light source is convenient for prototyping purposes, it is not a practical means of generating a shadow to support shadow-metaphor interactions on large displays.

### 3.4.2 Prototype 2: Virtual Shadows and Physical Interaction

The second Shadow Reaching prototype (Figure 3.7) explored generation of a virtual rather than a physical shadow. By “virtual,” we mean a shadow whose extent is somehow computed by the system and then rendered using the computer’s rendering capabilities. There are several possible benefits to this approach. First, a rendered shadow can be drawn in any form that is possible with the available rendering engine. The shadow could simulate a real-world transparent black shadow, or it could appear in some other form. Furthermore, we are not restricted by the physical apparatus of a lamp. The user will not be annoyed by the brightness or heat of a lamp, and the virtual light source, depending on the approach used, can

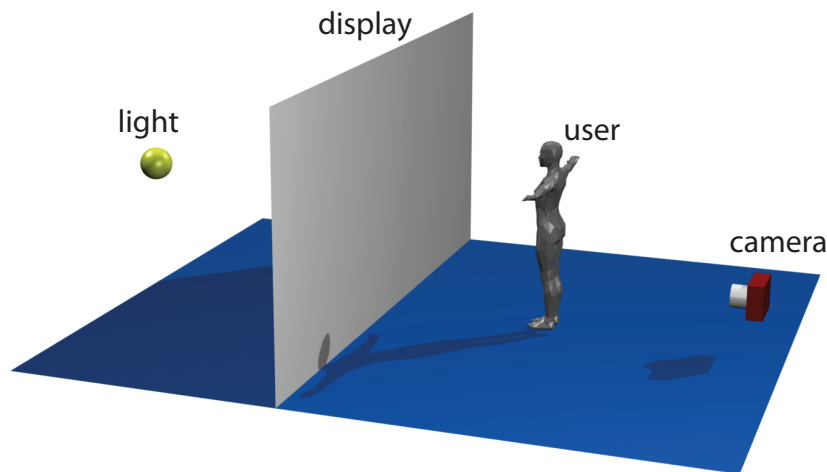


**Figure 3.7:** Shadow Reaching prototype 2. A user bounces balls using his virtual shadow.

potentially be placed anywhere in the room. It would even be possible to dynamically adjust the location of the light source based on input from the system or the user.

For the second prototype a virtual shadow was generated by placing an infrared light source behind the screen, captured with an infrared camera in front of the screen and behind the user (Figure 3.8) so that the camera's view of the light was blocked by the user. The camera was able to see the silhouette of the user against the screen and extract a model of the user's location using rudimentary computer vision techniques, in a manner similar to that of Tan and Pausch [190]. This approach generates a virtual shadow with some of our desirable properties, but not all. We can render the shadow however we want, and there is no physical lighting apparatus visible to the user, but the shadow can only be generated from one perspective, as defined by the locations of the wall, the user, and the camera.

In this prototype, the user's embodied shadow interacted with virtual balls that were programmed to bounce around the large display. The software constrains the balls to bounce off the shadow, but to otherwise follow physical laws. While the application was designed without any intended useful purpose, we found that



**Figure 3.8:** Placing a camera and infrared light source in the locations shown allows the camera to see users as a backlit shape. Vision algorithms can then extract the contour of the user.

users spontaneously developed their own tasks based on the possibilities presented by the system, consistent with both Klemmer et al.'s theme of “thinking through doing” [118] and the behaviours we observed in situations such as the one depicted in Figure 3.1. One user decided to trap balls in outstretched and joined arms, while another attempted to keep balls from hitting the ground. From this and other observations we conclude that whole body interaction, in this case enabled by body shadows, presents a host of affordances that might be exploited.

### 3.4.3 Prototype 3: Magic Lens Shadows

In our third prototype (Figure 3.9) we explored different shadow representations. We drew inspiration from Bier et al.'s [16] concept of Magic Lenses. Magic Lenses are movable see-through widgets that are used to visually filter on-screen data. They can perform arbitrary transformations on the data, including altering representation or presentation of secondary information. Magic Shadows, our third prototype, extend the concept of Magic Lenses. Magic Shadows are Magic Lenses whose



**Figure 3.9:** A mockup of users interacting using Magic Shadows. Their shadows display satellite photos, while the context display maps.

boundaries are defined by the user's shadow. They provide a natural means of defining personal views of the data, and of moving a lens about the workspace. As in the second prototype, we used a vision-based method for generating virtual shadows. The basic vision algorithm finds the contour of a body, and then the system renders one data set inside the bounds of the contour, while a second data set is rendered elsewhere. In our implementation the data was geographic information downloaded from Google Maps. Regular map data was rendered outside the contour, but the Magic Shadows showed satellite data inside the contour.

The power of Magic Shadows lies in the approach of combining the manipulation of data with body-centric interaction. In order to alter the view of data a user does not need to manipulate an input device or interact with onscreen widgets. The user's own body is the mechanism for manipulating the data. Magic Shadows is an interaction approach that is immediately understandable, and can likely be learned by anybody of almost any background. Furthermore, any third-party viewers should be able to interpret the interaction that is taking place. They will

Prototype	Shadow Generation	Shadow Projection	Interaction	Role of Shadow
Real Shadows	Real	True Perspective	Clicks	Awareness
Virtual Shadows	Virtual	Approx. Orthographic	Whole Body	Awareness + Input
Magic Lens	Image Processed	Approx. Orthographic	Implicit	Awareness + Visualization

**Table 3.3:** The Shadow Reaching design space as sampled by the three prototype systems.

easily be able to understand the mapping of user to shadow, and also understand not only why some data is being filtered, but how it came about to be filtered and who is filtering it. These properties are consistent with our body-centric model of interaction and they illustrate some of the power of the model.

### 3.5 Conclusions

We have described a technique, Shadow Reaching, that falls within what we have defined as our body-centric design framework. Shadow Reaching addresses significant hurdles that are specific to very large wall displays. It does so in a way that leverages what we intuitively understand is a powerful aspect of the human body in the physical world, namely body shadows. We explored three preliminary implementations of this interaction technique. The design space as explored is visualized in Table 3.3. The shadow served a variety of purposes, and interaction was supported in a number of different ways. We conclude that Shadow Reaching is promising as a basis for designing other interaction techniques.

#### 3.5.1 A Note About Shadow Geometries

The diagram in Figure 3.8 shows one particular geometry for the hardware to implement a system based on the shadow metaphor. In this configuration the light source is behind the display screen and the camera is behind the user. The positions of the light and the camera could be interchanged. Multiple lights or multiple cameras could be used and the software adapted accordingly. Visible light or some



other part of the spectrum, such as infrared (as we used), could be used if suitable cameras were employed.

The geometry for real shadows that was used in the first prototype is not as flexible. The physical nature of light and shadows requires that the light be placed behind the user and that it shine onto the display. With virtual shadows these requirements can be relaxed.

As we will see in later chapters, there are even more flexible ways to design systems that employ a shadow metaphor. There are many alternatives to casting light that can be used to generate realistic shadow images, including computed shadows that do not rely at all on light, and these need not be constrained to follow all of the rules of physical shadows.

### **3.5.2 Limitations of Our Initial Prototypes**

We learned through our prototypes that we were quite limited in what we could implement due to shortcomings of the hardware available to us. The buttons used to trigger events in Prototype 1 were awkward. The physical cables often became tangled, and they were sometimes too short. There was also a question of whether or not the concept of “click” events should even be part of a body-centric model of interaction. There are no “clicks” in the real world. Instead we reach, grasp, or move. These physical actions are much richer than a simple click at a single location. However, these physical actions are also much more difficult to model computationally. Existing software generally does not support rich physically based gestures, although there are exceptions [27], and with the recent introduction of products such as the iPhone and iPad, the landscape of physically based interfaces is evolving rapidly.

More importantly, the approaches we used to generate shadows were somewhat rudimentary. We have discussed the limitations of physical shadows, but the limitations of virtual shadows are more interesting because we have already seen that virtual shadows are the only practical means of implementing our approach. The limitation of our approach for generating virtual shadows is primarily due to the fact that the system develops a very sparse knowledge of the scene. The system can generate a virtual shadow, but only from one perspective. Furthermore, once a

shadow is generated the system has no knowledge of shadow specifics. It doesn't know which part is the hand, or foot, or head. That means that the system cannot generate a shadow for an arbitrary virtual light location, nor can it generate click events or other events registered to a particular body location. In short, our virtual shadow implementations are somewhat primitive and not robust enough to support the full set of interactions that we envision.

Lastly, we have not performed a formal evaluation of the system. We defer evaluation of our approach to Chapter 7, where we will describe an analysis of low-level pointing performance. The evaluation will not incorporate actual shadows, but will employ a perspective projection identical to what has been described.

From our analysis of virtual shadow generation we concluded that we require an implementation that has knowledge of the scene. It must know the geometry of the display, the users, and possibly other contextual objects in the scene. With a model of all relevant contextual objects the system will be able to generate an accurate virtual shadow from any perspective. Furthermore, it will be able to know the location of the user's hand (or any other body part) in real space and project it onto the display. This could lead to the development of a much richer set of interaction techniques. We will see this when we discuss further design considerations in Chapter 4 and Chapter 6.

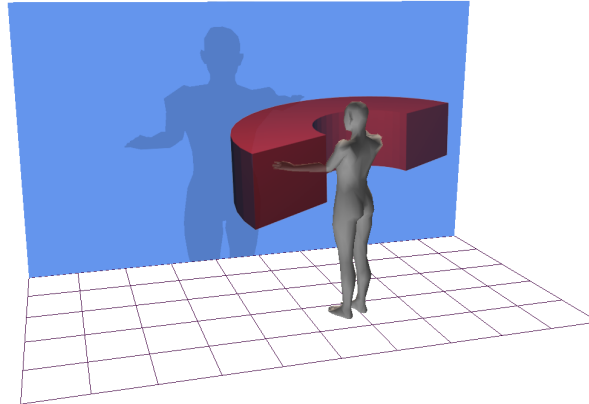
## Chapter 4

# Body-Centered Interaction

In the previous chapter we explored the use of a virtual shadow embodiment to support interaction and collaborative awareness in a very large wall display application context. In this chapter we describe a suite of interaction techniques that extend those that we previously presented. Common to these techniques is an awareness by the system of body location and activity. Shadows (as described in Chapter 3) are incorporated into these techniques as a unifying metaphor to provide awareness to users and collaborators. We also discuss two different applications that make use of these techniques. The applications share many commonalities with the use case scenarios described in Section 3.2, including the need to annotate and input text, navigate by panning and zooming, and interacting at a variety of distances. The techniques described in this chapter are made possible by a significantly more detailed model of the scene, including explicit representations of users and displays. This was not available in the previous prototypes. We argue that such a scene model should be a central and critical part of any body-centric design approach.

We start by drawing from experience in other fields, including psychology and sociology, that have explored the human body from different perspectives. We then take what we have learned and describe a number of individual interaction techniques and prototype systems that were developed to support an exploration of our body-centric approach.

Elements of this chapter have previously been published [184]. Design guidelines presented here are refinements of those presented earlier.



**Figure 4.1:** Peripersonal space (red volume) is that which is in reach of the arms, and often does not intersect with a large display during use.

## 4.1 Inspiration from Other Fields

Psychology and sociology have developed a very rich understanding of how human beings perceive the external world. An understanding of how the mind and body work is critical to the development of effective interaction techniques, especially when these techniques are intended to be body-centric. Based on our analysis of the literature we will not only develop a broad understanding of the issues underlying interaction, we will also define some specific design guidelines.

### 4.1.1 Interaction Spaces

The brain builds multiple representations of space in order to help it understand the world and coordinate operations. Neuropsychologists have discovered three representations that are of particular interest to our design context: *personal space*, *peripersonal space* (Figure 4.1), and *extrapersonal space* [91]. Personal space is the space occupied by the body. As Holmes and Spence discuss, peripersonal space is the space immediately surrounding our bodies [172]. This is the space where it is convenient for us to reach out and interact with our hands. Extrapersonal space is that which is not within easy grasp [169]. In order to physically reach into extrapersonal space it is normally necessary to move our bodies closer, turning ex-

trapezoidal space into peripersonal space. Although on a conscious level we don't always distinguish between peripersonal and extrapersonal, the brain seems to possess separate mechanisms for operating in each of them. Our ability to reach into the two spaces differs. The risk of colliding with objects in the two spaces differs as well. It is reasonable that the brain would construct different representations of the two spaces because separate representations could be optimized for the operations relevant to the different spaces. For example, the representation of extrapersonal space might be optimized for understanding and navigating, but not for interacting, whereas the representation of peripersonal space might be optimized for interaction. The word "might" is used here because psychologists are still in the process of developing their understanding of the different representations.

A slightly different interpretation of interaction spaces is offered by Colby [40]. She describes the distinction between *egocentric* and *allocentric* reference frames. Egocentric reference frames are those that are described in relation to the observer. Allocentric reference frames are described in relation to external objects. These reference frames are actually classes of reference frames, rather than individual reference frames. For example, a number of egocentric reference frames can be described, including: one in relation to the eyes, one in relation to the right hand, and one in relation to the torso. Similarly, multiple allocentric reference frames can exist in relation to different objects, or to a room.

What is the significance of multiple spaces to interaction with computers? As Cardinali points out [31], peripersonal space is "characterized by a high degree of multisensory integration between visual, tactile and auditory information." The significance of this is that peripersonal space "constitutes a privileged interface for the body to interact with nearby objects." In other words, the brain appears to be optimized for interaction in peripersonal space.

In human computer interaction, especially in situations involving large displays, it is often useful and possible for a user to interact with regions of a display that are outside of peripersonal space. Users may wish to interact with objects at the top of a very large wall display or at the center of a very wide table display, or they may wish to interact with either kind of display while standing at a distance from it. These operations are made possible through the use of indirect input techniques such as laser pointers, or the Shadow Reaching technique described in

Chapter 3. Such interactions are performed outside of the region where the “high degree of multisensory integration” exists. An important question is raised: does performance outside of peripersonal space really differ from that within peripersonal space? If so, can we design interfaces so that users can interact in extrapersonal space equally efficiently as in peripersonal space? Studies by Halligan and Marshall [79], and further explorations by McCourt and Garlinghouse [143], demonstrated that the performance of line bisection tasks differs based on the space (i.e. peripersonal vs. extrapersonal) within which the task is performed. Answering the second question, therefore, becomes important in our development of large display interfaces.

#### **4.1.2 Binding Spaces and Shadow Interaction**

We have discussed the different interaction spaces, and the problem of allowing users to interact in extrapersonal space using the optimized cognitive mechanisms normally used to interact in peripersonal space. Luckily, peripersonal space is flexible, and can change based on a number of factors. In this section we describe techniques for “binding” peripersonal and extrapersonal space with the goal of supporting interaction with computer displays.

Research by Vaishnavi et al. [202] has shown that the brain naturally “binds” personal and peripersonal space, so that the brain’s mechanisms for operating in one space are able to operate in the second space. This allows us to reach out and grasp an object in our immediate vicinity. Furthermore, Maravita et al. showed that a mirror image of a person serves to bind peripersonal and extrapersonal space [141]. It has also been shown that a shadow representation of a person similarly binds extrapersonal and peripersonal spaces [161], leading to the conclusion that a person’s “body schema” extends to include the body’s shadow. They note in particular that this can enhance a person’s ability to interact in virtual environments. There is thus substantial concrete evidence to suggest that shadows, and other personal embodiments, can enhance the interactive experience.

From this analysis of binding of spaces we formulate our first design guideline.

**D1** Where a large display system supports interaction at a distance, the interaction should be mediated through a representation that binds peripersonal and

extrapersonal space.

Researchers in HCI have been exploring shadows and other user representations [6, 99, 149, 151, 183, 195] as a means of providing expressive embodiment [76]. From our analysis of the psychology literature, it becomes clear why shadows have the power they seem to have.

While interaction spaces and bindings between those spaces are important topics, our understanding of these spaces is incomplete. As an example, early evidence that tool use can extend peripersonal space to beyond a hand's reach [98] has recently been contradicted by other work indicating that tools simply capture attention [90]. This indicates that it is as yet difficult to state absolute conclusions about how we function in different spaces, and the implications of these conclusions for human-computer interaction are also still not certain. We expect that an exploration of the topic in both psychology and HCI will continue for some time.

### **4.1.3 Proprioception**

Not all interactions need be performed in the space of the display. The human body itself can play an important role. Proprioception is a person's sense of their own body in space, using information gathered from muscles, skin, and joint receptors [66]. Cocchini et al. showed, using a "fluff test" of experiment participants removing stickers from their own body, that the brain has a separate mechanism for governing proprioceptively-guided self-touching [37]. It has also been shown that "eyes-free" proprioceptive reaching can outperform vision-guided reaching [46].

We conclude that proprioceptively guided reaching in personal space can augment parallel observation in extrapersonal space, and formulate our second design guideline.

**D2** Leverage the sense of proprioception by allowing some operations to be performed in the user's personal space without reliance on visual feedback from peripersonal or extrapersonal space.

#### 4.1.4 Social Conventions

Humans follow complex social rules that ease collaboration and co-existence in the physical world. Social conventions mediate means of communicating not only verbally but also physically, so that intentions are clarified. If a computing system is able to capture elements of user interaction that relate to known social conventions, users' existing knowledge and observation of the conventions can be used to improve interaction.

One important aspect of inter-user coordination relates to how people position themselves relative to one another during work. As Felipe and Sommer explained [58], there is a universal cross-cultural concept of private space<sup>1</sup>. Every person has a region of private space circumscribed around their body outside of which they attempt to keep other people. In work, it is generally only during direct collaboration that a person will comfortably allow another to enter into their private space. As described in a review by Sundstrom and Altman [188], however, the concept of private space is more complex and fluid than the simple dichotomy of private/non-private. In their model, the acceptable distance between two people is dependent on the shifting factors defining the interpersonal relationship. An interactive system that has a model of users and space, and a basic understanding of the concept of private space as it relates to collaboration, could use this knowledge to enhance the interactive experience.

Motivated by this, we formulate our third design guideline.

**D3** Interaction techniques should respect user models of private space, and when possible take advantage of them.

Social conventions are based on much more than just body position and orientation. It has been shown that non-verbal cues such as eye contact, body lean, smiling, and touch are all important in communicating trust and guiding decisions about conduct within the relationship [22]. Burgoon et al. identify a host of themes associated with these non-verbal cues that define relational interchanges, including *dominance or submission, emotional arousal, composure, similarity, task or*

---

<sup>1</sup>“Private space” in this context is sometimes referred to in the literature as “personal space.” We call it “private space” to disambiguate from the other definition of “personal space” used here.



*social orientation*, and others. They found in an experiment that frequent eye contact expresses greater intimacy, immediacy, and dominance. They further found that forward body lean and the presence of smiling communicate greater intimacy, whereas smiling and touch communicate intimacy and informality. They determined that close proximity was the cue carrying the most weight. These cues can be difficult for a computing system to capture, due to limitations in sensing. They can nevertheless be leveraged by developing interaction techniques that incorporate direct user-user interactions.

We thus formulate our fourth and final design guideline.

**D4** Where possible allow users to make direct use of body cues such as facial expression and posture in order to help manage coordination.

## **4.2 Supporting the Development of Rich Whole-Body Interaction Techniques**

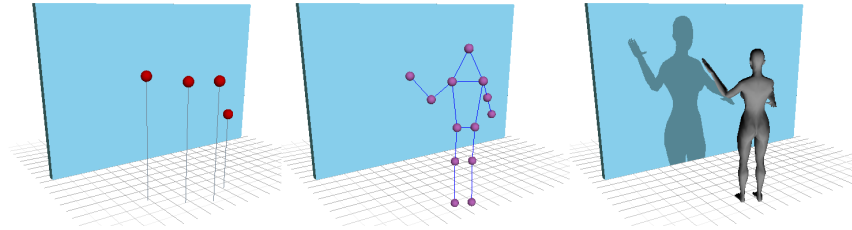
A robust implementation of whole-body interaction requires knowledge of the physical environment in which the interactions take place and also of the application that is being supported. We discuss each of these in turn.

### **4.2.1 Scene Model**

In our initial exploration of a body-centric interaction approach in Chapter 3, we limited ourselves to the development of shadow-metaphor techniques. Furthermore, our implementations were limited by the sensing approaches we used to capture user contours.

The interaction techniques we describe in this chapter rely on a much richer understanding of where users and displays are in relationship to each other. In order to support these techniques, we developed an approach for modeling the scene, based on data captured from magnetic sensors.

Our scene model is a full geometric model of users and displays. As shown in Figure 4.2, users are represented as both skeletons and contour meshes. The user models are generated by first capturing the locations of certain key user joints using magnetic position trackers (Polhemus Liberty Latus), and then generating an



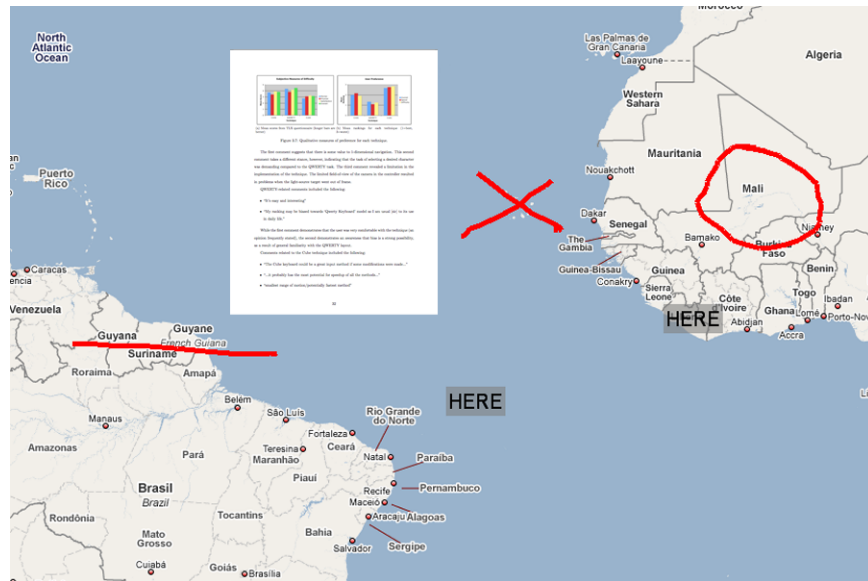
**Figure 4.2:** The pipeline used to generate the scene model. Hand and shoulder locations of the user are measured with magnetic location trackers, then a skeleton estimation of the user’s pose is generated, then a human mesh is mapped to the pose of the skeleton.

approximation of the entire skeleton using an inverse kinematic (IK) approach. The IK solver we used made sufficient assumptions that a single solution was always possible and easily computed using simple geometry. Finally, a 3D mesh of a generic user shape (generated using the MakeHuman open source human mesh generator) is manipulated to match the pose of the skeleton. The geometry of the displays in the scene are manually entered into the model after being measured in the workspace. If the displays were mobile it would be possible to track them in a manner similar to how we track users.

The scene model is a generic representation. It is agnostic to any individual interaction technique, and can be used for generating any individual technique. We will discuss the details of the scene model generation and use in Chapter 6.

## 4.2.2 Application Context

We wanted to explore our interaction techniques in the context of a realistic application. The application we developed is an interactive map exploration and editing tool (Figure 4.3). As we mentioned in a previous chapter, mapping is a common task for large wall displays, with examples in industrial control rooms [1], military command-and-control [179], and crisis management [104]. Our application makes use of 2D tiled data downloaded from Google Maps. It allows users to zoom in and out arbitrarily, and to pan to different views. The interaction techniques were integrated in order to allow users to more easily navigate and edit the map, including drawing free-form sketches and adding text.



**Figure 4.3:** Screenshot of the map exploration and editing application. Users can sketch geo-referenced annotations, type text, and insert documents.

Interaction was performed using Nintendo Wiimotes. The user held one device in each hand, and pressed buttons to initiate events. This is not ideal. As we have discussed before, “click” events do not have a real world equivalent. It would be more desirable to make use of real world actions such as grasping or pointing. The difficulties in implementing such approaches, however, drove us to use this simpler approach. We decided that exploring the “large-scale” aspects of body-centric interaction held more immediate potential than did dealing with the fine-grained aspects of gestural input.

### 4.3 Single User Interaction Techniques

The system-maintained scene model, described in section 4.2.1, includes body models of all users and relevant contextual objects. It supports the development of body-centric interaction techniques. We describe here several interaction techniques that make use of these models and leverage the themes explored in section 4.1. The techniques make use of the model by, for example, querying the 3D location of a user’s shoulder, the orientation of a user’s body, or the distance

between two users.

### **4.3.1 Virtual Shadow Embodiment**

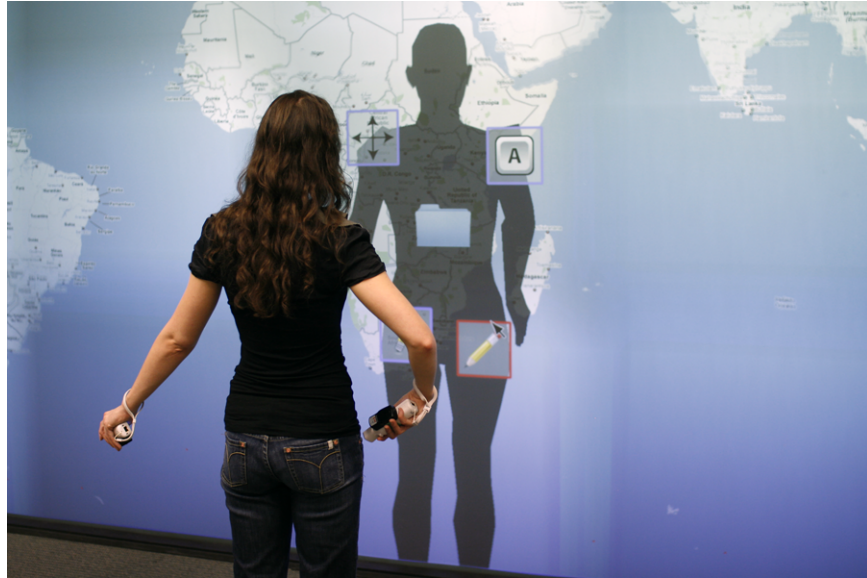
In Chapter 3 we described our early exploration of user shadow embodiments and our Shadow Reaching interaction technique. An important conclusion reached after developing several prototypes was that the potential of shadow embodiments can only be fully realized if they can be generated from any arbitrary perspective.

In the map exploration and editing tool we addressed some of the limitations of the Shadow Reaching prototypes. Rendering of the virtual shadow is accomplished by computing the projection of a virtual 3D model of the user onto a 2D surface representing the display. The projection can be done from any arbitrary location. The resulting projection of the user is then rendered onto the actual screen as a semi-transparent black shadow. This approach overcomes the major limitations of the original Shadow Reaching prototypes. Later in this chapter we will describe other extensions of the original Shadow Reaching concept.

In our conclusions in Chapter 3 we hypothesized that user shadows could be used as a platform on which to design other body-centric interaction techniques. This chapter will describe several techniques that do not inherently demand the use of a user body shadow, but we believe that they are made more powerful by being combined with a shadow representation.

### **4.3.2 Body-Based Tools**

Body-based tools are virtual tools that are stored at real physical locations on the user's body (Figure 4.4). To enter a mode or select an option in an application, the user places a hand at the corresponding body location and presses a button. This approach builds from everyday experiences. People frequently store items, such as keys or a wallet, in very specific locations, and can access these items easily with little physical or mental effort. It is this ease that we hoped to capitalize on. Body-based tools can be realized in a number of ways. In an application that also makes use of virtual shadow embodiments, the tools can be visually associated with the shadow embodiments. In this case the icons for the tools will appear on the shadows during tool selection in order to serve as visual aids. Alternately, if



**Figure 4.4:** A user accesses a tool stored on her right hip by placing her hand at that location. A variety of tools can be stored at different body locations.

the user has learned the locations of the tools on the body, the tools can have no visual representation on the display and the user can select them “blind.” This has the advantage of not requiring any visual feedback that might confuse other users.

This technique follows design guideline **D2**, allowing interaction in the user’s personal space and leveraging the proprioceptive sense. Compared to traditional toolbars and tool palettes this approach has several benefits. First, the user can select known tools without having to perform a visual search and targeting operation. Second, a user’s tools automatically follow the user and are always available, but don’t clutter the display. Third, in collaborative scenarios there is no confusion regarding who controls what tool, because each tool clearly corresponds to a single user’s shadow. We hypothesize that these advantages will simultaneously improve tool selection performance and reduce confusion.

In our implementation, body tools are normally not visible, but become visible if triggered by a button on the Wii Remote. The user can then hover over a tool and select it with a second button press. If the user knows where a tool is they can



**Figure 4.5:** A user accesses her personal data store. The data store is centered on the user's personal shadow embodiment. She can browse through the file hierarchy and move documents to the shared space.

select it directly with no toggling of visibility.

### 4.3.3 Body-Based Data Storage

Body-based data storage allows for convenient access to a user's personal data (Figure 4.5). There are many situations in which a user may want to retrieve personal data, such as a PDF file or photo, and then show it on the shared display. Body-based data storage provides a body-centric metaphor and mechanisms for accessing and sharing this information, consistent with design guideline **D2**.

Each user's torso serves as a virtual container, from which personal data files can be accessed. This virtual storage is mapped to a user's computer or network drive. A user can use his or her hands to open, expand, and search through files virtually stored in the torso. When the desired file is found the user can extract the file from their torso and drag it to the shared space. This approach has many of the same benefits as body-based tools. First, personal files are always in close proximity and readily accessible to the owner, and second, there is little possibility

for confusion regarding who “owns” which storage area.

There are several other advantages that are specific to the torso storage technique. Centering the navigation on the torso also centers it between the user’s arms. This makes it easy for the user to interact with the data, which is important because navigating through a complex file space is not a trivial task. We also note that the torso is simultaneously the most massive part of a person’s body, and the center of the person’s body. The mass of the torso lends itself to being a metaphorical container for vast amounts of information. The fact that it is central to the body also makes it a personal part of the body, which associates well with the private nature of the data being accessed, and follows design guideline **D3**.

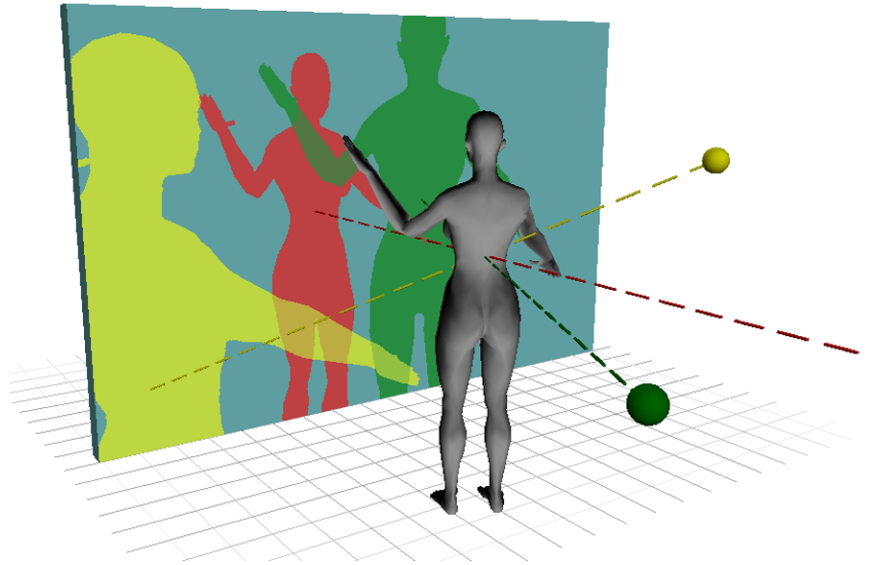
Visual feedback is provided through a data browsing widget in the form of a familiar hierarchical file browser shown in a grid layout. This is a suitable general purpose solution, however, if the application deals with only specific kinds of personal data, such as photos, a special-purpose widget could be designed.

#### **4.3.4 Dynamic Light-Source Positioning**

A single virtual light source is associated with every user, and the shadow cast of the user from the light source location onto the plane of the display is used to support interaction. Supporting dynamic light-source positioning can impact interaction in several meaningful ways. First, changing the projection of the shadow can allow the user to reach arbitrary locations on the screen. Moreover, altering the location of the light can be used to adjust the control-display ratio (CD) ratio, which can have a significant impact on pointing performance and error rates. CD gain is a smoothly varying function dependent on light ( $L$ ) and user ( $U$ ) distances to the display ( $gain = \frac{L}{L-U}$ ). We have developed several different light behaviours that govern how a light source moves (Figure 4.6), based on the scene model.

##### **User Following**

This light behaviour allows for easy manipulation over the entire surface of a very large display, without requiring the user to walk around. Based on the known location of the user’s shoulders, the behaviour places the light-source directly behind the user at a given distance. The result is that the user’s shadow moves as the user



**Figure 4.6:** Three possible light source behaviours, coded by colour. Green: user following. Red: orthographic. Yellow: manually positioned. This visualization was created in a modified version of the experimental system using real data.

turns, so that it is always directly in front of the user. This allows the user to perform continuous operations (such as dragging) across the entirety of a very large display simply by turning his or her body.

### **Orthographic**

This behaviour depends on the location of the user, and on the position of the display. The light source is placed at a very large distance directly behind the user, in a direction defined by the surface normal of the display. The result is a near-orthographic projection of the shadow onto the display.

The purpose of this behaviour is to provide a shadow mode of minimal distortion, with little risk of confusion. Confusion is minimized because the shadow is at the location on the display closest to the user. Close proximity minimizes the chance that the shadow will interfere with other users who are located elsewhere. The shadow does not move when the user turns, which can also minimize



confusion in multi-user situations.

### **Manually Positioned**

At times users may wish to manually position a virtual light source. The user may, for example, wish to optimize the shadow for interaction in a particular region on a very large display. A manually positioned light also provides a very stable projection, which can ease detailed work.

A variety of approaches can be taken for supporting user control of the light source. In our implementation the user points in the direction where the shadow is to appear and presses a button. The light source is then positioned behind the user in the direction opposite to the direction pointed. The distance  $d_l$  between the light source and the user is a function of the distance  $d_h$  of the user's hand to the user's body. Because the user is restricted by arm length, the distance is exaggerated by the system. For example,  $d_l = d_h^2 + c$ . This approach allows the user to control both the location of the shadow and its size, and as a result the CD ratio of the input.

### **Behaviour Transitioning**

This is a means of managing transitions between other behaviours. When switching from one behaviour to another it is undesirable for the light source to jump instantly from one position to another. This can cause confusion for the user and collaborators. Instead, the system transitions from the position calculated by the old behaviour function  $p = f_o$  to the the position calculated by the new behaviour  $p = f_n$  over a short period of time  $T$  by calculating a linear blend of the two functions  $p = (1 - t/T) \times f_o + (t/T) \times f_n$ . This results in continuity of the shadow projection.

## **4.4 Collaborative Interaction Techniques**

Large display systems are frequently used to support co-located collaboration, and ideally they should seamlessly support natural collaborative interactions. Although our current sensing and modelling approach focusses mostly on the *geometric* properties of users and environments, it is possible to extract an indication of collaborative intentions based solely on user geometry, and to further leverage this

through specific techniques.

#### 4.4.1 Synchronized Shadow Projections

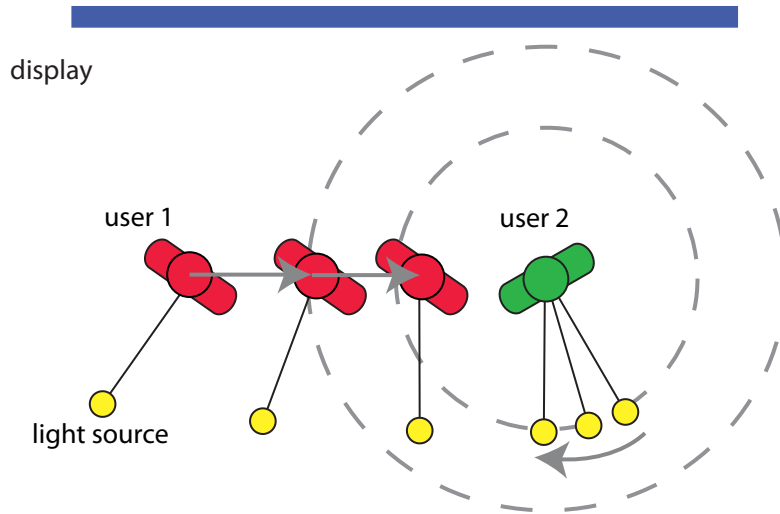
When users are collaborating, inter-user coordination is a concern equal in importance to raw interaction performance. However, the importance of collaboration at any isolated moment in time depends on how closely users are collaborating. Users positioned at opposite ends of a large display are likely working independently, whereas users positioned directly beside each other are likely collaborating closely. The synchronized shadows technique uses inter-user proximity, following design guideline **D3**, as an indicator of the degree of collaboration, and alters the shadow behaviour to change in a manner that supports each user's current collaborative state.

When users are not collaborating closely, the technique allows each user's shadow to follow its own behaviour independently (e.g. user following). As two users approach and enter each other's private space, however, the shadows synchronize (Figure 4.7). Synchronization means that the shadows alter their projection in order to be *consistent* and to *minimize conflict*. Consistency means that the shadows reflect a believable real-world lighting situation. For example, if User 1 is to the left of User 2, then User 1's shadow should be to the left of User 2's shadow. To minimize conflict, we enforce the condition that shadows not overlap significantly. The more shadows overlap, the more likely it is that users will be confused.

Once the user is judged to be within collaboration range the system transitions to a lighting model consistent with the set of requirements. The orthographic lighting model fills these requirements. Collaborative range can be defined as desired, but a good value is in the range of  $45\text{cm}$ – $120\text{cm}$ , identified by Hall [78] as a typical radius for private space.

#### 4.4.2 Access Control and Conflict Management

Management of private data is a concern in collaborative systems. Users must not only have a means of moving data between different privacy states, but the privacy state of all information artifacts must also be clear to users. We have built our access control protocols to center around the theme of *social awareness & skills*,



**Figure 4.7:** As a first user approaches a second user and enters that user’s private space, both users’ light sources transition to behaviours that are conducive to collaboration.

as defined by Jacob et al. [101]. We make use of standard social conventions to govern the handling of private data.

We enforce privacy by requiring all access to private data to take place in the literal body frame of reference (personal space), whereas access to public data takes place in the display’s frame of reference. For example, in order for a user to move private data from body storage to the display, the user must first directly access that storage through their torso. Once the file has been moved to the shared display, however, it can be accessed in the display’s frame of reference by any user. This follows design guideline **D3**.

In another scenario, if User 1 wants to grant User 2 permanent access to a personal file (or give User 2 a copy of the file), the user must physically and literally pass the file to the other user’s hand (Figure 4.8). Once the users’ hands move to within a certain distance of one another, the file is copied to the receiving user’s file store, and is then accessible to that user. This protocol of forcing private information access to occur in peripersonal space builds on a person’s sense of their own



**Figure 4.8:** A user passes a private document to a collaborator. The sharing protocol requires close physical proximity, and encourages direct eye contact. Feedback on the screen is a green circle surrounding the projection of the two users hands, to indicate a successful pass.

private space, and also allows users to observe each other directly, making use of often subtle human cues to aid in the coordination of the sharing task. This follows design guideline **D4**.

## 4.5 Preliminary Evaluation

We described a number of novel interaction techniques. Each one warrants a full controlled experiment, but that is beyond the scope of this work. We instead gathered preliminary user feedback from six users, with the goal of guiding future design iterations. Each user was introduced to the different application features and interaction techniques, and was then given an opportunity to explore the system. To simulate a collaborative environment the experimenter served as a colleague. Notes were taken about user behaviour, and feedback was gathered both during and following the session. Each session lasted approximately half an hour.

All users seemed able to understand the concepts behind the interaction tech-

niques. After one or two tries users were able to use the body-centric metaphor for tool selection, and similarly were able to navigate personal file space. Commenting on the body-centric approach in general, one user observed “you can’t mess up!” The different lighting behaviours were also easily understood, as were the collaboration protocols. This suggests that basing interactions on real-world body metaphors was a good decision. Nevertheless, there were several lessons learned that can guide improvements.

First, several participants commented that performance and realism are important in supporting the power of the shadow metaphor for interaction. The system exhibited occasional “hiccups”, where there was an observable delay before rendering refresh. These delays broke the users’ mental models of the reality of the shadow representation. There appears to be a threshold of accuracy that the shadow must achieve in order for the user to benefit from the embodiment and the binding of peripersonal and extrapersonal space. This could be related to the familiar concept of the “uncanny valley” from computer graphics [87], where something that is graphically close to real, but not quite real, can be seen as disconcerting.

Another recurring question related to shadow representation. Users wanted to know why the particular shadow visualization was chosen, and they were curious whether a different shadow representation might be superior. This is a valuable observation. We initially explored a simple shadow, but other variations of shadows should be explored. It was also observed that the shadow did not accurately represent user differences. This was necessary due to the generic mesh that we used to represent users. The noticeability of differences may be minimized with different shadow visualizations (e.g. with fuzzy edges), or alternately superior capture of user geometry (e.g. with visual hulls [147]) could result in more accurate user modelling.

An interesting comment relates to tool placement. A participant asked if it was better to place commonly used tools on the left side of the body for a right-handed user, in order to make selection with the dominant hand easier. The answer is unclear, as it has been shown that a person is able to reach more accurately using proprioception with their left hand, if they are right handed [45]. The difference between dominant and non-dominant sides in proprioceptive selection is something that should be further investigated.

Another issue that arose is that it was sometimes difficult for participants to remember the state of the two different hands. Each hand can be in a different mode, which is more complex than normal desktop systems where only a single cursor mode has to be remembered. It was suggested that the visualization of the input may change based on what mode a particular hand is in. It is not known if this would be sufficient feedback. In the real world we have tactile feedback from the tool being used to help us keep track of which tool is in which hand. A similar tactile approach may be desirable in our context.

## **4.6 Design Iteration**

A number of ideas for future development were identified in the initial informal evaluation of the prototype collection of body-centric interaction techniques. We performed a second iteration of design and development of the prototype system and related techniques. This involved improving some existing aspects of the system, and developing other new components.

### **4.6.1 Performance Improvements**

In the evaluation of the initial prototype, performance was identified as a critical factor in generating believable shadow visualizations and related interaction techniques. In reviewing the performance characteristics of the prototype, several performance bottlenecks were identified and addressed.

#### **Rendering**

The initial prototype made use of Microsoft's DirectX library for rendering visualizations. DirectX is very powerful and is a fully functional rendering platform, but it can be hard to use. There are many pitfalls that it is possible to fall into during application design, and these can have negative side effects in terms of performance. We determined that many of the performance hiccups in our application were related to non-optimal DirectX tunings.

In order to address the performance problems in rendering, we decided to move our rendering pipeline to Microsoft's XNA rendering platform. XNA is an abstraction layer on top of DirectX. While DirectX still performs all rendering, XNA hides

many of the vagaries of DirectX from the developer. XNA also eases development of graphical rendering shaders, and provides a convenient pipeline for managing and loading content such as textures, geometry, and meshes. After the move to XNA, many of the performance problems in the initial prototype were no longer evident.

### **Resource Loading**

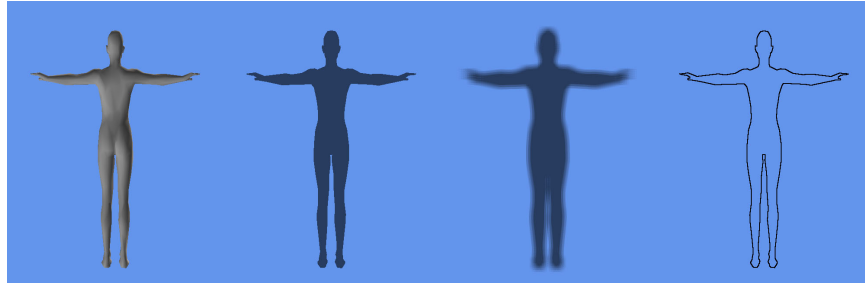
Another performance problem arose due to the problem of generating texture maps of dynamically loaded resources. The application uses map tiles that are either downloaded from Google Maps or stored in a cache on the machine. Because the application cannot anticipate the tiles that will be required it is not able to generate textures on loading of the application, so textures must instead be generated on the fly.

Generating textures from bitmaps in memory is a time intensive operation, and in normal circumstances is performed in the main application thread of an XNA or DirectX application. The result is that while the texture is being created the application comes to a halt, and no interactivity is possible. This is unacceptable from a user's standpoint. Our solution was to perform as much processing as possible in a separate thread. In the texture loading thread the raw bytes for the texture are loaded from the source (web or disk cache) into a memory buffer in the appropriate format. Then the only remaining processing to be performed on the primary thread is to load the already prepared texture. This reduces latency significantly.

### **4.6.2 Shadow Visualizations**

One recurring question from users in the initial evaluation regarded the shadow representations. Some users wanted to know what the ideal level of opacity of the shadow was. Other users wanted to know if a different representation would be superior to a solid shadow. After all, the shadow in our initial implementation darkens the region of interest, possibly making it harder to see.

We developed support for rendering novel shadow representations. Our architecture makes use of the High Level Shader Language (HLSL). HLSL is a Microsoft-developed shading language that allows for the specification of highly



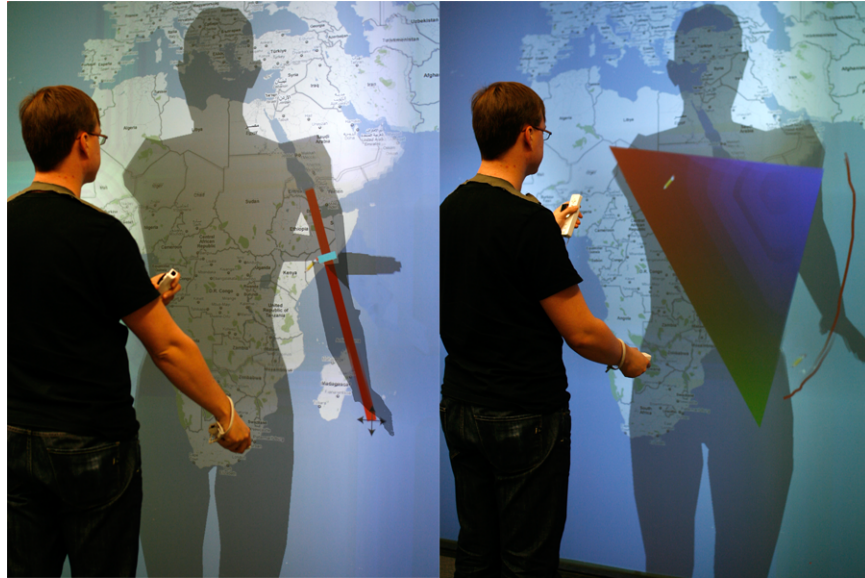
**Figure 4.9:** Four different styles of shadow rendering developed to explore shadow embodiments. From left to right: real geometry, sharp shadows, soft shadows, and body contour.

efficient and customizable shaders that run directly on graphics hardware. These shaders can be easily reconfigured and selected at run-time. We developed four different shadow renderers for exploration (Figure 4.9). These are: real geometry, sharp shadows, soft shadows, and body contour. The real geometry renderer shows an arbitrary real 3D model, fully lit, rather than an abstract shadow. The purpose of this is to explore more realistic user representations in the scene. It is possible that an embodiment with more realistic features, such as hair, eyes, and clothing, will be more powerful than an abstract representation, without being too distracting. The model shown in the Figure 4.9 is quite simple, but a more complex one could be used. The second renderer, sharp shadows, is visually the same as the original renderer described in Section 4.3.1. The third renderer, soft shadows, draws a shadow with a penumbra of configurable radius. This was developed to explore whether a shadow with a softer edge may be more believable. A number of users of the old sharp-edged shadow said that it seemed unrealistic due to its sharpness. The last shadow renderer, body contour, draws an outline around the contour of the body. The purpose of this is to minimize occlusion of regions of interest on the display.

### 4.6.3 Body-Based Control Surfaces

Adjustment of numeric values is a common task in any interactive system. In traditional UIs this is often done using 1D sliders or 2D widgets. Body-based control surfaces combine traditional easily understood widgets with a body-centered pro-





**Figure 4.10:** Two examples of body-base control surfaces. Left: a 1 dimensional slider mounted on the user's arm. He moves his hand up and down his arm to adjust a numeric value. Right: a 2 dimensional colour selector. The user selects a colour with one hand and draws with the other.

proceptive approach, following design guideline **D2**.

We implemented two different control surfaces (Figure 4.10). The first is a body-based 1D slider. The ends of the slider are connected to specific body joints. The joints chosen are ideally connected by a body part (e.g. elbow and hand connected by a forearm). The user can adjust a single numeric value by sliding a hand along the body part connecting the two joints. Feedback is shown on the display, but using proprioception the user can avoid relying on the feedback. In our application we implemented a slider that adjusts the darkness of the user's shadow.

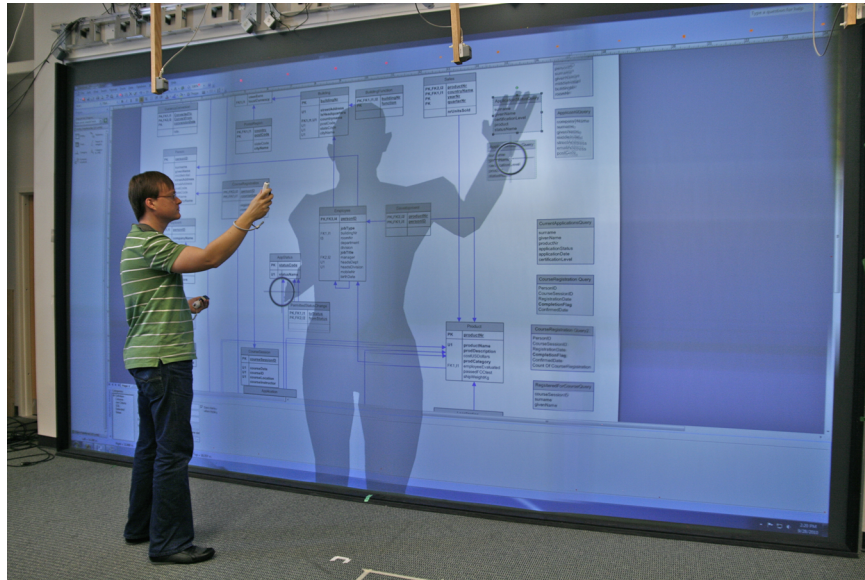
A 2D control surface can connect three or more joints. The surface visually connects the joints, and the user can adjust a multi-dimensional value by moving a hand over the surface. We implemented an RGB colour selector for adjusting the colour of sketch annotation.

## **4.7 A System for Supporting Universal Body-Centric Interaction in the Windows Operating System**

While it is useful to investigate new interaction approaches through the development of custom applications, this approach has limitations. It does not allow researchers to explore how the new interaction approaches might function in the context of existing applications and environments. Understanding how novel approaches can function in a familiar context can be critical, because new technologies are rarely introduced as part of entirely new working environments. New technologies often must first be introduced as part of established environments such as Microsoft Windows or Apple OS X.

Researchers have made many attempts to integrate new interaction approaches into existing environments. An early example is MIDDesktop, which supported interaction with Java Applets using multiple mice and multiple independent cursors [185]. A more recent example is Multi-pointer X (MPX), which integrates support for multiple mouse interaction into the X Windows system [95]. It is a success in that it is now officially part of the open-source code base. It is a shallow integration, however, in that it must work around the assumptions inherent in the operating system regarding the number of system cursors and input devices. Nevertheless, it is extremely useful for expanding the utility of the environment, as well as for raising awareness regarding the possibilities of new interaction techniques, making widespread and deep integration more likely in the future.

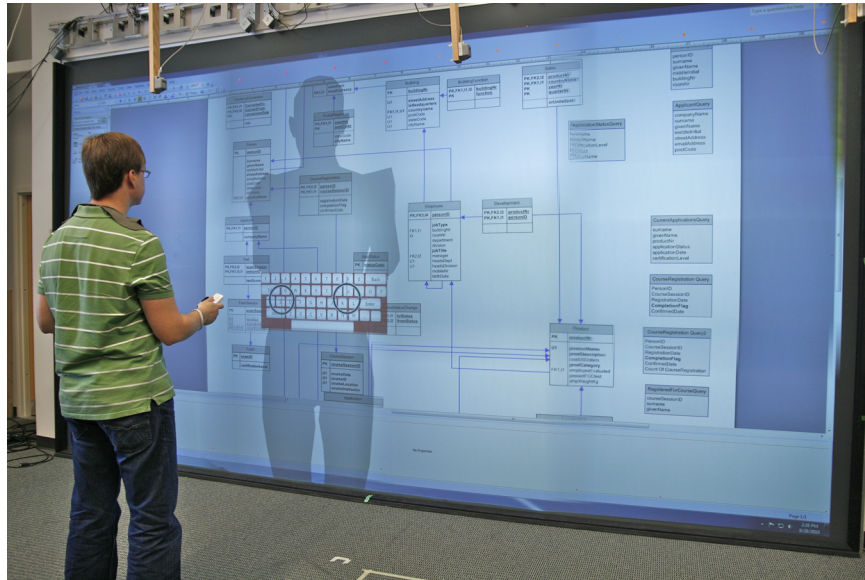
Towards the goal of exploring the use of our techniques in a familiar context, we developed an application that runs on the Microsoft Windows 7 operating system, and allows users to interact with the OS and native applications in much the same way as they do in our dedicated prototype. The application operates as a layer on top of all running applications, and captures actions from the user and generates native operating system mouse and keyboard events. It allows users to stand anywhere in the room and interact at a distance through a shadow embodiment of themselves. The system is shown in Figures 4.11 and 4.12.



**Figure 4.11:** A user interacting with Microsoft Visio. The interaction techniques supported by our system allow the user to select and drag objects in the application as he would normally do with a mouse.

### 4.7.1 Event Management

The application must trigger mouse events in the operating system in order to allow interaction with native applications. Because Windows 7 only has one system cursor the application must coalesce inputs from potentially multiple hands and multiple users into a single event stream. The approach it uses is summarized in Figure 4.13. The event coalescer component determines the individual input that controls the cursor based on priority. Upon a mouse down being triggered by a Wii Remote button press, the coalescer determines if the cursor is being controlled by another hand. If not, then it grants the input triggering the click authority to control the cursor. The input does so until the mouse button is released. If an input triggers a mouse down during a period when a different input has control, then the mouse down is not granted authority. While an input is controlling the cursor during a mouse down and mouse up event, the mouse cursor is sent events such that its motion tracks the projection of the controlling hand onto the display. Thus, the system cursor follows the user's hand's shadow. This approach allows for fairly seamless

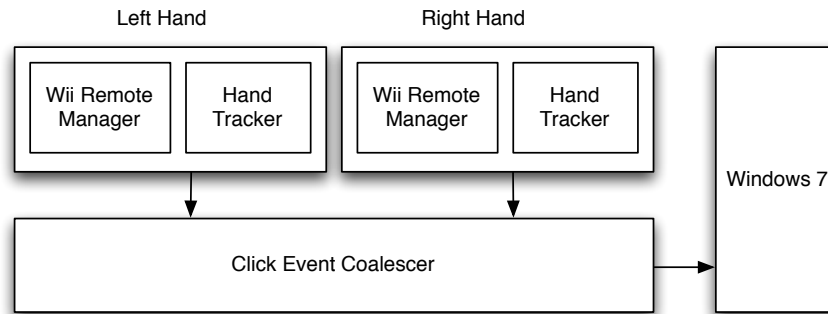


**Figure 4.12:** A user interacting with Microsoft Visio. A virtual keyboard integrated into the system allows the user to input text without using a physical keyboard, and without being within touch distance of the display.

transfer of control from input to input, or from user to user, but can frequently result in denied control when more than one input is triggered. This is unavoidable in a single cursor system such as Windows 7.

One drawback of the system is that the inputs only control the cursor when a click event has been triggered. Thus, click and drag events both function, but mouse movement events outside of mouse clicks do not. This can make it impossible for hover events to function. This is a problem that was initially described by Buxton in the development of his three-state model of interaction [24]. This limitation in interaction is similar to that used in stylus and touch based systems, where hovering does not exist. It can sometimes result in difficulties in applications designed for mouse use.

The event handling system can be considered to have analogs in the realm of explicit turn-taking in multiple mouse environments. Inkpen et al. investigated different protocols for mouse sharing [97]. They discovered that “give” and “take”



**Figure 4.13:** Flow of click events in the application. Clicks with the Wii Remote signal the event coalescer. The coalescer determines the Hand that currently has control of the cursor based on priority: first click grabs control until the click is released. The coalescer then determines the cursor location based on the location of the hand as projected to the display. The coalescer then sends mouse control events to the operating system.

protocols result in different levels of performance for children solving puzzles, and that the results depend on gender. Our protocol is a hybrid of the two protocols described by Inkpen et al. When idle, the pointer is “taken” by the initiation of an action, but cannot be taken when it is not idle.

We integrated only a few of the previously described interaction techniques into the system. These include the shadow representation and associated generation of click events, and the virtual keyboard for generating keyboard events. It would be possible to integrate some of the other techniques, such as body-based tools and body-based data access, but this would be more challenging, as these techniques don’t integrate easily into the existing OS workflow. although it can be done. For example, body-based tools could be associated with keyboard shortcuts, essentially working as macros to trigger changes to user state.

#### 4.7.2 Rendering

The application must provide rendered feedback to the user over top of the native application currently being used. Windows 7 provides an option to applications to render over top of all operating system content, even when the application is

not in focus. This allows the application to render at the front even when a native application is being interacted with. Unfortunately, there are a few exceptions to the render order. A few operating system components do not respect render order, and are rendered on top of all other components. These components obscure the shadows and cursors rendered by the application.

## 4.8 Conclusions

We have described a set of interaction techniques designed using the body-centric design approach. Each of the techniques makes use of a virtual scene model describing the location and pose of the users and displays in the workspace. The interaction techniques access the scene model in different ways, for example to determine a user's proximity to another user, the location of a user's hand, or the shape of a user's body as projected onto the display. The variety of interaction techniques that were possible using our geometric model of the scene is a testament to the power of our approach.

We also explored the feasibility of developing software applications using the interaction techniques we designed. A map browsing and editing application demonstrated that the techniques could be easily integrated into a common workflow. An informal evaluation of users' interaction with the application demonstrated the ability of users to explore and learn the interaction style, consistent with Klemmer's concept of *thinking through doing* [118]. A second application demonstrated that even though the Microsoft Windows 7 operating system is designed assuming a traditional WIMP style of interaction, a body-centric approach to interaction can be integrated into the OS. Some assumptions of the OS, in particular the assumption of a single cursor, impose limitations on how fully our techniques could be integrated, but the system worked to a satisfying degree.

The techniques we have developed follow our body-centric design approach, are rooted in hard research in the fields of psychology and sociology, can be integrated into new applications, and can even be integrated into existing operating systems to a limited degree. We believe that this demonstrates the power of these techniques and the body-centric approach in general, and that this should motivate further developments in the area.

### 4.8.1 Limitations

While the interaction techniques described appear to be compelling and powerful, and as a design exercise this should be considered a success, there are limitations to what we have accomplished. Most significantly, we have not performed any rigorous evaluations of the techniques. There are several reasons for this. First, an evaluation should ideally be performed on a prime example of an interaction technique. Our techniques are not prime examples, as they are hindered due to the limited sensing capabilities (i.e. noise and error) of the Liberty Latus wireless trackers, and the remaining performance problems in rendering. A formal evaluation would therefore underestimate the performance of an individual technique, compared to what it could be given optimal sensing. Second, it is unclear what evaluation should be performed for any one technique. Taking body-based tools as an example, they possess several theorized benefits over traditional toolbars, including mobility, the ability to be selected without visual feedback, and the ability to not interfere visually with collaborators' work. Designing an experiment to accurately measure all of these factors would be extremely challenging. These reasonings for not performing evaluations are consistent with the views of Greenberg and Buxton [70], who argue that evaluation at too early a stage of design can give misleading results and quash justifiably promising techniques.

Later in the dissertation we will isolate some aspects of body-centric interaction that can be thoroughly evaluated. In Chapter 5 we will investigate the task of inputting text, whereas in Chapter 7 we will perform a rigorous Fitts' law analysis of pointing performance. These evaluations will provide a foundation upon which higher level evaluations of specific techniques can be built.

In short, it is extremely desirable to evaluate this suite of techniques, but this is outside of the scope of this dissertation. In order for fair evaluations to be performed we must first have available better sensing hardware, and must also undertake extremely careful experimental design.

Another limitation is the reliance on "clicks" derived from traditional interaction techniques. Ideally we should be looking at new techniques, such as gesture based interfaces, that do not mimic button-based hardware. A more detailed discussion of this limitation appears in Section 3.5.2.

## Chapter 5

# Text Input

In this chapter we describe the design and evaluation of three text entry techniques for very large wall displays. Although the information represented by text is symbolic, and has no real-world physical equivalent, it is important that the physical means of producing text be properly integrated into an existing interaction approach (in this case body-centric interaction). Also, as observed in Section 3.2, many use cases for large wall displays include the inputting of text, either directly at the display or at a distance. Towards this goal, our design of the techniques takes into account the requirements specific to large wall displays that we previously identified as being a particular use case for body-centric interaction. These requirements include the ability to be used while standing, without any supporting surfaces, and the ability to be easily portable. As a result, the text techniques do not make direct use of the shadow metaphor underlying the other interaction techniques, but are able to be integrated harmoniously with these techniques.

We also identify two additional factors that are broadly applicable to all large display interaction techniques, and particularly relevant to text input: distance-independence and visibility-dependence. We describe a first evaluation that provides comparative performance results for three techniques. A second evaluation further investigates two of the techniques with the goal of understanding the importance of distance-independence on interaction.

Inputting text is a common requirement for many applications in nearly all computing contexts. It is considered to be one of the primitive interaction tasks [62].



For more than a century the standard QWERTY keyboard (in both typewriter and computer form) has dominated the mechanized creation of text in the English language [159]. Noyes points out that while the QWERTY distribution of keys on the keyboard, and indeed the physical layout of keyboard, may be far from optimal, the layout has emerged as a de facto standard. Layout ubiquity is a strength for use cases that are well supported by a traditional keyboard. A user familiar with the QWERTY key layout can sit down at nearly any desk in an English-speaking country and use the keyboard efficiently. A problem arises, however, in developing text entry techniques for situations where keyboards are not appropriate. In these cases it may be necessary to employ other interaction techniques to support text entry. For display form factors and work styles that differ significantly from traditional arrangements, the best outcome might not be achieved by simply building upon the familiarity of QWERTY-based input techniques.

As a general rule, each new computing platform must allow a user to input text for at least some applications. Different computing platforms will likely be best suited to different text input approaches. As new platforms emerge it is usually necessary to re-assess existing approaches, and perhaps design new ones. For example, smart-phones provide a variety of ways for text to be input: some phones support input using disambiguation techniques such as T9 on limited keypads, some phones support full keypads in either landscape or portrait orientation, and some phones support text entry using soft keyboards on a touchscreen. Makers and users have not converged on a single preferable technique. The development of text input techniques for large displays is also in its infancy. The use cases common to large wall displays, including multiple users moving freely in an open space, frequently without any table surface on which to operate, makes it difficult to develop a text entry mechanism that will be convenient and effective.

Large screen displays are an example of a new platform for interaction for which there is not yet a well-understood theory for developing or assessing text input techniques. This chapter presents some initial steps in this direction.

Elements of this chapter have previously been published [182].

## 5.1 Related Work

Text input possesses its own specific body of research. Because this body of work is relevant only to the current chapter, we begin with a section dedicated to a review of the literature on text input related to our work. Relevant previous research falls into two categories: text input techniques for large displays and text input techniques for small displays. Although our primary concern is text input for large displays, we will see that both are of interest for our work.

### 5.1.1 Techniques for Large Display Text Input

Large wall displays are physically similar to physical whiteboards and blackboards. It is natural to adopt text input techniques inspired by those physical surfaces. An obvious candidate is text input through direct writing using a stylus or other similar device. Many systems have taken this approach, including Flatland [157] and Tivoli [163]. While handwriting provides reasonable performance and would presumably be immediately understandable by a majority of users, it does have some drawbacks. It has been shown that writing speed with a pen is limited to around 20 words-per-minute (wpm) [9], which is inferior to many mechanized approaches. For example, touch typing on a keyboard is around 64.8 wpm for regular users [173]. It would be desirable to develop a text input technique with performance that approaches, or possibly even surpasses, that of typing. Another drawback with handwriting input is that it requires the user to be within physical reach of the display. The user can't be standing at a distance.

Because of the limitations of handwriting input, researchers have developed many alternate approaches. Pavlovych and Stuerzlinger evaluated text entry using direct touch with a variety of keyboard layouts that were shown on large displays [162]. They found that a standard QWERTY layout resulted in a mean text entry rate of 17.6 wpm, which is roughly comparable to handwriting performance. Magerkurth and Stenzel took a different approach, supporting text input for a large display using a small personal input device, with visual feedback provided on the large shared display [137]. The performance of their method was determined to range between 12.58 wpm and 21.27 wpm, depending on task and user experience. This is again in the range of, but somewhat slower than, hand-writing speed.

### 5.1.2 Techniques for Small Display Text Input

Small handheld devices, such as phones and personal digital assistants (and hybrids of the two), are increasingly being used for text-heavy tasks. Small display text entry approaches are relevant to large display input because small device use cases bear significant similarities to large display use cases. Users are often standing and walking, and need to input text in mid-air without relying on a fixed surface. Because of this, many small display entry techniques can be adapted for use on large displays.

It is difficult to integrate a full keyboard into a small display, so many small display text input approaches make use of text disambiguation techniques such as T9 or Multi-tap to support input on limited keyboards. These two techniques are widely deployed commercially, however, performance by any but expert users is poor compared to traditional text entry techniques such as typing, with typical performance being 7.98 wpm for Multi-tap and 9.09 wpm for T9 [102]. Other disambiguation techniques have been explored, including TiltText, which uses device tilt information from an accelerometer to filter which character from a set of possible characters is to be input into the system [212]. Performance of TiltText was found to be 11.76 wpm, marginally faster than the primary competitor, Multi-tap.

Techniques employing hand gestures can be particularly relevant to large displays. GesText is a technique that employs accelerometer-enhanced hand-held devices to support text entry [105]. Performance was found to be only around 3.5 wpm, perhaps because of the requirement that it operate on a device with only accelerometers. A similar technique relying on more sensitive and accurate sensing might very well perform better. For example, Amma et al. demonstrated how gyroscopes in addition to accelerometers could produce very reliable results, although they did not report performance results [5]. Castellucci and MacKenzie further explored the realm of accelerometer supported text input, with their UniGest technique that relies on gestures approximating actual characters [33]. The predicted level of performance for UniGest is 27.9 wpm, although this has not been validated.

It is clear that designing an easy-to-learn and efficient means of text input on small devices is not easy. Few systems come even close to being comparable to handwriting speed, let alone touch typing speed.

## 5.2 Limiting the Design Space

It is not immediately obvious how text input techniques can fit into the body-centric design framework. Body-centric interaction stresses physical world metaphors, leveraging reality-based properties and the capabilities of the human body. Text, however, is symbolic rather than physical. Language itself is abstract, a means of communication and a representation for knowledge. Language includes both spoken words and written words. The written word can take many forms, either a hand-written form or a printed form using one or more of any number of different fonts. The significance is that text is an abstract representation, and has no canonical physical-world equivalent. Nevertheless, any design for text input on large displays will need to take into account factors that are closely related to body-centric interaction if text input techniques are to be easily integrated into a comprehensive body-centric interaction framework.

The particularities of use cases for large displays must be considered when designing text input techniques. First, we must identify the contexts in which this type of text input might take place. Large wall displays are commonly used for giving presentations [126], for supporting brainstorming [36], and for supporting casual interactions in public spaces [192], but they are rarely used for long-term uninterrupted creation of documents or other data artifacts. This means that an appropriate text input technique should allow for easy transition between text input and other interaction, but it does not necessarily need to support entry of large blocks of text. It is also the case that while existing interaction techniques often force users of large wall display systems to stand within physical reach of the display, it is desirable to allow freedom of motion in the work environment, including outside of arms' reach of the display. We are especially interested in techniques that allow for interaction using the hands while not within physical reach of the display. We refer to these as “mid-air” techniques. Examples of such techniques include Soap [11], XWand [214], and VisionWand [25].

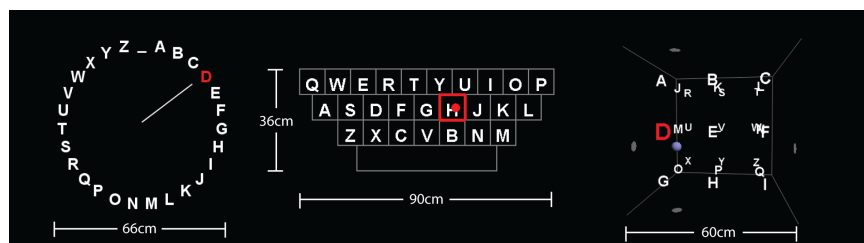
We limited ourselves to techniques that employ feedback on the large display, but do not need to take up a large portion of the display. In the interest of supporting collaborative awareness, we wanted to show enough feedback on the display that the collaborators of the typing user would understand what that user was doing.

A contrasting approach could involve a user typing on a small handheld device with limited feedback on the large display. This is a valid approach, but we chose not to take it due to the limited collaborative feedback provided, as well as the requirement that the typing user focus on a small isolated device, rather than the shared context. At the same time, however, we wanted to limit the size of the typing feedback on the display. We felt it was important that typing feedback take up a relatively small region on the display, so that collaborators could continue on with parallel work.

Two properties of mid-air input techniques become relevant once it is recognized that users have freedom of motion in the space. The first property is *distance-dependence*. The physical action required to perform a *distance-dependent* input changes as the distance between the user and display changes, whereas the action of a *distance-independent* technique is invariant with distance. As an example, pointing using ray-casting is *distance-dependent*: as the user moves farther from the display, the user's motions are magnified on the display surface. We hypothesize that large displays will benefit from the development of *distance-independent* techniques, because these techniques will not constrain the motions of the users within space due to physical input requirements varying with distance.

The second property we identify is *visibility-dependence*. A *visibility-dependent* technique requires that the user refer to visible feedback during use, whereas a *visibility-independent* technique does not. For example, touch-typing is *visibility-independent* due to haptic feedback, but input on touch screens (such as the iPhone) is usually not, because the user must confirm actions by looking at the display. We hypothesize that large display use will benefit from the development of *visibility-independent* techniques because these will allow the user to focus on the data being manipulated, rather than on the mechanics of interaction.

The remainder of this chapter describes our exploration of the design space of mid-air text input techniques for large wall displays, with a special emphasis on *distance-dependence*.



**Figure 5.1:** The three text input techniques as used in Experiment 1. From left to right: Circle, QWERTY, and Cube. Dimensions refer to the size of the feedback during the experiment.

### 5.3 Candidate Interaction Techniques

We designed three candidate interaction techniques for allowing mid-air text input on very large wall displays (Figure 5.1). These techniques were designed taking into consideration the properties of *distance-dependence* and *visibility-dependence*. They sample different combinations of these properties (Table 5.1). The QWERTY technique is distance- and visibility-dependent, the Circle technique is distance-independent but visibility-dependent, and the Cube technique is both visibility- and distance-independent. The fourth possibility, distance-independent but visibility-dependent, does not seem to be relevant because visibility dependence probably always implies at least some degree of distance dependence.

We designed our techniques to use only a limited region of the display for feedback. Because displays are often used collaboratively, it is undesirable for an interaction technique to monopolize large regions of the display for the purpose of supporting the interactions of just a single user. Thus, we assumed that in real scenarios the majority of the display would be used for shared content, and only a limited region would be used for feedback related to text entry.

As a starting point for designing our techniques, we assumed that there is some way for the user to specify 2D locations on the display by pointing, some way to determine the 3D location of the user's hand, and some way to trigger events. There are potentially many ways of doing this. For our implementation we made use of Nintendo Wii Remote devices, but any number of other devices could also be used. It would also be possible to use bare hands, if the hands were able to be sensed to a degree of accuracy where individual fingers could be identified in order

	Distance-dependent	Distance-independent
Visibility-dependent	<i>QWERTY keyboard</i> Laser pointer Put-that-there [17]	<i>Circle keyboard</i> Soap [11]
Visibility-independent		<i>Cube keyboard</i> Body Tools (chapter 4) Virtual shelves [127]

**Table 5.1:** Design space matrix of distance- and visibility-dependence, with some representative techniques. Emphasized techniques are evaluated in this chapter. One cell in the matrix is empty due to a lack of reasonable representative techniques.

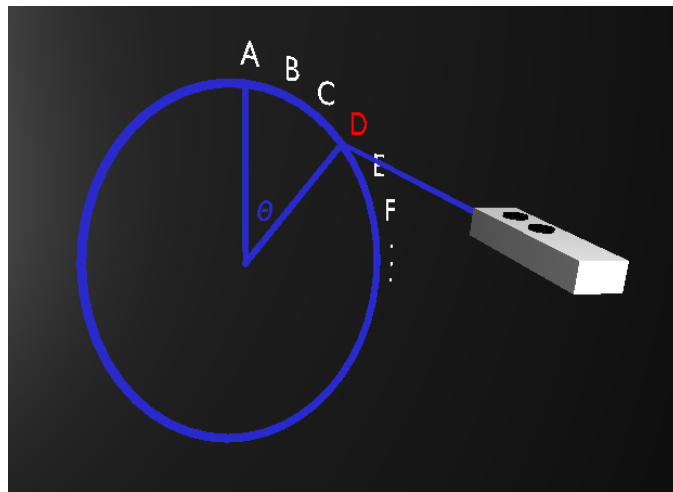
to trigger events. There is existing work that indicates we are close to being able to accomplish this [204]. Our results should have applicability when this capability is fully available.

### 5.3.1 QWERTY Keyboard

The QWERTY technique makes use of the familiar QWERTY key layout, and operates through a simple ray-casting metaphor. The visual feedback displays a standard keyboard layout and a dot cursor (Figure 5.1). The user controls the cursor by pointing the hand-held device at the display, and selects a character by hovering with the cursor over the appropriate key and then pressing a button.

The QWERTY technique is distance-dependent. As a user moves farther away from the display, the motion of the cursor is magnified, and thus the size of an individual button shrinks in motor space. The technique is also visibility-dependent, as the user almost certainly requires visual feedback during the character selection in order to aim the cursor at individual keys.

We hypothesized that while this technique will benefit from familiarity, the fact that it is both visibility- and distance-dependent will render its utility limited in the context of very large wall displays.



**Figure 5.2:** Selection of a character in the Circle technique is based on the point of intersection of a ray cast from the input device. The angle of the intersection point relative to the origin determines the selected character.

### 5.3.2 Circle Keyboard

In the Circle technique, letters of the alphabet are shown in a circular arrangement (Figure 5.1). A pointer line radiating from the center of the circle indicates the currently highlighted letter. Using the handheld device, the user moves the pointer to highlight the desired character before pressing a button to select the character. The angle of the pointer is defined by intersecting a ray cast from the handheld device with the display surface (Figure 5.2). The pointer line radiates from the center of the circle towards the point of intersection. This approach allows the user to move from one side of the circle to the other with relatively small arm motions. A small rotation about the wrist can cause the pointer to move quite dramatically.

Because character selection is defined by angle, rather than the absolute position of the ray-cast pointer on the display, input response is invariant with user distance from the display. Thus the technique is distance-independent. On the other hand, the technique is also visibility-dependent. There is a relatively small angle ( $13\frac{1}{3}$  degrees, for 26 letters plus space) from the whole circle of 360 degrees dedicated to each letter of the alphabet, so accurately selecting one character without



visual feedback is not practical.

Inspiration for the technique came both from a technique developed for touch wheel input [170], and from a similar technique used in the Nintendo Wii Game “Super Monkey Ball.” This prior work, plus the property of being distance-dependent, indicated the Circle technique would be worth investigating.

We hypothesized that this technique will benefit over the traditional QWERTY technique due to its distance-independence, although it may suffer in actual use due to its novelty.

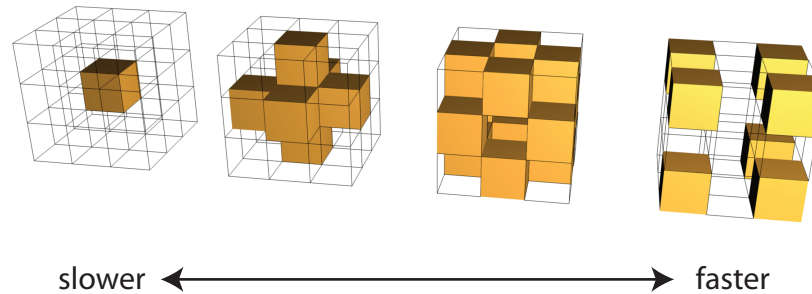
### **5.3.3 Cube Keyboard**

Visual feedback for the Cube technique is a 3D cube, subdivided into a  $3 \times 3 \times 3$  matrix of sub-cubes, within which are displayed the 26 letters of the English language and the space character (Figure 5.1). A dot cursor is also shown inside the cube. Movement of the hand-held device in 3D space maps directly to the 3D motion of the cursor within the cube. When the user moves the cursor into a character sub-cube, that character is highlighted in red. To input the highlighted character, the user presses a button.

The larger cube walls are “hard” in the sense that a ballistic movement of the controller in the direction of the desired character will cause the cursor to “stick” to the side of the larger cube. Such an impenetrable border results in a reduced Fitts’ index of difficulty and enhanced performance, as described by Walker and Smelcer [207]. Of course this benefit is only enjoyed by sub-cubes that border the outer faces of the larger cube.

We hypothesized that the corner sub-cubes, with three hard sides, will be easiest to hit, while the center sub-cube, with no hard sides, will be hardest to hit (Figure 5.3).

In our design of the Cube technique we went through several iterations. A problem specific to the technique emerged that had to be dealt with. This problem was that it was difficult for users to perceive the depth of the cube, both in terms of understanding which layer a desired character was located in, and where the current cursor was located. We made several attempts to address this problem. We first attempted to draw either opaque outlines for each sub-cube, or semi-transparent



**Figure 5.3:** Hypothesized relative performance of selecting sub-cubes using the Cube technique, based on number of impenetrable sub-cube sides.

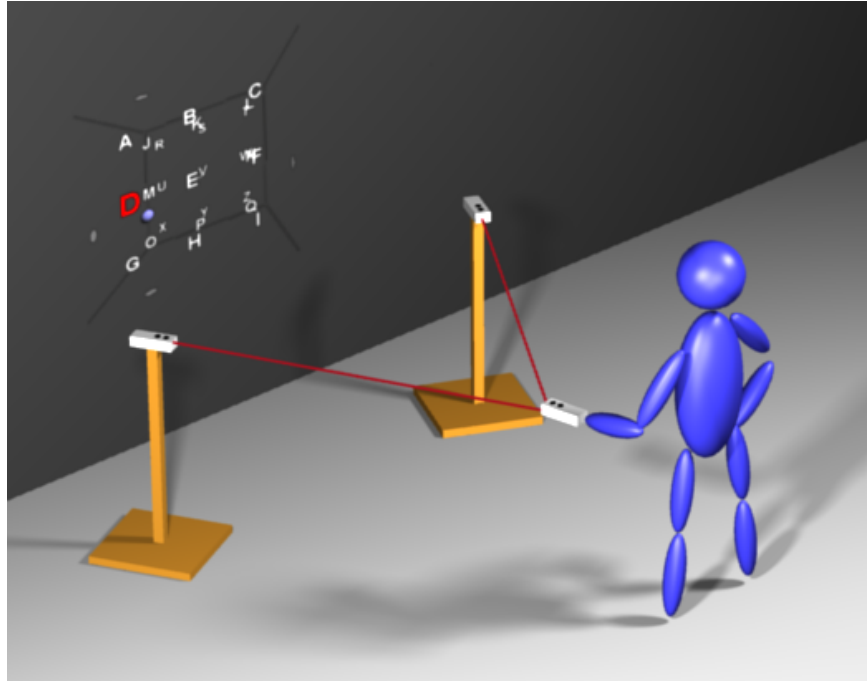
cubes for each sub-cube. We felt the cues of either of these shapes would help the user determine depth, but the feedback from both approaches was deemed to be too “busy.” Instead we provided lines outlining the single high-level cube, and then caused the cube to rotate as the cursor moved within it. The cube would rotate in a direction opposite to the cursor, allowing the user to employ parallax to identify where either the cursor or desired letters were located. The combinations of a sparse layout and active animated feedback appeared to be the best combination of the options explored.

## 5.4 Implementation

The three techniques have different sensing requirements in order to function. The QWERTY technique requires that knowledge of the absolute point of intersection of a ray with the display be determined. The Circle technique requires that the orientation of pointing relative to the display be determined. The Cube technique requires knowledge of 3D location in mid-air.

We were able to fill the sensing requirements of the QWERTY and Circle techniques using standard Nintendo Wii Remote devices. The infrared camera on the front of the device identifies the location of an infrared light source (provided by an LED) placed directly in front of the display. The location of this light source in the frame of the camera determines the pointing location of the device.

The Cube technique required the modification of a Wii Remote, because di-



**Figure 5.4:** Triangulating position of hand-held Wii Remote using 2 fixed Wii Remotes on stands. Red lines indicate vectors from detected IR light source to fixed Wii Remotes.

rection information from a single infrared camera is inadequate to produce 3D location information, and the accelerometers don't produce accurate enough information. To make the Cube technique possible, we modified the hand-held Wii Remote to act as an infrared emitter by soldering an infrared LED to its board and drilling a hole in the front of the device out of which the LED protruded. We then used two additional fixed Wii Remotes on stands to triangulate the position of the handheld Wii Remote in 3D (Figure 5.4). Using the known position, orientation, and field-of-view of the two sensing Wii Remotes, and the location of the hand-held Wii Remotes' LED in the field-of-view of the fixed cameras, we calculate the 3D position of the hand-held Wii Remote.

It is worth noting that the implementation of these techniques would be substantially easier if it were supported by a body-centric interaction architecture, as first introduced in Chapter 4, and later elaborated upon in Chapter 6.

## 5.5 Experiment 1: Exploring Mid-Air Techniques

We conducted a controlled experiment to compare performance of the QWERTY, Circle, and Cube techniques for text input on a very large wall display. By evaluating three techniques that provide contrasting points in the design space, this study provides a better understanding of mid-air input techniques in general and the properties of distance-dependence and visibility-dependence in particular.

### 5.5.1 Methodology

We followed a standard laboratory approach for the controlled experiment as an initial exploration of the design space.

#### Conditions

The experimental conditions were QWERTY, Circle, and Cube, as described in the prior section.

#### Task and Apparatus

The experimental task was to enter a set of English phrases as quickly and accurately as possible. Target phrases were shown one at a time above the text input feedback mechanism. As each character was entered correctly it appeared under the target phrase to provide visual feedback. Participants had to correctly enter each character before continuing on to the next character. Errors in input caused the Wii Remote to vibrate.

The phrase set was a randomly ordered version of that used by MacKenzie and Soukoreff [134]. The task was based closely on that used by Wigdor and Balakrishnan [212]. The same ordering of phrases was used in each session. Thus, each participant typed the same phrases, in the same order, for each of the three experimental conditions.

The experimental room contained a very large wall display that was approximately 4.9m×2.4m (16' × 9') in size. Only a small portion of the display was used for the text entry task, to simulate an isolated operation in a collaborative environment. Participants stood 2.44m (8') from the display. For the QWERTY and Circle conditions an infrared light source was placed in front of the display to support

the pointing functionality of the hand-held Wii Remote. For the Cube condition, two Wii Remotes on stands were used to measure the 3D position of the modified user-held Wii Remote.

The software, written in Java, ran on a Microsoft Windows XP computer. The software managed all interactive components and logged all timing and error data. Bluetooth support was developed using a combination of the BlueCove implementation of the Java JSR-82 specification, a WIDCOMM Bluetooth stack, and custom Wii Remote communications code.

### **Procedure**

Each experimental condition for a particular participant was administered on a different day, in a separate one-hour session. For each session, the participant completed as many task blocks as possible in 50 minutes, where a task block consisted of 10 predefined phrases from the larger phrase set. During 3-minute breaks between blocks, the participant sat at a table and completed a puzzle-building distractor task. The distractor task provided a mental and physical break from the primary task. This is consistent with real-world large wall display use, where it is unlikely that there will be lengthy, uninterrupted text entry. After each session, the participant completed a questionnaire for that condition.

At the beginning of the first session the participant was given a pre-questionnaire to collect demographic information. At the end of the third session, the participant completed a post-questionnaire that asked for rankings of the techniques, and comments.

### **Participants and Experimental Design**

Twelve participants (three female) were recruited through on-campus advertising. All participants were right-handed, although handedness was not a criterion for selection, and all were regular computer users (4+ hours weekly). Although a firm command of English was required of all participants, degree of fluency varied. When asked how long they had lived in English speaking countries, answers ranged from 1.5 years to 31 years (whole life).

The design was a single-factor within-subjects design. Order of presentation

was fully counterbalanced across subjects.

## Measures

Performance was measured using the standard words-per-minute metric, calculated as  $60 \times (|T| - 1) / (5 \times S)$ , where  $|T|$  is the number of characters in string  $T$ , and  $S$  is the completion time in seconds [135]. Because users had to correctly enter a character before moving on to the next one, speed contained an implicit error penalty. For completeness, however, we also calculated error rates as the percentage of all character events that were errors.

The pre-questionnaire collected demographic information and computer experience. Questionnaires for each condition collected preference data using a 5-point Likert scale based on the NASA Task Load Index [83], as well as comments from participants. Finally, a post-questionnaire collected comparative rankings on overall preference, speed and difficulty, as well as additional qualitative comments.

## Hypotheses

Experiment 1 was largely exploratory. We were primarily interested in the relative performance of the three techniques, but did not have specific hypotheses regarding this. We expected results to aid in determining the usefulness of mid-air text input techniques in general, and to help gauge the importance of distance-dependence and visibility-dependence as design factors.

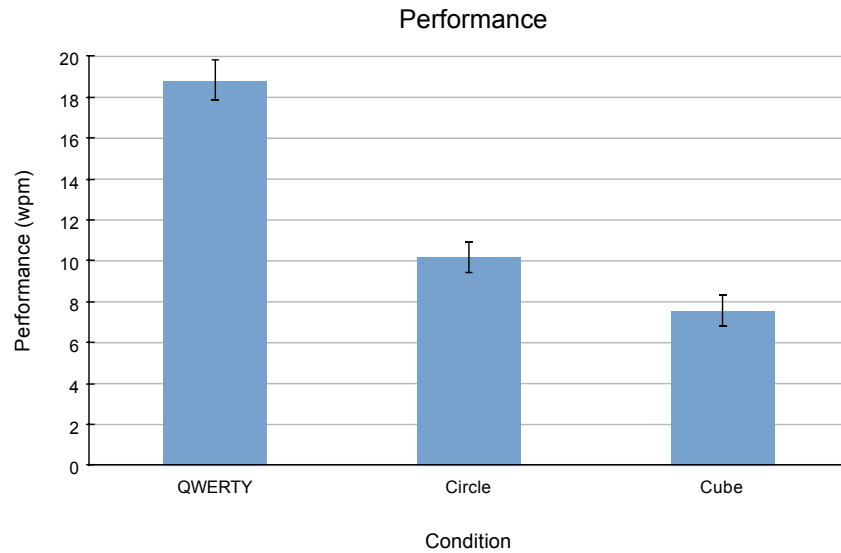
We did have two hypotheses specific to the Cube technique. These relate to the relative performance of selecting different kinds of sub-cubes in the larger cube:

**H1** Sub-cubes with more hard faces will be selected faster than sub-cubes with fewer hard faces.

**H2** Sub-cubes in layers closer to the user will be selected faster than sub-cubes in layers further away from the user.

### 5.5.2 Results

We ran a repeated measures ANOVA on the dependent variables of speed and errors. A Bonferroni adjustment was applied to all pairwise comparisons.



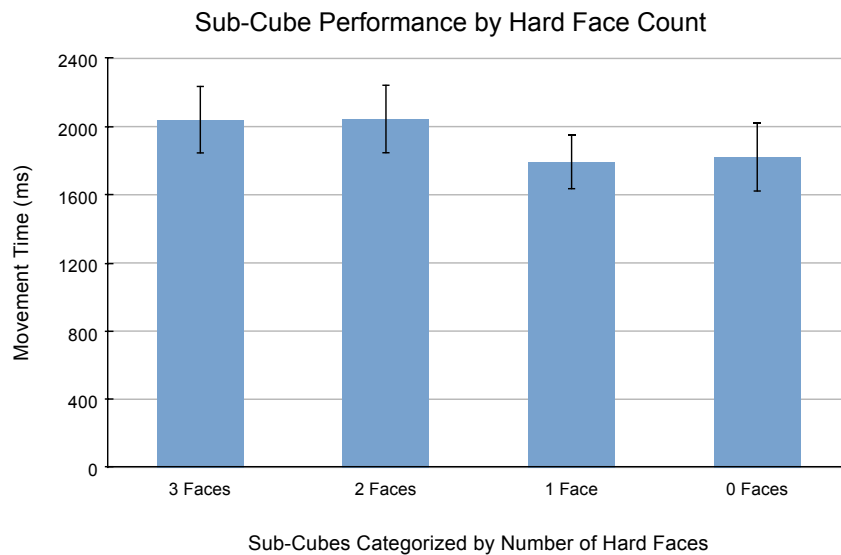
**Figure 5.5:** Mean input speed in words-per-minute for the three text input techniques. Error bars represent standard error.  $N = 12$ .

### Performance

As shown in Figure 5.5, the average input speed in words-per-minute was 18.9 for QWERTY, 10.2 for Circle, and 7.6 for Cube. A one-way repeated-measures ANOVA showed a significant main effect of technique ( $F_{2,22} = 291.556, p < 0.001$ ). We ran pairwise comparisons to compare between techniques. QWERTY was faster than both Circle ( $p < 0.001$ ) and Cube ( $p < 0.001$ ). Circle was also faster than Cube ( $p = 0.001$ ).

We were interested in the relative performance of pointing to the different sub-cubes in the Cube condition. We performed an ANOVA on average times of users pointing to sub-cubes with three, two, one, and zero hard faces (Figure 5.3). A summary of results is shown in Figure 5.6. It was found that there was a significant effect of number of hard faces to performance ( $F_{3,33} = 5.669, p = 0.003$ ). Significant pairwise comparisons are shown in Table 5.2.

To further investigate the role of sub-cube positions in movement time, we examined sub-cubes of different depths within the cube. Sub-cubes were either in the front layer, the middle layer, or the back layer. A repeated-measures ANOVA



**Figure 5.6:** Mean time for users to point to sub-cubes with three, two, one, and zero hard faces. Error bars represent standard error.  $N = 12$ .

# of side pairs	significance
zero-two	$p = 0.014$
zero-three	$p = 0.033$
one-two	$p = 0.007$
one-three	$p = 0.023$

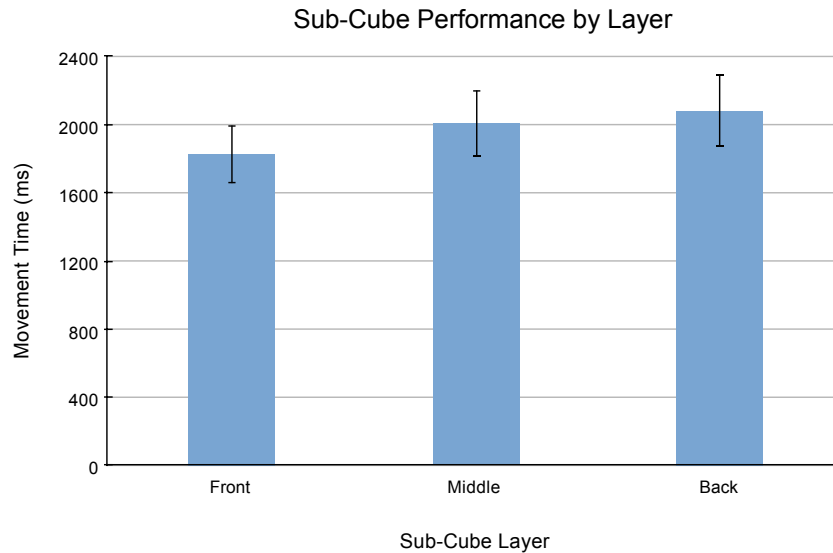
**Table 5.2:** Summary of significant pairwise comparisons for movement time to sub-cubes with different number of hard sides.

found a significant effect of layer on movement time ( $F_{2,22} = 6.017, p = 0.008$ ). Pairwise comparisons found that the front layer was faster than the middle layer ( $p = 0.021$ ), and was also faster than the back layer ( $p = 0.021$ ). Mean movement times to the three layers are shown in Figure 5.7.

### Error Rates

Mean error rates by condition were 2.4% for QWERTY, 6.3% for Circle, and 7.0% for Cube (Figure 5.8). A one-way repeated-measures ANOVA showed that technique significantly impacted error rate ( $F_{2,22} = 55.590, p < 0.001$ ). Pairwise com-





**Figure 5.7:** Mean time for users to point to sub-cubes in the front, middle and back layers. Error bars represent standard error.  $N = 12$ .

parisons showed that participants made fewer errors with QWERTY than with either Circle ( $p = 0.009$ ) or Cube ( $p < 0.001$ ).

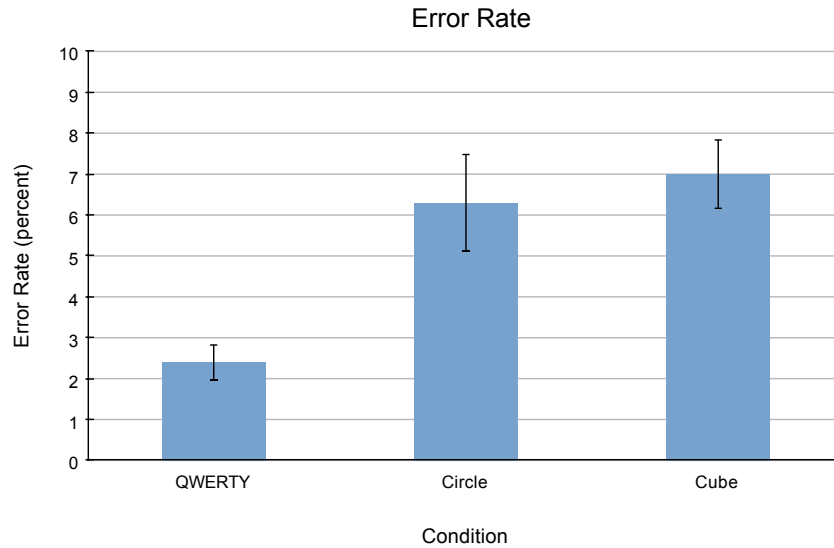
### Subjective Measures

A summary of results from participants' subjective ratings of the three conditions is shown in Figure 5.9. Results were fairly consistent across perceived speed, difficulty, and overall preference. Users found the QWERTY technique to be the easiest to use, followed by the Circle and then the Cube technique.

Results from the post-questionnaire, asking for rankings on speed, difficulty, and overall preference, are shown in Figure 5.10. It was clear that the QWERTY technique was favoured over the other two techniques.

### Comments

Free-form written comments provided important insight into the different techniques. The most consistent feedback stated a preference for the QWERTY technique. Other interesting comments included the following:



**Figure 5.8:** Mean error rates for the three text input techniques. Error bars represent standard error.  $N = 12$ .

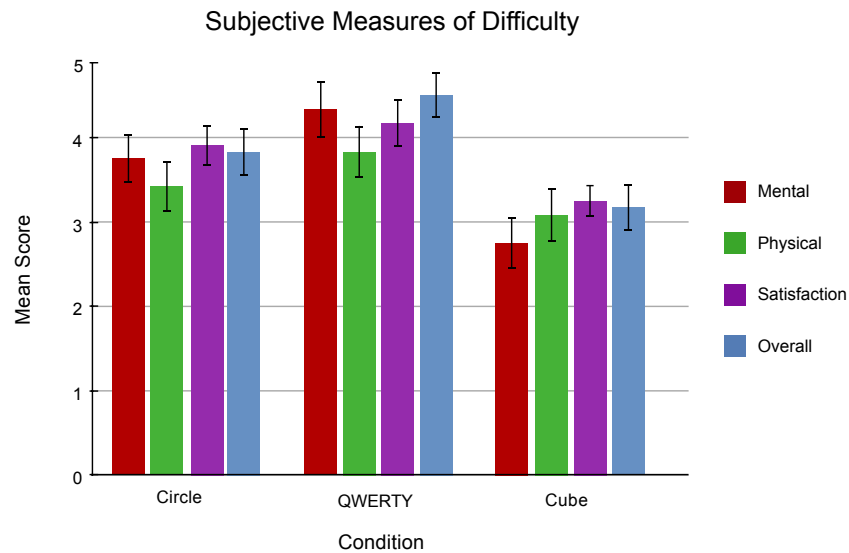
- “My ranking may be biased towards [the] QWERTY Keyboard model as I am usual [sic] to its use in daily life.”

This comment reveals an awareness of the biasing effect that familiarity with standard keyboards may have had on the user’s performance. This is a potential confound, which we discuss later.

The Circle technique garnered a combination of positive and negative comments:

- “Rotation alone was easier to manage than [rotation] + translation”
- “Seemed to require more accuracy than QWERTY technique”
- “I think if the sensor area was bigger, it would be easier”

The first comment suggests that the approach had potential, compared to the QWERTY technique. Unfortunately, the second comment confirms our fear that the angular accuracy required to select an individual character was a problem. The last comment revealed an unexpected shortcoming in our implementation of the



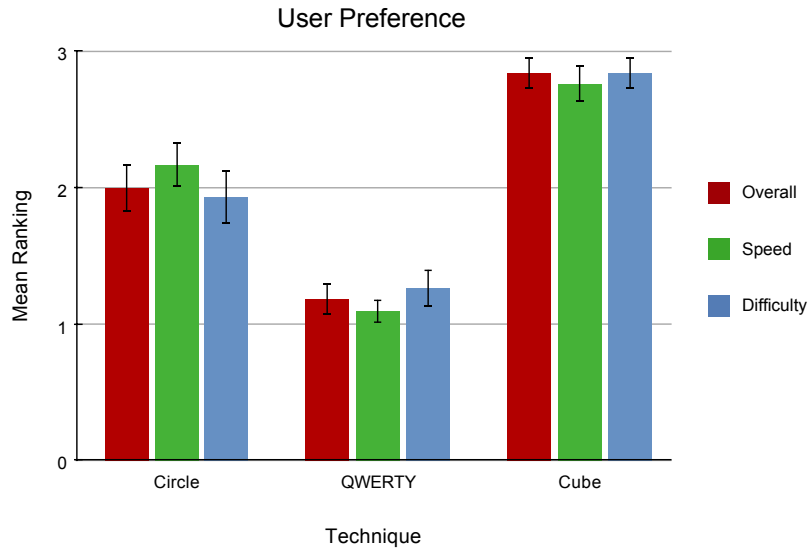
**Figure 5.9:** Mean scores for the three text input techniques from a NASA TLX based questionnaire. Ratings are on a scale of one to five (longer bars are better). Error bars represent standard error.  $N = 12$ .

technique. The field of view of the Wii Remote IR camera is limited, and several users reported problems caused by the Wii Remote losing sight of the IR light source.

Despite the relatively poor performance of the Cube technique, several users had positive comments:

1. “The Cube keyboard could be a great input method if some modifications were made...”
2. “...it probably has the most potential for speedup of all the methods...”
3. “smallest range of motion/potentially fastest method”

These comments indicate that participants saw value in the technique, but that it needs further design iterations, and may be difficult to learn.



**Figure 5.10:** Mean scores from user rankings of the three techniques from best to worst (1=best, 3=worst, shorter bars are better). Error bars represent standard error.  $N = 12$ .

### 5.5.3 Discussion

Each of the three text entry techniques performed differently. We discuss each in turn.

#### QWERTY

The QWERTY technique was significantly faster and had fewer errors than either the Circle or the Cube techniques. QWERTY performance of 18.9 wpm is competitive with both handwriting and pen-based typing, but is still well short of touch-typing. It is also convenient that the QWERTY technique can easily be adapted to work for both mid-air input and touch input, depending on the location of the user relative to the display. These results, combined with the positive subjective feedback on QWERTY, suggest that adapting the traditional QWERTY keyboard layout for mid-air interaction is a viable strategy for supporting text input with large wall display systems.

## Circle

The performance results of the Circle technique were less encouraging than those of the QWERTY technique, with input speeds averaging 10.2 wpm. This is low enough that it does not appear to be competitive with the traditional pen-based typing or touch-typing techniques. It maintains the advantage, however, of being practical while standing without any supporting surfaces. There are reasons to further investigate the Circle technique, however. Experiment 1 was largely exploratory, and did not isolate the factors of *distance-dependence* or *visibility-dependence*. We hypothesize that the Circle technique will hold benefits that become apparent when it is evaluated at a number of distances.

## Cube

The performance results for the Cube technique were the least encouraging, with mean input speeds of 7.6 wpm. This is slower than both of the other techniques that were evaluated, and well short of traditional techniques. We believe that this poor performance is due to initial difficulties in learning the technique, mostly because of difficulties learning three-dimensional key layouts and the three-dimensional gestures used to select the sub-cubes. It appears that none of the participants were able to enter the terminal stage of Fitts' pointing tasks, where they are able to move immediately to the desired target without thinking about what to do next. This hypothesis is supported by some of the comments made by users. We believe that the Cube technique holds potential, due in part to it being *visibility-dependent* and *distance-dependent*, but that this potential can only be revealed with a much longer-term evaluation. In addition, we believe this technique could hold potential when applied to other device contexts, specifically text entry on cell phones and other portable devices. The difficulty in supporting text entry on these devices is well documented, and a gestural technique such as the Cube technique could address these issues.

We were also able to reach conclusions regarding the Cube-specific hypotheses. We summarize these according to our hypotheses:

**H1** Sub-cubes with more hard faces will be selected faster than sub-cubes with fewer hard faces. *Not Supported.*

**H2** Sub-cubes in layers closer to the user will be selected faster than sub-cubes in layers further away from the user. *Supported*

A significant effect of hard face count was found on performance, but the impact was opposite of what we hypothesized. Sub-cubes with two and three hard faces were significantly slower to select than sub-cubes with one or zero hard faces. It is unclear why this is. One possibility is that the sub-cubes with more hard faces are further from the center of the cube. This could result in a larger distance to travel and longer selection time. The longer distance to travel may contribute more to movement time than the hard surfaces of the sub-cubes. Unfortunately there is no established model of 3D pointing performance that can be applied to analyze this task. Grossman and Balakrishnan [73] developed a trivariate model of pointing performance, but it did not incorporate hard faces and 2D visualizations of 3D manipulations.

A significant effect of sub-cube layer was also found on performance, with sub-cubes in the front layer faster to select than sub-cubes in other layers. We believe this to be due to users' reliance on visual feedback during the task. We hypothesize that if participants used the technique long enough they would no longer rely on visual feedback (*visibility-independence*) and the difference in layer performance would disappear.

## Summary

It is not surprising that the QWERTY technique produced the best results, but it is perhaps unfortunate. Any experiment comparing novel text input techniques to touch-typing and QWERTY-based key layouts faces the challenge, and possibly damaging confound, of comparing a previously unseen technique with the single dominant form of text input for the English language. It is nearly unheard of for new input techniques to best QWERTY in performance, despite substantial evidence that QWERTY is far from optimal. In order to perform a definitive evaluation of any new text input technique, a longitudinal evaluation is required, and that is beyond the scope of this work.

Based on the results of Experiment 1, we decided to further investigate the property of *distance-dependence*. We chose to do this through a further evaluation

of the QWERTY and Circle techniques. Cube was left out due to poor performance and the apparent difficulties for users in learning the technique.

## 5.6 Experiment 2: Investigating Distance Independence

Experiment 1 provided us with insight into performance for QWERTY, Circle and Cube text input techniques. However, it was not designed to isolate either visibility-dependence or distance-dependence as factors, due to wider differences between the three techniques. The goal of Experiment 2 was to determine how performance of Circle, hypothesized to be distance-independent, and QWERTY, hypothesized to be distance-dependent, differ as a user's distance from the display increases. An additional limitation of Experiment 1 was that previous experience with QWERTY keyboard layouts likely biased results in favour of QWERTY. Because of this, we modified the task to be a targeting task, simulating the terminal Fitts' stage of pointing in character entry.

The Cube technique was omitted from Experiment 2 due to the observed difficulties in learning the input technique. While we think it would be useful to evaluate in a longitudinal study, a shorter study would not be enough to properly evaluate the technique.

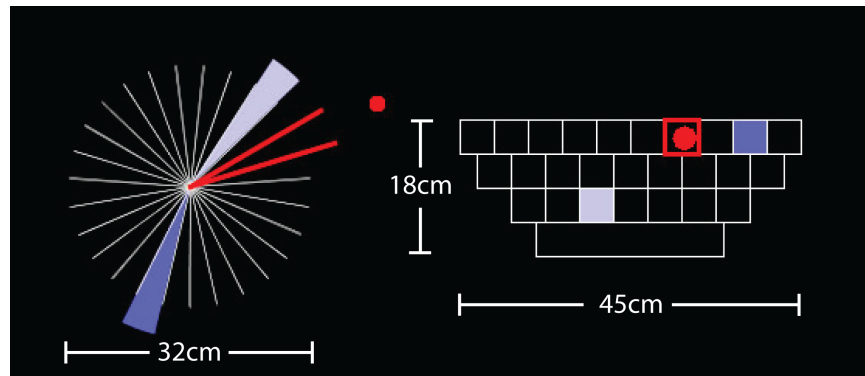
### 5.6.1 Methodology

The experimental setup was largely similar to that of Experiment 1. We highlight only the differences here.

#### Conditions

The first factor was input *technique*. We tested the QWERTY and Circle techniques from Experiment 1, with the differences that the “keys” were unlabelled for both techniques, and target keys were instead highlighted in white (Figure 5.11). A few minor graphical changes for the Circle technique were introduced, based on user feedback obtained in Experiment 1.

The second factor was *distance*. Participants interacted while standing either 2.74m (9') or 5.49m (18') from the large display.



**Figure 5.11:** The Circle keyboard (left) and QWERTY keyboard (right) interfaces as used in Experiment 2.

### Task and Procedure

The experimental task was to press highlighted keys on a blank (unlabelled) virtual keyboard as quickly and accurately as possible. Two keys were always highlighted: a white key indicated the current target, and a blue key indicated the next target. The purpose of the blue key was to allow the participants to plan ahead. The sequence of highlighted keys corresponded to the same input phrases as used in Experiment 1, although participants were not aware of the phrases because the keys were unlabelled.

The experiment was designed to fit in a single one-hour session. For each condition, participants completed four task blocks of 75 character input events, for a total of 300 character inputs for each condition. Between blocks was a 20 second break. Between conditions, participants sat at a table to fill out a questionnaire and work on a distractor task for a combined length of five minutes.

### Participants, Measures, and Experimental Design

Sixteen new participants (eight female) were recruited through on-campus advertising. Fifteen of the participants were right-handed, and all used their dominant hand throughout the experiment. All but two participants were regular computer users (4+ hours weekly). Participants were compensated \$10 for participating, and the fastest 50% of participants received an extra \$10.



Performance and error data were collected in the same manner as for Experiment 1. Participants again filled out pre-questionnaires, questionnaires for each condition, and a post-questionnaire.

The experiment was a  $2 \times 2$  within-subjects experimental design. The factors were technique (QWERTY and Circle) and distance (9 feet and 18 feet). Order of presentation was counterbalanced using a Latin square for the four combinations of technique and distance.

### **Hypotheses**

The following hypotheses are motivated by the discussion in the previous section on the impact of distance-dependence:

#### **H1 Performance and Error Rates**

1. Relative to Circle, QWERTY performance will decrease as distance increases.
2. Relative to Circle, QWERTY error rates will increase as distance increases.
3. Circle performance will not change with distance.

#### **H2 Preference**

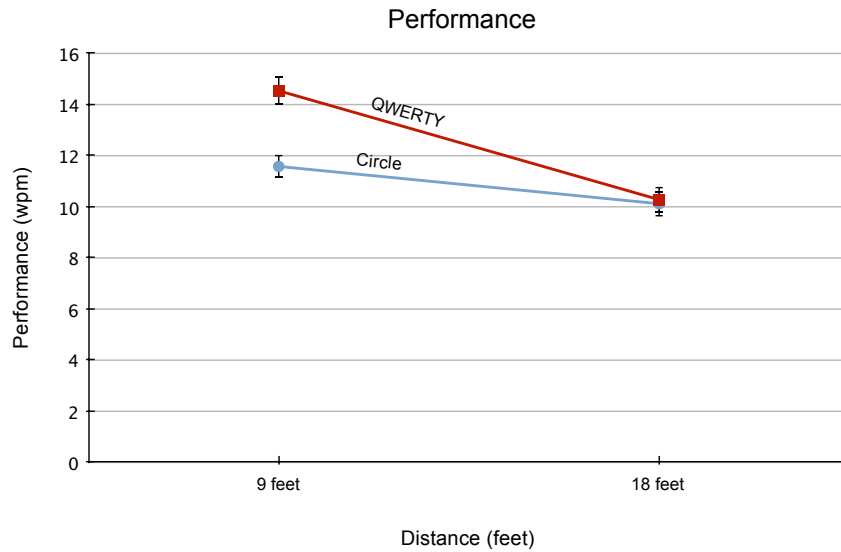
1. Circle will be rated better relative to QWERTY at the larger distance than the shorter one.

### **5.6.2 Results**

We ran a  $2 \times 2$  repeated measures ANOVA (*technique*  $\times$  *distance*) on each of the main dependent variables of speed and error rate. A Bonferroni adjustment was applied to all pairwise comparisons.

#### **Performance**

The average input speed for QWERTY was 14.5 wpm at 9 feet and 10.3 wpm at 18 feet, and for Circle it was 11.6 wpm at 9 feet and 10.1 wpm at 18 feet



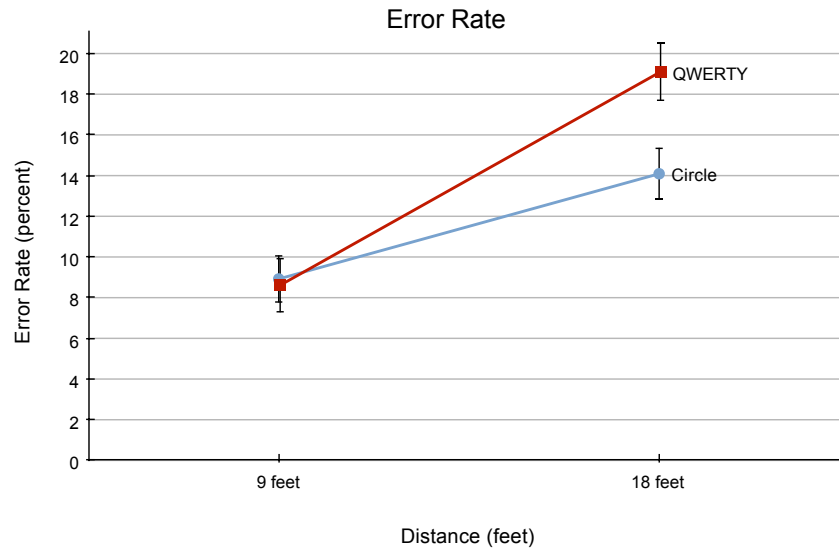
**Figure 5.12:** Performance for the four text input conditions in words per minute. Error bars represent standard error.  $N = 16$ .

(Figure 5.12). An ANOVA showed significant main effects of technique ( $F_{1,15} = 62.748, p < 0.001$ ) and distance ( $F_{1,15} = 244.573, p < 0.001$ ). A significant interaction of technique  $\times$  distance ( $F_{1,15} = 12.935$ ) was also found.

To understand how distance and input type impacted performance differently, we conducted post-hoc pairwise comparisons on the interaction effect. Distance had a significant impact on both QWERTY performance ( $p < 0.001$ ) and on Circle performance ( $p = 0.002$ ). At 9 feet, QWERTY performance was faster than Circle ( $p < 0.001$ ), but performance with QWERTY also degraded more quickly with distance and there was no difference found between the two techniques at 18 feet.

### Error Rate

Mean error rates by condition were calculated as the percentage of all character entry events that were incorrect (Figure 5.13). The average error rate for the QWERTY technique was 8.5% at 9 feet and 19.0% at 18 feet, and for the Circle technique was 8.9% at 9 feet and 14.1% at 18 feet. An ANOVA showed significant main effects of technique ( $F_{1,15} = 11.792, p = 0.004$ ) and distance ( $F_{1,15} =$



**Figure 5.13:** Mean error rates (percentage) for the four text input conditions. Error bars represent standard error.  $N = 16$ .

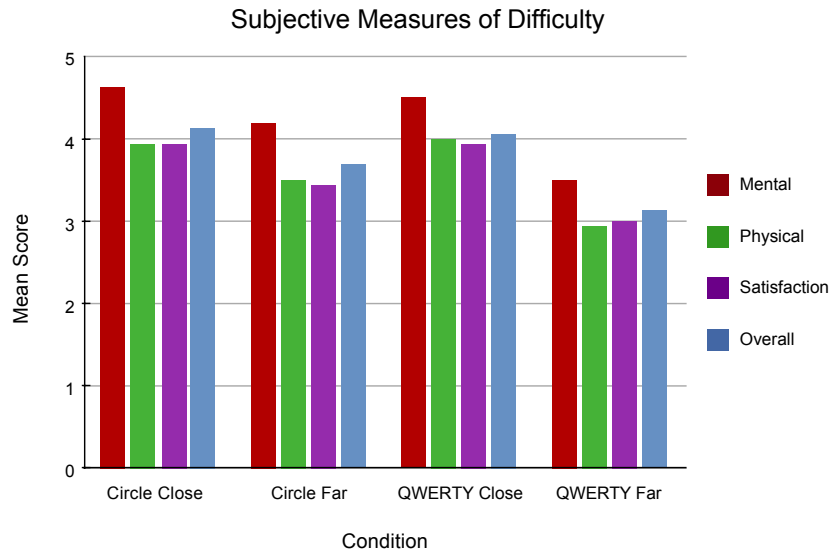
78.026,  $p < 0.001$ ). A significant interaction of technique  $\times$  distance was also found ( $F_{1,15} = 18.766$ ,  $p = 0.001$ ).

To understand how distance and input type impacted error rates differently, we conducted post-hoc pairwise comparisons on the interaction effect. Distance had a significant impact on error rates for both QWERTY ( $p < 0.001$ ) and Circle ( $p < 0.001$ ). These results showed the relative degradation of QWERTY as distance increases: at 9 feet there was no difference in error rate between Circle and QWERTY, but at 18 feet QWERTY had more errors than Circle ( $p < 0.001$ ).

### Subjective Measures

Questionnaires administered after each condition collected subjective ratings and comments. Responses to the TLX-based Likert questions are summarized in Figure 5.14.

Participants ranked the four conditions from best to worst in terms of perceived speed, difficulty, and overall preference (Figure 5.15). A Friedman test showed that technique significantly impacted rankings on all measures, including speed



**Figure 5.14:** Mean scores for the four text input conditions using a NASA TLX based questionnaire. Ratings are on a scale of one to five (longer bars are better).  $N = 16$ .

( $\chi^2_{(3,N=16)} = 30.675, p < 0.001$ ), difficulty ( $\chi^2_{(3,N=16)} = 27.675, p < 0.001$ ), and overall preference ( $\chi^2_{(3,N=16)} = 25.350, p < 0.001$ ). Pairwise comparisons using Wilcoxon Signed Ranks Tests found no differences between QWERTY and Circle at 9 feet on any of the measures. At 18 feet, however, Circle was perceived to be faster ( $z = -2.231, p = 0.026$ ) and less difficult ( $z = -2.349, p = 0.019$ ) than QWERTY and was preferred overall ( $z = -2.018, p = 0.044$ ).

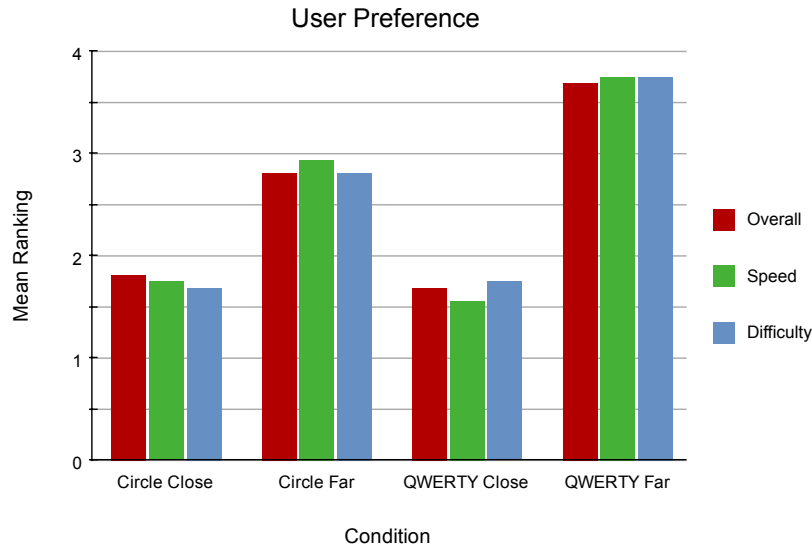
### 5.6.3 Discussion

We summarize our results according to our hypotheses:

**H1.1** Relative to Circle, QWERTY performance will decrease as distance increases.  
*Supported.*

**H1.2** Relative to Circle, QWERTY error rates will increase as distance increases.  
*Supported.*

**H1.3** Circle performance will not change with distance. *Not supported.*



**Figure 5.15:** Mean user rankings for the four text input conditions from best to worst (1 = *best*, 4 = *worst*, shorter bars are better).  $N = 16$ .

**H2.1** At the larger distance Circle will be rated better relative to QWERTY than at the shorter distance. *Supported.*

The quantitative results of Experiment 2 successfully answered a number of questions raised by Experiment 1. First, QWERTY performance was found to decrease significantly as the user moves away from the display, and is therefore distance-dependent. Contrary to our hypothesis, however, Circle performance also decreased as distance from the display increased. Thus, although the Circle technique is invariant with distance in motor space, it is not entirely distance-independent. The significant interaction showed, however, that Circle performance decreases less than QWERTY performance as distance increases, over the distances examined. In fact, we observed a crossover where Circle surpasses QWERTY performance: at 18 feet performance in wpm was no different for the two techniques, but errors were greater for QWERTY. We hypothesize that if the two techniques were compared at even greater distances, Circle would be superior to QWERTY in both measures of wpm and error rates. These results support the hypothesis that techniques possessing a degree of distance-independence, in this case the Circle in-

put technique, have value when applied to systems where users may be interacting from a variety of locations in the room.

Considering the QWERTY technique, the argument that it is unsuitable for interaction at large distances is strengthened by the error data and subjective responses from participants. The error rate for the QWERTY technique at 18 feet was the highest of all conditions. Perhaps more telling, that condition was subjectively ranked as the worst technique by almost all users. Comments related to the QWERTY 18 foot condition included: “...it wants very much concentration, mentally and physically!” and “The slight shake of the hand makes pointing a tough task.” These further highlight the limitations of distance-dependent techniques.

The second question raised by Experiment 1, whether the confound of key layout familiarity had any effect on performance of the Circle technique, appears to have been answered in the negative. While it is not appropriate to perform a statistical comparison of results between the two experiments, performance for both the labeled (Experiment 1) and unlabelled (Experiment 2) keyboards was in the range of 10–12 wpm. This suggests that in order to improve user performance, we will need to improve the fundamental design of the technique. It is unlikely that increased user familiarity with key layout over time will result in large performance gains.

## **5.7 Conclusions**

Large wall displays are well suited to interaction techniques that can be used outside of physical reach of the display, however, most research has focussed on techniques that rely on direct touching. We addressed this gap in the research by investigating the use of mid-air input techniques for the specific task of inputting text.

We developed three mid-air text input techniques: QWERTY, Circle, and Cube, which combine input using a hand-held device with visual display feedback on the large display. The techniques differ in regards to their distance- and visibility-dependence, which are design dimensions that we hypothesize are of significance to large displays. Our first experiment comparing the techniques showed that QWERTY performed significantly better than the other techniques, and that it is appro-

priate for deployment. However, there was also evidence that the Circle and Cube techniques hold promise in ways that the first experiment could not make evident. The inability of the three techniques in the first experiment to best the performance of touch-typing was unfortunate but not surprising. Touch-typing makes use of all ten fingers on a fixed device with rich tactile feedback. Developing a “mid-air” technique to match or beat touch typing will be a challenge.

A second experiment provided some answers to specific unanswered questions of the first experiment. Results showed that performance of the Circle technique degrades more gracefully than that of the QWERTY technique as distance between user and display increases, yet is not entirely distance-independent. This means that the Circle technique may be better than QWERTY for use in large rooms, such as lecture halls. A corollary of the impact of distance on performance is that pointing performance varies based on gain (as changes in distance effectively change gain), contrary to Fitts’ law. This suggests that it may be worthwhile to re-evaluate the impact of gain on pointing performance in general. More generally, this result suggests that the class of distance-independent techniques holds promise. In addition, we found that modifying the task to be one that simulates a Fitts’ task, but conceals an underlying text input task, made little difference to performance for either the QWERTY or Circle techniques.

We believe that future work should involve refining the techniques. The Circle technique’s main limitation, namely difficulty in character selection, might be dealt with using a detail-in-context approach to magnify the motor space extent of characters. It has even been shown that visual magnification without corresponding motor space magnification can improve targeting performance [38], and this may also be an applicable technique. Possible refinements of the Cube technique include improving navigation on the depth axis by varying transparency. Our results of the dependence of targeting time on sub-cube layer suggest that this is worthwhile. Alternate character layouts may need to be investigated, as well. It would be interesting to investigate the use of a Cube-like technique on mobile devices, as its visibility-independence may serve well in that context. Most importantly, a longitudinal study of Cube input performance should be undertaken. This will be a fundamental indicator of the value of the approach for full-time, expert users.

More generally, it would be worthwhile to further investigate the roles of distance-

dependence and visibility-dependence on interaction with large wall displays. As an example, the impact of distance-dependence on performance could suggest that pie menus may be superior to traditional menus for use at a distance, because pie menus are directional, and use is invariant with distance.



## Chapter 6

# Body-Centered API

In previous chapters we described a variety of body-centric interaction techniques. Developing interaction techniques of this type is typically quite onerous, with the need to integrate signals from one or more sensing technologies in a way that has often not been anticipated previously. This can require a significant amount of trial and error with different approaches before a suitable outcome is realized. In this chapter we describe a proposed architecture for supporting the development of novel, body-centric interaction techniques and applications. Our approach makes use of an abstract model of the scene, including users and displays, that the developer accesses in order to determine state. We believe this approach is superior to the status quo, and will benefit not only researchers, but also developers of commercial systems. We further believe that this approach is extensible in the sense that it robustly supports unanticipated use cases.

Numerous research toolkits exist for developing interaction techniques and applications using novel input devices and technologies. For example, PyMT supports the development of multi-touch applications [82], as does the architecture described by Echtler and Klinker [51], and also the DiamondSpin toolkit [181]. Other toolkits have been developed for supporting investigations into tangible interactions, including reacTIVISION and TUIO toolkits for tangible and tabletop integration [109], d.tools that emphasizes rapid iteration [84], Phidgets that emphasize rapid prototyping [71], DisplayObjects for tracking 3D styrofoam prototypes using a Vicon tracking system [4], and BOXES for building functioning prototypes

out of tin foil and other unusual materials [94]. For tracking of users in a space, it is possible to use toolkits provided with specific hardware platforms, such as the Vicon or PhaseSpace optical motion tracking systems, or the Polhemus magnetic tracking system. Other specialized hardware, such as the Cyberglove system [112], is used to measure detailed input from specific body-parts, such as hands. Some toolkits allow the explicit configuration of a variety of input devices [49]. Specialized toolkits have also been developed for such things as modeling proxemic interaction [47], recognizing gestures [209], supporting distributed peer-to-peer interfaces [146], and building mobile applications [108].

It is clear that there is a large heterogenous collection of toolkits for supporting the development of novel applications and interaction techniques. Some of these toolkits are specific to a particular hardware platform (e.g. Vicon). Most of these toolkits are specific to supporting some particular subset of interactive possibilities (e.g. tangibles on a table). What is true about all these toolkits is that no single toolkit is suitable to support exploration of a large set of interactive scenarios. If researchers wish to explore a novel idea, it is likely that they will have to either make use of several of these toolkits, or make their own toolkit from scratch. This is an undesirable situation that results in a proliferation of “one-off” toolkits that see very little reuse outside of their original application.

In this section we describe the Body-Centered Application Programming Interface (BAPI), which is an abstraction layer for supporting the development of novel computing applications and interaction techniques. The purpose of the BAPI is to conceal the complexity of hardware platforms and sensing technologies, and to allow the developer to focus on what is of primary interest, the users and their physical context. As Myers et al. [153] comment in their discussion of the future of interfaces, “tools will be needed that hide all of this complexity and provide an easy-to-use interface for programmers.”

Based on our experiences developing our body-centric interaction techniques, we have defined several requirements that a body-centric interaction architecture should fill. An interaction architecture should:

1. Maintain a single coherent scene model of the physical environment, including users, displays, and other contextual objects of relevance.

2. Provide a high-level interface to the scene model for use by developers that depends on the structure of the abstract model, and not on implementation details.
3. Provide a consistent interface that remains unchanged regardless of the hardware configuration producing the model.
4. Allow different sensing components to be combined arbitrarily in order to increase the fidelity or scope of the scene model.
5. Capture uncertainty in the model.

It is important to note that we are concerned primarily with a *geometric* model of users and context, rather than a model of user *intent*. Substantial research has already been performed on modeling user intent, but this is outside the scope of our work. It is useful to note, however, that many aspects of intent can be derived from a geometric model of a user. Examples are provided in chapter 4, where we discuss the use of user proximity, eye gaze, and other physical properties, to determine user attitudes and intent. The flip-side is also true, that knowing intent can help define and interpret a geometric model. We leave an exploration of this synergy for future work.

## **6.1 Drawing from Related Work**

Research from other fields is highly relevant to our work designing and implementing the BAPI. We describe here relevant related literature from databases, sensor fusion, and virtual reality.

### **6.1.1 Lessons from Databases**

Fields other than HCI have faced and overcome similar problems to those currently being experienced in HCI. In his seminal paper, Codd [39] described the relational model of data management, which he believed to be superior to the graph and network models of data management. Codd identified several properties of the relational model that he believed to be valuable. We believe that two of these are particularly relevant to our work in HCI:

1. “It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes.”
2. “...it provides a basis for a high level data language which will yield maximum independence between programs on the one hand and machine representation and organization of data on the other.”

Codd recognized the limitations of the network and graph models of data management, and introduced a new model that is simpler to understand and use, but still contains all the information that is needed. The power of his model is evident by the fact that products such as Oracle Database, Microsoft SQL Server, and MySQL, which make use of variants of Codd’s relational model, are now dominant in the commercial sphere.

The lesson to be taken from Codd’s model is that sometimes an abstract representation closer to that which is being modelled can be easier to use and understand than a representation mediated by a particular implementation, and that this abstract representation should be designed with a clear interface that focusses on what is relevant, rather than irrelevant details of implementation.

### **6.1.2 Lessons from Sensor Fusion**

On the surface, our goal of combining signals from multiple sensors to build a single representation of the scene is quite similar to the goals of sensor fusion. In their introduction on the topic, Hall and Llinas point out that humans make wide use of sensor fusion [77]. We combine input from taste, hearing, touch, and the rest of our senses in order to produce a more complete and accurate picture of the world than we would be able to construct using any one of our senses alone. We discussed this previously in section 4.1.1. This idea is the core of sensor fusion. Hall and Llinas describe how sensor fusion has found use in military applications for automated target recognition, remote sensing, and for guiding autonomous vehicles. In the civilian sphere, applications have been found for monitoring manufacturing processes, robotics, and in medical domains. Sensor fusion has not, however, been adopted in any significant way to support the development of computer interaction architectures.

A discussion of sensor fusion that comes very close to what is relevant to our topic is provided by Crowley and Demazeau [43]. They examine sensor fusion in the context of perception, where they define perception as being “the process of maintaining an internal description of the external environment.” They then provide a detailed description of sensor fusion particular to robotics and vision, where data from a number of cameras is combined to form a coherent model of the environment. They discuss algorithmic tools such as Kalman Filters that can be used for generating a model, and describe a pipeline for the iterative updating and refining of such a model. What is missing in this discussion, from our perspective, is the inclusion of sensors other than cameras, such as touch surfaces and magnetic trackers, as well as a discussion of how such a sensor fusion system might be used by a researcher developing new interactive technologies.

The lesson to be taken from the field of sensor fusion is first that there are established techniques that have been developed to merge multiple signals into a coherent whole, and that these techniques should be appropriated for our use as needed. The second lesson, unfortunately, is that producing a single coherent whole is a non-trivial task, and that there are major hurdles in developing a suitable approach.

### **6.1.3 Lessons from Augmented Reality and Virtual Reality**

Augmented Reality (AR) systems function by overlaying virtual elements on top of the real world, as seen from the perspective of the user. This is usually done using a combination of a head-mounted display (HMD) and some form of tracking. Toolkits such as ARToolkit [110] have been developed to support the development of such systems. These toolkits are differentiated from our approach by both scope and focus. First, AR toolkits are usually designed to sense very specific objects in the scene using specific sensors. For example, ARToolkit senses square black marker patterns using a camera. It is able to perform this task quite reliably, but is not designed to integrate input from a variety of sensors. It is also the case that ARToolkit can ignore most of the context of the work environment. It is meant to integrate virtual content into the real world, but it can safely remain ignorant of the vast majority of the real world. Thus, augmented reality toolkits are not designed

to generate a complete and coherent model of the entire work environment.

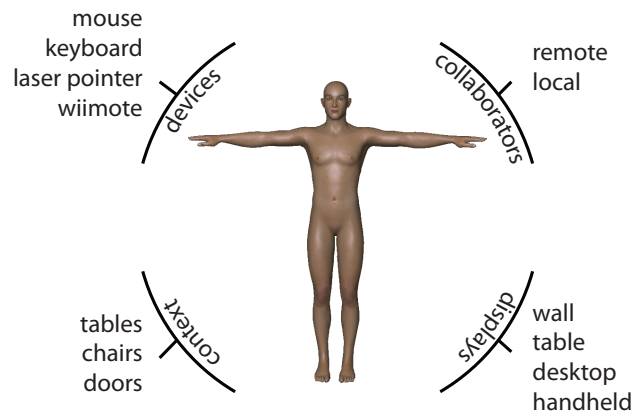
Virtual reality toolkits, such as the MR toolkit [180], on the other hand, have been developed to very accurately model and simulate interaction in virtual environments. They must integrate input such as head position/orientation, hand position/orientation, and body position/orientation in order to properly simulate interactions and update the geometric model of the scene. While it is clear that VR toolkits possess a deep ability to measure and model user body interactions, there is no focus on sensor abstraction and modularity, and the interactions supported by VR toolkits occur entirely in the virtual world, as opposed to our environment, where interactions occur in the physical world.

## **6.2 The Body-Centric Application Programming Interface**

In this section we describe a theoretical Body-Centric Application Programming Interface (API). In a later section we will describe the actual implementation of a subset of the theoretical BAPI described here.

Our approach for designing the BAPI was driven by our body-centric design philosophy. We want to put the user at the center of the model, where we believe he belongs, and consider other contextual objects and displays in relation to the user. Instead of a “device-centric” approach, where developers interface with devices and query their states, we stress a user-centric approach where the developer queries a model of the users in the scene. This is a change in focus of the development approach (Figure 6.1), and one that we think will benefit both developers and users. As Card et al. [29] point out, “an input device is a transducer from the physical properties of the world into logical parameters of an application.” Why rely on an understanding of the functioning of a transducer of physical properties when one can instead query a representation of the actual physical properties? As Blaise Pascal noted, “Les choses valent toujours mieux dans leur source [The stream is always purer at its source].” Our BAPI approach follows Pascal in spirit, by bringing the abstract virtual representation of the scene back full circle to the structure of the scene itself.

The high level architecture of the BAPI is very similar to the “autonomous



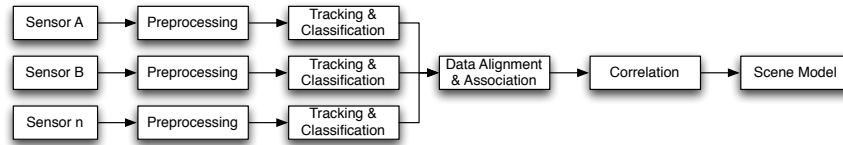
**Figure 6.1:** The design of our Body-centric application programming interface puts the user at the center of the model. Development is performed by primarily querying user states, rather than accessing devices such as mice and sensors and querying their states.

fusion” architecture (Figure 6.2) described by Hall and Llinas [77]. This architecture is more appropriate than the “centralized fusion” architecture that they also described. The centralized fusion architecture is most appropriate for architectures where the raw data streams from different sensors can be reasonably combined, and will result in improved classification. This is true of traditional data fusion scenarios where sensors collect compatible data streams (e.g. satellite imagery of different wavelengths). In our situation, however, data streams from sensors will often be heterogeneous. There is no meaningful way of combining raw data streams from a Vicon system, a touch surface, and a magnetic Polhemus sensor. Instead, preprocessing must first occur to transform data into “state vectors.” A state vector is a representation of features (e.g. an estimation of position, velocity, orientation), rather than raw data (e.g. pixels or magnetic field measurements).

The different stages of the data fusion pipeline are summarized here:

**Preprocessing** Any processing of the raw data that is required before identification of features can be performed.

**Tracking & Classification** An estimate of position, location, and any other relevant properties (captured in state vectors) is calculated for each object or



**Figure 6.2:** Data fusion pipeline for producing a single consistent scene model. Adapted from Hall and Llinas [77].

element the sensor measures.

**Data Alignment & Association** Results are transformed from sensor units and coordinate systems into a shared universal coordinate system.

**Correlation** Results from multiple sensors are related to objects identified in previous measurement iterations.

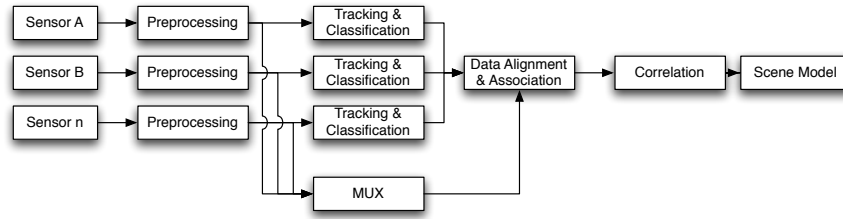
**Scene Model** Single coherent model is produced.

An alternate architecture (Figure 6.3) is possible that is a combination of the autonomous fusion and centralized fusion approaches. This architecture is desirable if there are a number of sensors of a similar type whose output streams can be meaningfully fused at the raw data level, rather than at the state vector stage. Fusing data at an earlier stage often results in superior results. An example of a scenario where a fusion approach would be appropriate is one where multiple video cameras are viewing a scene. The raw data from the cameras should be processed together by a MUX unit, in order to perform a multi-view reconstruction of the scene [147].

### 6.2.1 Uncertainty

When an abstract representation conceals one or multiple sensing approaches, the representation must incorporate knowledge of the inherent limitations of the underlying implementation. For example, a Vicon system relies on line-of-sight to cameras, which is not always available, and the signal from a Polhemus can become noisy as distance between the emitter and sensor increases. These limitations should be represented in the model without requiring the user of the model to possess knowledge of the underlying implementations.





**Figure 6.3:** Hybrid data fusion pipeline for producing a single consistent scene model. A hybrid model allows optimized early processing of raw data from similar sensors. Adapted from Hall and Llinas [77].

There has been work on representation of uncertainty in input. Mankoff et al. proposed that uncertainty be modelled as part of recognition toolkits [140]. In situations where ambiguity exists, their approach involves maintaining a hierarchy of possible events, and later resolving those events when more information becomes available. A mediation layer disambiguates possible events using one of a number of approaches. In later work, Mankoff et al. further explore the interactive possibilities that exist when ambiguity data exists for input [139]. The work described is important, but it doesn't explain how uncertainty might be integrated into a system where a potentially large number of sensing approaches of varying capabilities exist behind a layer of abstraction.

A more appropriate model of uncertainty can be drawn from work by Cheng and Prabhakar [35]. They describe a framework for storing sensor data in a database and then retrieving the sensor data. Central to their framework are three forms of uncertainty: *point uncertainty*, *interval uncertainty*, and *probabilistic uncertainty*. Point uncertainty is actually not uncertainty; it is the assumption that measurements of positions (or other values) represent exactly the actual values. Interval uncertainties are uncertainties where a value is known to be within some certain range, but nothing is known regarding the likelihood of the actual value being in different regions of the interval. Probabilistic uncertainty (explored in [34]) is similar to interval uncertainty, except an additional distribution function is known that specifies how likely it is that the actual value will be at given points within the interval. Key to their approach is that uncertainty in measurements is captured in queries to the database. This is an approach that fits well with our scene model of

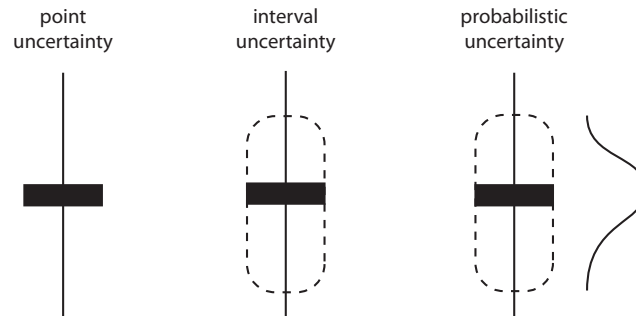
users and context. When an application developer accesses the model, the model can respond not only with measurements, but also with meta-information describing uncertainty.

The use of the three forms of uncertainty are best described using examples. Point uncertainty should be used only when the system is not capable of capturing uncertainty. Interval uncertainty should be used if data is known to be in a certain range, but there is no way of knowing the probability distribution of the sensor data within that range. Take as an example tracking a user's arm using optically tracked Vicon markers. There is one marker for each of the hand, elbow and shoulder. If all three markers are visible then the locations of the joints are known to within a very small error. However, if the elbow marker (for example) is not visible to the cameras, then the system must guess at the location of the elbow based on the known locations of the hand and shoulder. A possible range of values for the elbow can be computed using the known constraints of the person's arm, but there is no way of knowing where in that range the user has placed their elbow. Thus, interval uncertainty is an appropriate representation. A magnetic tracker, such as a Polhemus, provides a good example for probabilistic uncertainty. Such a tracker has a certain amount of noise in the signal. Based on a number of variables, such as distance from transmitter to sensor and battery state, the uncertainty may change. However, based on knowledge of the noise of the signal, a probability distribution of likely locations of the marker can be determined. The distribution provides more information than a simple interval, and is therefore more useful.

A description of how uncertainty can be captured in a query of model state is shown in Figure 6.4. It can be seen how a query to the model will return a single value defining the assumed location, an interval of possible values can also be returned (for the case of interval uncertainty), or a function can be returned (in the case of probabilistic uncertainty).

### **6.2.2 BAPI Query Language**

So far we have discussed the creation of the scene model. In order to be of any use, developers must be able to query the scene model to determine the state of users or contextual objects. Fundamental to the scene model are scene objects and



**Figure 6.4:** Uncertainty is captured in one of three ways in the response to the query for position of an object. *Point uncertainty* assumes that the position is exact and correct. *Interval uncertainty* provides a range of possible values. *Probabilistic uncertainty* provides a function describing the likelihood of the position being of a certain value.

properties. Objects are things like people, displays, and tools. Objects can also comprise other objects. A person is comprised of limbs and joints.

A developer using the BAPI should be able to query scene objects to determine their properties at any point in time. For example, a developer could query for the orientation of a display, the location of the top left corner of the display, or the width of the display. Orientation and location queries would return 3D vectors (and possibly associated meta-data concerning uncertainty), and the width query would return a scalar. For example, the following C++ code finds some basic properties of a specific display:

```
display->getOrientation ();
display->getTopLeftCorner ();
display->getWidth ();
```

Some objects are more complicated than displays. People, for example, not only have the basic properties, but also have complex geometry that changes over time. It can be useful to query very specific properties of a user. For example, the following code finds what direction a user is facing, and then iterates over the limbs and joints in the user, finding their positions:

```
user->getOrientation ();
foreach (Limb limb in user) {
    foreach (Joint joint in limb) {
        joint->getPosition ();
    }
}
```

```
    }  
  }  
  user->getMesh ();
```

It can also be useful to be able to determine relationships between different objects. The relationships can frequently be manually calculated, but convenience methods to perform the calculations can save a lot of time. For example, this code demonstrates finding the distance between two users, computing the angle between the forearm and upper arm, and computing the projection of a user's hand from a point to a display:

```
user1->proximityTo (user2);  
forearm = user1->getLimb (Limb:rightForeArm);  
upperArm = user1->getLimb (Limb:rightUpperArm);  
SceneObject.computeAngle (forearm , upperArm);  
rightHand = user1->getJoint (Joint:rightHand);  
display.project (lightOrigin , rightHand);
```

### 6.2.3 Limitations of the Design

We have described a general architecture for the BAPI, including definition of requirements, description of a processing pipeline, and high-level description of an interface. This serves as a general design, but many of the details of implementation have not been described. Unfortunately, the task of implementing such an architecture in order to anticipate all possible usage scenarios is a monumental task, and is far outside the scope of this dissertation. A reasonable parallel can again be drawn with Codd's initial definition of relational databases. His design was useful at a high level, but it was then necessary for it to be followed by decades of work building out the details of how relational databases function "under the hood."

We make no claim that the design we have offered is in any way easy to implement. The components of data alignment & association, and correlation, in particular, are expected to be very difficult to implement. What we do claim, however, is that once these components have been implemented, the architecture as a whole will be useful, without modification, to a broad set of application developers.

### 6.3 Usage Scenario

It is useful to provide an example of how the BAPI can be used in a real-world scenario. We describe here a scenario of users interacting with a realistic application, in a realistic context. It is worth noting that although realistic, this scenario was not designed with any actual display/sensing infrastructure (such as that which supported the techniques described in other chapters) in mind.

Our scenario involves two users interacting with a wall-based mapping application, with an additional touch table available. Large display mapping applications are commonly used in many contexts, including industrial control rooms [1] and command-and-control for military [179] and crisis management [104].

In our scenario, the application software requires detailed knowledge of the geometry of the two users, and the tools the users may be using. The knowledge requirements of the application include:

- R1** Where on the display is a user touching?
- R2** Is the user touching with a finger or a stylus?
- R3** Which one of the users is touching? What is the identity of that user?
- R4** If the user is touching with a finger, which finger of which hand is being used?
- R5** Where are each of the two users looking?
- R6** How far is each user from the display?
- R7** When does a third user enter the room? What is the identity of the third user?

No single sensor type can practically answer all the questions above. It is best to combine a variety of sensors that are each optimized for sensing particular things, and the architecture described in Figure 6.2 can then be used to construct a single coherent model that the software application can query.

In our scenario, the sensors used include a frustrated-total internal reflection (FTIR) touch wall [80], a number of video cameras placed about the room, eye trackers near the working display, microphones, and magnetically tracked styli. The abilities of the sensors map to desired functionality of the application is shown

application requirement	sensor
R1	FTIR touch wall ([80])
R2	Magnetically tracked styli + cameras ([86])
R3	Cameras ([13])
R4	Cameras + FTIR touch wall
R5	Cameras + Eye tracker ([50])
R6	Cameras ([61])
R7	Cameras

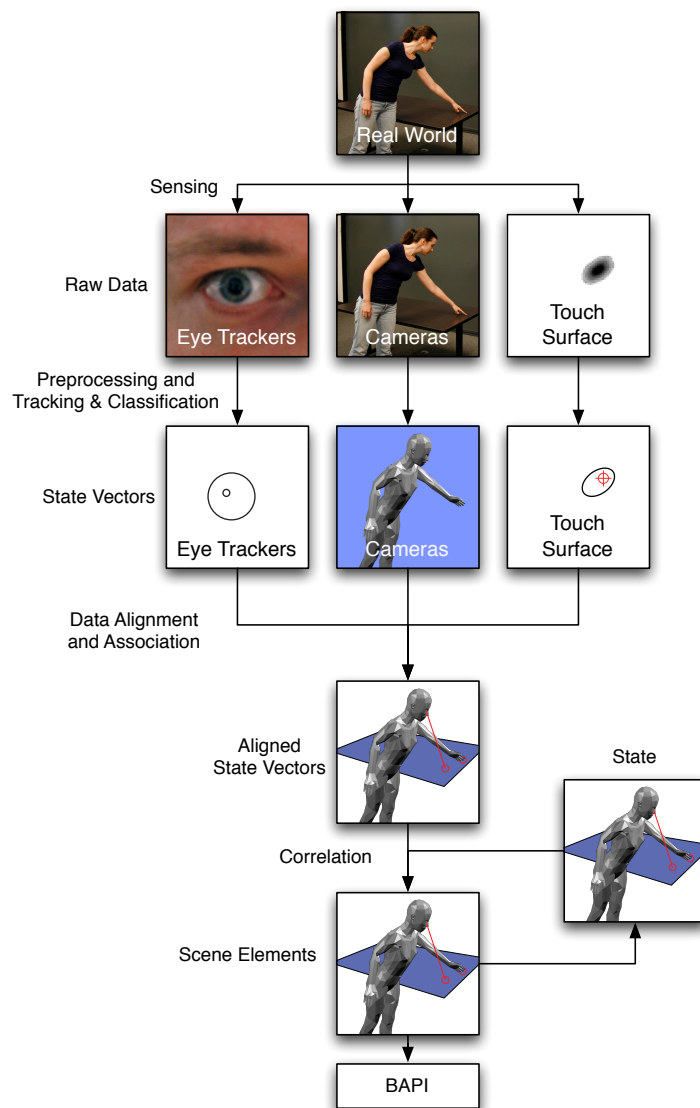
**Table 6.1:** Sensors mapped to the requirements of the application. Some requirements demand a fusion of data collected from multiple sensors.

in Table 6.1. Most of the requirements require a combination of input from different sensor types, as the abilities of each sensor type are limited.

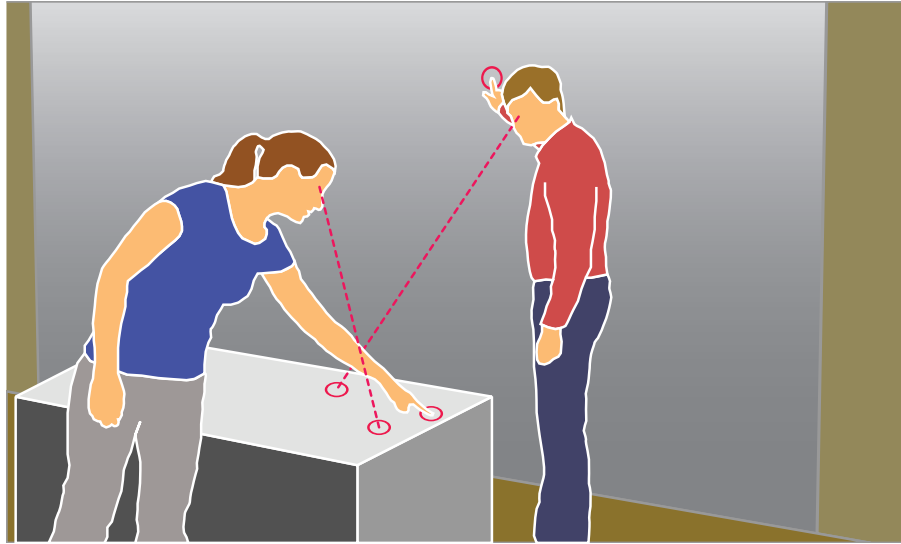
The input from the different sensors is processed by the fusion architecture as shown in Figure 6.5. The resulting scene model (Figure 6.6) is a single representation of the relevant aspects of the users and contextual objects. Allowance is made for retaining copies of the scene state for use by iterative algorithms, such as those based on Hidden Markov Models [12, 54], but when these are employed they will be invisible to the user of the model. Any robust model will likely include a predictive component based on retained state and a corrective component that uses feedback from the sensors to update the model. Hidden Markov Models are one of many classes of techniques that are in widespread use [21, 196].

The scene is not an entirely complete representation of the real world, as many things are not measured. It is, however, complete enough to fill the requirements of the application. If the requirements of the application were extended, or a new application with different requirements were developed, then extra sensors could be added, and a more complete scene model would be produced. Importantly, this expansion of the completeness of the scene model would not require changes to any of the existing application code, although the interaction architecture would need to know how to integrate the new sensors.

It is useful to note that the above scenario could be supported without the described fusion architecture or resulting scene model. A developer could access separate APIs for each of the different sensors, and perform ad-hoc fusion in the application. The downside of this approach is that the developer must have detailed



**Figure 6.5:** Processing pipeline of data from a variety of sensors to produce a coherent scene model for use by an application. In the first step raw data is captured. In the second step raw data is processed in isolation. In the third step state vectors are transformed into a consistent coordinate system. In the fourth step state vectors are associated with entities in the scene. Retained state information may be reused to refine further iterations of model construction.



**Figure 6.6:** Representation of users in a scene. Red circles represent some of the pertinent details that must be captured from different sensing systems. Examples in the image include touch locations and gaze fixations. Other data could include body contours and user identities.

knowledge of how each of the sensors works, and be able to directly manipulate the appropriate data. If any of the sensors changed, for example if the magnetically tracked styli were changed with optically tracked styli, the developer would need to make changes to the application. With the abstraction of the described architecture, this is not the case.

## 6.4 A BAPI Implementation

We implemented a preliminary version of a BAPI architecture in order to support our investigations into body-centric interaction techniques and applications. This implementation was used to support the work described in Chapter 4. Because the architecture evolved with the applications, and we had not yet finalized the design of our architecture, there are some differences between our theoretical architecture and what was actually implemented. There are also some omissions in the implementation, such as a lack of support for modelling uncertainty. However, our implementation largely follows the design in spirit, and could be extended to more



correctly fit the model.

The BAPI was implemented in C# using the .NET Framework version 3.5. Other languages, notably C, were used to communicate with legacy software libraries.

The classes implemented fall into two general class hierarchies. The first relates to the scene model itself (Figure 6.7), and represents items, tools, and users in the scene. The objects in this hierarchy are accessed by the application developer in order to query the state of the scene, and are accessed by the sensing components in order to update the state of the scene. The individual classes and important properties are summarized in Table 6.2. The classes are briefly described here:

**SceneModel** Container for all the scene objects in the scene. The developer can traverse the collection of objects using this class.

**SceneObject** Represents a single object existing in the scene.

**Body** Represents a single person.

**BodyIK** A body model that enforces certain constraints.

**BodyIKTest** A body model with autonomous motion. Used for testing.

**Limb** A part of a body.

**Joint** End points of a limb.

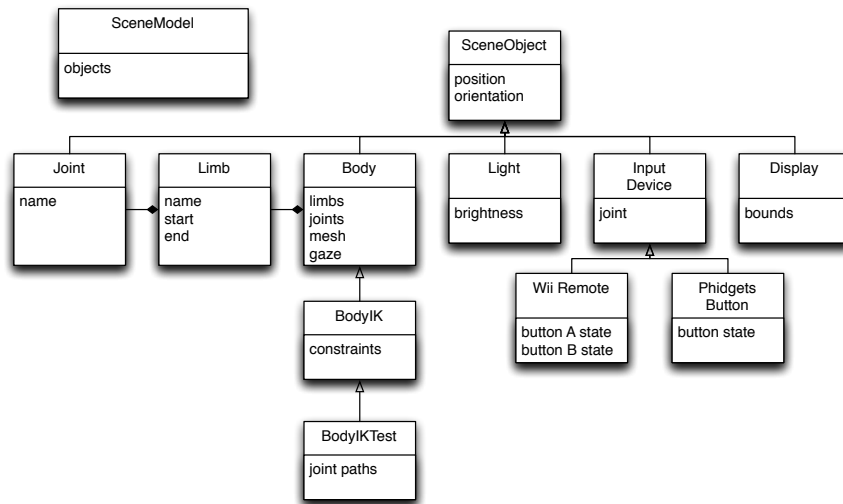
**Light** A virtual light source in the environment.

**Input Device** A tool used by the user to interact with the environment or virtual content of a display.

**Wii Remote** A Wii Remote used by clicking buttons.

**Phidgets Button** A simple single wired button.

**Display** A rectangular display in the environment.



**Figure 6.7:** UML class diagram of major classes in the modelling component of the implemented architecture.

The second class hierarchy (Figure 6.8) is devoted to the classes that will process data incoming from any sensors, and will update the scene based on what is sensed. The light modelling classes are a bit unusual in this hierarchy, as they are not associated with any form of sensing. Instead, they update the location of the virtual light sources in the scene (as described in section 4.3.4), based on the state of the scene as a whole. The individual classes and important properties are summarized in Table 6.3. The individual classes are briefly described below:

**LatusCapture** Collects position (X, Y, Z) data from any available Liberty Latus magnetic tracking sensors. Each tracker is associated with a joint on a body. The body position is updated as the position of the tracker changes.

**CameraCapture** Uses multiple cameras to identify the locations of coloured balls and triangulate their positions. Each ball is associated with the location of a user’s joint. The user body is updated to represent this.

**LightModel** A behaviour that describes the position of a virtual light source in the scene.

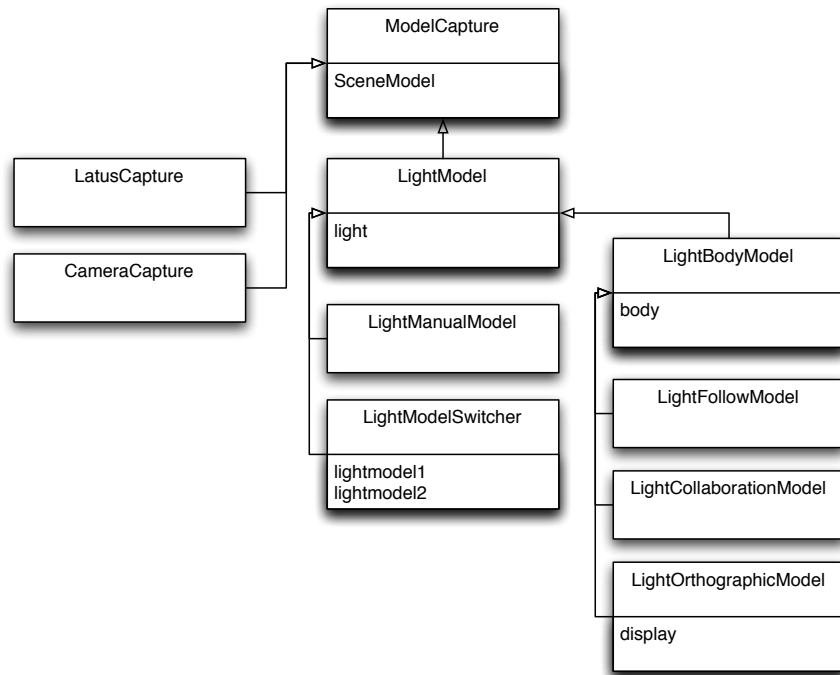
Class	Property	Description
SceneModel	objects	List of SceneObjects contained in the scene.
SceneObject	position orientation	Position of the object in the scene. Orientation of the object in the scene.
Body	limbs joints mesh  gaze	List of Limbs that comprise the Body. List of Joints that comprise the Body. Mesh that approximate the 3D contour of the Body.  Direction of gaze of the body.
Limb	name start end	Name of limb (e.g. left-lower-leg). Joint at one end of the Limb. Joint at the other end of the Limb.
Joint	name	Name of the joint (e.g. elbow)
BodyIK	constraints	Constraints used to calculate the pose of the Body.
BodyIKTest	joint paths	Predefined motion of certain Joints for debugging purposes
Light	brightness	Brightness of the light.
Input Device	joint	Joint (usually a hand) that is holding the device.
Wii Remote	button A state button B state	State of the A button (e.g. pressed). State of the B button.
Phidgets Button	button state	State of the button (e.g. pressed).
Display	bounds	Bounds of the surface of the Display.

**Table 6.2:** Major classes implemented as part of the scene model. Important properties of the classes are named and described.

**LightBodyModel** A behaviour that relies on the properties of a body in the scene in order to set the light position.

**LightFollowModel** A behaviour where the virtual light source is located behind the user, determined by a vector pointing out the user's back.

**LightCollaborationModel** A behaviour where light source positions change based on user proximity.



**Figure 6.8:** UML class diagram of major classes in the sensing component of the implemented architecture. These classes are generally responsible for updating the scene model to properly represent the physical scene. The LightModel classes are special purpose, and set the locations of virtual light sources according to specific behaviours.

**LightOrthographicModel** A behaviour where the virtual light source is positioned behind the user, relative to the display, at a very large distance.

**LightManualModel** A behaviour where the virtual light source is manually placed at any arbitrary location.

**LightModelSwitcher** A behaviour where the light source position changes from an old to a new behaviour in a smooth fashion.

Class	Property	Description
ModelCapture	SceneModel	The scene that should be updated by this class.
LatusCapture		
CameraCapture		
LightModel	light	The light that this class should update.
LightManualModel		
LightModelSwitcher	lightmodel1	The light model that is being switched from.
	lightmodel2	The light model that is being switched to.
LightBodyModel	body	The body that this model uses to set the light location.
LightFollowModel		
LightCollaborationModel		
LightOrthographicModel	display	The display that this model uses to set the light location.

**Table 6.3:** Major classes implemented as part of the sensing component. Important properties of the classes are named and described.

## 6.5 Conclusions

It is important that a new model of interaction be approachable by new developers. Regardless of how powerful a model can be for users, if it is difficult to implement, it is unlikely that it will see widespread adoption. We have described an interaction architecture that supports the development of body-centric interaction techniques.

Supporting the development of body-centric interaction techniques is particularly challenging. No single sensing approach is adequate to capture the range of data that is necessary to support the proper exploration of interactive possibilities. We have defined a theoretical interaction architecture that takes inspiration from the power of sensor fusion and the simplicity of databases, and applies the lessons learned from those fields to HCI. The theoretical pipeline we have described can possibly allow developers to explore the space of body-centric interactions.

We have also described a concrete implementation of a subset of the functionality described in the theoretical framework. This implementation was performed in parallel with our own design of interaction techniques. We were thus able to test and refine the approach as we went. As we progressed with our design iterations and the BAPI architecture matured, we found that we were spending less and less time writing code to interface with sensors and manage raw data, and more and more time focussing on the interaction techniques. In this sense our architecture can be considered a success.

### **6.5.1 Limitations**

As we noted earlier in the chapter, designing and implementing a fully functioning and extensible BAPI is a very large task, and well outside the scope of a single dissertation. We only implemented support for a handful of sensors, but there are many more that could be integrated. Coordination of the state vectors derived from a wide range of sensors is also potentially a very difficult task.

One question that will need to be answered is whether a generic approach such as that described by our BAPI will be able to provide a model of equal power and accuracy to an “ad-hoc” implementation where a developer accesses individual sensor systems directly. We believe that a scene model that is built with sufficiently sophisticated approaches in the data alignment & association and correlation stages of the data fusion process will be equivalent in quality to models that are built with literal sensor data.

We expect that some sort of BAPI architecture will emerge into the HCI community, but that it may take some time to do so. We expect that at a high level what ultimately emerges will draw inspiration from our approach, but the truth is that there are many internal details that remain to be addressed. This will be future work.

## Chapter 7

# Revisiting Fitts' Law When Gain Varies

The Shadow Reaching technique described in Chapter 3 relies on mid-air, physical pointing by the user, mediated by the shadow reaching software mapping between the user's physical actions (control) and the cursor and shadow representation on a large screen (display). The ratio of control movement to display movement is referred to as gain [8], which is inherent in the perspective projection underlying Shadow Reaching. In order to evaluate the effectiveness of such techniques, theoretical models that measure performance are necessary. We test a theoretical model of mid-air pointing performance for very large wall displays that uses the Welford two-part formulation of Fitts' law. This allows for an independent contribution to movement time of movement amplitude  $A$  and target width  $W$ . We demonstrate how the relative contributions of  $A$  and  $W$  can be mathematically captured in an exponent  $k$ . We then provide new experimental data suggesting that the exponent  $k$  increases monotonically as control-display gain increases, and that it appears to increase linearly. We conclude that to accurately model pointing performance on interactive displays more robust models, such as the Welford two-part formulation, should be adopted to take into account control-display gain. A significant contribution of the work reported in this chapter is the finding that although most of the standard formulations of Fitts' law do not take gain into account, it appears that for physical pointing on large screen displays gain plays an important role that can be

modeled by the Welford formulation.

Developing theoretical models of low level performance is an important task in aid of developing new interaction techniques and new display form factors. As Card et al. argued in their discussion of the Model Human Processor [30], theoretical models of human action support the ability of designers to perform approximate engineering calculations when creating systems. In addition, as pointed out by I.S. MacKenzie et al. [129], as the link between user and machine gets more “direct,” speed-accuracy models for actions in the physical world become more relevant to actions taken in computing contexts. This becomes even more true when adopting newer interaction techniques, such as the Reality-Based Interfaces of Jacob et al. [101], the body interfaces of Klemmer et al. [118], and the body-centric interaction techniques introduced in this dissertation.

We initially set out to develop a theoretical model of pointing performance for mid-air pointing on large wall displays. This is consistent with the overall theme of the dissertation, specifically body-centric interaction, because of the connection of large-scale interaction to user motion within the work context. Such a theoretical model of performance would apply specifically to the kinds of techniques that we developed using a body-centric approach. What we discovered, however, was that our investigations arrived at conclusions with much wider implications. They apply not only to our specific domain but to others as well, including mouse input on large displays, and also pointing on small displays. We decided, due to the importance of the findings, to expand the scope of this chapter, while still focusing on the elements that are specific to our theme of body-centric interaction.

The Fitts’ law model of pointing performance [60] has proven to be extremely robust. As Pratt et al. [168] point out, it has been applied to physical pointing underwater [111], in near-zero gravity [65], with microscopic targets [124], and when pointing with one’s feet [88], among other things. Fitts’ law has found a home in the field of Human Computer Interaction, with a nearly thirty-year legacy of use [186]. When evaluating new pointing devices or new display form factors one of the first research tasks is often to perform a Fitts’ law evaluation, producing a predictive model of performance and determining throughput for the particular approach.

However, there are exceptions to Fitts’ law. Not long after Fitts’ original paper,



Welford observed [208] that some Fitts-like tasks result in data that does not follow Fitts' model. Welford proposed an alternate two-part model of performance to take into account this deviation from expected performance. Welford's model allows for independent contributions to movement time of target width  $W$  and movement amplitude  $A$ , rather than only considering the ratio of  $A$  and  $W$  captured in Fitts' "index of difficulty" that we describe in the next section.

More recently, there are other indications that Fitts' law has limitations. Pratt et al. [168] discovered that allocentric spatial information can modulate pointing performance, with the result that pointing to a farther target can in some cases take less time than pointing to a closer target. As Pratt et al. conclude: "it now seems unlikely that a single equation will be able to accurately capture all aspects of speed-accuracy trade-offs." From these two examples it is clear that Fitts' law, while widely useful and validated, should not be taken as gospel. Alternate explanations should be considered when appropriate. Two such situations are when the size of the display varies significantly from those traditionally studied, and when widely varying values of the control/display gain ratio are employed.

In this chapter we explore how control-display gain influences mid-air pointing performance on very large wall displays, and how this can be best modeled using the two-part Welford formulation of Fitts' law. Such a model can be used to help in the design of interaction techniques such as the body-centric techniques described in chapters 3 and 4, and the text input techniques described in chapter 5.

We start by performing a re-analysis of data obtained from other researchers. This re-analysis serves to frame our own experiments and expectations. The re-analysis demonstrates that Fitts' law breaks down when modelling pointing on some computing devices, and that the magnitude of this breakdown depends on the control-display gain. This breakdown may not have been identified by some other researchers because of limitations in experimental design and analysis. The re-analysis further suggests that the two-part pointing model proposed by Welford can correct the breakdown of Fitts' original model.

Motivated by the re-analysis, we then present results from a new experiment that investigated physical pointing at a distance on a very large wall display. These results indicate that the Welford two-part formulation models pointing performance much better than the original Fitts' formulation, and that pointing performance

does indeed vary based on gain in a manner consistent with the results of the re-analysis.

The contributions of this chapter are three-fold. First, we demonstrate through a re-analysis of data from other researchers that Fitts' law falls short in modelling pointing performance on a computer and we suggest why this shortcoming may have been overlooked for so long in the HCI community. Second, we analyze the data used in the re-analysis to show that Welford's two-part formulation more accurately models pointing performance at all levels of control-display gain. Third, we present results from a new experiment that support the findings of our re-analysis by demonstrating how the coefficients in the Welford two-part formulation depend on control-display gain for mid-air pointing on a very large wall display.

## **7.1 Related Work**

Literature related to Fitts' law is specifically relevant to the research reported in this chapter.

### **7.1.1 Fitts' Law for Computer Pointing**

Fitts' law [60], originally developed as a tool for modelling the performance of human physical pointing, was applied to pointing on a computing device by Card et al. [28]. The empirically determined parameters of a Fitts model for computer pointing depend largely on the device used. Researchers have performed numerous evaluations of devices, including investigations into traditional mouse, pad, and trackball devices [55], stylus input [63], direct touch on tables [64], and pointing with a laser pointer [154]. Researchers have also extended Fitts' law to special cases. Variations on Fitts' model have been developed for 2D pointing [130], 3D pointing [73], pointing to expanding targets [144], and pointing to dynamically revealed targets [26], among others. Researchers have even investigated such subtle points as the impact of cursor orientation on performance [166], and the independence of throughput on the speed/accuracy tradeoff [131].

There are still some open questions, however, regarding how Fitts' law should be applied in HCI. Guiard [74] raised the question of consistency in the design of Fitts' law experiments and introduced a new interpretation, form and scale, for

Fitts' law experimentation. Outside of the field of HCI, questions have also been raised regarding the applicability of Fitts' law. Pratt et al. [168] discovered that allocentric information can modulate pointing performance, suggesting that there is more to pointing than the low-level motor movement modelled by Fitts' law. These explorations, combined with Keulen et al.'s [113] identification of multiple reference frames used for reaching, suggest that our understanding of pointing performance and Fitts' law is far from complete, and that we should continue to re-evaluate our understanding and use of Fitts' law.

## 7.2 One-Part and Two-Part Models of Pointing Performance

Several formulations of Fitts' law have been posited. We discuss here four important variants.

$$MT = a + b \log_2 \left( \frac{2A}{W} \right) \quad (7.1)$$

$$MT = a + b \log_2 \left( \frac{A}{W} + 1 \right) \quad (7.2)$$

$$MT = a + b \log_2 \left( \frac{A}{W} + 0.5 \right) \quad (7.3)$$

$$MT = a + b_1 \log_2 A - b_2 \log_2 W \quad (7.4)$$

The original version due to Fitts (eq. 7.1) defines movement time as depending on the distance between targets ( $A$ ) and the size of targets ( $W$ ), as well as two experimentally determined constants. The  $\log_2(2A/W)$  term is known as the index of difficulty ( $ID$ ). In Fitts' formulation it is the dimensionless ratio of  $A$  and  $W$  that matters; the individual values of  $A$  and  $W$  are not in isolation important.

Soukoreff and I.S. MacKenzie [186] have promoted the use of a formulation (eq. 7.2) where  $ID$  is more consistent with a Shannon-inspired information-theoretic interpretation of Fitts' law. This is similar to what we refer to as Welford's one-part model (eq. 7.3). Both have an additive constant within the logarithmic term.

The fourth important variation, and the one we will focus on, is the Welford two-part model (eq. 7.4), which allows for separable contributions of  $A$  and  $W$  to movement time. By separable, we mean that the individual values of  $A$  and  $W$  are of significance, rather than just the ratio  $A/W$ . Welford introduced his two-part model to account for deviations from Fitts' law that he observed in data collected from an improved version of Fitts' original experiments [208, page 158]. Welford's conclusions have been indirectly supported by more recent work in kinesiology, where it was found that trajectories of limbs and velocity profiles during pointing depend independently on  $A$  and  $W$  [128], and that movement times can be independent of target size in virtual grasping tasks [142]. Welford's two-part model seems to have been largely overlooked in the HCI literature, although there is work, including that by Wobbrock et al. [217], that hints at the need for a model of pointing other than Fitts'. We think Welford's model is potentially powerful in its generality, especially when applied to interaction with large displays.

Several properties of large display interaction techniques differentiate them from the direct touch motor movement that was originally studied by Fitts. First, interaction and feedback are often located in different spaces. For example, a user may manipulate an input device (e.g. a mouse) in one space while visual feedback, including a visual cursor, is shown on a display in a separate space. Second, there is not necessarily a one-to-one correspondence between input movements and resultant feedback. The control-display gain (CD gain) can be manipulated in different ways, and motion can be either relative or absolute. Third, manipulation can be performed outside of a person's physical reach, using devices such as laser pointers. It is known that humans use different cognitive mechanisms to operate inside and outside of physical reach [91], and these different mechanisms may result in different performance profiles. Because of this, a single model, such as Fitts' law, may not be adequate in describing a task with many influencing variables.

In order to understand the performance properties of different interaction techniques for very large wall displays, it is useful to determine whether Fitts' law applies and how it might need to be generalized. Important questions include: does Fitts' original formulation (or alternately one of the Shannon-inspired formulations) apply to distance pointing on a large display? If not, does an alternate model, such as Welford's two-part model, better explain performance? Are con-

clusions generalizeable across multiple interaction techniques? To answer these questions it helps to first develop a clearer understanding of existing work involving the different formulations, and different interaction scenarios.

### 7.2.1 Pointing at a Distance and the Kopper $k$ Exponent

In this section we analyze recent work by Kopper et al. on distance laser-pointer interaction and relate it to the much earlier Welford two-part separable model of pointing performance. Kopper et al. [121] examined distance pointing with a laser pointer on large displays. They developed a model of performance based on angular measurements  $\alpha$  for amplitude and  $\omega$  for target width (eq. 7.5).

$$MT = a + b \log_2 \left( \frac{\alpha}{\omega^k} + 1 \right) \quad (7.5)$$

The use of angular measurements is consistent with their technique, where rotation of the input devices, rather than translation, results in cursor motion. The exponent  $k$  they introduce allows  $\alpha$  and  $\omega$  to have separate degrees of impact on the movement time. We will call this exponent the “Kopper  $k$ ” and we will generalize the use of a  $k$  exponent to other variants of Fitts’ law. We will similarly refer to eq. 7.5 as the “Kopper angular formulation” or simply the “Kopper formulation.”

A formulation analogous to the Kopper angular formulation can be constructed using linear units (eq. 7.6).

$$MT = a + b \log_2 \left( \frac{A}{W^k} + 1 \right) \quad (7.6)$$

In this version of the Kopper formulation, linear amplitude  $A$  replaces angular  $\alpha$ , and linear width  $W$  replaces angular  $\omega$ . A second formulation is also possible (eq. 7.7), where we omit the Shannon-inspired “+1” term. This omission aids in further manipulation of the equation; it will be justified later when we present our re-analysis and our new experimental data.

$$MT = a + b \log_2 \left( \frac{A}{W^k} \right) \quad (7.7)$$

An important observation, not noted by Kopper et al., is that the linear analog of their formulation (eq. 7.7) can be derived directly from Welford’s two-part model

(eq. 7.8), meaning they are equivalent. The  $k$  and  $b$  values in the linear version of Kopper's formulation are equal to  $b_2/b_1$  and  $b_1$  from Welford's formulation, respectively. We therefore have a second means of expressing Welford's two part model, with different constants.

$$\begin{aligned}
 MT &= a + b_1 \log_2(A) - b_2 \log_2(W) \\
 &= a + b_1 \left[ \log_2 A - \frac{b_2}{b_1} \log_2(W) \right] \\
 &= a + b_1 \left[ \log_2(A) - \log_2(W^{b_2/b_1}) \right] \\
 &= a + b_1 \log_2 \left( \frac{A}{W^{b_2/b_1}} \right) \\
 &= a + b_1 \log_2 \left( \frac{A}{W^k} \right) \tag{7.8}
 \end{aligned}$$

The significance of the linear version of the Kopper formulation lies in the exponent  $k$ . The exponent  $k$  is a single constant that conveniently encapsulates the relative magnitude of the separable contributions of the independent variables  $A$  and  $W$  to the overall movement time. If experimental results determine that  $k = 1$  the model is simply Fitts' law (without the factor of 2 multiplying  $A$ , a minor detail), and Fitts' law will model the experimental data as well as does the two-part model. However, in cases where experimental data dictate that  $k$  deviate significantly from unity, Fitts' formulation will do a poor job of modelling results. Thus  $k$  is useful not only for gauging the relative contributions of  $A$  and  $W$ , but is also a good indicator of the applicability of Fitts' formulation. We adopt the use of  $k$  for much of the remaining chapter to illuminate the separable contributions of  $A$  and  $W$ .

### 7.2.2 Unit Dependence in Two-Part Models of Pointing

The empirically determined constants of a Fitts model are independent of the choice of units (e.g. cm or mm) used in measurement. This is not the case using the Welford two-part or linear version of the Kopper formulations. As shown in eq. 7.9, Fitts' formulation is not impacted by changing units used in measure-

ment (represented by scaling both  $A$  and  $W$  by some constant value  $c$ ), because the scaling constants cancel out. The empirically derived constants  $a$  and  $b$  will therefore be the same regardless of units used.

$$\begin{aligned} MT &= a + b \log_2 \left( \frac{2cA}{cW} \right) \\ &= a + b \log_2 \left( \frac{2A}{W} \right) \end{aligned} \quad (7.9)$$

In Welford's and Kopper's formulations, however, the units chosen will impact the constants derived. As seen in eq. 7.10, a multiplicative constant  $c$  applied to measurements will result in a change in the empirically determined additive constant  $a$ .

$$\begin{aligned} MT &= a + b_1 \log_2(cA) - b_2 \log_2(cW) \\ &= a + b_1(\log_2 c + \log_2 A) - b_2(\log_2 c + \log_2 W) \\ &= (a + b_1 \log_2 c - b_2 \log_2 c) + b_1 \log_2 A - b_2 \log_2 W \\ &= a' + b_1 \log_2 A - b_2 \log_2 W \end{aligned} \quad (7.10)$$

A multiplicative scale  $c$  will result in a new additive constant,  $a' = a + b_1 \log_2 c - b_2 \log_2 c$ . Importantly, the  $b_1$  and  $b_2$  constants, and therefore the  $k$  value, all remain independent of the units chosen. Thus the magnitude of the separable effects of  $A$  and  $W$  as captured in the exponent  $k$  is independent of the units chosen to represent amplitude and target width.

This explanation glosses over an important point. Both  $A$  and  $W$  appear within logarithms. This is not mathematically valid because the logarithm of whatever units are used is not really an admissible quantity. As Graham explains [69], Welford anticipated this objection and postulated nominal values  $A_0$  and  $W_0$  that "normalize"  $A$  and  $W$  respectively and eliminate the problem of logarithmic units.

$$MT = a + b_1 \log_2 \left( \frac{A}{A_0} \right) - b_2 \log_2 \left( \frac{W}{W_0} \right) \quad (7.11)$$

We will assume that suitable constants  $A_0$  and  $W_0$  are used as in eq. 7.11 but will simply write eq. 7.4 with the understanding that suitable normalization has taken place and that only the intercept value  $a$  might be affected by the choice of normalization values  $A_0$  and  $W_0$ .

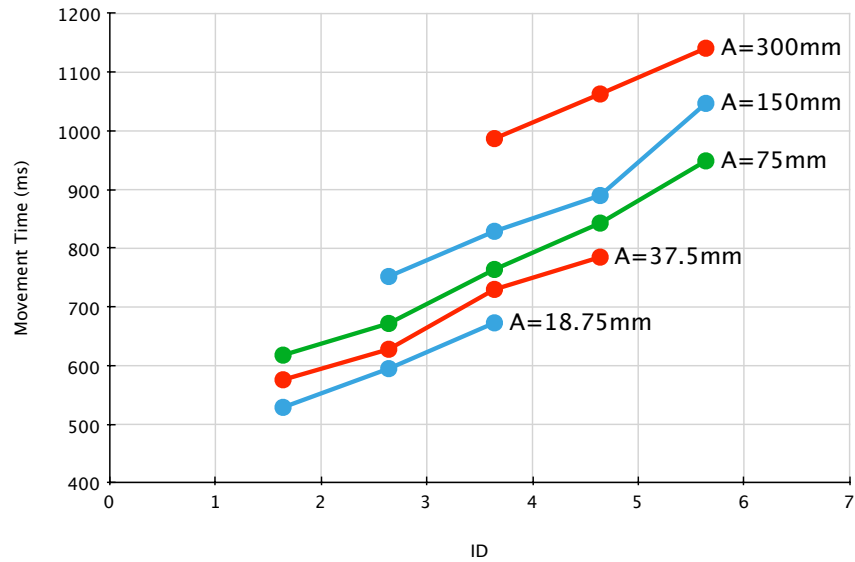
### **7.2.3 Alternate Models**

The Fitts and Fitts-like (e.g. Welford) models are not the only models of pointing performance. Earlier models include those by Woodworth [219] and Hollingworth [89]. Another model is that due to Schmidt et al. [177], which developed a set of relationships between movement amplitude, precision, time, and the mass of objects being moved. We used Fitts as the basis for our work for several reasons. Most importantly, Fitts is accepted as the gold standard, and is used most widely by HCI researchers and practitioners. Because of this, any new work on pointing must necessarily be compared to Fitts. In addition, Fitts is much more concise and easily understandable than the older models due to Woodworth and Hollingworth. Fitts is also more relevant to computer pointing than the Schmidt model, which deeply integrates the physical properties of pointing (such as the mass of the item being moved) which are not clearly transferable to virtual pointing operations.

## **7.3 A Re-Analysis of Pointing Experiments Selected from the Literature**

One-part models of pointing based on Fitts' original formulation have garnered much more attention in the HCI literature than have two-part models such as those by Welford and Kopper. This may be because one-part models appear to serve us well and they are simpler and more concise than two-part models. We might well ask, is there a need for two-part models of pointing performance? In order to answer this question we undertook a re-analysis of published experimental data. We delved into aspects of the data that were left unexplored by the authors of three studies. We found that one-part models possess previously unappreciated limitations, and that two-part models of pointing performance are more useful than was previously thought. We completed the re-analysis by reviewing 19 papers to see whether the lack of appreciation for two-part models might be due to limitations



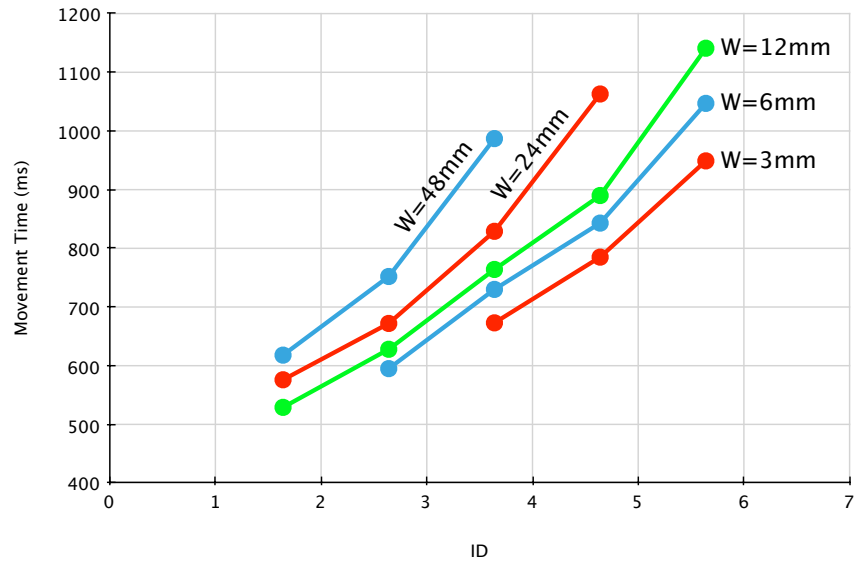


**Figure 7.1:** Lines connect points representing tasks with the same amplitude. With  $A$  held constant movement time varies roughly linearly with  $ID$ . Adapted from Graham [69, page 47].

in the experimental designs and analyses that were used.

### 7.3.1 Physical Pointing on a Small Display

Graham described a series of experiments dealing with indirect (virtual) control of an on-display pointer using a motion-tracked finger [69]. His first experiment investigated pointing at varying levels of control-display gain on a traditional sized display. He found that Fitts' original formulation does not accurately model pointing because of a separable effect of  $A$  and  $W$  on performance. A visualization of Graham's results (Fig. 7.1 and 7.2) reveals a pattern very similar to that found by Welford in his reproduction of Fitts' experiment. Movement time scales roughly linearly with  $ID$  within either a single  $A$  or  $W$  value, but not across changes in both the  $A$  and  $W$  values. As with Welford's analysis, Graham found that a two-part model (eq. 7.4) was necessary to compensate for the patterns observed, and to accurately model the movement time. The validity of applying this two-part model was further supported by Graham's analysis of hand velocity and acceleration pro-

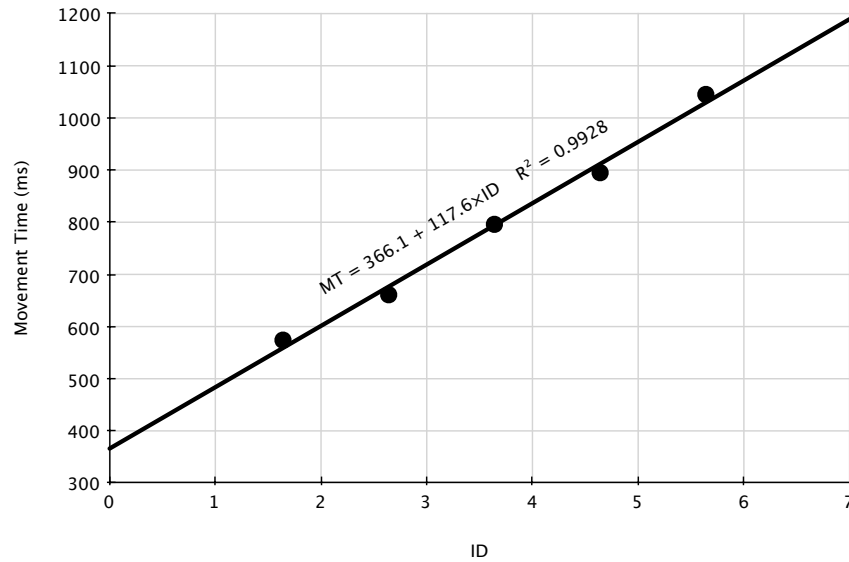


**Figure 7.2:** Lines connect points representing tasks with the same target width. With  $W$  held constant movement time varies roughly linearly with  $ID$ . The data points are exactly the same as in Fig 7.1, only the lines have been drawn differently. Adapted from Graham [69, page 47].

files during pointing, which again revealed separable effects of  $A$  and  $W$  in different temporal segments of the motion.

The fact that both Welford and Graham found a two-part formulation to be necessary to accurately model performance of their respective real-world and virtual pointing tasks is intriguing. But what does this mean for evaluation of other forms of computer pointing? After all, Fitts' original formulation (and those closely related to it) have repeatedly been found to accurately model pointing on a computer. However, Graham points out that even for cases where there is a separable effect of  $A$  and  $W$ , Fitts' formulation holds for certain subsets of data, and can also hold for data that is averaged across all of the  $A$  and  $W$  values for each  $ID$ .

In explanation, Graham notes that an experiment and related analysis must be performed carefully to isolate the contributions of  $A$  and  $W$  in order to potentially reveal the described effect. As an example, Fig. 7.3 shows a regression analysis of data points from the Graham data that are averaged within  $ID$  values, whereas

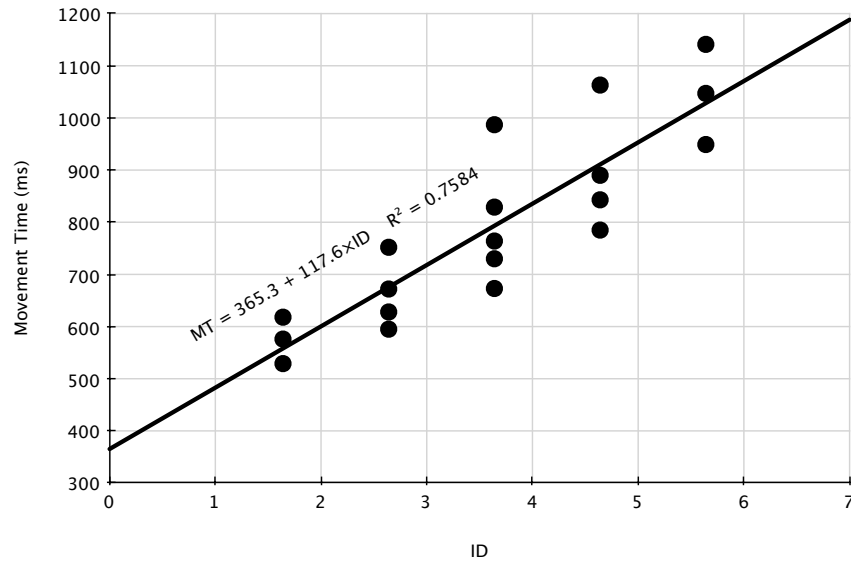


**Figure 7.3:** A regression analysis of the Graham data using average MT results for every  $ID$  value at  $gain = 1$ . Poor fit is concealed due to averaging of data points.

Fig. 7.4 shows a regression analysis of the same data, but where data points are not averaged. It is evident that averaging data points conceals the poor fit of the data to a linear function, a danger that should be avoided. The importance of careful experimental design is backed up by arguments made by Guiard [74], who notes that standard Fitts' law experimental designs tend to contain a confound related to the concomitant variation of  $A$  and  $W$  with  $ID$ . The importance of this is that past experiments that found Fitts' law to correctly model pointing may have been masking separable effects because of experimental design limitations.

We further analyzed the results of Graham's experiment<sup>1</sup>. This is summarized in Table 7.1. For completeness, we performed regression analyses for the Fitts (eq. 7.1), Shannon (eq. 7.2), and Welford (eq. 7.4) formulations, providing the regression coefficients and  $R^2$  values for each. For the Welford formulation we also compute  $k$ , the ratio  $b_1/b_2$ . There is no universally accepted threshold for a good  $R^2$  values, but I.S. MacKenzie [129] suggests  $R^2 = 0.9$  as a guideline when evaluating

<sup>1</sup> We thank Evan Graham for providing us with access to the data he collected.

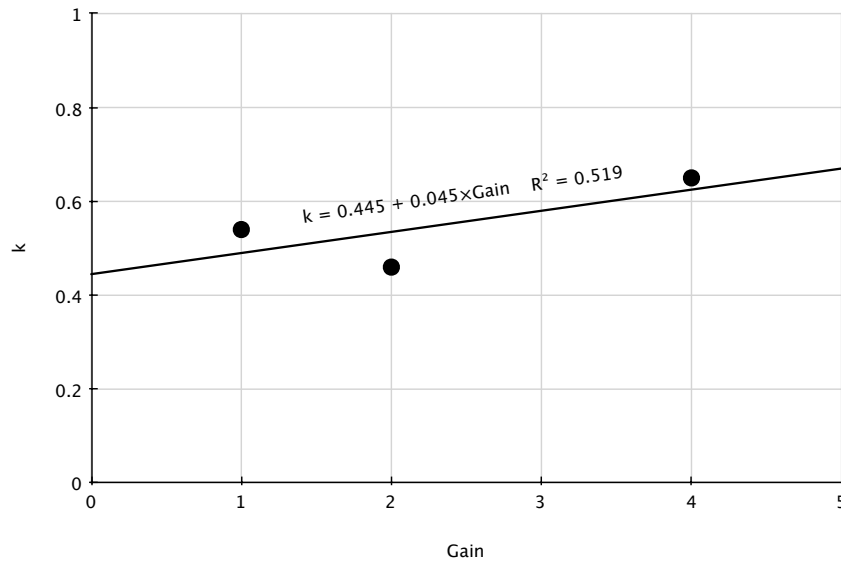


**Figure 7.4:** A regression analysis of the Graham data using MT results of every combination of  $A$  and  $W$  at  $gain = 1$ . Poor fit is evident as a result of analyzing all data points. Adapted from Graham [69, page 45].

Gain	Fitts ( $2A/W$ )			Shannon ( $A/W + 1$ )			Welford				
	$a$	$b$	$R^2$	$a$	$b$	$R^2$	$a$	$b_1$	$b_2$	$k$	$R^2$
1	0.365	0.117	0.759	0.384	0.139	0.761	0.133	0.153	0.082	0.54	0.967
2	0.346	0.111	0.789	0.364	0.132	0.793	0.018	0.151	0.070	0.46	0.981
4	0.349	0.112	0.952	0.371	0.133	0.945	0.188	0.136	0.089	0.65	0.992

**Table 7.1:** Modelling of movement time from Graham data using the Fitts formulation, the Shannon formulation, and the Welford two-part formulation. For the Fitts and Shannon formulations  $R^2$  decreases with lower gain. For the Welford formulation  $R^2$  is consistently good. Fitts formulation data is from Graham [69].

Fitts' law results. We follow this advice. Graham's original analysis of his data (limited to the case of  $gain = 1$ ) revealed a poor fit for the Fitts formulation but a good fit using the Welford two-part formulation. Our analysis of Graham's data for all three gain levels measured (Table 7.1) demonstrates that the Welford two-part formulation produces an accurate fit for all gain levels, ranging from  $R^2 = 0.967$  at  $gain = 1$  to  $R^2 = 0.992$  at  $gain = 4$ , whereas acceptable fits using the Fitts and Shannon formulations are only achieved at  $gain = 4$ . The  $k$  values for each gain level are also shown for the Welford two-part formulation. They are computed



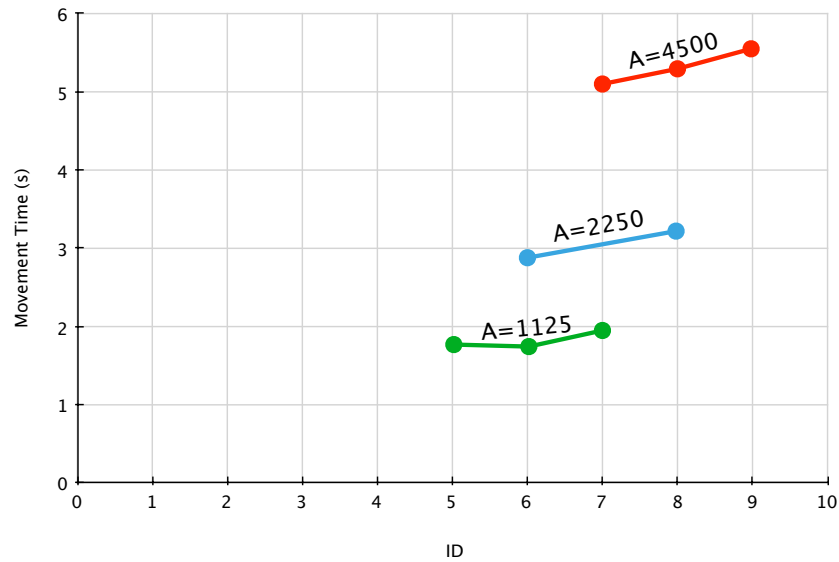
**Figure 7.5:** The  $k$  values for the Graham data.

directly from the ratio of two Welford model constants in the manner previously described.

The  $k$  values for each gain level of the Graham data are shown graphically in Figure 7.5. No clear pattern is evident. A linear regression of the data points for  $k$  produced a poor value of  $R^2 = 0.519$  ( $a = 0.44$ ,  $b = 0.046$ ). It is unsurprising that no clear trend emerged, however. Just three gain values were investigated, limiting the number of data points, and only a small number of participants (six) were used in the experiment.

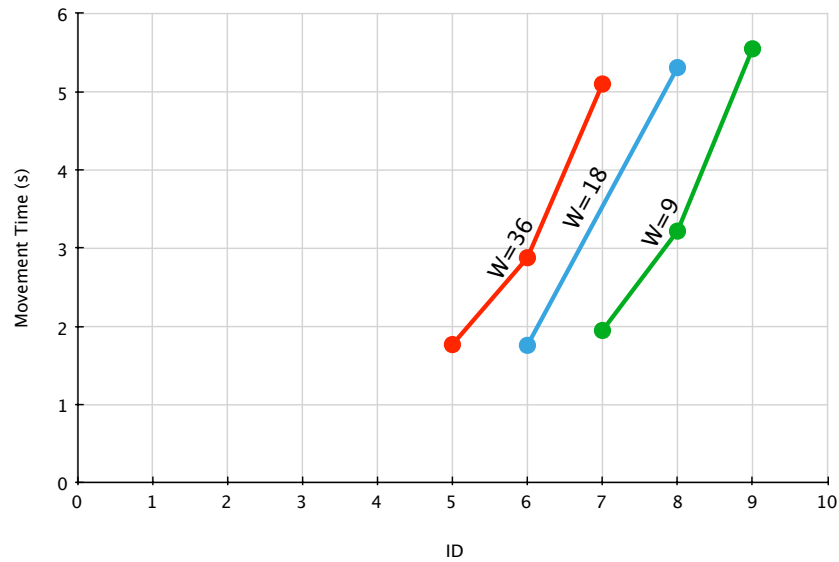
### 7.3.2 Mouse Pointing on a Large Display

The analysis of Graham's results leaves some open questions. Most importantly, Graham's results relate to indirect (virtual) finger pointing mapped to an on-screen cursor. This is not a common input mechanism in computing systems. What might we find if we analyze pointing performance using a mouse, instead? Other researchers have found Fitts' law to be adequate to describing this task, but will we be able to observe a similar separable contribution of  $A$  and  $W$  to pointing time for this task?



**Figure 7.6:** Movement time results for all  $A/W$  combinations for the Casiez data at  $gain = 2$ . Lines connect points representing tasks with the same amplitude.

Casiez et al. recently performed a very thorough analysis of mouse pointing performance that clarifies the landscape of both small and large display pointing [32]. They evaluated pointing performance for constant CD gain levels as well as for different levels of pointer acceleration, on both small and large displays. They examined a wide range of  $ID$  values, using a variety of  $A$  and  $W$  values for each  $ID$ . In their first experiment on small display interaction they analyzed each gain level separately using the original Fitts' one part model, and found regression  $R^2$  values ranging from 0.956 to 0.984, indicating a close match between model and measurements. However, the results from their second experiment, performed on a large display, found regression  $R^2$  fits ranging from 0.577 to 0.959, with a roughly direct relationship between gain and  $R^2$ . Casiez et al. argue this relationship was due to excessive mouse clatching at low gain levels. From our analysis of results by Graham and Kopper we suspect that the cause might be due to the separable contributions of  $A$  and  $W$  to  $MT$  across all gains. To determine if this was the case we performed further analysis on the Casiez data.



**Figure 7.7:** Movement time results for all  $A/W$  combinations for the Casiez data at  $gain = 2$ . Lines connect points representing tasks with the same target width. The data points are exactly the same as in Fig 7.6, only the lines have been drawn differently.

We first produced two visualizations of movement times for individual  $A$  and  $W$  pairs for the  $gain = 2$  condition, which was the condition with the worst  $R^2$  in the original analysis. Figure 7.6 shows that when lines are drawn connecting data points with the same  $A$  values, each line is approximately straight and parallel to the other lines and when the points with the same  $W$  values are connected again the lines are approximately straight and parallel (Figure 7.7). This is evidence of a separable effect of  $A$  and  $W$ , and is the same pattern found by both Welford and Graham in their data. Both used similar evidence to justify applying the Welford two-part model (eq. 7.4). We believe that this is appropriate in many situations where  $A$  and  $W$  both vary.

Results of a linear regression analysis of the Casiez results are shown in Table 7.2. The original analysis from Casiez et al. using the Fitts formulation shows a poor fit at lower gains. Our analysis<sup>2</sup> using the Welford two-part formulation pro-

<sup>2</sup> We thank Gery Casiez for providing us with access to the data he and his colleagues collected.

Gain	Fitts ( $2A/W$ )			Shannon ( $A/W + 1$ )			Welford				
	$a$	$b$	$R^2$	$a$	$b$	$R^2$	$a$	$b_1$	$b_2$	$k$	$R^2$
2	-4.125	0.950	0.577	-3.259	0.960	0.577	-15.286	1.742	0.159	0.091	0.977
5	-1.405	0.412	0.734	-1.029	0.417	0.734	-4.300	0.628	0.196	0.312	0.935
8	-0.841	0.308	0.805	-0.560	0.311	0.805	-2.608	0.444	0.172	0.387	0.961
12	-0.431	0.243	0.891	-0.209	0.246	0.892	-1.320	0.317	0.169	0.533	0.974
16	-0.393	0.232	0.936	-0.181	0.234	0.936	-1.009	0.287	0.177	0.632	0.989
20	-0.569	0.264	0.959	0.329	0.266	0.960	-0.876	0.301	0.226	0.750	0.978

**Table 7.2:** Modelling of movement time from Casiez et al. data using the Fitts, Shannon, and Welford formulations.  $R^2$  values for the Fitts and Shannon formulations decrease with lower gain, but the Welford formulation is consistently good. The data was provided by Casiez et al.

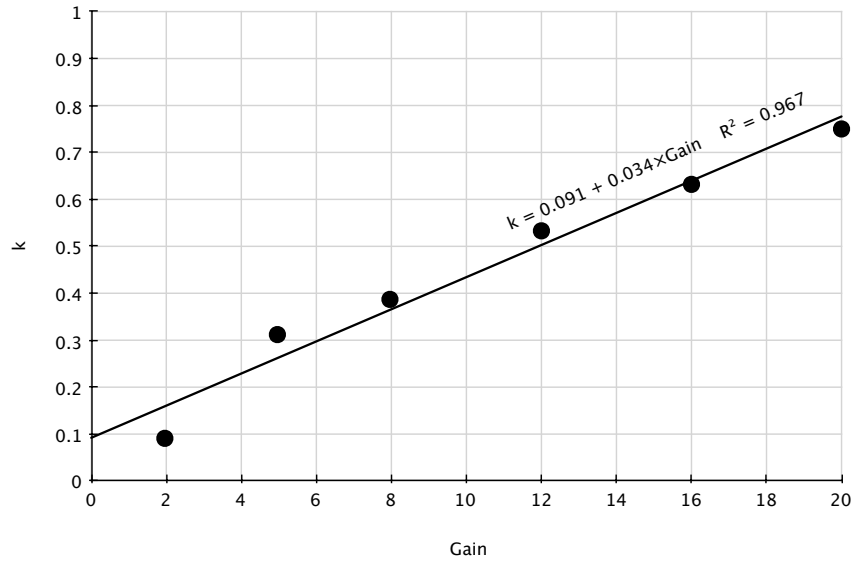
duces a good fit for all gain levels, ranging from a low of  $R^2 = 0.935$  at  $gain = 5$  to a high of  $R^2 = 0.989$  at  $gain = 16$ . The  $k$  values for each gain level are also computed using the Welford formulation.

As with Graham’s data, the  $k$  values serve to quantify the separability of contributions of  $A$  and  $W$ . We therefore decided to further analyze these values with the goal of determining a pattern. The  $k$  values for each gain level of the Casiez data are shown in Figure 7.8. The values increase monotonically with gain, and the dependence of  $k$  on gain appears to be linear. To confirm this hypothesis we ran a linear regression and found the fit to be very close,  $a = 0.091$ ,  $b = 0.034$ ,  $R^2 = 0.967$ . Using this regression model to capture the variance of  $k$  we can state a model of large display mouse pointing performance using the linear version of the Kopper formulation, this time with  $gain$  as an independent variable (eq. 7.12). This equation is especially powerful, as it models pointing performance at all levels of gain, with gain as a variable. This is an advancement over previous work, where it was not possible to incorporate gain as a variable in the formula for movement time.

$$MT = a + b \log_2 \left( \frac{A}{W^{0.091 + 0.034 \times gain}} \right) \quad (7.12)$$

The Welford formulation has an extra degree of freedom, compared to the Fitts formulation, and is therefore guaranteed to produce an  $R^2$  fit that is at least as good as that of the Fitts model, regardless of whether  $A$  and  $W$  are really making independent contributions to  $MT$ . The pattern of  $k$  varying linearly with gain, as shown in Figure 7.8, strongly suggests that what we have observed is a real





**Figure 7.8:** The  $k$  values for the Casiez data.

phenomenon and is not due to chance, but it is still desirable to perform a statistical test to determine if the improvement in fit is statistically significant. An  $F$ -test, as shown in eq. 7.13, can compare two linear models, where one is nested in the other, in the sense that they are equivalent except for an additional degree of freedom in one formulation [213].

$$F = \frac{\frac{RSS_1 - RSS_2}{p_2 - p_1}}{\frac{RSS_2}{n - p_2}} \quad (7.13)$$

The original Fitts formulation is nested within the Welford formulation, and we can therefore perform the  $F$ -test (this test is not possible between Welford and Shannon, as they do not nest). For comparing Welford and Fitts (where  $p_1 = 2$  and  $p_2 = 3$ ) using the Casiez data ( $n = 8$ ), we use eq. 7.14.

$$F = \frac{SSE_{Fitts} - SSE_{Welford}}{SSE_{Welford} / 5} \quad (7.14)$$

The results of the test comparing the Fitts and Welford models are shown in Table 7.3. The test indicates that the Welford formulation models pointing time

Gain	Fitts SSE	Welford SSE	n	SSE	F-ratio	p	Sig?
2	7.953	0.438	8	0.0876	85.8	< 0.001	yes
5	0.740	0.180	8	0.0360	15.5	0.010	yes
8	0.276	0.056	8	0.0112	19.6	0.006	yes
12	0.087	0.021	8	0.0042	15.7	0.010	yes
16	0.044	0.008	8	0.0016	22.5	0.005	yes
20	0.036	0.019	8	0.0038	4.47	0.088	no

**Table 7.3:** Results of a statistical  $F$ -test comparing regressions using the Welford formulation to those using the Fitts formulation for the Casiez data. For gain levels with significant results, the Welford formulation models the data significantly better than the Fitts formulation.

significantly better than the Fitts formulation at all levels of gain except  $gain = 20$ . The failure to better model the data at  $gain = 20$  is not surprising, because this is where  $k$  is approaching 1, as shown in Figure 7.8. When  $k = 1$ , the Welford and Fitts formulations are identical.

Our analysis of the Casiez data revealed two valuable insights that were not discussed in their original paper. First, the departure of pointing performance from the Fitts formulation in the large display condition is corrected by the use of Welford’s two-part formulation, suggesting separable impacts of  $A$  and  $W$  on mouse pointing performance. Second, we discovered that the  $k$  values vary linearly depending on gain for the levels of gain examined.

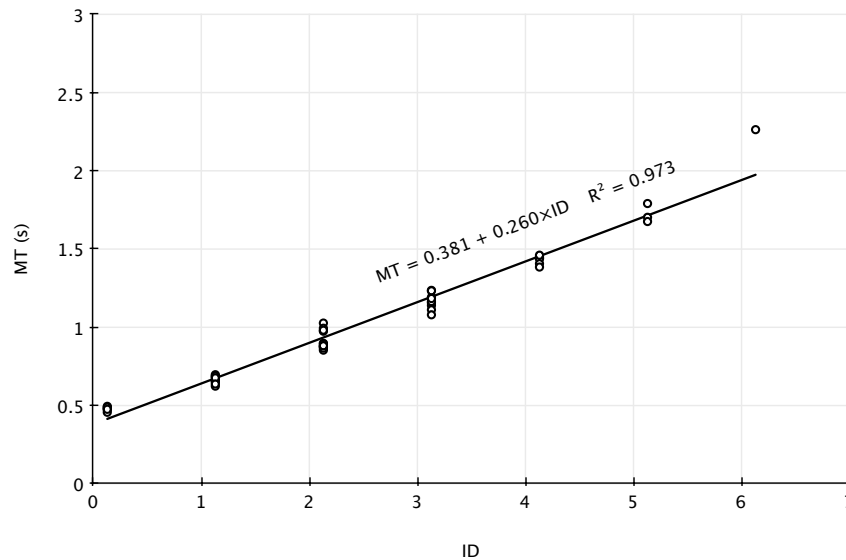
### 7.3.3 Mid-Air Pointing on a Large Display

In a different paper, Tsukitani et al. [199] (including the author) evaluated performance of users undertaking a serial 1D pointing task on a very large wall display. Their goal was to investigate the distinction between “display-space” (interaction as measured on the display) and “hand-space” (interaction as measured in user coordinates) interaction. Their experimental design was such that results serve to provide a partial picture of the applicability of two-part formulations to modelling pointing performance on large wall displays.

In the Tsukitani et al. experiment, users stood at one of three distances (1.0m, 2.5m, 3.25m) from a 5m×3m wall display. The cursor was placed on the display

Gain	Fitts ( $2A/W$ )			Shannon ( $A/W + 1$ )			Welford				
	a	b	$R^2$	a	b	$R^2$	a	$b_1$	$b_2$	k	$R^2$
1.33	0.149	0.247	0.966	0.145	0.331	0.971	-0.044	0.281	0.212	0.755	0.985
2.66	0.158	0.241	0.988	0.171	0.294	0.991	0.260	0.250	0.230	0.919	0.990
5.33	0.105	0.270	0.975	0.134	0.321	0.986	0.428	0.266	0.273	1.027	0.976

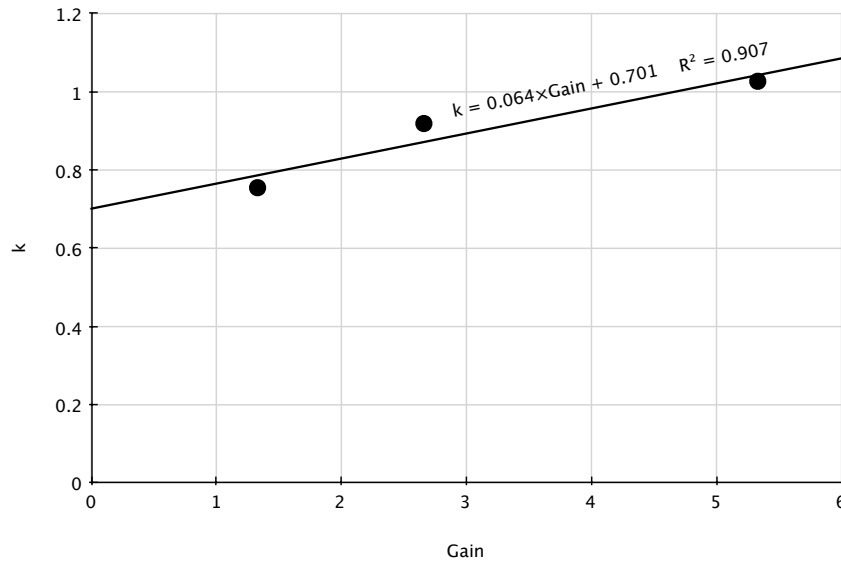
**Table 7.4:** Modelling of movement time from Tsukitani data using the Fitts formulation, the Shannon formulation, and the Welford two-part formulation.



**Figure 7.9:** Regression analysis of the Tsukitani et al. results using a Fitts' model.

at a point determined by a vector originating at a point 4.0m from the display, and passing through the hand of the user. The origin was located at a height such that the cursor would appear near the middle of the display when the hand was held at waist height. Effective gain values of 1.33, 2.66, and 5.33 can be computed from the projection geometry used in the experiment.

Fitts' original formulation appeared to do a good job of modelling pointing performance ( $R^2 = 0.973$ , Fig. 7.9), which would suggest there was no need for a two-part formulation. We decided to delve deeper into the data, and computed linear regression results for each gain level independently (Table 7.4). When the  $k$  values for the three gain levels are visualized (Fig. 7.10) a pattern of  $k$  mono-



**Figure 7.10:** The  $k$  values for the Tsukitani data.

tonically increasing with gain is evident. This is consistent with the results of our analyses of the other two data sets, particularly the Casiez data. The reason for the good fit with Fitts' original formulation is also evident. The Tsukitani experiment only evaluated pointing at three gain levels, and these gain levels are all close to where the regression line of best fit for  $k$  as it varies with gain passes through  $k = 1$ . This is where Welford's formulation is effectively a one-part model, so not much evidence of separability of  $A$  and  $W$  would be present.

While the Tsukitani data reveals no need for a two-part model, we suspect that had a larger number and wider spread of gain values been evaluated such a need would emerge. A range of gain levels similar to those explored by Casiez et al. would be desirable.

### 7.3.4 Speculation About Limitations in Previous Work

Our belief that interactive computer pointing performance, specifically mouse pointing, does not follow Fitts' law may be surprising. Many papers have successfully applied Fitts' law to mouse pointing and other computer tasks. Why is it that these papers did not find an effect similar to what we found? There are several answers.

First, as just noted, for some levels of gain  $k$  will be close enough to unity that modelling performance using Fitts' law produces good  $R^2$  values. Second, as Graham observed, experimental design and analysis choices can serve to partially mask or even completely obscure the separable impacts of  $A$  and  $W$  on performance, even when  $k$  is not close to 1.

In order to get a clearer picture of why the shortcomings of Fitts' law were not previously appreciated we performed an analysis of 19 papers directly related to device pointing drawn from a review by Soukoreff and I.S. MacKenzie [186] of 27 years of Fitts' law papers. We found that 15 of these papers incorporated limitations at either the experimental design phase or the data analysis phase that could conceal a separable effect of  $A$  and  $W$ . Of these 15 papers, seven [3, 19, 75, 92, 106, 107, 132] included limited combinations of  $A$  and  $W$  in their experimental designs. Having only one value for either  $A$  or  $W$  will completely conceal any separable impact of  $A$  and  $W$  on performance. In addition, eight papers [28, 55, 96, 136, 148, 165, 176, 220] aggregated data points for the same  $ID$  in their analysis. Only four of the nineteen papers [81, 130, 132, 133] avoid these two pitfalls and have potential to reveal the effect through robust experimental design and analysis.

Another explanation for why a two-part formulation has not been more thoroughly explored is the fact that the effect is clearest when examining a wide range of gains. This is most common when experimenting on large displays. For example, in the Casiez work the poor fit using the Fitts formulation was clear in the second experiment that was performed on a large display, but was not evident in the first experiment on a traditional-sized display.

We are not criticizing the previous research. The analyses were reasonable under the assumption that Fitts' law is robust, which has been the functioning assumption for the last several decades. However, with the realization that a two-part model may be more appropriate, it becomes necessary to evaluate a broad range of both  $A$  and  $W$  values, and to include each resulting  $A/W$  pair in the analysis.

### **7.3.5 Re-Analysis Conclusions**

We performed analyses of data from two studies by other researchers (Graham [69] and Casiez et al. [32]), and one recent study in which the author collaborated [199].

Experiment	Properties		Results		
	display size	gain	$k$ values	$k$ slope	$k$ slope $R^2$
Graham	small	1–4	0.54–0.65	0.045	0.519
Casiez	large	2–20	0.091–0.75	0.034	0.967
Tsukitani	large	1.33–5.33	0.755–1.027	0.064	0.907

**Table 7.5:** Summary of Fitts’ law experiments analyzed, and conclusions drawn.

We summarize the analysis in Table 7.5, which provides a clearer picture of pointing on both small and large displays.

The key insight was derived from the work by Graham, namely that  $A$  and  $W$  contribute independently to the magnitude of movement time, and that a Fitts-like model where only the ratio of  $A$  and  $W$  is considered is inadequate to accurately model pointing time. Instead, the two-part Welford formulation is more appropriate for describing pointing performance on a display.

Applying this insight to other studies helped clarify the results from those studies. We found that recent results of Casiez et al. and Tsukitani et al. are best modelled using the two-part Welford model, and that this fixes a problem of poor model fit for conditions of low gain on large displays. We further demonstrated that  $k$  varies predictably over different mouse gain values.

We draw the following conclusions from our re-analysis:

1. The original Fitts’ law formulation does not accurately model computer pointing performance using either a finger or mouse, except at some specific gain levels.
2. Welford’s two-part formulation of pointing performance does accurately model computer pointing performance using either a finger or a mouse.
3. Welford’s formulation is equivalent to the linear version of Kopper’s formulation, and the magnitude of the separable impacts of  $A$  and  $W$  on performance can be captured in a constant  $k$ .
4. The value of  $k$  varies linearly with gain in a large display mouse pointing task. The rate at which  $k$  varies with gain depends on the technique.

## 7.4 Evaluation of Pointing Performance on Large Displays

When we began this research, our goal was to develop a model of mid-air pointing performance for very large wall displays. Our re-analysis of other work suggested that it would be desirable to design experiments so that separable contributions of  $A$  and  $W$  to pointing time could be revealed. We thus decided to examine a wide range of gain values in addition to using multiple pairs of  $A$  and  $W$  values. The general design of our experiment is inspired by those of Casiez et al. and Tsukitani et al., which were discussed in the previous section.

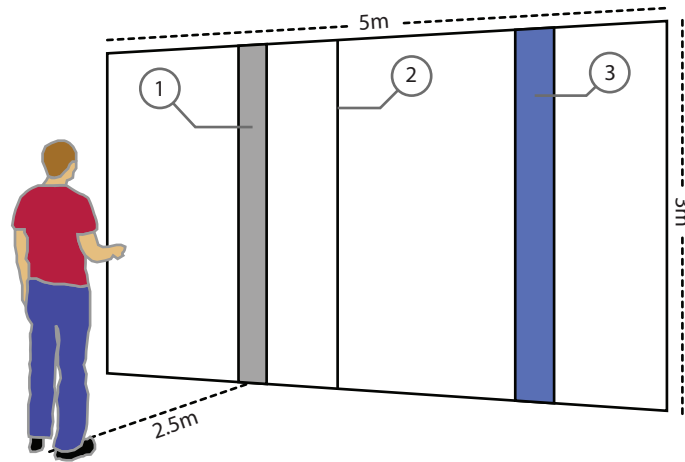
### 7.4.1 Apparatus

Users viewed a large vertical glass screen approximately  $5\text{m}\times 3\text{m}$  in size. The screen was rear-projected by a  $4\times 3$  array of  $800\times 600$  resolution projectors (Figure 7.11 and 7.12). The images of neighbouring projectors overlapped 160 pixels with a blending function to minimize discontinuities due to possible misalignment. Overall resolution was  $2720\times 1480$  pixels.

The software ran on a computer running the Windows XP operating system and was written in C# using the Microsoft XNA Game Studio library and .NET 3.5. The WiimoteLib library was used to communicate with the Wii Remote device. The same computer ran the Vicon tracking software. Logging of events was performed in real time and stored on the machine.

Click events were performed using the thumb (A) button on a Nintendo Wii Remote.

Tracking of the Wii Remote was performed using a Vicon motion capture system because the native Wiimote sensing capabilities were not accurate enough for our needs. The Wii Remote was outfitted with reflective markers for this purpose (Figure 7.13). Tracking accuracy is important, and Casiez et al. [32] develop a model for calculating the usable range of gain values given sensing accuracy and display resolution. With the resolution of the experimental display, and Vicon signal noise measured to be not more than 0.07mm (capture volume was very small, making high accuracy possible), we applied Casiez's model and calculated the maximum usable gain of our apparatus to be 26. Therefore every screen pixel



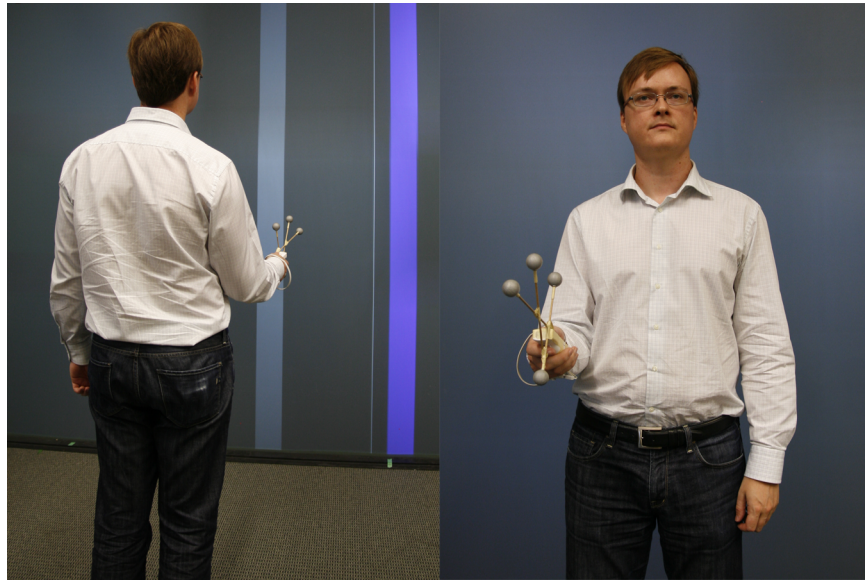
**Figure 7.11:** Layout of experimental apparatus. Labelled components are: (1) center target, currently not active. (2) cursor. (3) right target, currently active.

was addressable at gain values up to 26, and all values of gain investigated in the experiment are considered “usable.”

#### 7.4.2 Task and Stimuli

The experimental task was a serial 1D tapping task between two targets of variable amplitude and width, modelled closely after tasks used by Casiez et al. [32] and Fitts [60]. It was decided to use a traditional 1D task similar to what was originally used by Fitts, rather than a 2D task such as that defined by ISO 9241-9 [48], because we were concerned with the fundamental applicability of Fitts’ law. For each target pair a participant first clicked the start target and then performed a sequence of 8 reciprocal taps between the two targets. The current target was always blue, and the current non-target was always grey. One target was directly in front of the participant, while the other target was to the right of the participant at the given amplitude. This arrangement was chosen to avoid any possible impact of cross-lateral inhibition, which is a difficulty in motions where the hand crosses the body’s midline [178]. The participant was required to correctly click the start target to initiate the trial. Missed clicks for the following eight taps were recorded and





**Figure 7.12:** A user interacting with the experimental system. The view from the back of the user shows the elements on the screen: the center target in grey, the right target in blue, and the cursor in between the two targets.

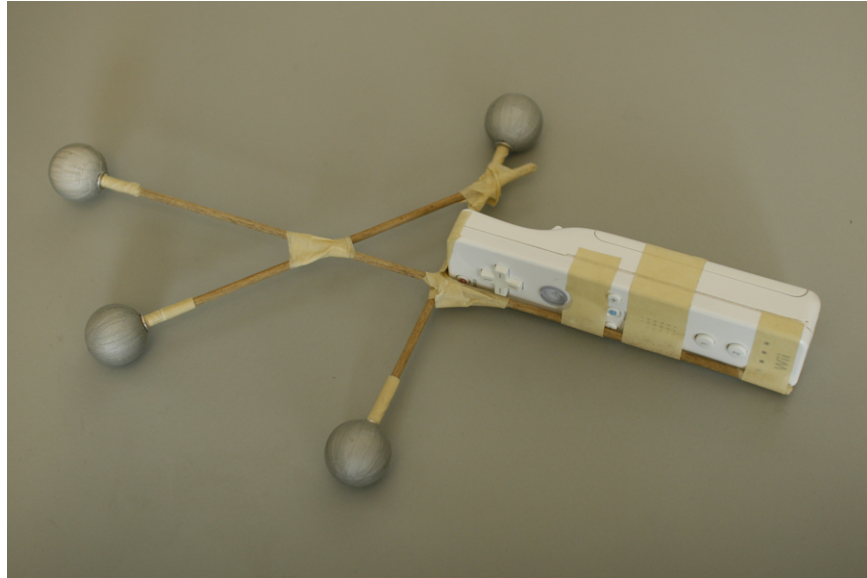
there was no requirement to correct errors. After a click the target briefly flashed green to indicate success, or red to indicate an error.

### 7.4.3 Participants

Nineteen participants (two female) were recruited through on-campus advertising. All were right handed, a requirement for participation in the experiment. Ages ranged from 20 to 42, mean 26.4, SD 5.7. All participants were regular computer users (9+ hours per week). They were compensated \$10 for participating, and the half with the best performance were later compensated an extra \$10.

### 7.4.4 Design

A within-subjects design was used. The independent variables were gain (2, 5, 8, 12, 16, 20), target size (5cm, 10cm, 20cm), target amplitude (25cm, 50cm, 100cm, 250cm), and trial block (1, 2, 3). *A/W* combinations were fully crossed, except the



**Figure 7.13:** The Wii Remote mounted with reflective Vicon tracking markers.

250cm amplitude was only used at gains 16 and 20, because it was not reachable at lower gains. We chose an incomplete design in the interest of investigating a wider range of gain,  $A$ , and  $W$  values, as demanded by our motivation. Such a partial design is not ideal, but it does support an investigation of parameters outside of what would be possible with a full design.

Each gain level was presented during each block of trials. Gain levels were randomly ordered during each block. Within each gain level each  $A/W$  pair was presented.  $A/W$  pairs were randomly ordered during each gain level. Eight taps were performed for each  $A/W$  pair.

- 19 participants
- × 3 blocks
- × (4 gains × 9  $A$  and  $W$  combinations)
- + (2 gains × 12  $A$  and  $W$  combinations)
- × 8 taps
- = 27,360 total taps

## Procedure

The experiment was performed in a single session for each participant, lasting approximately 50 minutes. Participants arrived and filled out a pre-questionnaire gathering demographic information. They were introduced to the system and the pointing task was explained. Participants were told to complete the task as quickly as possible with a goal of 95% accuracy. They each practiced at least five  $A/W$  combinations (40 taps), and were invited to practice more if they felt the need.

Participants then completed the three experimental blocks. Whenever the gain level was changed a practice  $A/W$  pair was presented to the participant. The purpose was to allow the participant to grow accustomed to the new gain level. The participant was not informed that the  $A/W$  pair was a practice pair. It was presented in the flow of the experiment, but the data for these pairs were not analyzed.

Between each block the participants sat at a table and played a distractor puzzle task for three minutes. They were invited to take extra time to rest, but none did so.

After all conditions were completed a participant filled out a post-questionnaire that gathered qualitative feedback on particular aspects of the experiment.

## Measures

Performance was measured as the time taken to perform each individual click action. Timing began for each  $A/W$  pairing when the participant clicked the start target. Errors were measured as click events that occurred outside of the current target. The location of each click was also recorded.

## Hypotheses

We derived our hypotheses from our re-analysis of related research.

**H1** The Fitts formulation will not accurately model pointing performance at all gain levels.

**H2** The Welford two-part formulation will accurately model pointing performance at each individual gain level.

**H3a** (*weak*) The exponent  $k$  will vary monotonically with gain.

Factor	F-ratio	Significance	Partial $\eta^2$
<i>gain</i>	$F_{3,0,53.8} = 36.9^*$	$p < 0.001$	0.672
<i>A</i>	$F_{1,1,20.4} = 778.5^*$	$p < 0.001$	0.977
<i>W</i>	$F_{1,1,20.1} = 408.3^*$	$p < 0.001$	0.958
<i>gain</i> $\times$ <i>A</i>	$F_{10,180} = 4.3$	$p < 0.001$	0.194
<i>gain</i> $\times$ <i>W</i>	$F_{4,3,77.6} = 15.0^*$	$p < 0.001$	0.455
<i>A</i> $\times$ <i>W</i>	$F_{2,5,45.8} = 18.5^*$	$p < 0.001$	0.506

\*Greenhouse-Geisser correction for violated sphericity applied.

**Table 7.6:** Significant ANOVA results for movement time in the mid-air large display pointing experiment.

**H3b** (*strong*) The exponent  $k$  will vary linearly with gain.

#### 7.4.5 Results

We were concerned with the possible impact of learning effects. Before our main analysis we performed a repeated measures ANOVA to determine if there was an effect of block. We found no effect of block on either movement time ( $F_{2,36} = 0.943$ ,  $p = 0.399$ ) or error rate ( $F_{1,382,24.873} = 0.117$ ,  $p = 0.814$ , with a Greenhouse-Geisser correction for violation of sphericity). We therefore included all blocks in our analysis.

#### Movement Time

Significant main effects of *gain*, *A*, *W* were found. Significant interactions of *gain*  $\times$  *A*, *gain*  $\times$  *W*, and *A*  $\times$  *W* were also found. Results are summarized in Table 7.6.

We performed a linear regression using data aggregated from all participants. Regression constants calculated using a Fitts' model ( $MT = a + b \log_2(2A/W)$ ), a Shannon model ( $MT = a + b \log_2(A/W + 1)$ ), and a Welford two-part model ( $MT = a + b_1 \log A - b_2 \log W$ ) are shown in Table 7.7. In order to adjust for accuracy we performed a second analysis of the results, this time using effective target size  $W_e$ . Soukoreff and I.S. MacKenzie [186] argue that effective width, as computed using the distribution of click events rather than the actual target size, more accurately represents the task actually performed. These results are presented in

Gain	Fitts ( $2A/W$ )			Shannon ( $A/W + 1$ )			Welford				
	a	b	$R^2$	a	b	$R^2$	a	$b_1$	$b_2$	k	$R^2$
2	0.070	0.233	0.991	0.084	0.286	0.998	0.269	0.236	0.229	0.97	0.992
5	0.089	0.204	0.989	0.100	0.252	0.998	0.336	0.200	0.209	1.05	0.989
8	0.076	0.209	0.982	0.087	0.258	0.994	0.388	0.198	0.221	1.12	0.985
12	0.032	0.242	0.980	0.082	0.282	0.990	0.402	0.232	0.263	1.13	0.983
16	0.022	0.256	0.972	0.075	0.299	0.982	0.517	0.239	0.296	1.24	0.982
20	0.035	0.275	0.964	0.091	0.321	0.974	0.653	0.250	0.331	1.32	0.983
2-20	0.012	0.252	0.936	0.051	0.299	0.949	0.342	0.245	0.264	1.08	0.937

**Table 7.7:** Linear regression constants determined when using the Fitts formulation, the Shannon formulation, and the Welford two-part formulation. Movement times were averaged over all participants. Actual movement amplitude  $A$  and actual target width  $W$  were used.

Gain	Fitts ( $2A/W$ )			Shannon ( $A/W + 1$ )			Welford				
	a	b	$R^2$	a	b	$R^2$	a	$b_1$	$b_2$	k	$R^2$
2	0.133	0.223	0.989	0.132	0.280	0.994	0.214	0.240	0.210	0.875	0.993
5	0.041	0.228	0.982	0.049	0.285	0.990	0.274	0.227	0.228	1.00	0.982
8	-0.002	0.241	0.973	0.009	0.300	0.984	0.358	0.229	0.260	1.14	0.978
12	-0.050	0.278	0.937	0.000	0.328	0.947	0.613	0.264	0.365	1.38	0.961
16	-0.028	0.292	0.910	0.021	0.348	0.916	0.985	0.273	0.457	1.67	0.970
20	0.013	0.314	0.891	0.065	0.373	0.893	1.275	0.290	0.529	1.82	0.975
2-20	-0.017	0.278	0.861	0.013	0.336	0.873	0.332	0.274	0.291	1.06	0.862

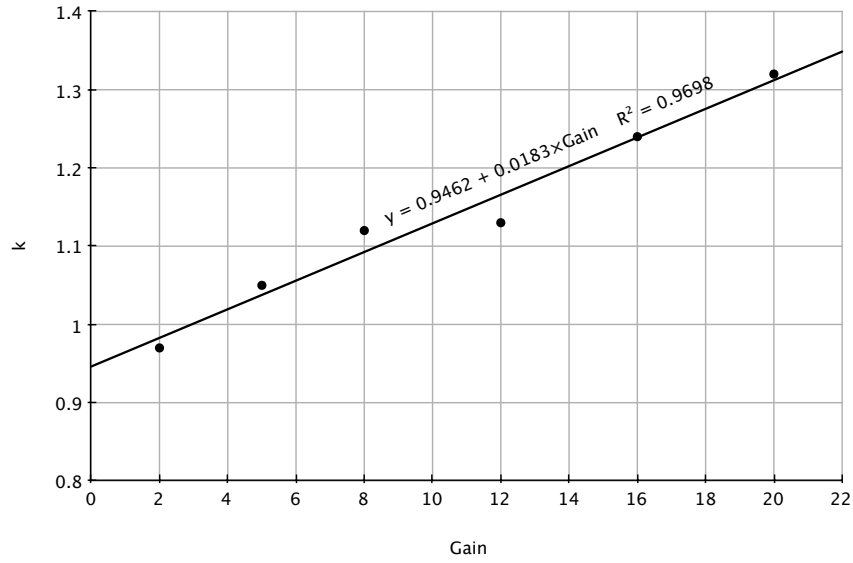
**Table 7.8:** Linear regression constants determined when using the Fitts formulation, the Shannon formulation, and the Welford two-part formulation. Movement times were averaged over all participants. Actual movement amplitude  $A$  and effective target width  $W_e$  values were used.

Table 7.8.

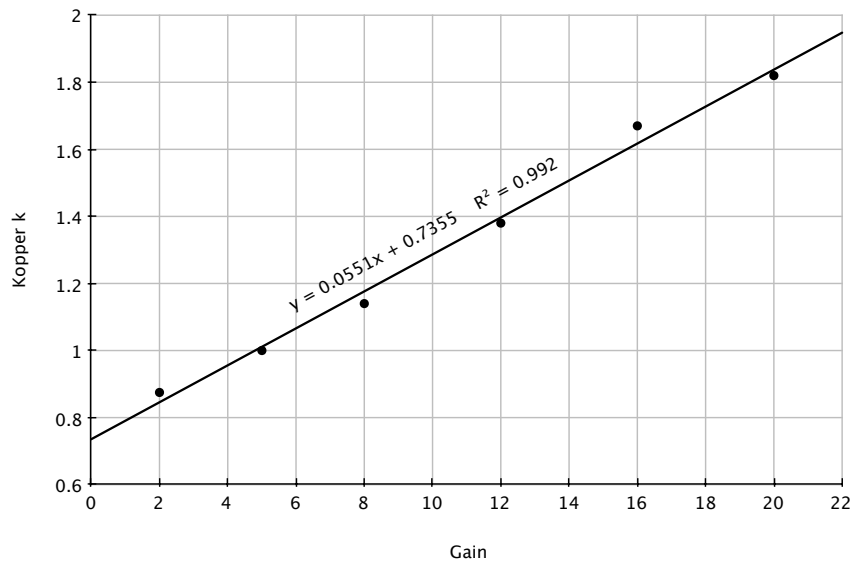
To test the hypothesis that the  $k$  value arising from the Welford two-part model will vary based on gain we performed a linear regression analysis on the  $k$  values computed for each gain level (Figure 7.14). The linear function of best fit was found to be  $k = 0.95 + 0.018 \times \text{gain}$ , with a fit of  $R^2 = 0.97$ . We recomputed the linear regression using effective width  $W_e$  (Figure 7.15). The linear function of best fit was found to be  $k = 0.735 + 0.055 \times \text{gain}$ , with a fit of  $R^2 = 0.99$ .

As with our analysis of the Casiez data, we ran  $F$ -tests to determine if the Welford model was significantly better in describing the data than was the Fitts model. Results using actual target width are summarized in Table 7.9. Results using effective target width are summarized in Table 7.10. Higher gains resulted in significant differences in the models, whereas lower gains didn't. This is expected, as  $k$  is close to unity at lower gains, and diverges from unity at higher gains.

Although  $k$  conveniently captures the *relative* contributions of  $A$  and  $W$  on performance, it can still be useful to investigate the *individual* contributions of  $A$



**Figure 7.14:** The  $k$  values relative to gain computed using actual  $A$  and  $W$ .



**Figure 7.15:** The  $k$  values relative to gain computed using actual  $A$  and effective  $W_e$ .

Gain	Fitts SSE	Welford SSE	n	SSE	F-ratio	p	Sig?
2	5721	5553	9	925.5	0.181	0.685	no
5	5813	5542	9	923.7	0.293	0.608	no
8	9437	7857	9	1309.5	1.207	0.314	no
12	31678	26531	12	2947.9	1.746	0.219	no
16	48920	31170	12	3463.3	5.125	0.049	yes
20	72757	35741	12	3971.2	9.321	0.013	yes

**Table 7.9:** Results of a statistical  $F$ -test comparing regressions using the Welford formulation to those using the Fitts formulation for our experimental data. Data analyzed was actual width data from the mid-air pointing experiment. For gain levels with significant results, the Welford formulation models the data significantly better than the Fitts formulation.

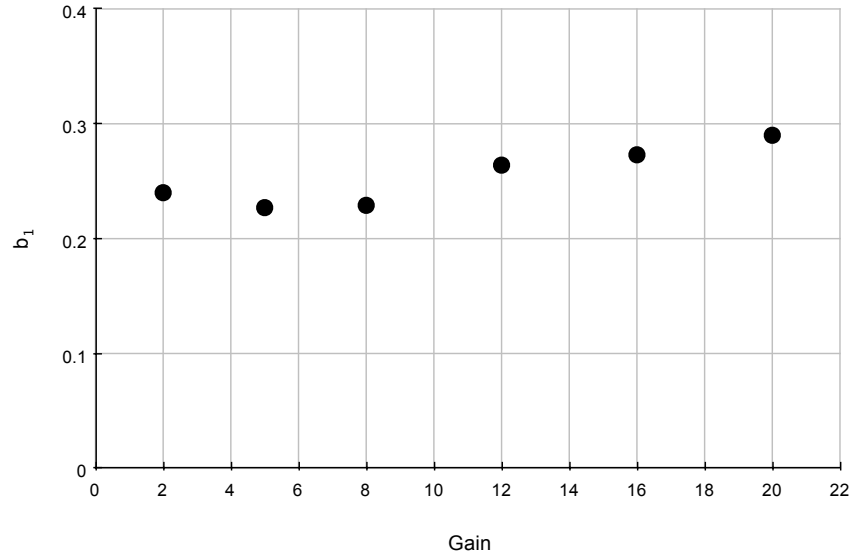
Gain	Fitts SSE	Welford SSE	n	SSE	F-ratio	p	Sig?
2	7471	4401	9	733.5	4.185	0.087	no
5	9193	9190	9	1531.7	0.001	0.966	no
8	14219	12029	9	2004.8	1.092	0.336	no
12	98234	60982	12	6775.8	5.50	0.044	yes
16	158752	52477	12	5830.8	18.23	0.002	yes
20	223278	51797	12	5755.2	29.80	< 0.001	yes

**Table 7.10:** Results of a statistical  $F$ -test comparing regressions using the Welford formulation to those using the Fitts formulation for our experimental data. Data analyzed was effective width data from the mid-air pointing experiment. For gain levels with significant results, the Welford formulation models the data significantly better than the Fitts formulation.

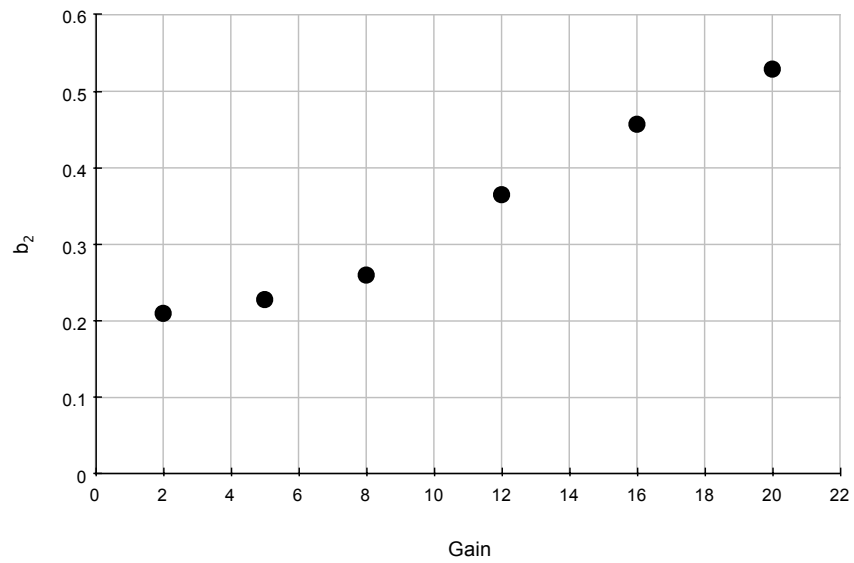
and  $W$  to movement time. Towards this goal we examined how both  $b_1$  and  $b_2$  from the Welford formulation varied dependent on gain. These results are shown in Fig. 7.16 and Fig. 7.17. What is revealed is that as gain changes  $b_2$  varies much more than does  $b_1$ .

### Error Rate

Mean error rates were found to be 7.8%. An ANOVA found significant main effects of  $gain$ ,  $A$ , and  $W$ . The interaction of  $gain \times W$  was also significant. Results are summarized in Table 7.11.



**Figure 7.16:** Dependence of  $b_1$  on gain.



**Figure 7.17:** Dependence of  $b_2$  on gain.



Factor	F-ratio	Significance	Partial $\eta^2$
gain	$F_{2.9,52.8} = 28.5^*$	$p < 0.001$	0.613
A	$F_{2,36} = 13.6$	$p < 0.001$	0.431
W	$F_{1.4,24.3} = 96.0^*$	$p < 0.001$	0.842
gain $\times$ W	$F_{4.2,75.8} = 16.6^*$	$p < 0.001$	0.480

\*Greenhouse-Geisser correction for violated sphericity applied.

**Table 7.11:** Significant ANOVA results for error rate in the mid-air large display pointing experiment.

### Subjective Measures

A summary of results from participants' subjective ratings of the difficulty of the task is shown in Figure 7.18. A Friedman test comparing ratings for low (2, 5), medium (8, 12) and high gain levels (16, 20) showed a significant impact of gain on difficulty ( $\chi^2_{(2,N=19)} = 30.958, p < 0.001$ ). Pairwise comparisons using a Wilcoxon Signed Ranks Test showed significant differences between high and low gains ( $z = -3.882, p < 0.001$ ) and high and medium gains ( $z = -3.882, p < 0.001$ ).

### 7.4.6 Discussion

It is useful to first discuss the results of the ANOVA on the movement time data.

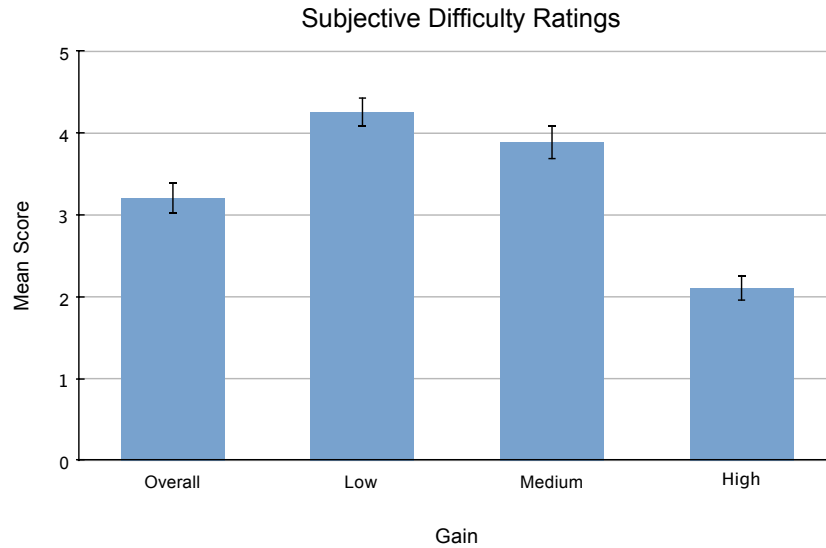
The finding that there was a significant effect of both  $A$  and  $W$  on movement time is not surprising. The fact that both distance between targets and size of targets impact movement time is fundamental to any discussion of the speed-accuracy trade-off.

The finding of a significant impact of  $gain$  is also unsurprising because similar effects have been found by other researchers [32, 120, 132].

The three significant interactions ( $gain \times A$ ,  $gain \times W$ , and  $A \times W$ ) are similarly as anticipated.

The interaction of  $A \times W$  is explained, even when using a Fitts formulation, by the logarithmic nature of  $ID$ . Changing  $A$  (or  $W$ ) by a fixed amount is not expected to result in a fixed change in movement time.

The interactions of  $gain \times A$  and  $gain \times W$  are explained by the expectation that, given a main effect of  $gain$ , changing  $A$  will not produce fixed changes in movement time at different gain levels.



**Figure 7.18:** Mean scores of task difficulty overall, at low gain (2, 5), medium gain (8, 12) and high gain (16, 20) levels, with standard error. Ratings on a scale of one (impossible) to five (easy). N=19.

We summarize the results according to our hypotheses, based on the results of the linear regressions.

**H1** The Fitts formulation will not accurately model pointing performance at all gain levels. *Somewhat supported.*

**H2** The Welford two-part formulation will accurately model pointing performance at each individual gain level. *Supported.*

**H3a** (*weak*) The exponent  $k$  will vary monotonically with gain. *Supported.*

**H3b** (*strong*) The exponent  $k$  will vary linearly with gain. *Supported.*

Fitts' one-part model of pointing performance had mixed success in characterizing pointing performance. Using actual  $W$  values, Fitts formulation gave linear fits ranging in accuracy from a low of  $R^2 = 0.964$  to  $R^2 = 0.991$  at different levels of gain. For the levels of gain examined these  $R^2$  values are good, surpassing the somewhat arbitrary 0.9 threshold. However it is clear that the  $R^2$  values are

decreasing as gain increases. The Shannon formulation fares better at modelling the data. It provides very good  $R^2$  fits, but it is clear that the quality of fit is again decreasing at higher levels of gain.

Using effective  $W$ , Fitts' law is less successful. Using the Fitts formulation, linear quality of fit in this case ranges from a low of  $R^2 = 0.861$  to a high of  $R^2 = 0.989$  at different levels of gain, failing to produce acceptable linear fits at some levels of gain. Again, the Shannon formulation fares better than the Fitts formulation, but performance again drops off at higher gains. It is these results that are more significant, due to the nature of effective width as an accurate representation of the task. The reason for the good fits using actual  $W$  and the poor fits using effective  $W$  is evident from Figures 7.14 and 7.15. It is clear that the slope of the dependence of  $k$  on gain is much lower for actual width than for effective width. Thus,  $k$  does not deviate nearly as much from unity for actual width as it does for effective width.

We thus conclude that hypothesis **H1** is somewhat supported. The results behaved as expected, however, in the case of the actual width analysis the contributions of  $A$  and  $W$  to performance did not differ enough to result in a poor fit using Fitts' original formulation, at least in the range of gains examined. There is more support for the hypothesis if effective width is used.

Welford's two-part model of pointing performance produced a good fit at each level of gain for both actual widths and effective widths. Linear regressions for actual widths ranged from a low of  $R^2 = 0.982$  to a high of  $R^2 = 0.992$ . For effective width linear regressions ranged from a low of  $R^2 = 0.961$  to a high of  $R^2 = 0.993$ . Thus, hypothesis **H2** is supported. It is worth noting that Welford's model did not produce a good fit when all data at all levels of gain were analyzed together ( $R^2 = 0.862$ ). This suggests that, even when using Welford's model, each level of gain should be modelled separately.

The  $k$  values were observed to vary monotonically and linearly, according to gain, supporting hypothesis **H3a** and **H3b**. For actual width results, the  $k$  values followed a linear model to an accuracy of  $R^2 = 0.970$ . For the effective width results, the  $k$  values followed a linear model to an accuracy of  $R^2 = 0.992$ . Interestingly, the slopes for the two sets of results were noticeably different, with  $k$  varying more in the effective width set of data. The intercept of the slope at  $gain = 0$  was also noticeably different, although  $gain = 0$  is meaningless in an interactive setting,

suggesting that the intercept may not be of much significance.

As part of our analysis we examined how  $b_1$  and  $b_2$  vary depending on gain (Figure 7.16 and 7.17). It is interesting to note that  $b_2$  increases quite consistently with gain, while  $b_1$  stays relatively constant. This would suggest that the effect of  $W$  on decreasing movement time tends to dominate over the effect of  $A$  to increase movement times, at higher gains. This is an issue to investigate in depth in future work.

## 7.5 Conclusions

Fitts' law has been widely used as a tool for analyzing the performance of pointing tasks on computer systems, both for forming predictive models and for determining performance as characterized by throughput. Over the years Fitts' law has become so entrenched that researchers rarely ever question the fundamental assumptions underlying the use of Fitts' law, most significantly whether or not there are limitations to its applicability to modelling pointing on interactive displays where the control-display gain varies widely.

Our conclusions fall into two categories. First, from a re-analysis of results reported by other researchers we were able to develop a deeper understanding of how Fitts' applies to pointing on different types of interactive displays. Second, results of our own experiment provide a theoretical model of mid-air pointing on a very large wall display that is more accurate than a standard Fitts' law explanation.

### 7.5.1 Rethinking Fitts' Law for Modelling Pointing on Interactive Displays

A perhaps surprising conclusion is that Fitts' law is fragile when applied to computing pointing. We determined this through a re-analysis of data obtained from other researchers. First, we explored a number of different models of pointing performance. Several variations on Fitts' original formulation have been well explored in the HCI literature, but we added a discussion of Welford's two-part formulation that was originally developed to account for shortcomings in Fitts' model. We were also able to relate a model described by Kopper et al. [121] directly to the Welford two part formulation. Importantly, we were able to relate a constant  $k$  to the coef-

ficients  $b_1$  and  $b_2$  in Welford's formulation. We concluded that  $k$  is a convenient means of capturing the relative contributions of  $A$  and  $W$  to movement time.

An analysis of data from Graham [69] showed that Fitts' law does not accurately model absolute pointing with a tracked finger at constant gain on small displays. Our subsequent analysis of his data reveals that instead the Welford two-part formulation is required to accurately model pointing performance. Our analysis of data from Casiez et al. [32] revealed that Fitts' law also fails to accurately model mouse pointing at constant gain on a large display, with particular shortcomings at lower gain levels. In this case as well it was found that the Welford two-part formulation accurately modelled pointing performance at all gain levels. Furthermore, a linear dependence of  $k$  on gain was observed. Our analysis of data from Tsukitani et al. [199] showed a similar pattern to those found in the other data, although the linear regression of  $k$  showed a different intercept with the  $k = 1$  line.

Our conclusions regarding the modelling of pointing tasks in general on interactive displays are as follows: Fitts' law cannot be relied on to accurately model pointing performance in computing systems, especially at widely varying levels of gain. Welford's two-part formulation of pointing performance corrects for the discovered shortcomings of Fitts' law. Movement amplitude  $A$  and target width  $W$  are both of significance in modelling performance, and should be considered as separable, independent variables when developing models of pointing performance. The relative contributions of  $A$  and  $W$ , as captured by  $k$ , appear to vary linearly with gain, although the intercept of the line, and possibly the slope, vary by interaction technique.

### **7.5.2 Developing a Model for Interaction with Very Large Wall Displays**

We applied the conclusions of our re-analysis towards developing a model of mid-air pointing on a very large wall display at a variety of gain levels. Our experimental results demonstrated that Welford's two-part model accurately models pointing performance at all gain levels explored, whereas Fitts' model produces mixed results. In isolation our experimental results might be not quite convincing, due to the fairly shallow  $k$  slope, but when considered in the context of our re-analysis of others' data (especially the Casiez data), the results are compelling. As with

the Casiez data, we discovered a linear dependency of  $k$  on gain. The fact that the slopes and intercepts of the  $k$  lines vary between our data and Casiez's is interesting, and worthy of future consideration.

The two most valuable analyses of the chapter were those of the Casiez data and our new experimental data. These two data sets provided a good coverage of  $A/W$  pairings over a wide range of gains, with a suitable number of participants. The findings of these two experiments were consistent, that  $k$  increases linearly with gain, although the intercept and possibly the slope of the  $k$  line vary. The data from the Graham and Tsukitani analyses lend some support, although the lack of data points, and possibly the small number of participants, limit their contribution.

The work presented in this chapter was an essential component of the dissertation, in that it provided a validated theoretical basis for the further development of interaction techniques for large displays and other situations where gain may vary widely. The work has more widespread relevance, however, and can stand alone as a significant contribution to HCI. The need to re-evaluate the use of Fitts law as a tool, and our discovery of some concrete shortcomings of Fitts, impacts how we interpret thirty years of past research, and should influence researchers in how they design and analyze future evaluations of pointing performance.

### 7.5.3 Future Work

With the realization of the shortcomings of Fitts' law in some situations, and the applicability of a two-part Welford model of pointing performance, it is clear that there is much work still to be done. First, researchers may need to reevaluate conclusions that were long thought to be sound. Existing models of pointing performance for some input device or display combinations may be faulty, due to limitations in either experimental design or analysis.

Our knowledge of how  $k$  varies with gain is only partially complete. It appears that  $k$  varies linearly with gain, but it also appears that the linear function differs for different input approaches. For example, at  $gain = 2$  the Casiez et al.  $k$  was at a very low value of 0.091, whereas for our large display mid-air pointing task,  $k$  was at 0.875 for the same gain level. It is important that we examine the  $k$  dependence for different devices and display types. We should also strive to develop a deeper

understanding of the influencing factors on  $k$ . For example, Casiez hypothesized that mouse clutching was responsible for the deviation of their data from Fitts' law. We may find that clutching is a variable that influences the  $k$  slope and intercept. There are also almost certainly other factors, because none of the other pointing methods we investigated allowed clutching, yet the  $k$  formulation was still found to be relevant.

The significance of  $k$  to other variables relevant to pointing should also be investigated. We may discover that  $k$  itself is a primary measure of pointing performance, on par with gain. Our introduction of  $k$  as a parameter was motivated by Kopper et al., who introduced it for their angular formulation. It is worth noting that  $k$  was implicit in the earlier Welford two-part formulation. As we demonstrated, the linear analog to the Kopper formulation follows directly from a mathematical manipulation of Welford's formula. All we did was make  $k$  explicit.

As we noted earlier in this chapter, alternate theoretical models for physical pointing exist. The work of Schmidt et al. [177] is a good candidate for further investigations that might explore how gain can be incorporated into their model, and whether additional changes are required for computer-mediated physical pointing on large screens. Just as we saw benefits in revisiting the assumptions about Fitts's law, and how the Welford two-part formulation already accommodates gain as an implicit parameter, it may be that the model of Schmidt et al. has a similar potential.

Related to our particular interest in large wall displays, there is again much work to be done. We limited our examination to constant control-display gains. Variable gains, such as the pointer acceleration investigated by Casiez et al., are important to consider. This may allow us to expand our model of the Welford two-part model and the  $k$  relationship to include pointer acceleration.

## Chapter 8

# Conclusions

In this dissertation, we have argued that a body-centric model for interaction is well suited for use with very large wall displays. We started by observing that while traditional interaction techniques and form factors, i.e. mouse and keyboard with a monitor on a desk, are well suited to some use cases, they are poor at supporting many other kinds of tasks. We also observed that the limitations of traditional computing systems are largely due to the technical limitations of early hardware.

In our discussion of related work, we discovered that there has been substantial research into the development of interfaces based on real-world physical behaviour, in particular how human brains perceive the world and how human bodies function in the real world. These topics have been the basis for investigations into reality-based interfaces and whole body interfaces. In particular, we concluded that the themes explored in these research areas are particularly relevant to use with very large wall displays, because large displays are (a) often used collaboratively, (b) are often used while users are standing and moving about, and (c) are at a scale consistent with a human body.

We set out to explore a theory of body-centric interaction through investigations in three relevant sub-areas. These areas are: the development of novel interaction techniques, the design of supporting interaction architectures, and investigations into low-level human performance properties. These areas are all inter-related, and conclusions from one area serve to inform advances in the others. Indeed, conclusions regarding the applicability of body-centric interaction cannot be made



without an understanding of all of these areas. We summarize our conclusions as they relate to these three areas, and then draw some more general conclusions.

## **8.1 Interaction Techniques**

In Chapters 3, 4, and 5 we described a number of novel body-centric interaction techniques developed specifically for use with very large wall displays. The design of these techniques was founded on literature from psychology and sociology, in particular results related to how people perceive the world in different spaces, and how people relate socially to one another.

Our early investigations into the Shadow Reaching technique explored the use of a shadow embodiment of users, to overcome the problems of distance reaching and awareness support that are particular to very large wall displays. We concluded that this approach was powerful, and opened up the potential for the development of more interaction techniques, using Shadow Reaching as a foundation.

Our later investigations built a host of interaction techniques on top of the Shadow Reaching metaphor. These interaction techniques included methods for selecting modes, managing personal data, exchanging data with collaborators, adjusting numerical values, and managing the presentation of user embodiments. All of this was completed with a minimum use of arbitrary icons, and a maximum use of physical body movement and real-world representations.

We also performed investigations into techniques for text input. Text input is a relevant topic of investigation due both to the fact that it is a nearly universal task in computing systems, and because language has no physical equivalent, so it is not clear how it might be integrated directly into a body-centric interaction approach. We concluded that techniques can be developed that support text input in our use context consistent with a body-centric framework. Our techniques allowed for free user motion in a space, and supported input using hands in mid-air, without the requirement of specialized input devices such as mice or keyboards.

## **8.2 Interaction Architecture**

In Chapter 6 we investigated the practicalities of supporting the development of body-centric interaction techniques. We framed an approach that stresses the use

of a computational representation of the geometry of the scene, including displays, users, and other relevant contextual items. We concluded that this approach can provide a convenient generic interface for use by application developers, but that the development of such an architecture would be difficult with current technology.

We described an initial implementation of a subset of the features described in our theoretical design. This implementation supported the development of the interaction techniques described in Chapter 4. Our implementation tracked users using either Polhemus Liberty Latus magnetic markers, or vision tracked coloured balls. It also supported the modelling of virtual light sources by different lighting behaviours.

### **8.3 Theoretical Models of Performance**

We derived our first theoretical model from our work on text input, described in Chapter 5. In that work we were able to isolate the impact of user distance from the display on performance of text input techniques that are either distance-dependent or distance-independent. We concluded, first, that while distance-independent techniques are invariant with distance in motor space, performance nevertheless degrades as user distance to the display increases. This may be due to the change in perceived size of the visual feedback on the display. Second, we concluded that distance-dependent techniques degrade in performance at a faster rate than distance-independent techniques as distance increases. Third, we were able to determine, in the case of our two distance-independent and distance-dependent text input techniques, the distance from the display at which the distance-independent technique outperformed the normally superior distance-dependent technique. These findings have potentially broader implications on distance-dependency, outside of just text input.

Our second theoretical model related to our investigation of Fitts' law pointing performance on large displays. We first demonstrated, through a re-analysis of data from other researchers, that Fitts' law possesses previously unappreciated limitations in modelling computer pointing, and that these limitations appear to be closely tied to control-display gain. In analyzing the data we found that a two-part model of pointing performance, due to Welford, corrects for these issues. In our

own investigations of mid-air pointing on a large display we observed a similar pattern of performance, with Fitts' model falling short of providing accurate predictions, but Welford's model correctly predicting performance. Our conclusions on pointing performance are relevant to body-centric interaction with large wall displays, but have potentially broad implications to all pointing interactions within computing systems.

## **8.4 Final Words**

The need for a well-defined body-centric model for interaction on large wall displays is indicated strongly by numerous observations in the field of human-computer interaction. We have undertaken to fill this need, however, the development of such a model is an ongoing process. We make no claim to have performed a complete investigation into the topic, but we believe we have performed a rigorous investigation into the critical areas of research that will eventually coalesce into a unified model of interaction.

The study of body-centric interaction, both for large wall displays and other form factors, is likely to continue. We believe that the true potential for this area of research will become clear as superior sensing technologies mature, and once an interaction architecture similar to our BAPI is adopted. This will usher in a period where researchers and designers are free to explore ideas with minimum hindrance from development difficulties. We hope that in the event of a "golden age" of body-centric interaction researchers will take heed of what we have learned here in regards to theoretical foundations for interaction techniques, the design of interaction techniques themselves, and the importance of developing accurate theoretical models of performance.

# Bibliography

- [1] ABLA, G., FLANAGAN, S., PENG, Q., BURRUSS, J., AND SCHISSEL, D. Advanced tools for enhancing control room collaborations. *Fusion Engineering and Design* 81 (2006), 2039–2044.
- [2] ABOWD, G., AND DIX, A. Integrating status and event phenomena in formal specifications of interactive systems. In *Proceedings of SIGSOFT '94* (1994).
- [3] AKAMATSU, M., MACKENZIE, I. S., AND HASBROUQ, T. A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics* 38 (1995), 816–827.
- [4] AKAOKA, E., GINN, T., AND VERTEGAAL, R. DisplayObjects: Prototyping functional physical interfaces on 3D styrofoam, paper or cardboard models. In *Proceedings of TEI '10* (2010), pp. 49–56.
- [5] AMMA, C., GEHRIG, D., AND SCHULTZ, T. Airwriting recognition using wearable motion sensors. In *Proceedings of the 1st Augmented Human International Conference* (2010), pp. 1–8.
- [6] APPERLEY, M., MACLEOD, L., MASOODIAN, M., PAINE, L., PHILLIPS, M., ROGERS, B., AND THOMSON, K. Use of video shadow for small group interaction awareness on a large interactive display surface. In *Proceedings of AUIC 2003* (2003), pp. 81–90.
- [7] APPLE COMPUTER INC. *Macintosh human interface guidelines*. Addison-Wesley, 1992.
- [8] ARNAUT, L. Y., AND GREENSTEIN, J. S. Is display/control gain a useful metric for optimizing an interface? *Human Factors: The Journal of the Human Factors and Ergonomics Society* 13 (1990), 651–663.

- [9] BAILEY, R. N. *Human performance engineering: Designing high quality, professional user interfaces for computer products, applications, and systems*, 3rd ed. Prentice Hall, 1996.
- [10] BAUDEL, T., AND BEAUDOUIN-LAFON, M. Charade: Remote control of objects using free-hand gestures. *Communications of the ACM* 36, 7 (1993), 28–35.
- [11] BAUDISCH, P., SINCLAIR, M., AND WILSON, A. Soap: A pointing device that works in mid-air. In *Proceedings of UIST '06* (2006), pp. 43–46.
- [12] BAUM, L. E., AND PETRIE, T. Statistical inference for probabilistic functions of finite state markov chains. *Annals of Mathematical Statistics* 37, 6 (1966), 1554–1563.
- [13] BERNARDIN, K., AND STIEFELHAGEN, R. Audio-visual multi-person tracking and identification for smart environments. In *Proceedings of MULTIMEDIA '07* (2007), pp. 661–670.
- [14] BEWLEY, W. L., ROBERTS, T. L., SCHROIT, D., AND VERPLANK, W. L. Human factors testing in the design of Xerox's 8010 "Star" office workstation. In *Proceedings of CHI '83* (1983), pp. 72–77.
- [15] BEZERIANOS, A., AND BALAKRISHNAN, R. The Vacuum: Facilitating the manipulation of distant objects. In *Proceedings of CHI '05* (2005), pp. 361–370.
- [16] BIER, E. A., STONE, M. C., PIER, K., BUXTON, W., AND DEROSE, T. D. Toolglass and Magic Lenses: The see-through interface. In *Proceedings of SIGGRAPH '93* (1993), pp. 73–80.
- [17] BOLT, R. A. "Put-that-there": Voice and gesture at the graphics interface. In *Proceedings of SIGGRAPH '80* (1980), pp. 262–270.
- [18] BORING, S., BAUR, D., BUTZ, A., GUSTAFSON, S., AND BAUDISCH, P. Touch Projector: Mobile interaction through video. In *Proceedings of CHI 2010* (2009), pp. 2287–2296.
- [19] BORITZ, J., BOOTH, K. S., AND COWAN, W. B. Fitts' law studies of directional mouse movement. In *Proceedings of Graphics Interface '91* (1991), pp. 216–223.
- [20] BRAGDON, A., ZELEZNIK, R., WILLIAMSON, B., MILLER, T., AND LAVIOLA, JR., J. J. GestureBar: Improving the approachability of

- gesture-based interfaces. In *Proceedings of CHI '09* (2009), pp. 2269–2278.
- [21] BRUBAKER, M. A., SIGAL, L., AND FLEET, D. J. Video-based people tracking. In *Handbook of Ambient Intelligence and Smart Environments*, H. Nakashima, H. Aghajan, and J. C. Augusto, Eds. Springer US, 2010, pp. 57–87.
- [22] BURGOON, J. K., BULLER, D. B., HALE, J. L., AND DETURCK, M. A. Relational messages associated with nonverbal behaviors. *Human Communication Research* 10, 3 (1984), 351–378.
- [23] BUXTON, B. Surface and tangible computing, and the “small” matter of people and design. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers* (2008), pp. 24–29.
- [24] BUXTON, W. A three-state model of graphical input. In *Proceedings of INTERACT '90* (1990), pp. 449–456.
- [25] CAO, X., AND BALAKRISHNAN, R. VisionWand: Interaction techniques for large displays using a passive wand tracked in 3D. In *Proceedings of UIST '03* (2003), pp. 173–182.
- [26] CAO, X., LI, J. J., AND BALAKRISHNAN, R. Peephole pointing: Modeling acquisition of dynamically revealed targets. In *Proceedings of CHI '08* (2008), pp. 1699–1708.
- [27] CAO, X., WILSON, A. D., BALAKRISHNAN, R., HINCKLEY, K., AND HUDSON, S. E. ShapeTouch: Leveraging contact shape on interactive surfaces. In *Proceedings of TABLETOP '08* (2008), pp. 139–146.
- [28] CARD, S. K., ENGLISH, W. K., AND BURR, B. J. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for selection on a CRT. *Ergonomics* 21 (1978), 601–613.
- [29] CARD, S. K., MACKINLAY, J. D., AND ROBERTSON, G. G. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems* 9, 2 (1991), 99–122.
- [30] CARD, S. K., MORAN, T. P., AND NEWELL, A. *The psychology of human-computer interaction*. Lawrence Erlbaum Associates, 1983.
- [31] CARDINALI, L., BROZZOLI, C., AND FARNÈ, A. Peripersonal space and body schema: Two labels for the same concept? *Brain Topology* 21 (2009), 252–260.

- [32] CASIEZ, G., VOGEL, D., BALAKRISHNAN, R., AND COCKBURN, A. The impact of control-display gain on user performance in pointing tasks. *Human-Computer Interaction* 23, 3 (2008), 215–250.
- [33] CASTELLUCCI, S. J., AND MACKENZIE, I. S. UniGest: Text entry using three degrees of motion. In *Extended Abstracts of CHI '08* (2008), pp. 3549–3554.
- [34] CHENG, R., KALASHNIKOV, D. V., AND PRABHAKAR, S. Evaluating probabilistic queries over imprecise data. In *Proceedings of SIGMOD '01* (2001).
- [35] CHENG, R., AND PRABHAKAR, S. Managing uncertainty in sensor database. *SIGMOD Record* 32, 4 (2003), 41–46.
- [36] CHERUBINI, M., VENOLIA, G., DELINE, R., AND KO, A. J. Let's go to the whiteboard: How and why software developers use drawings. In *Proceedings of CHI '07* (2007), pp. 557–566.
- [37] COCCHINI, G., BESCHIN, N., AND JEHKONEN, M. The fluff test: A simple task to assess body representation neglect. *Neuropsychological Rehabilitation* 11, 1 (2001), 17–31.
- [38] COCKBURN, A., AND BROCK, P. Human on-line response to visual and motor target expansion. In *Proceedings of Graphics Interface '06* (2006), pp. 81–87.
- [39] CODD, E. F. A relational model of data for large shared data banks. *Communications of the ACM* 13, 6 (1970), 377–387.
- [40] COLBY, C. L. Action-oriented spatial reference frames in cortex. *Neuron* 20 (1998), 15–24.
- [41] COLLOMB, M., HASCOËT, M., BAUDISCH, P., AND LEE, B. Improving drag-and-drop on wall-size displays. In *Proceedings of Graphics Interface 2005* (2005), pp. 25–32.
- [42] COMSTOCK, J., JONES, L., AND POPE, A. The effectiveness of various altitude indicator display sizes and extended horizon lines on altitude maintenance in a part-task simulation. In *Proceedings of the Human Factors and Ergonomics Society* (2003), pp. 108–112.
- [43] CROWLEY, J., AND DEMAZEAU, Y. Principles and techniques for sensor data fusion. *Signal Processing* 32, 1–2 (1993), 5–27.

- [44] CZERWINSKI, M., SMITH, G., REGAN, T., MEYERS, B., ROBERTSON, G., AND STARKWEATHER, G. Toward characterizing the productivity benefits of very large displays. In *Proceedings of Interact '03* (2003), pp. 9–16.
- [45] DANIEL J. GOBLE, S. H. B. Task-dependent asymmetries in the utilization of proprioceptive feedback for goal-directed movement. *Exp Brain Research* 180 (2007), 693–704.
- [46] DARLING, W. G., AND MILLER, G. F. Transformations between visual and kinesthetic coordinate systems in reaches to remembered object locations and orientations. *Exp Brain Res* 93 (1993), 534–547.
- [47] DIAZ-MARINO, R., AND GREENBERG, S. The Proximity Toolkit and ViconFace: The video. In *Extended Abstracts of CHI '10* (2010), pp. 4793–4798.
- [48] DOUGLAS, S., KIRKPATRICK, A., AND MACKENZIE, I. Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard. In *Proceedings of CHI '99* (1999), pp. 215–222.
- [49] DRAGICEVIC, P., AND FEKETE, J.-D. The input configurator toolkit: Towards high input adaptability in interactive applications. In *Proceedings of AVI '04* (2004), pp. 244–247.
- [50] DUCHOWSKI, A. T. *Eye Tracking methodology: Theory and practice*, 2nd ed. Springer, 2007.
- [51] ECHTLER, F., AND KLINKER, G. A multitouch software architecture. In *Proceedings of NordiCHI '08* (2008), pp. 463–466.
- [52] ENGELBART, D. C., AND ENGLISH, W. K. A research center for augmenting human intellect. In *Proceedings of the AFIPS '68 fall joint computer conference* (1968), pp. 395–410.
- [53] ENGLAND, D., RANDLES, M., FERGUS, P., AND TALEB-BENDIAB, A. Towards an advanced framework for whole body interaction. In *Virtual and Mixed Reality*, R. Shumaker, Ed., vol. 5622 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009, pp. 32–40.
- [54] EPHRAIM, Y., AND MERHAV, N. Hidden Markov processes. *IEEE Transactions on Information Theory* 48, 6 (2002), 1518–1569.



- [55] EPPS, B. W. Comparison of six cursor control devices based on Fitts' law models. In *Proceedings of the Human Factors Society 30th annual meeting* (1986), pp. 327–331.
- [56] ETFORCASTS. Worldwide pda and smartphone forecast. [http://www.etforecasts.com/products/ES\\_pdas2003.htm](http://www.etforecasts.com/products/ES_pdas2003.htm), Accessed Dec 2010.
- [57] FASS, A., FORLIZZI, J., AND PAUSCH, R. MessyDesk and MessyBoard: Two designs inspired by the goal of improving human memory. In *Proceedings of DIS '02* (2002), pp. 303–311.
- [58] FELIPE, N. J., AND SOMMER, R. Invasions of personal space. *Social Problems* 14, 2 (1966), 206–214.
- [59] FINKE, M., KAVIANI, N., WANG, I., TSAO, V., FELS, S., AND LEA, R. Investigating distributed user interfaces across interactive large displays and mobile devices. In *Proceedings of AVI '10* (2010), pp. 413–413.
- [60] FITTS, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 6 (1954), 381–391.
- [61] FOCKEN, D., AND STIEFELHAGEN, R. Towards vision-based 3-D people tracking in a smart room. In *Proceedings of ICMI '02* (2002).
- [62] FOLEY, J. D., WALLACE, V. L., AND CHAN, P. The human factors of computer graphics interaction techniques. *IEEE Comput. Graph. Appl.* 4, 11 (1984), 13–48.
- [63] FORLINES, C., AND BALAKRISHNAN, R. Evaluating tactile feedback and direct vs. indirect stylus input in pointing and crossing selection tasks. In *Proceedings of CHI '08* (2008), pp. 1563–1572.
- [64] FORLINES, C., WIGDOR, D., SHEN, C., AND BALAKRISHNAN, R. Direct-touch vs. mouse input for tabletop displays. In *In Proceedings of CHI '07* (2007), pp. 647–656.
- [65] FOWLER, B., MEEHAN, S., AND SINGHAL, A. Perceptual-motor performance and associated kinematics in space. *Human Factors* 50, 6 (2008), 879–892.
- [66] GANDEVIA, S. C., REFSHAUGE, K. M., AND COLLINS, D. F. Proprioception: Peripheral inputs and perceptual interactions. *Adv Exp Med Biol* 508 (2002), 61–68.

- [67] GARTNER RESEARCH. Competitive landscape: Mobile devices, worldwide, 1Q10, May 2010.
- [68] GEISSLER, J. Shuffle, throw or take it! Working efficiently with an interactive wall. In *Proceedings of CHI '98* (1998), pp. 265–266.
- [69] GRAHAM, E. D. *Pointing on a computer display*. Doctoral dissertation, Simon Fraser University, 1996.
- [70] GREENBERG, S., AND BUXTON, B. Usability evaluation considered harmful (some of the time). In *Proceedings of CHI '08* (2008), pp. 111–120.
- [71] GREENBERG, S., AND FITCHETT, C. Phidgets: Easy development of physical interfaces through physical widgets. In *Proceedings of UIST '01* (2001), pp. 209–218.
- [72] GREENBERG, S., AND ROUNDING, M. The notification collage: Posting information to public and personal displays. In *Proceedings of CHI '01* (2001), pp. 514–521.
- [73] GROSSMAN, T., AND BALAKRISHNAN, R. Pointing at trivariate targets in 3D environments. In *Proceedings of CHI '04* (2004), pp. 447–454.
- [74] GUIARD, Y. The problem of consistency in the design of Fitts' law experiments: Consider either target distance and width or movement form and scale. In *Proceedings of CHI '09* (2009), pp. 1809–1818.
- [75] GUIARD, Y., BEAUDOUIN-LAFON, M., AND MOTTET, D. Navigation as multiscale pointing: Extending Fitts' model to very high precision tasks. In *Proceedings of CHI '99* (1999), pp. 450–457.
- [76] GUTWIN, C., AND GREENBERG, S. A descriptive framework of workspace awareness for real-time groupware. *CSCW 11*, 3 (2001), 411–446.
- [77] HALL, D., AND LLINAS, J. An introduction to multisensor data fusion. *Proceedings of the IEEE 85* (1997), 6–23.
- [78] HALL, E. T. *The Hidden Dimension*. Peter Smith Publisher Inc, 1992.
- [79] HALLIGAN, P., AND MARSHALL, J. Left neglect in for near but not far space in man. *Nature* (350), 498–500.

- [80] HAN, J. Y. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of UIST '05* (2005), pp. 115–118.
- [81] HAN, S. H., JORNA, G. C., MILLER, R. H., AND TAN, K. C. A comparison of four input devices for the macintosh interface. In *Proceedings of the Human Factors Society 34th annual meeting* (1990), pp. 267–271.
- [82] HANSEN, T. E., HOURCADE, J. P., VIRBEL, M., PATALI, S., AND SERRA, T. PyMT: A post-WIMP multi-touch user interface toolkit. In *Proceedings of ITS '09* (2009), pp. 17–24.
- [83] HART, S., AND STAVELAND, L. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Human Mental Workload*, P. Hancock and N. Meshkati, Eds. North Holland Press, Amsterdam, 1988, pp. 139–183.
- [84] HARTMANN, B., KLEMMER, S. R., BERNSTEIN, M., AND L. Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of UIST '06* (2006), pp. 299–308.
- [85] HASCOET, M. Throwing models for large displays. *HCI 2* (2003), 73–77.
- [86] HINCKLEY, K., YATANI, K., PAHUD, M., CODDINGTON, N., RODENHOUSE, J., WILSON, A., BENKO, H., AND BUXTON, B. Manual deskterity: An exploration of simultaneous pen + touch direct input. In *Extended abstracts of CHI '10* (2010), pp. 2793–2802.
- [87] HODGINS, J., JÖRG, S., O'SULLIVAN, C., PARK, S. I., AND MAHLER, M. The saliency of anomalies in animated human characters. *ACM Trans. Appl. Percept.* 7, 4 (2010), 1–14.
- [88] HOFFMAN, E. A comparison of hand and foot movement times. *Ergonomics* 34 (1991), 397–406.
- [89] HOLLINGWORTH, H. L. The inaccuracy of movement. *Archives of Psychology* 13 (1909), 1–87.
- [90] HOLMES, N. P., SANABRIA, D., CALVERT, G. A., AND SPENCE, C. Tool-use: Capturing multisensory spatial attention or extending multisensory peripersonal space? *Cortex* 43 (2007), 469–489.
- [91] HOLMES, N. P., AND SPENCE, C. The body schema and multisensory representation(s) of peripersonal space. *Cognitive Processing* 5, 2 (2004), 94–105.

- [92] HORNOF, A. J. Visual search and mouse pointing in labeled versus unlabeled two-dimensional visual hierarchies. *ACM Transactions on Computer-Human Interaction* 8, 3 (2001), 171–197.
- [93] HUANG, E. M., MYNATT, E. D., RUSSELL, D. M., AND SUE, A. E. Secrets to success and fatal flaws: The design of large-display groupware. *IEEE CG&A* 26, 1 (2006), 37–45.
- [94] HUDON, S. E., AND MANKOFF, J. Rapid construction of functioning physical interfaces from cardboard, thumbtacks, tin foil and masking tape. In *Proceedings of UIST '06* (2006), pp. 289–298.
- [95] HUTTERER, P., AND THOMAS, B. H. Enabling co-located ad-hoc collaboration on shared displays. In *Proceedings of Australasian User Interface '08* (2008), pp. 43–50.
- [96] INKPEN, K. M. Drag-and-drop versus point-and-click mouse interaction styles for children. *ACM Transactions on Computer-Human Interfaces* 8, 1 (2001), 1–33.
- [97] INKPEN, K. M., BOOTH, K. S., GRIBBLE, S. D., AND KLAWE, M. Give and take: Children collaborating on one computer. In *Conference companion on Human factors in computing systems* (1995), pp. 258–259.
- [98] IRIKI, A., TANAKA, M., AND IWAMURA, Y. Coding of modified schema during tool use by macaque postcentral neurones. *Neuroreport* 7 (1996), 2325–2330.
- [99] ISHII, H., AND KOBAYASHI, M. ClearBoard: A seamless medium for shared drawing and conversation with eye contact. In *Proceedings of CHI '92* (1992), pp. 525–532.
- [100] JACKSON, R., AND FAGAN, E. Collaboration and learning within immersive virtual reality. In *Proceedings of Collaborative Virtual Environments* (2000), pp. 83–92.
- [101] JACOB, R. J., GIROUARD, A., HIRSHFIELD, L. M., HORN, M. S., SHAER, O., SOLOVEY, E. T., AND ZIGELBAUM, J. Reality-Based Interaction. In *Proceedings of CHI '08* (2008), pp. 201–210.
- [102] JAMES, C. L., AND REISCHEL, K. M. Text input for mobile devices: Comparing model prediction to actual performance. In *Proceedings of CHI '01* (2001), pp. 365–371.

- [103] JIANG, H., OFEK, E., MORAVEJI, N., AND SHI, Y. Direct pointer: Direct manipulation for large-display interaction using handheld cameras. In *Proceedings of CHI '06* (2006), pp. 1107–1110.
- [104] JIANG, X., HONG, J. I., TAKAYAMA, L. A., AND LANDAY, J. A. Ubiquitous computing for firefighters: Field studies and prototypes of large displays for incident command. In *Proceedings of CHI '04* (2004), pp. 679–686.
- [105] JONES, E., ALEXANDER, J., ANDREOU, A., IRANI, P., AND SUBRAMANIAN, S. GesText: Accelerometer-based gestural text-entry systems. In *Proceedings of CHI '10* (2010), pp. 2173–2182.
- [106] JONES, T. Psychology of computer use: XVI. Effect of computer-pointing devices on children's processing rate. *Perceptual and Motor Skills* 69 (1989), 1259–1263.
- [107] JONES, T. An empirical study of children's use of computer pointing devices. *Journal of Educational Computing Research* 7, 1 (1991), 61–76.
- [108] JOSEPH, A. D., AND KAASHOEK, M. F. Building reliable mobile-aware applications using the Rover Toolkit. *Wireless Networks* 3, 5 (1997), 405–419.
- [109] KALTENBRUNNER, M. reacTIVISION and TUIO: A tangible tabletop toolkit. In *Proceedings of ITS '09* (2009), pp. 9–16.
- [110] KATO, H., AND BILLINGHURST, M. Marker tracking and HMD calibration for a video-based augmented reality conference system. In *2nd IEEE and ACM International Workshop on Augmented Reality* (1999), pp. 85–94.
- [111] KERR, R. Movement time in an underwater environment. *Journal of Motor Behavior* 5 (1973), 175–178.
- [112] KESSLER, G. D., HODGES, L. F., AND WALKER, N. Evaluation of the CyberGlove as a whole-hand input devices. *ACM Transactions on Computer-Huan Interaction* 2, 4 (1995), 263–283.
- [113] KEULEN, R. F., ADAM, J. J., FISCHER, M. H., KUIPERS, H., AND JOLLES, J. Selective reaching: Evidence for multiple frames of reference. *J. Exp Psychol Hum Percept Perform* 28, 3 (2002), 515–526.

- [114] KHAN, A., FITZMAURICE, G., ALMEIDA, D., BURTONYK, N., AND KURTENBACH, G. A remote control interface for large displays. In *Proceedings of UIST '04* (2004), pp. 127–136.
- [115] KHAN, A., MATEJKA, J., FITZMAURICE, G., AND KURTENBACH, G. Spotlight: Directing users' attention on large displays. In *Proceedings of CHI '05* (2005), pp. 791–798.
- [116] KIN, K., AGRAWALA, M., AND DEROSE, T. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proceedings of Graphics Interface '09* (2009), pp. 119–124.
- [117] KIRSTEIN, C., AND MULLER, H. Interaction with a projection screen using a camera-tracked laser pointer. In *Proceedings of Multimedia Modeling '98* (1998), pp. 191–192.
- [118] KLEMMER, S., HARTMANN, B., AND TAKAYAMA, L. How bodies matter: Five themes for interaction design. In *Proceedings of DIS '06* (2006), pp. 140–149.
- [119] KLEMMER, S. R., NEWMAN, M. W., FARRELL, R., BILEZIKJIAN, M., AND LANDAY, J. A. The designers' outpost: A tangible interface for collaborative web site. In *Proceedings of UIST '01* (2001), pp. 1–10.
- [120] KOESTER, H. H., LOPRESTI, E., AND SIMPSON, R. C. Toward Goldilocks' pointing device: Determining a “just right” gain setting for users with physical impairments. In *Proceedings of Assets '05* (2005), pp. 84–89.
- [121] KOPPER, R., BOWMAN, D. A., SILVA, M. G., AND MCMAHAN, R. P. A human motor behavior model for distal pointing tasks. *International Journal of Human Computer Studies* (2010).
- [122] KRUEGER, M. W., GIONFRIDDO, T., AND HINRICHSSEN, K. VIDEOPLACE – an artificial reality. In *Proceedings of CHI '85* (1985), pp. 35–40.
- [123] LANDY, M. S., MALONEY, L. T., JOHNSTON, E. B., AND YOUNG, M. Measurement and modeling of depth cue combination: In defense of weak fusion. *Vision Research* 35, 3 (1995), 389–412.
- [124] LANGOLF, G. D., CHAFFIN, D. B., AND FOULKE, J. A. An investigation of Fitts' law using a wide range of movement amplitudes. *Journal of Motor Behavior* 8 (1976), 113–128.

- [125] LANIR, J., BOOTH, K. S., AND FINDLATER, L. Observing presenters' use of visual aids to inform the design of classroom presentation software. In *Proceeding of CHI '08* (2008), pp. 695–704.
- [126] LANIR, J., BOOTH, K. S., AND TANG, A. MultiPresenter: A presentation system for (very) large display surfaces. In *Proceedings of MultiMedia '08* (2008), pp. 519–528.
- [127] LI, F. C. Y., DEARMAN, D., AND TRUONG, K. N. Virtual shelves: Interactions with orientation aware devices. In *Proceedings of UIST '09* (2009), pp. 125–128.
- [128] MACKENZIE, C. L., MARTENIUK, R. G., DUGAS, C., LISKE, D., AND EICKMEIER, B. Three dimensional movement trajectories in Fitts' task: Implications for control. *Quarterly Journal of Experimental Psychology* 39, 4 (1987), 629–547.
- [129] MACKENZIE, I. S. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction* 7 (1992), 91–139.
- [130] MACKENZIE, I. S., AND BUXTON, W. Extending Fitts' law to two-dimensional tasks. In *Proceedings of CHI '92* (1992), pp. 219–226.
- [131] MACKENZIE, I. S., AND ISOKOSKI, P. Fitts' throughput and the speed-accuracy tradeoff. In *Proceedings of CHI '08* (2008), pp. 1633–1636.
- [132] MACKENZIE, I. S., AND RIDDERSMA, S. Effects of output display and control-display gain on human performance in interactive systems. *Behaviour and Information Technology* 13 (1994), 328–337.
- [133] MACKENZIE, I. S., SELLEN, A., AND BUXTON, W. A comparison of input devices in elemental pointing and dragging tasks. In *Proceedings of CHI 91* (1991), pp. 161–166.
- [134] MACKENZIE, I. S., AND SOUKOREFF, R. W. Phrase sets for evaluating text entry techniques. In *Extended Abstracts of CHI '03* (2003), pp. 754–755.
- [135] MACKENZIE, I. S., AND TANAKA-ISHII, K. *Text Entry Systems*. Morgan Kaufmann, 2007.

- [136] MACKENZIE, I. S., AND WARE, C. Lag as a determinant on human performance in interactive systems. In *Proceedings of CHI '93* (1993), pp. 488–493.
- [137] MAGERKURTH, C., AND STENZEL, R. A pervasive keyboard - separating input from display. In *Proceedings of PERCOM '03* (2003), p. 388.
- [138] MALONE, T. W. How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst.* 1, 1 (1983), 99–112.
- [139] MANKOFF, J., HUDSON, S. E., AND ABOWD, G. D. Interaction techniques for ambiguity resolution in recognition-based interfaces. In *Proceedings of UIST '00* (2000), pp. 11–20.
- [140] MANKOFF, J., HUDSON, S. E., AND ABOWD, G. D. Providing integrated toolkit-level support for ambiguity in recognition-based interfaces. In *Proceedings of CHI '00* (2000), pp. 368–375.
- [141] MARAVITA, A., SPENCE, C., SERGENT, C., AND DRIVER, J. Seeing your own touched hands in a mirror modulates cross-modal interactions. *Psychological Science* 13, 4 (2002), 350–355.
- [142] MASON, A. H., WALJI, M. A., LEE, E. J., AND MACKENZIE, C. L. Reaching movements to augmented and graphic objects in virtual environments. In *Proceedings of CHI '01* (2001), pp. 426–433.
- [143] MCCOURT, M. E., AND GARLINGHOUSE, M. Asymmetries of visuospatial attention are modulated by viewing distance and visual field elevation: Pseudoneglect in peripersonal and extrapersonal space. *Cortex* 36 (2000), 715–731.
- [144] MCGUFFIN, M. J., AND BALAKRISHNAN, R. Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM Transactions on Computer-Human Interaction* 12, 4 (2005), 388–422.
- [145] MEISNER, E. M., ÀBANOVIC, S., ISLER, V., CAPOREAL, L. R., AND TRINKLE, J. ShadowPlay: A generative model for nonverbal human-robot interaction. In *Proceedings of Human Robot Interaction '09* (2009), pp. 117–124.
- [146] MELCHIOR, J., GROLAUX, D., VANDERDONCKT, J., AND ROY, P. V. A toolkit for peer-to-peer distributed user interfaces: Concepts,



implementation, and applications. In *Proceedings of Engineering Interactive Computing Systems '09* (2009), pp. 69–78.

- [147] MILLER, G. *High Quality Novel View Rendering from Multiple Cameras*. Doctoral dissertation, University of Surrey, UK, CVSSP, SEPS, University of Surrey, Guildford, GU2 7XH, 2007.
- [148] MITHAL, A. K., AND DOUGLAS, S. A. Differences in movement microstructures of the mouse and the finger-controlled isometric joystick. In *Proceedings of CHI '96* (1996), pp. 300–307.
- [149] MIWA, Y., AND ISHIBIKI, C. Shadow communication: System for embodied interaction with remote partners. In *Proceedings of CSCW '04* (2004), pp. 467–476.
- [150] MORAVEJI, N., INKPEN, K., CUTRELL, E., AND BALAKRISHNAN, R. A mischief of mice: Examining children’s performance in single display groupware systems with 1 to 32 mice. In *Proceedings of CHI '09* (2009), pp. 2157–2166.
- [151] MUELLER, F. F., AGAMANOLIS, S., GIBBS, M. R., AND VETERE, F. Remote impact: Shadow boxing over a distance. In *Extended Abstracts of CHI '08* (2008), pp. 2291–2296.
- [152] MÜLLER, J., ALT, F., MICHELIS, D., AND SCHMIDT, A. Requirements and design space for interactive public displays. In *Proceedings of MM '10* (2010), pp. 1285–1294.
- [153] MYERS, B., HUDSON, S. E., AND PAUSCH, R. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction* 7, 1 (2000), 3–28.
- [154] MYERS, B. A., BHATNAGAR, R., NICHOLS, J., PECK, C. H., KONG, D., MILLER, R., AND LONG, A. C. Interacting at a distance: Measuring the performance of laser pointers and other devices. In *Proceedings of CHI '02* (2002), pp. 33–40.
- [155] MYERS, B. A., STIEL, H., AND GARGIULO, R. Collaboration using multiple PDAs connected to a PC. In *Proceedings of CSCW '98* (1998), pp. 285–294.
- [156] MYNATT, E. D. The writing on the wall. In *Proceedings of INTERACT '99* (1999), pp. 196–204.

- [157] MYNATT, E. D., IGARASHI, T., EDWARDS, W. K., AND LAMARCA, A. Flatland: New dimensions in office whiteboards. In *Proceedings of CHI '99* (1999), pp. 346–353.
- [158] NIELSEN, J. Noncommand user interfaces. *Communications of the ACM* 36, 4 (1993).
- [159] NOYES, J. The QWERTY keyboard: A review. *International Journal of Man-Machine Studies* 18, 3 (1983), 265–281.
- [160] PARKER, J. K., MANDRYK, R. L., AND INKPEN, K. M. TractorBeam: Seamless integration of local and remote pointing for tabletop displays. In *Proceedings of Graphics Interface '05* (2005), pp. 33–40.
- [161] PAVANI, F., AND CASTIELLO, U. Binding personal and extrapersonal space through body shadows. *Nature Neuroscience* 7 (2004), 13–14.
- [162] PAVLOVYCH, A., AND STUERZLINGER, W. An analysis of novice text entry performance on large interactive wall surfaces. In *Human-Computer International* (2005).
- [163] PEDERSEN, E. R., MCCALL, K., MORAN, T. P., AND HALASZ, F. G. Tivoli: An electronic whiteboard for informal workgroup meetings. In *Proceedings of CHI '93* (1993), pp. 391–398.
- [164] PLAUE, C., STASKO, J., AND BALOGA, M. The conference room as a toolbox: Technological and social routines in corporate meeting spaces. In *Proceedings of Communities and Technologies '09* (2009), pp. 95–104.
- [165] PO, B. A., FISHER, B. D., AND BOOTH, K. S. Mouse and touchscreen selection in the upper and lower visual fields. In *Proceedings of CHI '04* (2004), pp. 359–366.
- [166] PO, B. A., FISHER, B. D., AND BOOTH, K. S. Comparing cursor orientations for mouse, pointer, and pen interaction. In *Proceedings of CHI '05* (2005), pp. 291–300.
- [167] PORTER, S., MARNER, M. R., ECK, U., SANDOR, C., AND THOMAS, B. H. Rundle Lantern in miniature: Simulating large scale non-planar displays. In *Proceedings of ACE '09* (2009), pp. 311–314.
- [168] PRATT, J., ADAM, J. J., AND FISCHER, M. H. Visual layout modulates Fitts law: The importance of first and last positions. *Psychonomic Bulletin & Review* 14, 2 (2007), 350–355.

- [169] PREVIC, F. H. The neuropsychology of 3-D space. *Psychol Bull* 124 (1998), 123–164.
- [170] PROSCHOWSKY, M., SCHULTZ, N., AND JACOBSEN, N. E. An intuitive text input method for touch wheels. In *Proceedings of CHI '06* (2006), pp. 467–470.
- [171] REETZ, A., GUTWIN, C., STACH, T., NACENTA, M., AND SUBRAMANIAN, S. Superflick: A natural and efficient technique for long-distance object placement on digital tables. In *Proceedings of Graphics Interface '06* (2006), pp. 163–170.
- [172] RIZZOLATTI, G., FADIGA, L., FOGASSI, L., AND GALLESE, V. The space around us. *Science* 277 (1997), 190–191.
- [173] ROEBER, H., BACUS, J., AND TOMASI, C. Typing in thin air: The Canesta projection keyboard - a new method of interaction with electronic devices. In *Extended abstracts of CHI '03* (2003), pp. 712–713.
- [174] ROGERS, Y., AND LINDLEY, S. Collaborating around vertical and horizontal large interactive displays: Which is best? *Interacting With Computers* 16 (2004), 1133–1152.
- [175] ROGERS, Y., AND RODDEN, T. *Configuring spaces and surfaces to support collaborative interactions*. Kluwer Publishers, 2003, pp. 45–79.
- [176] RUTLEDGE, J. D., AND SELKER, T. Force-to-motion functions for pointing. In *Proceedings of INTERACT '90* (1990), pp. 701–706.
- [177] SCHMIDT, R. A., ZELAZNIK, H., HAWKINS, B., FRANK, J., AND JR., J. Q. Motor-output variability: A theory for the accuracy of rapid motor acts. *Psychological Review* 86, 5 (1979), 415–451.
- [178] SCHOFIELD, W. N. Do children find movements which cross the body midline difficult? *The Quarterly Journal of Experimental Psychology* 28, 4 (1976), 571–582.
- [179] SCOTT, S. D., WAN, J., RICO, A., FURUSHO, C., AND CUMMINGS, M. Aiding team supervision in command and control operations with large-screen displays. In *Proceedings of HSIS '07* (2007).
- [180] SHAW, C., LIANG, J., GREEN, M., AND SUN, Y. The decoupled simulation model for virtual reality systems. In *Proceedings of CHI '92* (1992), pp. 321–328.

- [181] SHEN, C., VERNIER, F. D., FORLINES, C., AND RINGEL, M. DiamondSpin: An extensible toolkit for around-the-table interaction. In *Proceedings of CHI '04* (2004), pp. 167–174.
- [182] SHOEMAKER, G., FINDLATER, L., DAWSON, J. Q., AND BOOTH, K. S. Mid-air text input techniques for very large wall displays. In *Proceedings of Graphics Interface '09* (2009), pp. 231–238.
- [183] SHOEMAKER, G., TANG, A., AND BOOTH, K. S. Shadow Reaching: A new perspective on interaction for large displays. In *Proceedings of UIST '07* (2007), pp. 53–56.
- [184] SHOEMAKER, G., TSUKITANI, T., KITAMURA, Y., AND BOOTH, K. S. Body-centric interaction techniques for very large wall displays. In *Proceedings of NordiCHI '10* (2010), pp. 463–472.
- [185] SHOEMAKER, G. B. D., AND INKPEN, K. M. MIDDesktop: An application framework for Single Display Groupware investigations. Tech. Rep. TR 2000-1, School of Computing Science, Simon Fraser University, 2000.
- [186] SOUKOREFF, R. W., AND MACKENZIE, I. S. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *Int. J. Human-Computer Studies* 61 (2004), 751–789.
- [187] STØDLE, D., BJØRNDALLEN, J. M., AND ANSHUS, O. J. A system for hybrid vision- and sound-based interaction with distal and proximal targets on wall-sized, high-resolution tiled displays. In *Proceedings of HCI'07* (2007), pp. 59–68.
- [188] SUNDSTROM, E., AND ALTMAN, I. Interpersonal relationships and personal space: Research review and theoretical model. *Human Ecology* 4, 1 (1976), 47–67.
- [189] TAN, D. S., MEYERS, B., AND CZERWINSKI, M. WinCuts: Manipulating arbitrary window regions for more effective use of screen space. In *Proceedings of CHI '04* (2004), pp. 1525–1528.
- [190] TAN, D. S., AND PAUSCH, R. Pre-emptive shadows: Eliminating the blinding light from projectors. In *Extended abstracts of CHI '02* (2002), pp. 682–683.

- [191] TANG, A., BOYLE, M., AND GREENBERG, S. Display and presence disparity in mixed presence groupware. *Journal of Research & Practice in Information Technology* 37, 2 (2005), 71–88.
- [192] TANG, A., FINKE, M., BLACKSTOCK, M., LEUNG, R., DEUTSCHER, M., AND LEA, R. Designing for bystanders: Reflections on building a public digital forum. In *Proceedings of CHI '08* (2008), pp. 879–882.
- [193] TANG, A., LANIR, J., GREENBERG, S., AND FELS, S. Supporting transitions in work: Informing large display application design by understanding whiteboard use. In *Proceedings of GROUP '09* (2009), pp. 149–158.
- [194] TANG, A., NEUSTAEDTER, C., AND GREENBERG, S. VideoArms: Embodiments for mixed presence groupware. In *Proceedings of the 20th British HCI Group Annual Conference* (2006), pp. 85–102.
- [195] TANG, J. C., AND MINNEMAN, S. VideoWhiteboard: Video shadows to support remote collaboration. In *Proceedings of CHI '91* (1991), pp. 315–322.
- [196] TAYLOR, G., SIGAL, L., FLEET, D., AND HINTON, G. Dynamical binary latent variable models for 3D human pose tracking. In *IEEE Conference on Computer Vision and Pattern Recognition* (2010).
- [197] TORY, M., STAUB-FRENCH, S., PO, B., AND WU, F. Artifact-mediated coordination in building design. *Journal of Computer Supported Cooperative Work* 17, 4 (2008), 311–351.
- [198] TSE, E., HANCOCK, M., AND GREENBERG, S. Speech-filtered bubble ray: Improving target acquisition on display walls. In *Proceedings of ICMI '07* (2007), pp. 307–314.
- [199] TSUKITANI, T., SHOEMAKER, G., BOOTH, K. S., TAKASHIMA, K., ITOH, Y., KITAMURA, Y., AND KISHINO, F. A Fitts' law analysis of shadow metaphor mid-air pointing on a very large wall display. In *Proceedings of Information Processing Society of Japan Interaction '10* (2010).
- [200] ULLMER, B., AND ISHII, H. Emerging frameworks for tangible user interfaces. In *Human-computer interaction in the new millenium*, J. M. Carroll, Ed. Addison-Wesley, 2001, pp. 579–601.

- [201] UTTERBACK, C., AND ACHITUV, R. Text Rain. In *SIGGRAPH Electronic Art and Animation Catalog* (2002), p. 78.
- [202] VAISHNAVI, S., CALHOUN, J., AND CHATTERJEE, A. Binding personal and peripersonal space: Evidence from tactile extinction. *Journal of Cognitive Neuroscience* 13, 2 (2001), 181–189.
- [203] VAN DAM, A. Post-WIMP user interfaces. *Communications of the ACM* 40, 2 (1997), 63–67.
- [204] VOGEL, D., AND BALAKRISHNAN, R. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of UIST '05* (2005), pp. 33–42.
- [205] VOGEL, D., AND BALAKRISHNAN, R. Occlusion-aware interfaces. In *Proceedings of CHI '10* (2010), pp. 263–272.
- [206] VOGEL, D., AND BAUDISCH, P. Shift: A technique for operating pen-based interfaces using touch. In *Proceedings of CHI '07* (2007), pp. 657–666.
- [207] WALKER, N., AND SMELCER, J. B. A comparison of selection time from walking and pull-down menus. In *In Proceedings of CHI '90* (1990), pp. 221–226.
- [208] WELFORD, A. T. *Fundamentals of Skill*. Methuen, London, 1968.
- [209] WESTEYN, T., BRASHEAR, H., ATRASH, A., AND STARNER, T. Georgia tech gesture toolkit: Supporting experiments in gesture recognition. In *Proceedings of Multimodal Interfaces '03* (2003), pp. 85–92.
- [210] WHITTAKER, S., AND HIRSCHBERG, J. The character, value, and management of personal paper archives. *ACM Trans. Comput.-Hum. Interact.* 8, 2 (2001), 150–170.
- [211] WIENSS, C., NIKITIN, I., GOEBBELS, G., TROCHE, K., GÖBEL, M., NIKITINA, L., AND MÜLLER, S. Sceptre: An infrared laser tracking system for virtual environments. In *Proceedings of VRST '06* (2006), pp. 45–50.
- [212] WIGDOR, D., AND BALAKRISHNAN, R. TiltText: Using tilt for text input to mobile phones. In *Proceedings of UIST '03* (2003), pp. 81–90.

- [213] WIKIPEDIA. F-test.  
[http://en.wikipedia.org/wiki/F-test#Regression\\_problems](http://en.wikipedia.org/wiki/F-test#Regression_problems), Accessed Dec 2010.
- [214] WILSON, A., AND SHAFER, S. XWand: UI for intelligent spaces. In *Proceedings of CHI '03* (2003), pp. 545–552.
- [215] WILSON, A. D., AND BENKO, H. Combining multiple depth cameras and projectors for interactions on, above, and between surfaces. In *Proceedings of UIST 2010* (2010).
- [216] WILSON, E. O. *Sociobiology: The new synthesis*, 25th anniversary ed. Harvard, 2000.
- [217] WOBROCK, J. O., CUTRELL, E., HARADA, S., AND MACKENZIE, I. S. An error model for pointing based on Fitts' law. In *Proceedings of CHI '08* (2008), pp. 1613–1622.
- [218] WOBROCK, J. O., MORRIS, M. R., AND WILSON, A. D. User-defined gestures for surface computing. In *Proceedings of CHI '09* (2009), pp. 1083–1092.
- [219] WOODWORTH, R. S. The accuracy of voluntary movement. *Psychological Review* 3 (1899), 1–114.
- [220] ZHAI, S., CONVERSY, S., BEAUDOUIN-LAFON, M., AND GUIARD, Y. Human on-line response to target expansion. In *Proceedings of CHI '03* (2003), pp. 177–184.

# Appendices



## **Appendix A**

# **Text Experiment 1 Questionnaires**

## Pre Questionnaire

1. How old are you? (circle one)

19-25

26-30

31-35

36+

2. What is your gender? (circle one)

Male

Female

Other

3. How much time do you spend per week typing on a keyboard? (circle one)

<1 hour

1-3 hours

4-8 hours

9+ hours

4. How much time in total have you spent playing a Nintendo Wii? (circle one)

Never played

<1 hour

1-10 hours

10+ hours

5. Do you own a cell phone? (circle one)

Yes

No

If you answered “yes” to question 5, answer the next 2 questions :

6. How many text messages do you send per month from your phone? (circle one)

None

1-10

11-100

101+

7. What technique do you use for sending phone text messages? (circle one)

I don't know

T9

Multitap

Other: \_\_\_\_\_

## QWERTY Keyboard Questionnaire

1. How mentally demanding was the task? (circle one number)

(Easy)      1      2      3      4      5      (Impossible)

2. How physically demanding was the task? (circle one number)

(Easy)      1      2      3      4      5      (Impossible)

3. Overall, what was the level of difficulty of the task? (circle one number)

(Easy)      1      2      3      4      5      (Impossible)

4. How successful were you in accomplishing what you were asked to do? (circle one number)

(Perfect)      1      2      3      4      5      (Failure)

5. How hard did you have to work to accomplish your level of performance? (circle one number)

(Not Hard)      1      2      3      4      5      (Very Hard)

6. How insecure, discouraged, irritated, and annoyed were you, versus secure, gratified, content, and complacent? (circle one number)

(Exasperated) 1      2      3      4      5      (Fulfilled)

7. Please write any comments you have regarding your experience with this interaction technique:

---

---

---

---

## Circle Keyboard Questionnaire

1. How mentally demanding was the task? (circle one number)

(Easy)      1      2      3      4      5      (Impossible)

2. How physically demanding was the task? (circle one number)

(Easy)      1      2      3      4      5      (Impossible)

3. Overall, what was the level of difficulty of the task? (circle one number)

(Easy)      1      2      3      4      5      (Impossible)

4. How successful were you in accomplishing what you were asked to do? (circle one number)

(Perfect)      1      2      3      4      5      (Failure)

5. How hard did you have to work to accomplish your level of performance? (circle one number)

(Not Hard)      1      2      3      4      5      (Very Hard)

6. How insecure, discouraged, irritated, and annoyed were you, versus secure, gratified, content, and complacent? (circle one number)

(Exasperated) 1      2      3      4      5      (Fulfilled)

7. Please write any comments you have regarding your experience with this interaction technique:

---

---

---

---

## Cube Keyboard Questionnaire

1. How mentally demanding was the task? (circle one number)

(Easy)      1      2      3      4      5      (Impossible)

2. How physically demanding was the task? (circle one number)

(Easy)      1      2      3      4      5      (Impossible)

3. Overall, what was the level of difficulty of the task? (circle one number)

(Easy)      1      2      3      4      5      (Impossible)

4. How successful were you in accomplishing what you were asked to do? (circle one number)

(Perfect)      1      2      3      4      5      (Failure)

5. How hard did you have to work to accomplish your level of performance? (circle one number)

(Not Hard)      1      2      3      4      5      (Very Hard)

6. How insecure, discouraged, irritated, and annoyed were you, versus secure, gratified, content, and complacent? (circle one number)

(Exasperated) 1      2      3      4      5      (Fulfilled)

7. Please write any comments you have regarding your experience with this interaction technique:

---

---

---

---

## Post Questionnaire

This questionnaire asks you to rank the three techniques you used from best to worst in several categories. The questions are asking you for your personal judgment, so there are no wrong answers.

1. Rank the three techniques from overall best to worst (1=best, 3=worst).

Circle Keyboard \_\_\_\_\_

Cube Keyboard \_\_\_\_\_

QWERTY Keyboard \_\_\_\_\_

2. Rank the three techniques from fastest to slowest (1=fastest, 3=slowest)

Circle Keyboard \_\_\_\_\_

Cube Keyboard \_\_\_\_\_

QWERTY Keyboard \_\_\_\_\_

3. Rank the three techniques from easiest to use, to hardest to use (1=easiest, 3=hardest)

Circle Keyboard \_\_\_\_\_

Cube Keyboard \_\_\_\_\_

QWERTY Keyboard \_\_\_\_\_

## **Appendix B**

### **Text Experiment 2**

### **Questionnaires**

## Pre Questionnaire

1. How old are you? \_\_\_\_\_
  
2. What is your gender? (circle one)  
Male                      Female                      Other
  
3. What handedness are you? (circle one)  
Left-handed              Right-handed              Ambidextrous
  
4. How much time do you spend per week typing on a keyboard? (circle one)  
<1 hour                  1-3 hours                  4-8 hours                  9+ hours
  
5. How much time in total have you spent playing a Nintendo Wii? (circle one)  
Never played              <1 hour                  1-10 hours                  10+ hours
  
6. Do you own a cell phone? (circle one)  
Yes                          No
  

If you answered “yes” to question 6, answer the next 2 questions :

  
7. How many text messages do you send per month from your phone? (circle one)  
None                          1-10                          11-100                          101+
  
8. What technique do you use for sending phone text messages? (circle one)  
I don't know              T9                          Multitap                          Other: \_\_\_\_\_



## Questionnaire

1. How mentally demanding was the task? (circle one number)

(Impossible) 1      2      3      4      5      (Easy)

2. How physically demanding was the task? (circle one number)

(Impossible) 1      2      3      4      5      (Easy)

3. Overall, what was the level of difficulty of the task? (circle one number)

(Impossible) 1      2      3      4      5      (Easy)

4. How successful were you in accomplishing what you were asked to do? (circle one number)

(Failure)      1      2      3      4      5      (Perfect)

5. How hard did you have to work to accomplish your level of performance? (circle one number)

(Very Hard) 1      2      3      4      5      (Not Hard)

6. How insecure, discouraged, irritated, and annoyed were you, versus secure, gratified, content, and complacent? (circle one number)

(Exasperated) 1      2      3      4      5      (Fulfilled)

7. Please write any comments you have regarding your experience with this interaction technique:

---

---

---

---

## Post Questionnaire

1. Rank the four techniques from overall best to worst (1=best, 4=worst).

Circle Close \_\_\_\_\_

Circle Far \_\_\_\_\_

QWERTY Close \_\_\_\_\_

QWERTY Far \_\_\_\_\_

2. Rank the four techniques from fastest to slowest (1=fastest, 4=slowest)

Circle Close \_\_\_\_\_

Circle Far \_\_\_\_\_

QWERTY Close \_\_\_\_\_

QWERTY Far \_\_\_\_\_

3. Rank the four techniques from easiest to use, to hardest to use (1=easiest, 4=hardest)

Circle Close \_\_\_\_\_

Circle Far \_\_\_\_\_

QWERTY Close \_\_\_\_\_

QWERTY Far \_\_\_\_\_

4. Please write any comments or thoughts you have regarding any of the techniques you used, or the experiment in general:

---

---

---

---

## **Appendix C**

# **Fitts' Law Experiment Questionnaires**

**Targeting Experiment 2  
Pre Questionnaire**

Participant: \_\_\_\_\_

1. How old are you? \_\_\_\_\_

2. What is your gender? (circle one)

Male

Female

Other

3. Which is your dominant hand? (circle one)

Right

Left

Both

4. How much time do you spend per week using a computer? (circle one)

<1 hour

1-3 hours

4-8 hours

9+ hours

5. Have you ever used a very large (greater than 3m diagonal) wall or table computer display? (circle one)

Yes

No

If you answered "yes," please explain what you used it for:

---

---

---

6. How much time have you spent playing the Nintendo Wii? (circle one)

Never played

<2 hours

2-10 hours

10+ hours

**Targeting Experiment 2**  
**Post Questionnaire**

Participant: \_\_\_\_\_

1. Overall, what was the level of difficulty of the task? (circle one number)

(Impossible) 1      2      3      4      5      (Easy)

2. How difficult were the least sensitive gain (gain = 2 or 5) levels? (circle one number)

(Impossible) 1      2      3      4      5      (Easy)

3. How difficult were the middle sensitivity gain (gain = 8 or 12) levels? (circle one number)

(Impossible) 1      2      3      4      5      (Easy)

4. How difficult were the highest sensitivity gain (gain = 16 or 20) levels? (circle one number)

(Impossible) 1      2      3      4      5      (Easy)

5. Did you employ any particular strategy in completing the task? Please explain.

---

---

---

---

6. Please write any other comments you have regarding your experience with this task:

---

---

---

---

## Appendix D

# Large Display Luminance Properties

Much of the work described in this dissertation was performed on a large wall display located in X715 of the ICICS/CS building at the University of British Columbia. It is useful to know the luminance properties of this display. For example, evaluating human perception of onscreen elements requires that a model of luminance is known so that differences in RGB values can be converted into accurate contrast values. It is also important to know how luminance varies with angle to the screen. This is especially true for very large displays, as users frequently look at onscreen elements that deviate significantly from orthogonal presentation.

We describe a model of luminance for the large wall display used for prototype 1 in chapter 3, and all work in chapters 4, 5, 7. We measured luminance values of different RGB triplets in order to map RGB values to luminance values. We also examined the impact on luminance of angle of the observer to the display.

Specific values of luminance almost certainly do not generalize beyond the particular display used. The nature of the display, including type of glass used and the coating used on the glass will have an impact on luminance behaviour.

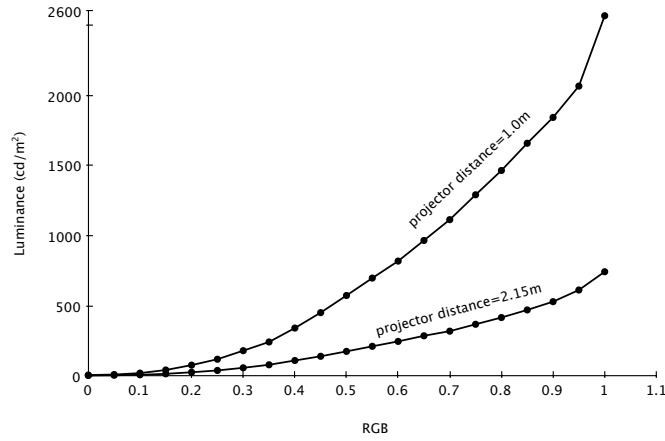
Measurements were made using a Photo Research Inc. PR650 SpectraScan colorimeter mounted on a tripod.

RGB	Raw		Normalized	
	2.15m	1.0m	2.1m	1.0m
0.00	5.13	8.42	0.007	0.003
0.05	6.92	12.0	0.009	0.005
0.10	10.2	22.2	0.014	0.009
0.15	17.1	44.3	0.023	0.017
0.20	28.3	79.1	0.038	0.031
0.25	41.2	121	0.055	0.047
0.30	60.0	182	0.080	0.071
0.35	81.5	244	0.110	0.095
0.40	112	343	0.151	0.134
0.45	142	453	0.191	0.176
0.50	177	575	0.238	0.224
0.55	213	699	0.286	0.272
0.60	248	820	0.333	0.319
0.65	288	967	0.387	0.377
0.70	321	1115	0.431	0.434
0.75	370	1292	0.497	0.503
0.80	418	1465	0.562	0.571
0.85	472	1659	0.634	0.646
0.90	531	1843	0.713	0.718
0.95	614	2066	0.825	0.805
1.00	744	2567	1.0	1.0

**Table D.1:** Luminance as it varies based on RGB and distance of projector from display.

## D.1 Luminance as RGB Varies

Different grayscale values were explored. R, G, and B values were identical for each measurement. RGB triplets ranging from 0 (black) to 1.0 (white) in increments of 0.05 were measured. The screen was rear-projected, and luminance values were measured from the front of the display. Two distances of projector to display were explored: 1.0m and 2.15m. Raw and normalized results are shown in Table D.1, and raw results are visualized in Figure D.1. Normalized values were determined by treating the value at RGB=1.00 as 1.0 and scaling all other values.



**Figure D.1:** Luminance as it varies based on RGB and distance of projector from display.

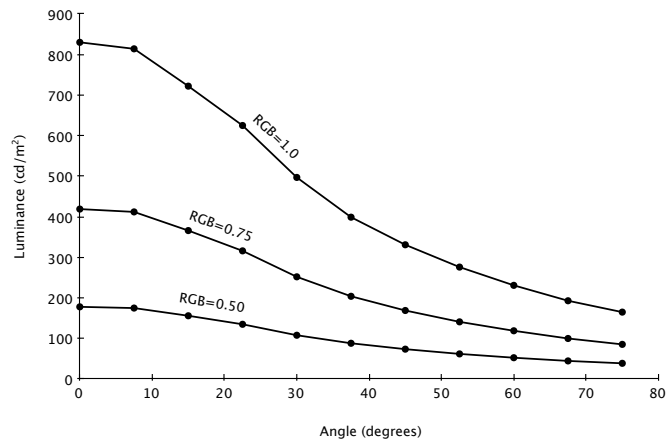
Angle	Raw			Normalized		
	RGB 1.0	RGB 0.75	RGB 0.50	RGB 1.0	RGB 0.75	RGB 0.50
0.00	830	419	178	1.0	1.0	1.0
7.50	814	412	175	0.981	0.983	0.983
15.0	722	366	156	0.870	0.874	0.876
22.5	625	316	135	0.753	0.754	0.758
30.0	497	252	108	0.599	0.601	0.607
37.5	399	204	88.2	0.481	0.487	0.496
45.0	331	169	73.7	0.399	0.403	0.414
52.5	276	141	62.0	0.333	0.337	0.348
60.0	231	119	52.6	0.278	0.284	0.296
67.5	193	100	44.7	0.233	0.239	0.251
75.0	165	85.5	38.8	0.199	0.204	0.218

**Table D.2:** Luminance as it varies based on angle of meter to display.

## D.2 Luminance as Angle Varies

We measured luminance at a fixed point on the display as measured from a point a certain distance and certain angle from the fixed point. Raw and normalized results are shown in Table D.2 and raw results are visualized in Figure D.2.





**Figure D.2:** Luminance as it varies based on angle.