

# **User Interaction in Hybrid Multi-Display Environments**

Hrvoje Benko

Submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy  
in the Graduate School of Arts and Sciences



COLUMBIA UNIVERSITY  
2007

© 2007

Hrvoje Benko  
All Rights Reserved

## ABSTRACT

### User Interaction in Hybrid Multi-Display Environments

Hrvoje Benko

This dissertation presents the design, implementation, and evaluation of novel pointer-based and gesture-based interaction techniques for multi-display environments (MDEs). These techniques transcend the constraints of a single display, allowing users to combine multiple displays and interaction devices in order to benefit from the advantages of each.

First, we introduce a set of pointer warping techniques, which improve existing mouse pointer interaction in an MDE, by allowing users to instantaneously relocate the cursor to an adjacent display, instead of traversing the bezel. Formal evaluations show significant improvements in user performance when compared to standard mouse behavior.

Next, we focus on a particular kind of MDE, in which 3D head-worn augmented reality displays are combined with handheld and stationary displays to form a *hybrid MDE*. A key benefit of hybrid MDEs is that they integrate and utilize the 3D space in which the 2D displays are embedded, creating a seamless visualization environment. We describe the development of a complex hybrid MDE system, called *Visual Interaction Tool for Archaeology (VITA)*, which allows for collaborative off-site analysis of archaeological excavation data by distributing the presentation of data among several head-worn, handheld, projected tabletop, and large high-resolution displays.

Finally, inspired by the lack of freehand interactions for hybrid MDEs, we designed gestural techniques that address the important issues of interacting across and within 2D and 3D displays. First, we present a set of *Cross-Dimensional Gestures*, which facilitate transitioning and associating the data between devices, displays, and dimensionalities, by synchronizing the recognition of gestures between a 2D multi-touch-sensitive projected display and a tracked 3D finger-bend sensor glove. Second, we describe a set of *Dual Finger Selection* techniques that allow for precise and accurate selection of small

targets within 2D displays, by exploiting the multi-touch capabilities of a tabletop surface. Third, we present the *Balloon Selection* technique, which is three times more accurate than standard wand-based selection when selecting small 3D objects above a tabletop surface. Balloon Selection decouples the 3DOF selection task into a 2DOF task and a 1DOF task, while grounding the user's hands on a touch-sensitive tabletop surface.



# Dissertation Committee

Chair

---

Prof. John R. Kender, Ph.D.  
Columbia University, New York, NY

Adviser

---

Prof. Steven K. Feiner, Ph.D.  
Columbia University, New York, NY

Internal Member

---

Prof. Julia B. Hirschberg, Ph.D.  
Columbia University, New York, NY

External Member

---

Andrew D. Wilson, Ph.D.  
Microsoft Research, Redmond, WA

External Member

---

David J. Sturman, Ph.D.  
Massive Incorporated, New York, NY

May, 2007

This document differs from the original Columbia University dissertation deposit version. Although it contains identical content, it has been formatted to reduce the total number of pages and save some trees.

# Table of Contents

<b>List of Figures .....</b>	<b>vi</b>
<b>List of Tables.....</b>	<b>xiv</b>
<b>Acknowledgements .....</b>	<b>xv</b>
<b>1 Introduction .....</b>	<b>1</b>
1.1 Why MDEs? .....	3
1.2 Why Hybrid MDEs?.....	6
1.3 Why Freehand Interaction? .....	7
1.4 Contributions .....	9
1.4.1 Pointer Warping with Multi-Monitor Mouse .....	11
1.4.2 VITA Hybrid MDE .....	12
1.4.3 Cross-Dimensional Gestures .....	13
1.4.4 Dual Finger Selection Techniques.....	14
1.4.5 Balloon Selection.....	15
<b>2 Related Work.....</b>	<b>17</b>
2.1 MDE Implementations .....	17
2.1.1 Early MDEs .....	17
2.1.2 Ubiquitous Computing Environments .....	18
2.1.3 Peripheral and Ambient Displays.....	20
2.1.4 Tangible User Interfaces.....	21
2.1.5 Augmented Reality and Hybrid MDEs .....	22
2.1.6 Display Clusters.....	25
2.1.7 Multi-Display Consumer Systems.....	25
2.2 MDE Display Classification.....	27
2.3 Interacting in MDEs .....	29
2.3.1 Interacting across Displays.....	29
2.3.1.1 Mouse Pointer Interactions.....	29
2.3.1.2 Interacting with Pens, Wands, and PDAs.....	31
2.3.2 Interacting through Displays .....	33
2.3.3 Display Association.....	34
2.3.4 Experimental Studies of MDE Interactions.....	35
2.4 Freehand Interaction.....	36

2.4.1.1	Hand Gesture Theory .....	37
2.4.1.2	Gestural Interaction with Standard Displays .....	38
2.4.1.3	Gestural Interaction in 3D Virtual Environments .....	39
2.4.1.4	Bimanual Interaction .....	39
2.4.1.5	Multi-Touch Technology .....	40
<b>3</b>	<b>Improving Pointer Interaction in MDEs .....</b>	<b>43</b>
3.1	MDE Bezel Traversal Issues .....	44
3.1.1.1	Increased Distances .....	44
3.1.1.2	Gaps between Displays .....	45
3.1.1.3	Display Mismatch .....	45
3.2	Multi-Monitor Mouse (M <sup>3</sup> ) .....	48
3.2.1	M <sup>3</sup> Frame-Switching Alternatives .....	49
3.2.1.1	Mouse-Button Switch (MB) .....	49
3.2.1.2	Keyboard Switch (KB) .....	50
3.2.1.3	Mouse-Location Switch (ML) .....	50
3.2.1.4	Head-Orientation Switch (HEAD) .....	51
3.2.1.5	Head-Orientation–Mouse Switch (HEADMB) .....	51
3.2.2	M <sup>3</sup> Pointer-Placement Strategies .....	52
3.2.2.1	Fixed-Location Center Strategy (C) .....	52
3.2.2.2	Frame-Relative Strategy (FR) .....	52
3.2.2.3	Frame-Memory Strategy (FM) .....	54
3.3	User Evaluations of M <sup>3</sup> .....	54
3.3.1	Experiment 1: Pointer Warping in Homogeneous MDEs .....	54
3.3.1.1	Setup .....	55
3.3.1.2	Method .....	55
3.3.1.3	Procedure .....	56
3.3.1.4	Task .....	56
3.3.1.5	Hypotheses .....	57
3.3.1.6	Results .....	58
3.3.1.7	Subjective Evaluations .....	60
3.3.2	Experiment 2: Pointer Warping in Heterogeneous MDEs .....	61
3.3.2.1	Setup .....	62
3.3.2.2	Method .....	65
3.3.2.3	Procedure .....	65
3.3.2.4	Task .....	65
3.3.2.5	Hypotheses .....	67
3.3.2.6	Results .....	67
3.3.2.7	Subjective Evaluations .....	72
3.4	Discussion .....	73
<b>4</b>	<b>VITA: A Hybrid MDE Implementation .....</b>	<b>77</b>
4.1	Motivation .....	78
4.1.1	Design Considerations .....	80
4.2	Visual Interaction Tool for Archaeology (VITA) .....	81
4.2.1	VITA Hardware Components .....	82
4.2.2	VITA Modular Framework .....	83

4.3	Interacting in VITA .....	86
4.3.1	Tabletop Mode.....	86
4.3.2	Life-Size Mode.....	88
4.3.2.1	3D Multimodal Interaction .....	90
4.4	Usability Evaluations of VITA.....	92
4.4.1	Site Exploration Scenario .....	92
4.4.1.1	Results .....	93
4.4.2	Teacher-Student Scenario.....	94
4.4.2.1	Results .....	95
4.5	Discussion.....	96
4.5.1	Hybrid MDE Collaboration Issues .....	97
4.5.2	Hybrid MDE Interaction Issues.....	98
<b>5</b>	<b>Cross-Dimensional Gestural Interaction in Hybrid MDEs .....</b>	<b>101</b>
5.1	Motivation .....	101
5.2	Cross-Dimensional Gestures .....	102
5.2.1	Integration of Gestural Primitives .....	103
5.2.2	Design Considerations.....	105
5.2.3	Facilitating Transitions: Cross-Dimensional Pull and Push.....	106
5.2.4	Facilitating Associations: Cross-Dimensional Pin and Connect .....	108
5.2.5	Gesture Modifiers: Cross-Dimensional Privacy Pull and Push.....	110
5.2.6	Synchronization of Gesture Recognition across Devices.....	112
5.3	Usability Evaluation of Cross-Dimensional Gestures .....	113
5.3.1.1	Procedure.....	114
5.3.1.2	Setup.....	114
5.3.1.3	Results and Feedback .....	115
5.4	Discussion.....	118
<b>6</b>	<b>Improving the Precision of Touch Screen Interaction.....</b>	<b>121</b>
6.1	Touch-Screen Selection Issues .....	122
6.2	Previous Single-Touch Solutions .....	124
6.3	Dual Finger Selection Techniques.....	125
6.3.1	Interaction Requirements.....	125
6.3.1.1	Multi-Finger Order-Based Labeling Approach .....	126
6.3.2	Dual Finger Offset.....	127
6.3.3	Dual Finger Midpoint.....	127
6.3.4	Dual Finger X-Menu .....	128
6.3.5	Dual Finger Slider .....	131
6.3.6	Dual Finger Stretch.....	133
6.4	Clicking Stabilization Technique .....	135
6.5	Prototype Multi-Touch Tabletop Device.....	136
6.6	User Experiment.....	138
6.6.1	Participants .....	138
6.6.2	Task .....	138
6.6.3	Part 1: Evaluation of SimPress Clicking .....	139
6.6.3.1	Method.....	139
6.6.3.2	Hypothesis .....	140

6.6.3.3	Results.....	140
6.6.4	Part 2: Comparison of Four Dual Finger Selection Techniques.....	141
6.6.4.1	Method.....	142
6.6.4.2	Hypotheses.....	142
6.6.4.3	Results.....	143
6.6.5	Subjective Evaluation.....	145
6.7	Discussion.....	146
<b>7</b>	<b>Improving the Precision of 3D Selection in Hybrid MDEs.....</b>	<b>149</b>
7.1	Selection Issues in Scaled-Down 3D Models.....	150
7.1.1	Adhering to the 3D Interaction Design Guidelines.....	151
7.2	Related 3D Selection Interactions.....	151
7.3	Balloon Selection Technique.....	153
7.3.1	Interaction Description.....	154
7.3.2	String Height-Clutching.....	156
7.3.3	Implementation Details.....	157
7.3.4	Visual and Audio Feedback.....	158
7.4	User Experiment.....	160
7.4.1	Baseline Comparison Techniques.....	161
7.4.1.1	WAND Technique.....	162
7.4.1.2	KEYBOARD Technique.....	162
7.4.2	Setup.....	162
7.4.3	Method.....	163
7.4.4	Task.....	164
7.4.5	Procedure.....	164
7.4.6	Hypotheses.....	165
7.4.7	Results.....	166
7.4.7.1	Completion Time Analysis.....	166
7.4.7.2	Error Rate Analysis.....	167
7.4.7.3	Cursor Radius Analysis.....	169
7.4.7.4	Subjective Evaluations.....	170
7.5	Discussion.....	172
<b>8</b>	<b>Conclusions and Future Work.....</b>	<b>175</b>
8.1	Summary of Contributions.....	175
8.2	Design Guidelines.....	177
8.2.1	Hybrid MDE Implementation Guidelines.....	177
8.2.2	Freehand Interaction Guidelines.....	178
8.3	Future Work.....	181
8.3.1	MDE Pointer Warping Interactions.....	182
8.3.2	Hybrid MDE Improvements.....	183
8.3.3	Collaboration, Personalization, and Privacy.....	184
8.3.4	Freehand Gestural and Multi-Touch Interaction.....	185
8.4	Final Remarks.....	186
<b>Appendix A</b>	<b>Multi-Monitor Mouse Widget.....</b>	<b>189</b>
A.1	Implementation Details.....	190

A.2	Widget Distribution .....	190
<b>References</b>	<b>.....</b>	<b>191</b>

# List of Figures

Figure 1.1: Multi-display environment hierarchy used in this dissertation. ....	2
Figure 1.2: Examples of two MDE categories: (a) homogeneous, (b) heterogeneous. ....	2
Figure 1.3: The VITA hybrid MDE [Benko 2004] (see Chapter 4) consists of four different kinds of displays: head-worn, hand-held (not shown), top-projected touch-sensitive tabletop, and high-resolution vertical. ....	3
Figure 1.4: Comparing (a) standard bezel traversal and (b) $M^3$ pointer warping. Instantaneously relocating (warping) the cursor to the next screen, shown as dashed line, reduces the distance traversed by conventional mouse movement. Note: “Sonar” circles are displayed to increase cursor visibility at warp destination. ....	12
Figure 1.5: Available displays in the VITA system (left to right): tracked head-worn, projected tabletop, tracked handheld, and vertical high-resolution display. ....	13
Figure 1.6: Using the Cross-Dimensional Pull gesture to transform a 2D image of a bowl found on the table into a 3D model of the same bowl above the table. ....	14
Figure 1.7: Dual Finger X-Menu technique allows the user to interactively slow down the cursor speed to improve the overall accuracy of selection. ....	15
Figure 1.8: Balloon Selection technique. ....	16
Figure 2.1: Early MDEs: (a) the first computer-supported conference work by Engelbart and colleagues [Bootstrap 2003], and (b) Xerox PARC Colab meeting room [Stefik 1987]. ....	18
Figure 2.2: Roomware® components consisting of several InteractTables® and a DynaWall® [Streitz 2002]. ....	20
Figure 2.3: The original <i>hybrid user interface</i> [Feiner 1991]: (a) a combination of a tracked see-through head-worn display with a	



vertical and stationary flat-panel display, (b) the screen capture of a user's view.....	22
Figure 2.4: Collaborative augmented reality environments: (a) Augmented Surfaces [Rekimoto 1999] and (b) EMMIE hybrid MDE [Butz 1999]......	23
Figure 2.5: Handheld multi-display commercial devices: (a) Nintendo DS Lite game system where the bottom display is touch and stylus sensitive [Nintendo 2006] and (b) Toshiba dual screen eBook stylus-sensing prototype presented at CeBIT 2006 [TechBlog 2006]. .....	26
Figure 2.6: Using hyperdragging to interact with real and virtual objects [Rekimoto 1999]: (a) the user can use hyperdragging to select a physical object (business card), (b) an image of an object can be taken via the ceiling mounted camera, and (c) the user can then hyperdrag the image to their computer desktop. ....	30
Figure 2.7: The use of PIP in StudierStube hybrid MDE [Schmalstieg 2000]: (a) the hardware of PIP is composed out of a lightweight (wooden) panel and a tracked pen, (b) multiple users each using their own PIP to interact with 3D and 2D content. ....	32
Figure 2.8: Using hand gestures to interact with map data on the TouchLight multi-touch prototype [Wilson 2004]......	41
Figure 3.1: Comparing standard bezel traversal and pointer warping in a homogeneous MDE: (a) <i>Bezel Traversal</i> – Using standard pointer movement to move between monitors from S to T; (b) <i>Pointer warping</i> – Instantaneously relocating (warping) the cursor (dashed line) to the next screen, reduces the distance traversed by conventional mouse movement. Note: “Sonar” circles are displayed to increase cursor visibility at warp destination. ....	43
Figure 3.2: Comparison of four different display spaces or coordinate systems (possible units of measurement are given in parentheses)......	46
Figure 3.3: Devices used to trigger the pointer warp: (a) side mouse buttons, (b) keyboard key combination, (c) mouse location (on different mouse-pad-shaped areas on a location sensing surface), and (d) head-orientation (3DOF orientation tracker fixed to the headphones)......	49
Figure 3.4: Traversing between S, T1 and T2 locations using different $M^3$ pointer placement strategies: a) fixed-location center, b) frame-relative, c) one of many possible frame-memory scenarios. Dashed lines indicate warping; solid lines indicate conventional movement. ....	53
Figure 3.5: Experiment 1 setup: Homogeneous MDE test setup consisted of four monitors and four corresponding “mouse pads” used by mouse-location switch. The background is set to an inactive spreadsheet image to simulate a visually noisy working environment. ....	55

Figure 3.6: Start (S) and target (T) button layout in our user study. To eliminate strategy bias, T is located at the half-way point between the warping locations for frame-relative (FR) and center fixed-location (C) strategies. Control path (CTRL) is also shown.....	57
Figure 3.7: Mean movement times (ms) for different conditions ( $F_{(6,42)}=3.88, p<0.01$ ), shown with 95% confidence intervals. ....	58
Figure 3.8: Interaction of Direction and Condition for mean movement times (ms) ( $F_{(6,42)}= 6.1391, p<0.001$ ), shown with 95% confidence intervals.....	59
Figure 3.9: Interaction of Distance and Condition for mean movement times (ms) ( $F_{(12,84)}=9.509, p<0.001$ ), with 95% confidence intervals. ....	60
Figure 3.10: Heterogeneous MDE used in Experiment 2 consisting of a small low-resolution near-horizontal display (A), a medium high-resolution vertical display (B), and a large low-resolution vertical display (C).....	61
Figure 3.11: The differences in device space (top) and visual space (bottom) representations of our experimental setup. Traversing a continuous device-space path (top left) with a standard pointer can be a cognitively demanding task in visual space (bottom left). Pointer warping aids the user by removing the need to traverse the bezels (right). Concentric “sonar” circles help increase the visibility of the cursor after the warp by highlighting the destination. ....	63
Figure 3.12: Our experimental task setup consists of nine paths (top), shown here as dashed lines, each connecting two of nine targets, distributed over three screens. Note that portions of paths 1, 2, 7, and 8 are blocked by the edges of the screens. The actual paths (bottom) in device space that a pointer could follow in CTRL mode on path 1 (from A to B), 4 (from A to C), and 7 (from B to C).....	66
Figure 3.13: Aggregated movement mean times (ms) for the Condition factor ( $F_{(4,36)}=11.46, p<0.001$ ). ....	68
Figure 3.14: Movement mean times (ms) for the Path factor ( $F_{(8,72)}=7.625, p<0.001$ ). ....	70
Figure 3.15: Mean screen transition times (ms), computed as aggregated path times (e.g., A–B screen transition is the aggregated times of paths 1, 2, and 3). ....	70
Figure 3.16: Mean screen transition times (ms) for each Condition.....	71
Figure 4.1: The author with the team of Columbia University archaeology, range-scanning, and visualization experts at the Stanford University archaeological excavations at Monte Polizzo, Sicily in July, 2003. ....	78

Figure 4.2: Archaeological data collected at Monte Polizzo and incorporated into the VITA hybrid MDE. Data not typically collected at an excavation includes panoramic images, and 3D object and 3D site models. ....	79
Figure 4.3: Available displays in the VITA hybrid MDE: (a) tracked head-worn (AR), (b) touch-sensitive projected tabletop (TABLE), (c) vertical high-resolution display (VERTICAL), and (d) tracked handheld (LENS).....	81
Figure 4.4: VITA user’s wearable components.....	82
Figure 4.5: VITA modular architecture diagram.....	83
Figure 4.6: A portion of the code that illustrates how modules can subscribe to particular messages of interest. ....	84
Figure 4.7: Two users simultaneously collaborate in VITA. While one user is inspecting the 3D virtual model of the ceramic vessel above the table, comparing it with the high-resolution image on the screen, the second user is looking at the 3D miniature terrain model next to the interactive tabletop. ....	86
Figure 4.8: Temporal and spatial representations of data in the tabletop mode of VITA: (a) a Harris Matrix visualization displayed on the TABLE display (designed and implemented by Edward Ishak) represents temporal connections between layers, and (b) the 3D world-in-miniature site model places excavated finds in a spatial context. ....	87
Figure 4.9: Real vs. virtual: (a) Real image of a portion of the excavated structure taken at the site. (b) User exploring the same section of the site in life-size mode of VITA.....	89
Figure 4.10: 3D poses recognized by the VITA system.....	90
Figure 4.11: The user selects an object of interest (an Elymian Ritual Pot) by pointing at it and saying “select that object.” ....	91
Figure 4.12: The VITA system being evaluated in our teacher-student scenario. The teacher (Trinity Jackman of the Stanford Archaeology Center) is Assistant Director of the Monte Polizzo excavation. She is working with a Columbia University archaeology student. ....	94
Figure 5.1: The dimensionality border between a 2D surface display and a 3D volume (imaged through head-worn displays). The 2D–3D dimensionality border in our hybrid MDE can be defined as a space where the user’s tracked hands operate in 3D and simultaneously touch the 2D surface. ....	102
Figure 5.2: Frames from a sequence of Cross-Dimensional Pull and Push gestures. (a) Cross-Dimensional Pull is performed by selecting the 2D object and then forming a grabbing gesture. (b) The 2D object	

disappears, while the 3D object appears from the table. (c) Holding the 3D object. (d) Cross-Dimensional Push is performed by pressing on the table with a flat hand through the 3D object. (e) The 3D object disappears and the 2D object appears. (f) The 2D object (an image), projected on the table. ....	107
Figure 5.3: Cross-Dimensional Pin associates the 3D object with the surface: (a) Using a 3D point gesture to pin the object to the surface. (b) Rings appear once the object is pinned, providing visual feedback to the user. ....	109
Figure 5.4: Cross-Dimensional Connect: (a) The user performs the gesture by tapping the table while holding a 3D object. (b) The leader line connects two representations of the same object. ....	109
Figure 5.6: Sample synchronization communication between 2D and 3D modules during the recognition of Cross-Dimensional Pull. ....	113
Figure 5.7: Mean rating scores for ease of use of cross-dimensional gestures. Participants ranked each gesture on a five-point scale (1-difficult, 5-easy). ....	115
Figure 5.8: Mean rating scores for intuitiveness of cross-dimensional gestures. Participants ranked each gesture on a five-point scale (1-confusing, 5-intuitive). ....	116
Figure 6.1: State diagram for simulating standard mouse interaction on a touch screen (adapted with modifications from [Buxton 1985]). ....	123
Figure 6.2: This image shows the use of the Dual Finger X-Menu in selecting “slow 10×” mode: the primary finger (right) controls the cursor position, while the secondary finger (left) adjusts the cursor speed. ....	127
Figure 6.3: Dual Finger Midpoint technique positions the cursor at exactly the halfway point between the two fingers, giving the user both a cursor offset as well as a variable reduction of cursor speed. ....	128
Figure 6.4: Dual Finger X-Menu enables the user to adjust the cursor speed by crossing over a particular area of the on-screen menu. Here, “freeze mode” is selected, making the cursor completely immobile. ....	129
Figure 6.5: Dual Finger X-Menu contains four selection areas for cursor speed control (normal, slow 4×, slow 10× and freeze), and two toggle areas (snap and magnify). Magnify mode presents an integrated magnification widget in the middle of the menu, while Snap mode removes the current cursor offset. ....	130
Figure 6.6: Cursor notification widget signals the current amount of cursor speed reduction: 4× reduction (left), 10× reduction (middle), and frozen cursor (right). ....	131

Figure 6.7: Dual Finger Slider. The right finger (primary) controls the cursor, and the left finger (secondary) is invoking the invisible slider. Speed reduction modes are achieved by moving the fingers closer together: (a) normal, (b) slow 4×, (c) slow 10×, (d) frozen cursor mode. ....	132
Figure 6.8: Dual Finger Stretch technique adaptively scales the user interface. (a) The secondary finger specifies the square zooming area centered at the primary finger’s location, (b) The primary finger performs precise selection while, simultaneously, the secondary finger adjusts the level of magnification. ....	134
Figure 6.9: A small rocking motion of the user’s finger triggers the SimPress clicking technique: (a) tracking (hover) state, (b) dragging (click) state. (The inset images show the actual area of contact detected by our device as a black blob, as well as the stabilized cursor location as a white cross.).....	135
Figure 6.10: The difference in the amount of cursor movement when the cursor is attached to the center of mass or to the top middle location of the user’s touch. The axis-aligned touch bounding boxes are given for reference.....	136
Figure 6.11: The multi-touch back-projected tabletop display prototype uses an infra-red illuminant and camera to detect contacts through a diffuse surface. ....	137
Figure 6.12: Mean error rate (%) using the SimPress technique alone without assistance ( $F_{(4,44)} = 62.598, p < 0.001$ ). ....	140
Figure 6.13: Interaction of Noise × Technique for error rate (%) ( $F_{(6,66)} = 8.025, p < 0.001$ ). ....	143
Figure 6.14: Interaction of Technique × Width for error rate (%) ( $F_{(9,99)} = 29.473, p < 0.001$ ). ....	144
Figure 6.15: Mean performance time (s) with respect to target widths. (Notice: time data is not shown for Offset technique at 1 pixel due to lack of successfully completed trials.).....	144
Figure 6.16: Subjective preferences for the best technique across different noise levels.....	145
Figure 7.1: Balloon Selection being used to select an object of interest in the world-in-miniature model within the VITA system. Note that the objects representing the excavated finds tend to be extremely small and closely grouped together, making their selection difficult at this scale. ....	149
Figure 7.2: The basic principle of Balloon Selection. ....	154
Figure 7.3: Balloon Selection interaction sequence. (a) Placing two fingers (the anchor and the stretching finger) adjacent to each other	

on the tabletop instantiates the 3D cursor (balloon). (b) Moving the stretching finger away from the anchor stretches the virtual string between the two fingers. (c) Moving the fingers closer together raises the balloon from the surface. (d–e) Moving the thumb on the anchor’s hand towards or away from the anchor scales the balloon up (for easier selection) or down (for more precise selection). (f) Moving the anchor on the surface translates the balloon parallel to the plane of the table, while varying the distance between the anchor and the stretching finger determines the balloon’s height. Placing the thumb of the stretching finger’s hand adjacent to the stretching finger triggers the selection of the target cube. ....155

Figure 7.4: Instrumented gloves (right hand glove shown) designed by the author that produce an independent report for each of the user’s two thumbs and two index fingers on the MERL DiamondTouch surface. ....157

Figure 7.5: State transition diagram for the implementation of Balloon Selection. Pinch is defined as placing two fingers next to each other on the surface. Therefore P(a,b) means that fingers *a* and *b* are *pinched* together, while !P(a,b) means that those same fingers are apart.....159

Figure 7.6: Interaction devices used in our experiment (top to bottom): A 3DOF tracked three button wand (Logitech TrackMan Live), a pair of specially constructed gloves with two active fingers on each hand, and the keyboard. The devices are laid out on the DiamondTouch surface. ....161

Figure 7.7: The “cube-of-cubes” target setup being used with a WAND technique (3D cursor is a semi-transparent white sphere). (a) The successful selection of a goal target (red middle target). (b) The unsuccessful selection of more than one target by a larger 3D cursor.....164

Figure 7.8: Average completion times (s) for the three techniques ( $F_{(2,22)}=22.334$ ,  $p<0.001$ ): KEYBOARD was significantly slower than BALLOON ( $t_{(11)}=5.07$ ,  $p<0.001$ ), while WAND and BALLOON performed without statistically significant differences overall. ....167

Figure 7.9: Interaction of Technique  $\times$  Size for average completion times (s). ....167

Figure 7.10: Average error rate (%) of the three techniques ( $F_{(2,22)}=18.707$ ,  $p<0.001$ ). We found no statistically significant difference between the error rates of BALLOON and KEYBOARD. ....168

Figure 7.11: Average error rate (%) across target sizes ( $F_{(3,33)}=4.672$ ,  $p<0.01$ ). On average, selecting the smallest target (4 mm) caused participants to commit twice as many errors as they did for the larger targets. ....169

Figure 7.12: Average cursor radius (mm) across techniques. ....	169
Figure 7.13: Subjective user ratings for technique speed, ease of use, and arm fatigue (ratings on a 1–5 Likert scale).....	171
Figure 7.14: Balloon Selection being used in a dense point cloud model of Fort Jay at Governors Island, NY. (Range scan data courtesy of Paul Blaer and Peter Allen of the Columbia University Robotics Lab.).....	173
Figure A.1: The screenshot of the control interface for the Multi-Monitor Mouse widget application.....	189

## List of Tables

Table 2.1: Comparison of various MDEs with regards to the displays they incorporate. The author’s hybrid MDE system VITA is highlighted in gray. ....	28
Table 3.1: Displays used in the experiment. ....	62
Table 3.2: Display transition characteristics in our experiment. ....	64
Table 3.3: Summary of two M <sup>3</sup> experiments. Check symbol (✓) indicates comparison was tested. ....	73
Table 5.1: A set of basic hand posture primitives and their interpretations in 2D and 3D. Our 2D touch recognition is based on the largest axis-aligned rectangle that bounds all user’s touches. Our 3D pose recognition is based on finger-bend-sensing. Finger notation is similar to [Baudel 1993]: (finger bent), (finger unbent), and (thumb unbent). ....	104
Table 5.2: The vocabulary of Cross-Dimensional Gestures. Note: The arrow (→) indicates a transition between two hand poses. ....	105
Table 6.1: Comparison of small target widths to an average right hand index finger width at the fingertip (estimated as 18.3 mm for an adult). Note: Values for the 30" diagonal screen correspond to the display characteristics used in this chapter. ....	122
Table 7.1: The description of each state from Figure 7.5. In all balloon states, the position of the primary (anchor) finger determines the (x,y) position of the balloon. ....	160



## Acknowledgements

Completing this work would not be possible without all the great people around me, sharing their ideas, their work, their support, and their friendship.

First, I thank my advisor Steven Feiner for supporting me throughout my graduate school endeavors, allowing me to tackle any whacky idea I had, supplying me with tons of gadgets to play with, giving me a chance to travel around the world (from Japan to Sicily to South Africa), supporting me even when I had to take a long personal leave, and providing me with an environment full of brilliant minds and free from worrying about funding. I am grateful for his suggestions on all the different projects that I have worked on. Most of all, I thank him for believing in me and helping me get through this rather difficult graduate school experience.

Next, I would like to thank all the people in the CGUI lab that we have spent innumerable hours with, coding, talking, thinking, drinking, playing, and laughing. I thank Alex Olwal for a great trip to Japan and for helping me develop the original framework that most of this thesis is based on. I also thank Edward Ishak, with whom I had the honor of chasing cheetahs and elephants in South Africa and climbing mountains in Sicily, all while doing computer science research. I thank him for the amazing time we spent together, his friendship, and his keen eye for detail that made our projects and demos so much better. Tobias Höllerer's work continues to be an inspiration for a lot of my research. Blaine Bell taught me the true meaning of "work hard, play hard" attitude and Simon Lok taught me everything I know about computer maintenance and networking (and a bit about snowboarding). Sinem Guven always provided a smile and a supportive face in a lab full of often crude boys.

I also had the privilege of sharing my office with several remarkable individuals. Gábor Blaskó was instrumental in getting me to think outside of the box and Mark Eaddy helped me tremendously with all my programming questions, and kept me focused on “good” research and getting my Ph.D. Finally, Sean White was invaluable in helping me preserve a positive outlook when things looked murky and I am truly lucky for sharing with him some of my most inspiring conversations.

Next in the long CGUI Ph.D. line are Steven Henderson and Ohan Oda, who gave me valuable feedback and were always more than eager to serve as my guinea pigs.

Having an opportunity to work with Andrew Wilson was truly an inspiring experience, and I am forever thankful for his mentorship, his incredible ideas, and great talks about research, industry, academia, brewing beer, and flying kites.

I am honored to have John Kender, Julia Hirschberg, and David Sturman on my committee. All of them provided helpful guidance, inspiration, and advice at different times throughout this difficult process.

I am also thankful to Peter Allen, Alejandro Troccoli, Benjamin Smith, and Paul Blaer, from the Robotics Lab for some wonderful trips around the world and around the New York harbor, and for providing me with amazing 3D models that made my demos look great.

The archaeological expeditions that we took part in would not have been possible without all the hard work of James Conlon and Lynn Meskell.

I am indebted to many fellow researchers and colleagues who shared their ideas with me and provided helpful encouragement: Patrick Baudisch, Ken Hinckley, Ed Cuetrell, Kris Thorisson, Aner Ben-Artzi, Sameer Maskey, Christian Sandor, Ed Kaiser, Phil Cohen, David McGee, Andrea Corradini, and many others.

Mary Van Starrex, Patricia Hervey, Twinkle Edwards, Rosemary Addarich, Alice Cueba, Elias Tesfaye, and Anne Fleming provided me with administrative assistance, getting my paychecks, my packages, my reimbursements, my equipment purchasing, as well as all the support for the CS BBQ.

I thank Tiberiu Chelcea for giving me a role model of what a BBQ Czar should be like and Andrew Howard for all the great tennis, hiking, racquetball, and soccer that provided welcome distractions from the research-only existence.

My work would not be possible without the enthusiasm of several students who worked on projects with me: Sajid Sadi, Zachariah Munoz, Shezan Baig, and Franz Coriand.

I am thankful to my in-laws, Suzanne and Lee, for standing behind me 110% and especially to Lee for his unending curiosity about my research and the desire to read my papers.

I am forever grateful to my parents, Vesna and Damir, my first role models, who even while an ocean apart, always made sure that I feel loved and supported.

Most of all, I would like to thank Elizabeth who in the course of my Ph.D. changed from my girlfriend to my wife. She survived my grad school years and kept unselfishly providing love, care, and support when we needed it most. I apologize for all the neglect that this long and hard process has caused you, and I am happy to say “It is done!” Thank you!

This work was supported in part by National Science Foundation grants IIS-01-21239 IIS-00-82961, and EIA-02-02063, Office of Naval Research contracts N00014-04-1-0005, N00014-99-1-0394, N00014-99-1-0683, and N00014-99-1-0249, Microsoft Research, Mitsubishi Electric Research Labs, Autodesk, and NVIDIA.

To Liz

# 1 Introduction

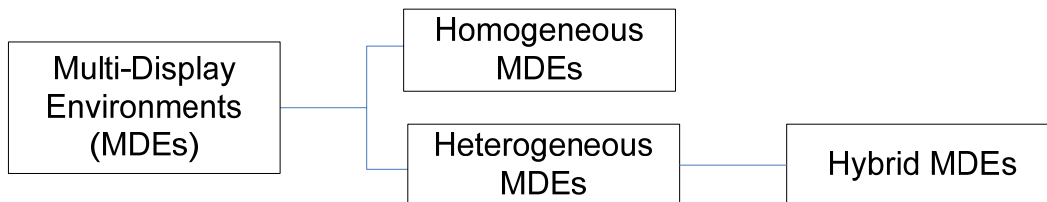
*Multi-display environments* (MDEs) [Hutchings 2004a] are computing environments consisting of more than one connected display. Improvements in display technology, significant reductions in display costs, and increasingly ubiquitous networking availability, are making it possible to create various kinds of MDEs. However, current user interaction techniques are designed mostly for computer systems with a single independent display, and have not yet evolved to take into account the constraints and affordances of such environments. The primary contributions of this dissertation are the design, implementation, and evaluation of several novel gesture-, touch-, and pointer-based interaction techniques that transcend the single display constraints, and therefore allow the user to combine multiple displays and interaction devices and to benefit from the advantages of each.

MDE configurations can be characterized as either homogeneous or heterogeneous (Figure 1.1). The most frequently encountered examples are *homogeneous*, where two or more displays of the same size, resolution, and relative orientation to the user, are tiled next to one another (Figure 1.2a). When displays of different size, resolution, orientation, or dimensionality\* are used together, they form a *heterogeneous* multi-monitor configuration (Figure 1.2b). These heterogeneous combinations often include projected wall-sized displays, laptops, tablets, tabletop displays, personal digital assistants (PDAs), and smart-phones, in addition to existing desktop displays. Not only are these displays located in the same physical space, but also each individual display in an MDE provides just one physical window into the networked data space. This *connection* be-

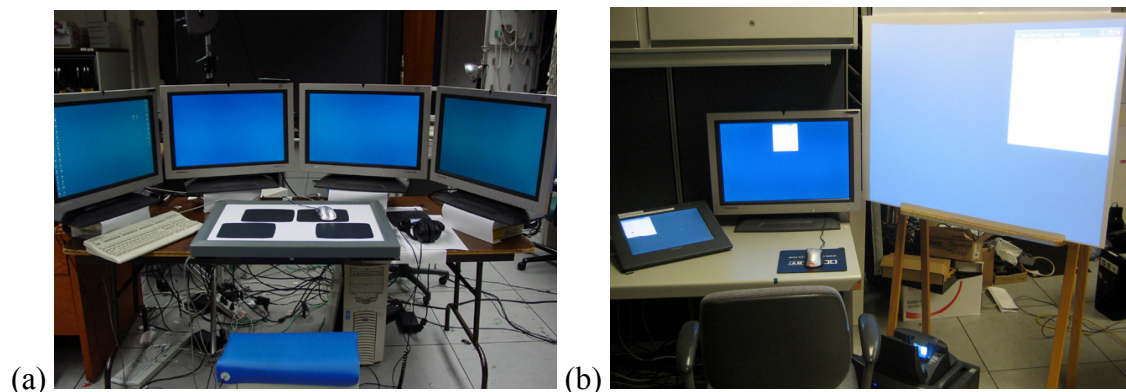
---

\* Dimensionality here refers to whether a display is capable of displaying two-dimensional or three-dimensional data, where a three-dimensional display can present imagery in stereo, showing a different view to each eye.

tween the displays is a crucial component of an MDE, without which an MDE would just be a collection of various independent collocated displays. The connection can be established either through the network protocol (*distributed display environments* [Hutchings 2005]) or by virtue of being powered by the same computer device (*multi-mon* systems [Grudin 2001]).



**Figure 1.1: Multi-display environment hierarchy used in this dissertation.**



**Figure 1.2: Examples of two MDE categories: (a) homogeneous, (b) heterogeneous.**

This dissertation focuses on a subset of heterogeneous MDEs that we call *hybrid MDEs* (Figure 1.1). This term is inspired by the notion of a *hybrid user interface* [Feiner 1991], which defines a cooperative computing environment, in which “multiple heterogeneous 2D and 3D displays and interaction devices are used in synergistic combination to benefit from the advantages of each.” Hybrid MDEs combine 3D augmented reality (AR) displays with additional 2D displays, an example of which is shown in Figure 1.3. In addition to being able to present the user with true 3D stereoscopic data and completely personalized views, AR displays can be used as the “ether” that embeds all other displays in the environment [Butz 1999]. Ultimately, hybrid MDEs have the potential to present a truly seamless user interface.



**Figure 1.3: The VITA hybrid MDE [Benko 2004] (see Chapter 4) consists of four different kinds of displays: head-worn, hand-held (not shown), top-projected touch-sensitive table-top, and high-resolution vertical.**

## 1.1 Why MDEs?

In recent years, the typical computer workspace has been experiencing a very significant transformation, from a single desktop computer with a single attached display, to an MDE with multiple connected devices and displays. From the user's perspective, all these devices are simply becoming windows (or displays) into the networked data space, and users increasingly expect that it is easy to produce, share, transfer across, and interact with any kind of data on any kind of device.

While this transformation has not yet been completely realized, it is not uncommon today to see a user's desktop with two or three tiled flat-screen displays or a laptop computer and a smart-phone on the side. Currently, these devices might operate on different networks, run different applications, have drastically different processing power, and share only a small amount of data consisting mostly of email, calendar appointments, and pictures. However, even in this "disjoint" case, the user clearly benefits from the availability of additional displays. In the following paragraphs we outline the most important benefits to using more than one display.

*Extra pixels:* The simplest and probably most important benefit of having multiple displays is that they enlarge the available displayable area. Just as there is a benefit to using a larger higher-resolution display, there is a benefit from having an additional display, since the user has more pixels at their disposal. With more displayable space, more information can be presented simultaneously. In addition, two displays usually cost less than a single high-resolution display with the sum of their resolutions or screen area.

*Extra views and partitions:* Having multiple displays results in many additional benefits that cannot be replicated by simply providing more pixels on a larger display. The extra displays can be used to organize the user's desktop space, provide additional views or perspectives, or allow some material to be displayed privately, as discussed below. Grudin points out that rather than treating multiple monitors as a single large display space, the users tend to treat multi-monitor systems as ways to partition their desktop space, with each display eventually assuming a particular role [Grudin 2001]. For example, in observing CAD designers and computer programmers, he noted that one display usually contained only their work application, with an additional display often being dedicated for miscellaneous applications, such as email, instant messaging, music player, and documentation.

*Personalization and privacy:* In addition to offering a different viewpoint and perspective, additional views can offer different levels of privacy or personalization. Collaborative meeting scenarios today often involve several personal displays (e.g., laptops) in addition to one or more large public displays. Furthermore, different display types often imply different levels of privacy, with head-worn displays probably being the most private of all. Thus, how the data is shared and distributed across various displays might have more to do with the privacy settings, than with the available space on the displays.

*Mobility and placement:* Different displays can have different mobility characteristics. Some are completely stationary (e.g., wall-mounted displays), some are meant to be placed on a table or lap, while others can be completely mobile (e.g., handhelds, cell-phones) or even head- or wrist-worn. The simple fact that a particular display can be rotated can have a significant effect on how it eventually gets used and how much pri-



vacy it offers. In addition, the placement of displays makes them more or less suitable for certain tasks. Horizontal displays tend to be more suited for small group collaborations or for some direct manipulations, while large vertical ones are better for lectures and presentations to larger audiences.

*Ambient information:* The ability to “arrange” multiple displays in the environment pushes some displays into the focus of a user’s attention, while simultaneously relegating others to their visual periphery. However, even those displays on the periphery can provide useful information in the form of notification indicators or better awareness tools. In addition, the proliferation of “smart appliances” with built-in displays is one extreme case of a heterogeneous MDE where a lot of information might be conveyed by such devices embedded in the environment.

*Input and interaction capabilities:* Some displays rely on additional input devices (e.g., mice, keyboard, and pens) and some have sensing technologies built in (e.g., touch-sensitivity, inertial sensors, and cameras). Furthermore, some displays can be completely tracked, as is the case with augmented reality or virtual reality head-worn displays. In this case, they are able to provide information correctly registered with respect to the user’s view. Being able to integrate various input and interaction capabilities into one environment, makes the MDE space richer. However, effort must be taken to properly manage various input techniques, and potentially make them available across displays.

*Technology and intrinsic capabilities:* Displays also differ in technology (e.g., LCD, CRT, Plasma, OLED, and DLP) and have different intrinsic properties, such as color, resolution, contrast, brightness, transparency (see-through or opaque), stereo and volumetric capabilities. For a more in-depth discussion of display technology, see Section 2.2.

However, having multiple displays can have its drawbacks too. It requires more desktop space and can be more distracting to the user to monitor a larger set of screens. The applications that run in an MDE also tend to be more complex, since the applications and the system need to be aware of the existence of multiple displays and have networking and middleware support for distributed operation. This also implies the need for content management [Butz 1999, Bell 2001] that manages the data and the interac-

tions across screens. The existence of monitor frames (bezels) and the task of combining displays with differing resolutions, sizes, orientations and dimensionalities can make it more difficult for the user to interact across the monitors [Baudisch 2004, Benko 2005a, Benko 2007b]. In addition, having different physical partitions can be a problem when the data or the application does not easily lend itself to partitioning.

Even so, in most cases, the benefits of MDEs greatly outweigh their shortcomings, and this dissertation presents the design of different MDE interactions that further minimize those negative effects.

## 1.2 Why Hybrid MDEs?

*Augmented reality* (AR) interfaces make it possible to visualize computer-generated graphics overlaid on the surrounding environment. In contrast to *virtual reality* (VR), which provides an entirely synthetic computer-generated sensory experience, AR interfaces attempt to enhance, rather than replace, the user's perception of the surrounding environment. Given the many similarities between the devices used for AR and VR and their common goal of immersion in a 3D environment, many AR interfaces borrow heavily from existing VR interaction research and often consist of purely virtual elements, such as 3D widgets or 3D interaction metaphors. However, since the user is not completely immersed in the virtual world, AR also allows for the simultaneous use of all existing interfaces that surround the user. These surrounding interfaces are often better suited for some tasks than the AR user interface, but without simple ways to move data to and from them, the AR user is discouraged from interacting with them.

Ideally, a hybrid MDE should allow the user to take advantage of the capabilities of different displays and interaction devices in a complementary fashion, using the best available interface for a particular task. There are three specific benefits of AR displays that bring some unique capabilities to hybrid MDEs.

First, because of the close proximity of head-worn AR displays to the user's head, such displays offer a truly *private or personal experience*. This is of particular importance in collaborative scenarios where each participant's view could be augmented with additional information that might be private or personalized based on the user's exper-

tise or preferences. While the other displays in the environment could also be considered private to some extent, only the head-worn AR displays can guarantee truly private views.

Second, by combining the position and orientation tracking and their stereoscopic capabilities, AR displays are also capable of presenting *3D views* that are perspectively correct with respect to the user.

The third unique characteristic of AR displays is that the user actually looks “*through*” them rather than “*at*” them, as is the case with other displays. This feature implies that there can be frequent cases of display overlap in a hybrid MDE, which presents interesting opportunities when the displayed contents complement each other across displays. By providing the contextual and visual overlays on top of other displays, the AR display is uniquely capable of visually connecting all other displays, therefore, “gluing” together information presented on individual displays within a hybrid MDE.

Ideally, the AR display would offer very high resolution, wide field of view, minimal lag, adjustable focus, and be precisely tracked, transparent, very bright for outdoor use, and not more encumbering than a regular pair of eyeglasses. However, such displays are still far from being available today. Even without the perfect AR display, the benefits outlined in the previous paragraphs can still be achieved and therefore make the hybrid MDE systems compelling for tasks requiring combinations of 2D and 3D visualizations and private or personalized views.

### **1.3 Why Freehand Interaction?**

As outlined in section 1.1, one of the major benefits of MDEs is that they have the potential to bring together various input and interaction metaphors and thus make the MDE interaction space richer and more diverse. A key issue in designing MDEs with diverse interaction capabilities is how to provide a consistent way of interacting across displays. This is particularly the case in hybrid MDEs, where there is a large interaction gap between the AR interface and other, more traditional, 2D displays. Most of the AR interaction metaphors borrow heavily from the VR interaction research (e.g., gloves,

wands, and 3D widgets), while many of the other displays rely on more standard input devices (e.g., keyboards, mice and tablets). Providing a consistent interaction metaphor across such different displays is a difficult challenge.

Most of the existing MDEs rely on some *physical controller device* (PCD) as a main interaction method in the environment. A PCD can be implemented as a mouse [Johanson 2002b, Andrews 2004], pen [Rekimoto 1997], wand [Butz 1999, Wilson 2003a, Wilson 2003b], or, more-recently, a personal digital assistant (PDA) device [Nichols 2002, Regenbrecht 2002, Slay 2006]. A PCD provides a control of any particular display by combining some sort of direct or remote pointing mechanism with additional capabilities, such as different control buttons (e.g., [Wilson 2003b]) or a graphical user interface for a set of controls (e.g. [Slay 2006]).

In this dissertation, we free the user’s hands from a particular device and allow them to use intrinsic human gestural or touch capabilities to interact on and across displays. We use the term *freehand interaction* to encapsulate all hand-, gesture-, and touch-based interactions that do not depend on any intermediary tangible devices such as wands, pens, or mice. Given the extensive research in freehand interaction for control of various computer interfaces, it is rather surprising that very little effort has been put into incorporating and evaluating such interactions in MDEs.

The work in this dissertation fills this void and in doing so we are inspired by the work of many researchers who explored human gesture interaction for control of traditional single-monitor systems [Bolt 1980, Baudel 1993, Cutler 1997, Rekimoto 2002, Wu 2003, Wilson 2004].

The goal of this work is not a pure “device-less” user interface, where no tangible devices are used to interact with the system. In fact, we would argue that such a goal is ultimately neither realistic nor possible given the extensive human history of creating various tools to simplify certain tasks. Instead, we explore freehand interactions as a baseline hybrid MDE interaction method. By relying on intrinsic human gesture- and touch-based capabilities as a baseline interaction method across and within the displays, the user is given a direct and simple interaction tool set that transcends the display boundaries and is device-independent. At the same time, by virtue of having their hands

free of a physical device, the user is able to easily acquire additional input devices for certain tasks when there is a clear benefit to doing so.

In order to facilitate this freehand approach, the displays in the hybrid MDE need to be user-aware; that is, the user's hands need to be tracked both on and above the displays. While this constraint seems rather significant at this point, the continuing improvement in multi-touch-sensitive prototypes [Lee 1985, Dietz 2001, Rekimoto 2002, Wilson 2004, Han 2005, Wilson 2005a], the introduction of many mainstream handheld devices that support touch input (e.g. the Sony UX series of ultra-mobile PCs, Nintendo DS, or Apple iPhone), and many glove- and vision-based methods for hand gesture tracking (e.g., [Krueger 1991, Baudel 1993, Kjeldsen 1997, Wilson 2006]), make it likely that most displays of the future will be capable of sensing the user's hands both on and above the display surface. We discuss the different touch and gesture sensing approaches in more detail in Section 2.4.

## 1.4 Contributions

This dissertation attempts to address interaction issues in MDEs through two distinct directions. First, we explore the current commercially available MDE configurations and we address the shortcomings that are present with the standard mouse-based pointer interactions across multiple displays. Second, we focus on a more “exotic” hybrid MDE space that might have a longer time horizon for introduction into mainstream use. Within the hybrid MDE framework we have developed, we present several freehand techniques that improve the interaction within and across the 2D and the 3D displays.

Overall, this thesis makes five contributions toward improving user interaction in systems that span multiple displays, devices, and dimensions:

- *Design, implementation, and evaluation of the Multi-Monitor Mouse (M3) pointer warping techniques* [Benko 2005a, Benko 2007b]: The M<sup>3</sup> techniques accelerate the standard mouse interaction in an MDE by instantaneously relocating the mouse cursor between displays instead of traversing the monitor bezels.

- *Design, implementation, and evaluation of a complex hybrid MDE, Visual Interaction Tool for Archaeology (VITA)*<sup>†</sup> [Benko 2004]: The VITA hybrid MDE combines see-through head-worn AR displays with several additional displays of differing sizes and orientations into an MDE visualization environment for exploration of archaeological excavation data. VITA is built on top of a modular hybrid MDE framework designed to facilitate simple integration of various devices and displays into a cohesive system.
- *Design, implementation, and evaluation of novel gestural interaction techniques, called Cross-Dimensional Gestures*<sup>†</sup> [Benko 2005b]: Cross-Dimensional Gestures facilitate the association and the transition of data across displays, devices, and dimensionalities in a hybrid MDE, by synchronizing the recognition of gestures on both the 2D surface and in the 3D environment.
- *Design, implementation, and evaluation of novel multi-touch interaction techniques, called Dual Finger Selection techniques* [Benko 2006]: These multi-touch interactions allow for precise and accurate selection of small targets on touch screens within an MDE.
- *Design, implementation, and evaluation of a novel hybrid MDE interaction technique, called Balloon Selection* [Benko 2007a]: Balloon Selection improves the accuracy with which 3D targets can be selected by the 3D cursor by decoupling the 3 degree-of-freedom (DOF) selection task into a 2DOF and a 1DOF task, while grounding the user's hands on a multi-touch-sensitive display surface.

The order in which these contributions are presented in this dissertation does not correspond to the chronological order in which they were created. Instead, they are presented in order of increasing complexity of interaction, starting with the simplest homogeneous multi-monitor combinations and ending with significantly more complex hybrid MDE interactions. However, the original motivation for all this work stems from

---

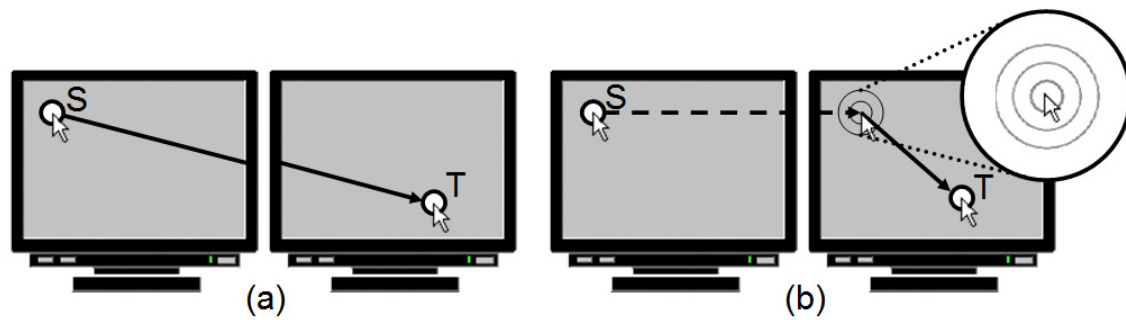
<sup>†</sup> The VITA hybrid MDE and Cross-Dimensional Gestures developed in VITA have been designed in collaboration with Edward Ishak. The contributions discussed in this dissertation focus on particular aspects of those projects that were designed and implemented by the author.

our work in the VITA system (discussed in Chapter 4). While working on VITA, we spent numerous hours programming in a hybrid MDE and we experienced firsthand many of the difficulties with pointer-based interactions across heterogeneous displays. This motivated our investigations into the pointer warping concepts that are described in Chapter 3. In addition, when analyzing the use of VITA with target archaeology users, we quickly realized that the users were extremely limited by the lack of available free-hand interactions for interacting in a hybrid MDE. In particular, our users complained about the lack of methods for transitioning and associating data across displays. This directly motivated us to create novel multimodal techniques that address such issues (Chapter 5). In addition, given the precise nature of archaeological work, the archaeologists demanded more precise interactions for selecting points of interest in both 2D and 3D displays. The techniques presented in Chapters 6 and 7 are our multi-touch solutions to these problems, which try to account for affordances of displays with which the user is interacting and use those affordances to best assist the user in performing precise freehand selection. The focus of all our work is to enlarge the interaction vocabulary available to the user when interacting in an MDE.

In the following sections we describe each of our contributions in greater detail.

#### 1.4.1 Pointer Warping with Multi-Monitor Mouse

Multiple-monitor computer configurations significantly increase the distances that users must traverse with the mouse when interacting with existing applications, resulting in increased time and effort. Chapter 3 presents a set of *Multi-Monitor Mouse* ( $M^3$ ) techniques, which virtually simulate having one mouse pointer per monitor when using a single physical mouse device.  $M^3$  techniques allow for conventional control of the mouse within each monitor's screen, while permitting immediate warping across monitors when desired to increase mouse traversal speed (see Figure 1.4).  $M^3$  incorporates four different implementations of display switch triggering, as well as three different pointer placement strategies after the warp.



**Figure 1.4: Comparing (a) standard bezel traversal and (b)  $M^3$  pointer warping. Instantaneously relocating (warping) the cursor to the next screen, shown as dashed line, reduces the distance traversed by conventional mouse movement. Note: “Sonar” circles are displayed to increase cursor visibility at warp destination.**

The  $M^3$  techniques have been extensively tested in two formal user studies that tested both homogeneous and heterogeneous multi-display configurations. In homogeneous MDEs,  $M^3$  techniques showed significant performance improvements when crossing two or more bezels [Benko 2005a], while in heterogeneous MDEs,  $M^3$  techniques’ benefits grew in proportion to both the distance and the amount of visual-device space mismatch between monitors, and showed significant performance improvements even when crossing a single bezel [Benko 2007b]. Overall, our studies found that pointer warping can improve the mouse performance up to 30%, and all of the users preferred it to the standard mouse behavior. The standalone  $M^3$  utility has been released for free download for Microsoft Windows XP platform (see Appendix A).

#### 1.4.2 VITA Hybrid MDE

Chapter 4 presents a design and implementation of the complex hybrid MDE, called Visual Interaction Tool for Archaeology (VITA) [Benko 2004]. VITA is an experimental collaborative mixed reality system designed for off-site visualization of an archaeological excavation. The VITA system allows multiple users to visualize the dig site in a hybrid MDE environment in which tracked, see-through, head-worn displays are combined with a multi-user, multi-touch, projected surface, a large high-resolution display, and a tracked handheld display (Figure 1.5).





**Figure 1.5: Available displays in the VITA system (left to right): tracked head-worn, projected tabletop<sup>‡</sup>, tracked handheld, and vertical high-resolution display.**

VITA focuses on augmenting the existing archaeological analysis methods with new ways to organize, visualize and combine standard 2D information available from an excavation (drawings, pictures, notes, and videos) with textured, laser range-scanned 3D models of objects and the site itself. To facilitate easy and flexible management of various displays in a hybrid MDE, the VITA framework incorporates a publish-and-subscribe-based messaging framework as well as a multimodal interaction integrator that coordinates speech-, 3D gesture- and touch-based interactions across users, devices, and displays. Section 4.4 presents feedback from the two informal usability evaluations of VITA with archaeological experts.

### 1.4.3 Cross-Dimensional Gestures

The vocabulary of freehand interaction techniques available for interacting across displays and devices in a hybrid MDE, such as VITA, is very limited. In part, that is due to a hard problem of integrating different interaction devices and providing seamless interaction space across different displays. Particularly lacking are transitional interac-

---

<sup>‡</sup> Throughout this dissertation, the white surface of the touch-sensitive projected tabletop is covered with black paper to provide a darker background for imaging the projected display.

tions in hybrid MDEs, which allow the user to seamlessly select and move an object from the 2D environment display to their 3D head-worn display and vice-versa.

Chapter 5 addresses that problem and presents a set of gestural interaction techniques, called *Cross-Dimensional Gestures* [Benko 2005b], that facilitate two fundamental MDE tasks: transitioning and associating data across displays. We prefix our techniques with the term *cross-dimensional* to denote that they not only act across displays and devices, but also across the 2D–3D dimensionality border in hybrid MDEs.

The main contribution of these interactions is that they synchronize the recognition of gestures between the 2D multi-touch display and the 3D tracked glove thus providing the connection between the 2D and the 3D environments (Figure 1.6). In addition to transitioning and connecting the data, we explore gestural modifiers that facilitate of interacting with private data in a collaborative hybrid MDE. The ease of use and intuitiveness of these interactions were tested in an informal usability study and the results are presented in Section 5.3.



**Figure 1.6:** Using the Cross-Dimensional Pull gesture to transform a 2D image of a bowl found on the table into a 3D model of the same bowl above the table.

#### 1.4.4 Dual Finger Selection Techniques

When relying on gestural and touch based interactions in a hybrid MDE, the size of human fingers and the lack of sensing precision can make precise touch screen interactions difficult. By leveraging the capabilities of multi-touch sensitive displays, we have developed a set of five interactions, called *Dual Finger Selection* techniques [Benko 2006], that help the user select very small targets. These interactions, presented in detail in Chapter 6, facilitate pixel-accurate targeting by adjusting the control-display ratio with a secondary finger while the primary finger controls the movement of the on

screen cursor (Figure 1.7). In addition to the dual finger techniques, we present a “clicking” enhancement, called *SimPress*, which reduces the motion errors during clicking and simulates a hover state on devices unable to sense proximity.

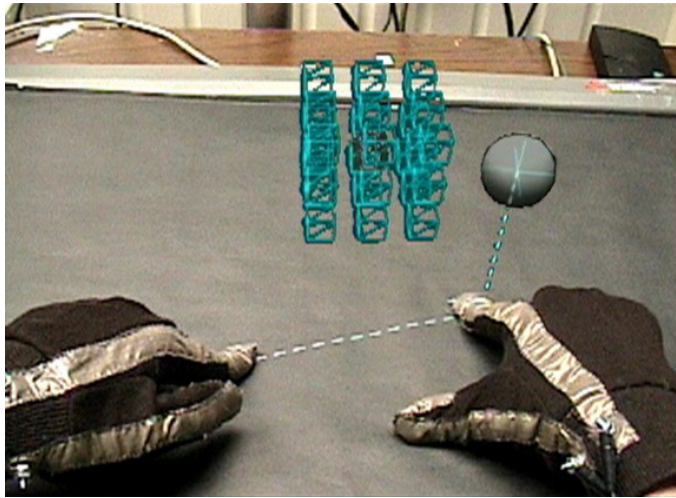


**Figure 1.7: Dual Finger X-Menu technique allows the user to interactively slow down the cursor speed to improve the overall accuracy of selection.**

Two formal user studies evaluated the performance and the error rate of the three most promising techniques (*Stretch*, *X-Menu*, and *Slider*) against our baseline (*Offset*), on four target sizes and three input noise levels. All these chosen techniques significantly outperformed the control technique and were preferred by our participants, with *Stretch* being the overall performance and preference winner (see Section 6.6).

### 1.4.5 Balloon Selection

Inspired by the concept of cross-dimensional interaction and the performance improvements and the error-rate reduction of Dual Finger Selection, we have designed another cross-dimensional technique, called *Balloon Selection* [Benko 2007a], which allows for precise 3D selection in the volume above a tabletop surface by using multiple fingers on a multi-touch-sensitive surface. This interaction, described in Chapter 7, fully exploits the benefits of a hybrid MDE, by combining the interaction capabilities of one display and the visualization capabilities of the other in a synergistic manner.



**Figure 1.8: Balloon Selection technique.**

Balloon Selection is modeled after the real world metaphor of manipulating a helium balloon attached to a string and this metaphor allows the user to effectively decompose a 3 degree-of-freedom (DOF) selection task into a 2DOF positioning task performed by one finger on the tabletop in an absolute 2D Cartesian coordinate system and a 1DOF positioning task performed by another finger on the tabletop in a relative 2D polar coordinate system (Figure 1.8).

The formal user evaluation of Balloon Selection compared it to two well-known interaction techniques for selecting a static 3D target: a 3DOF tracked wand and keyboard cursor keys often found in 3D computer games. Overall, Balloon Selection exhibited the “best-of-both worlds” behavior, showing that users were able to achieve significant reduction in error rate compared to a more direct wand-based 3D interaction (up to three times less), with comparable completion time. Similarly, Balloon Selection was found to be twice as fast as keyboard selection, and without significant differences in the number of errors committed.

## 2 Related Work

This chapter provides an overview of the existing research related to this dissertation. When considering existing MDE research, we restrict our review to systems designed for collocated users and displays (i.e. “same place, same time”). A large research effort is also focused on remote collaboration systems, which connect multiple displays at a distance, but those systems are beyond the scope of this dissertation.

We begin this chapter by reviewing the existing MDE research. Of particular relevance are MDE developments in the areas of ubiquitous computing environments, augmented reality and hybrid user interfaces. Next, we elaborate on the interaction techniques used in various MDEs as well as review some of the ethnographic and empirical studies that explore MDE use. We conclude with an overview of the related gesture- and touch-based interactions that significantly influenced the design of our MDE interactions described in the later chapters. While the related work presented in this chapter is relevant to the entire dissertation, Chapters 3–8 highlight additional related research that is specific to the topics discussed in those chapters.

### 2.1 MDE Implementations

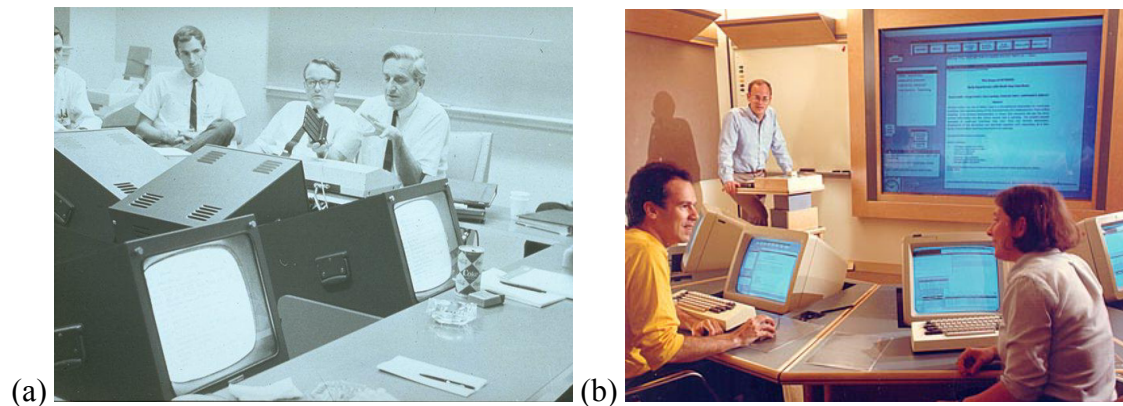
In following sections, we present several research areas that explore MDEs and human-computer interactions within them.

#### 2.1.1 Early MDEs

The earliest systematic exploration of a multi-user MDE can be attributed Douglas Engelbart and his colleagues at the Augmentation Research Center at Stanford Research



Institute [Engelbart 1968]. Their NLS project contained many breakthrough innovations such as a computer mouse, hypertext, object referencing, and dynamic file linking. In addition, they demonstrated these innovations in the first shared-screen collaboration environment. This same system was also used in what probably was the first computer-supported meeting [Bootstrap 2003]. Each participant in this collocated meeting used a mouse to interact with a distributed hypermedia agenda and briefing materials, which were distributed over multiple screens (Figure 2.1a).



**Figure 2.1: Early MDEs: (a) the first computer-supported conference work by Engelbart and colleagues [Bootstrap 2003], and (b) Xerox PARC Colab meeting room [Stefik 1987].**

A more extensive research on computer support for collaborative face-to-face meetings was later conducted at the Xerox PARC *Colab* meeting room [Stefik 1987]. The Colab system was comprised of one large wall-sized display as the interaction focus and many interconnected workstation displays (Figure 2.1b).

### 2.1.2 Ubiquitous Computing Environments

Following the early experiments within the Colab room, Mark Weiser and his colleagues at Xerox PARC began to explore how computers could be better integrated into the everyday office environment. Weiser coined the term *ubiquitous computing* to describe the notion of seamlessly embedding many computers into the physical environment around the user, making the computer essentially “invisible to the user” [Weiser 1991]. A key step in this process is the transformation of the traditional computer workstation as the dominant computing interface, into a collection of many interconnected

devices that become part of the user's surroundings and facilitate typical human activities. The initial research focused on three sizes of display devices, so called "boards" (wall-sized), "pads" (notepad-sized) and "tabs" ("Post-it note"-sized) [Weiser 1993]. Much of the initial effort was spent designing the prototype hardware and middleware software that implemented and connected various devices and displays with the ultimate goal of creating environments with hundreds of available devices.

In many ways, the terms multi-display environment and ubiquitous computing environment have been used to describe similar concepts, but they are not the same. Weiser describes "ubiquitous computing" as a particular style of computing environment where the focus is on human activity and the displays and devices are embedded in the environment with the primary purpose of facilitating those human activities. On the other hand, the term MDE is less general, since it focuses specifically on displays. It describes any kind of computing environment where multiple displays coexist in a synergistic fashion. MDEs range from a simple dual-display multi-monitor configuration on a user's desktop to many complex distributed display environments proposed by ubiquitous computing.

A number of research initiatives have explored combining multiple users, displays and devices into ubiquitous computing environments: *Intelligent Room* at MIT [Brooks 1997], *i-Land* at GMD-IPSI Darmstadt [Streitz 1999], *EasyLiving* at Microsoft Research [Brumitt 2000], *Gaia* at University of Illinois at Urbana-Champaign [Roman 2000], *Aura* at Carnegie Mellon University [Garlan 2002], and *iRoom* at Stanford University [Johanson 2002a].

The Intelligent Room project [Brooks 1997] and the EasyLiving project [Brumitt 2000] aimed to make environments more aware of the human activities going on within them and to embed computation and information services into such intelligent spaces for people engaged in everyday activities. In order to provide "ready-at-hand" computation to the users, these projects focused on creation of self-aware spaces for casual interaction. They used advanced vision-based perception techniques to reduce the dependence on typical mouse and keyboard interfaces.

The i-Land project [Streitz 1999] and its derivatives (*ConnecTables* [Tandler 2001], *RoomWare* [Streitz 2002]) explored embedding displays directly into movable furniture

and investigated collaboration and pen interaction issues across such displays (Figure 2.2). These projects are of particular interest because they allowed the user to dynamically connect two displays to overcome the restrictions of display sizes and borders. Connecting displays in such an ad-hoc manner allowed i-Land researchers to begin exploring the effects of display orientations and visibility on user collaboration.



**Figure 2.2: Roomware® components consisting of several InteractTables® and a DynaWall® [Streitz 2002].**

The focus of Gaia [Roman 2000], Aura [Garlan 2002], and iRoom [Johanson 2002a] projects was on providing the distributed middleware for easy implementation of heterogeneous MDEs. For example, the iRoom collaborative meeting environment supported data and control sharing across devices through a distribution framework, called *iROS*. The *iROS* framework provided an event heap communication mechanism through which all iRoom components can send and receive events. This publish and subscribe mechanism allowed components to remain very loosely coupled, which ensured that the system was robust to potential device failures, as well as open to easy addition and removal of devices from the system [Johanson 2002a].

### 2.1.3 Peripheral and Ambient Displays

Ishii and Ullmer designed *ambientROOM* [Ishii 1997] as a concept prototype environment where standard non-active surfaces (e.g., a ceiling, wall, or closet door) can be used to present subtle peripheral cues that notify the user of a particular event without



disrupting the task at hand. The *Everywhere Displays* project [Pinhanez 2001] enabled any surface in the environment to become both a display and an interaction surface. This projector and camera combination system uses a steerable mirror to transform any surface into an interactive display. Everywhere Displays corrected projection distortions by using computer graphics to warp the projected image so that it appears undistorted on the surface. Pointing and clicking interactions are allowed by detecting hand movements with a video camera. Simultaneously combining multiple steerable projector-camera units was explored by Ehnes and colleagues [Ehnes 2005].

MacIntyre and colleagues designed the *Kimura* office environment [MacIntyre 2001] that uses several large peripheral displays to assist users in managing multiple tasks. Their peripheral displays present interactive montage of images that provide contextual cues of pending tasks and past actions. Recently, another peripheral MDE prototype was created by Forlines and colleagues in which they explored the multi-user and multi-display interaction with a geospatial visualization tool (a *Multi-User Google Earth* [Forlines 2006a]). The focal point of their environment is the multi-user sensing tabletop display, while other peripheral displays provide different 3D views of the geospatial information.

#### 2.1.4 Tangible User Interfaces

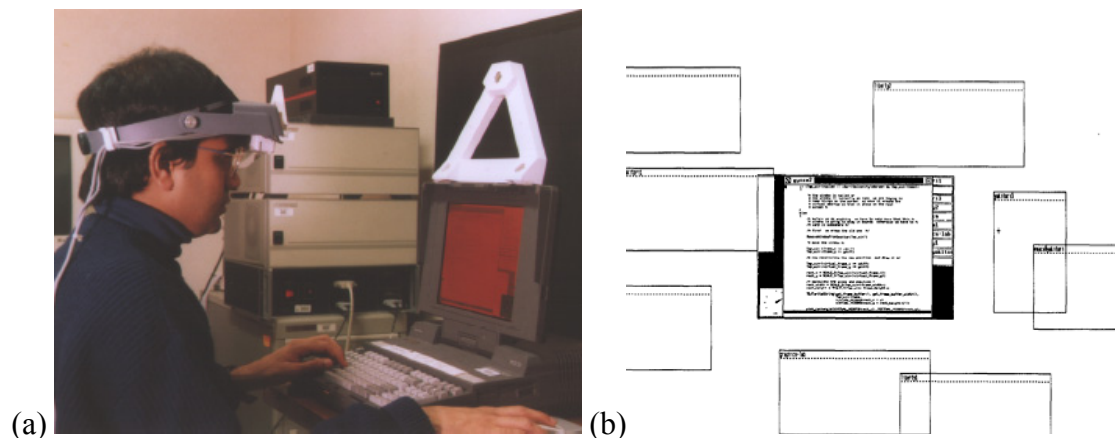
In addition to ambient interfaces discussed in the previous section, Ishii and colleagues explored the concept of *tangible user interfaces*, which allow the user to control digital information by handling and manipulating tangible physical objects such as abstract physical models (e.g., wooden blocks, architectural models, plastic triangles), everyday objects (e.g., dolls, bottles), and sculpting material (e.g., sand) [Ishii 1997, Ullmer 1998, Underkoffler 1999].

Many tangible user interfaces aim to reduce the use of standard graphical user interfaces and instead, use real physical objects to interact with the digital media. In many instances, those physical objects can be used within an MDE as a powerful interaction metaphor for interacting in such an augmented space. For example, the *Urp* project [Underkoffler 1999] explored an urban planning scenario where physical architectural

models were used for easy layout of a city block, while the wind currents or sun shadows were projected on top of them, and a wall sized display presented a virtual 3D view from the first-person perspective from within the model.

### 2.1.5 Augmented Reality and Hybrid MDEs

The idea of using computer generated imagery to overlay additional information on top of the surrounding physical environment dates back to the 1960s, when Ivan Sutherland built the first head-worn optical see-through stereoscopic graphics display [Sutherland 1968]. He demonstrated pioneering AR concepts by presenting simple vector graphics in context of his lab. However, it was not until the 1990s, and the research on presenting instructions for aircraft cable assemblies by Caudell and Mizell at Boeing Corporation [Caudell 1992], that *augmented reality* became a widely accepted term to describe computer graphics overlays on top of the real world. The extensive review of AR and VR technologies is beyond the scope of this thesis, and a more detailed review of these technologies can be found in [Bowman 2004]. We focus here on particular aspects of AR research which explore the combinations of personal head-worn or hand-held displays with additional displays in an environment to form hybrid MDEs.

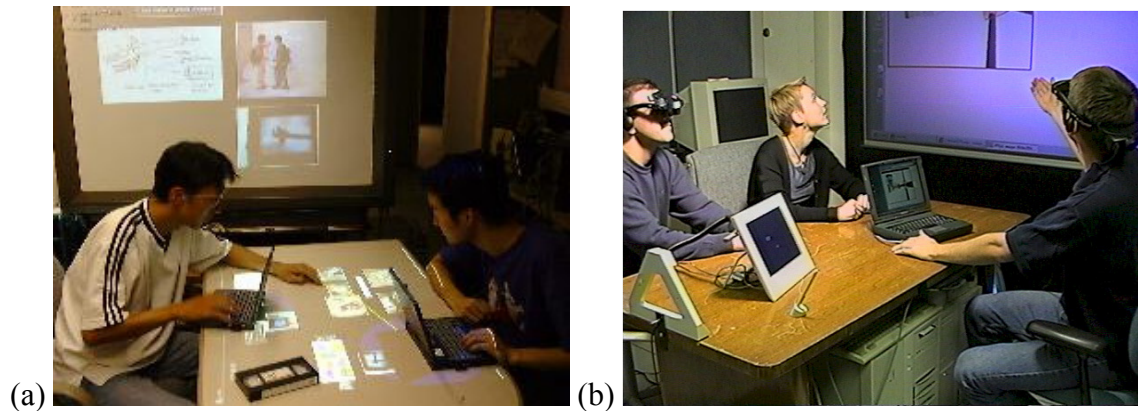


**Figure 2.3: The original *hybrid user interface* [Feiner 1991]: (a) a combination of a tracked see-through head-worn display with a vertical and stationary flat-panel display, (b) the screen capture of a user’s view.**

In 1991, Feiner and Shamash presented the first research on hybrid MDEs, and coined the term *hybrid user interface* [Feiner 1991] (Figure 2.3). They embedded a

small, stationary, desktop, flat-panel “detail” display within the field of view of a monocular, head-tracked, see-through, head-worn “context” display, and they reported on the benefits of simultaneously being able to view a high-resolution “focus” display in addition to the low-resolution contextual information around it.

This *focus-plus-context* approach was later extended by Baudisch who designed an MDE consisting of two stationary displays: a large projected screen surrounding a small high-resolution monitor [Baudisch 2001]. Low and colleagues reversed the approach taken by Feiner and Shamash, using a stereo, head-tracked, opaque, head-worn display to provide detail within the context of a larger monocular, projected display [Low 2003]. The projected display could be seen by the user’s peripheral vision around the loose fitting head-worn display and this increased the user’s field of view.



**Figure 2.4: Collaborative augmented reality environments: (a) Augmented Surfaces [Rekimoto 1999] and (b) EMMIE hybrid MDE [Butz 1999].**

In 1999, two independent projects jumpstarted the development of more complex and collaborative augmented reality environments: *Augmented Surfaces* at Sony [Rekimoto 1999] and *EMMIE* at Columbia University [Butz 1999]. Rekimoto and Saitoh created the Augmented Surfaces prototype [Rekimoto 1999], which connected various portable computers and wall display via an augmented projected tabletop. Their users were not required to wear any head-worn or handheld displays because the projection display overlaid the additional information directly on top of the tabletop surface including any other device that was placed there (Figure 2.4a). Their system allowed the users to smoothly move digital information among the displays by using the built-in

touchpad or pointing stick on portable laptop devices. A camera-based object recognition system was used to track and identify movable devices via attached visual markers.

Butz and colleagues explored a collaborative hybrid MDE meeting scenario in the *EMMIE* project [Butz 1999] (Figure 2.4b). Their system integrated a variety of tracked handheld and stationary displays, head-worn displays, and physical input devices. Optical see-through head-worn displays were used to overlay virtual information on the physical environment, including the displays within it, while the movable hand-held displays in *EMMIE* were used as either “magic lenses” [Bier 1993] or “magic mirrors” to allow modifications to the 3D data visualized in the user’s head-worn displays. The magic lens concept uses a handheld display to present an illusion of being able to see through the display by displaying perspectively corrected computer graphics. The magic mirror concept does the opposite: the display acts as a mirror, i.e. it displays a perspectively corrected computer graphics as a reflection of the environment in front of the display.

The *StudierStube* project [Schmalstieg 2000, Schmalstieg 2002] created a distributed framework for the development of mobile, collaborative and ubiquitous AR applications. Their framework is based on a distributed scene graph architecture and they demonstrated a collaborative storyboard design application in which multiple users (each wearing a head worn display) interact on a large wall-sized display. This application is a prototype cinematic design tool in which each 3D scene is represented as a kind of “world in miniature” and the users can interact with it using tangible tracked props.

The use of tangible props for interacting in a collaborative hybrid MDE has also been explored by Regenbrecht and colleagues in *MagicMeeting* [Regenbrecht 2002]. They present data on several stationary and head-worn displays, and their tangible tracked tools can be passive (e.g., a “lazy Susan” turntable device for inspecting and rotating virtual 3D objects) and active (e.g., a PDA device that is used to transfer the objects between displays).

Combining tangible props with speech commands and interactive tracked displays was explored in the *Herding Sheep* prototype by MacWilliams and colleagues [MacWilliams 2003]. This interactive game demonstration for children was built on top

of a component-based *Distributed Wearable Augmented Reality Framework (DWARF)*, which facilitates rapid prototyping of multi-display AR applications.

Recently, Slay and Thomas explored the idea of combining see-through handheld AR displays with additional displays in a hybrid MDE and they focused their efforts on the development of the *Universal Interaction Controller (UIC)* device [Slay 2006]. The UIC is a tracked PDA device that allows for pointing and selection in a hybrid MDE as well as some simple manipulation of selected data through the graphical user interface on the PDA.

### 2.1.6 Display Clusters

Most of the systems described in previous sections fall into the category of distributed display environments with many different devices and displays. There is another important class of MDEs whose multiple displays are tiled together in an attempt to create a larger seamless display space. Such displays can be powered by a single machine, but more frequently they are powered by a cluster of machines running a highly parallel rendering infrastructure (e.g., *Chromium* [Humphreys 2002] or *SAGE* [Jeong 2006]). The main focus of the research effort in this domain is not to synergistically combine different displays to benefit from the characteristics of each, but rather to use many identical displays in an “as-seamless-as-possible” way to create one larger or higher-resolution display. Many researchers are exploring wall-sized tiled displays [Czemuszenko 1997, Humphreys 1999, Raskar 1999, Leigh 2006] and multi-wall CAVE displays [Cruz-Neira 1992], but the detailed discussion of such systems is beyond the scope of this dissertation.

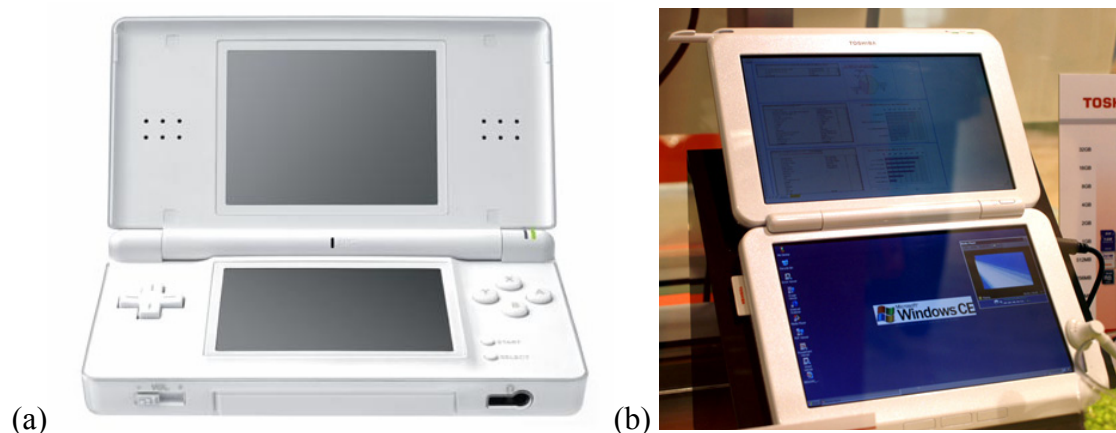
### 2.1.7 Multi-Display Consumer Systems

All previously discussed MDE systems fall under a category of research prototypes and while several companies are commercializing such ideas, they have not yet reached wide market adoption. However, there are several multi-display configurations that are already available on the consumer market today. Apple Macintosh machines started

supporting multiple monitors back in 1988; however, only in recent years did such configurations start becoming increasingly popular in the workplace and at home, thanks mostly to the drop in prices of flat panel displays and the support for dual monitor output on most modern computer graphics cards.

In addition, MDE configurations have become widely adopted in many industries that require their workers to have instantaneous access to large amounts of visual information, e.g., financial industry trading desks, emergency dispatch centers, and air-traffic control towers. For many of those specialized application domains, there has been a steady increase in software and hardware that explicitly take advantage of multi-monitor configurations. For example, Realtime Soft produced the *UltraMon* utility [Realtime 2007] that provides a multi-display-aware window manager that optimizes the use of windows, applications, and taskbars in a multi-monitor setup.

Recently, small portable devices are also beginning to emerge in multi-display configurations. Already, many mobile phones come with a main display and an additional smaller notification display for displaying time and caller identification. Microsoft Windows Vista operating system has extended this concept onto laptop and tablet computers. Their Windows *SideShow* technology [Microsoft 2007] allows the secondary screen to display important notification information (like email, calendar, etc.) whether the device is on, off, or in sleep mode.



**Figure 2.5: Handheld multi-display commercial devices: (a) Nintendo DS Lite game system where the bottom display is touch and stylus sensitive [Nintendo 2006] and (b) Toshiba dual screen eBook stylus-sensing prototype presented at CeBIT 2006 [TechBlog 2006].**

In 2004, Nintendo released a handheld game console, called *Nintendo DS* [Nintendo 2006], which features a clamshell design with two color displays. In addition to the standard button controls, this device features a touch- and stylus-sensitive bottom display which allows the game designers to engage the users in novel play scenarios. The success of the system prompted Nintendo to release an updated version in 2006 called *Nintendo DS Lite* (Figure 2.5a). Interestingly, the Nintendo DS game designers often highlight the display differences by taking into account the display orientation in their game design and often place a map interface on a horizontal (bottom) display and a first-person 3D view on the vertical (top) display.

Several other manufacturers are developing early product versions of dual-display handheld or laptop devices. For example, Toshiba Corporation exhibited a prototype dual-display eBook device at CeBIT in 2006 [TechBlog 2006], which featured two color displays, both of which are stylus-sensitive (Figure 2.5b). The enhanced experience that these kinds of devices offer to their users depends on how successful their designers are in creating interfaces that are truly aware of the multi-display configuration.

## 2.2 MDE Display Classification

We now provide a brief summary of the various kinds of displays that are found in MDEs. Display devices can be characterized along many important internal and external factors. The internal factors convey intrinsic technological capabilities and depend on how displays are built. They include display technology, resolution, color gamut, brightness, contrast, refresh rate, transparency, size, and depth or stereo capabilities. The external factors deal with how displays get integrated into devices and how they are used. They include form factor, mobility, and placement (e.g., worn, held, placed, or mounted).

In analyzing MDEs, we are primarily interested in how multiple displays can best be used together, and, therefore, we extend the ubiquitous computing classification (tabs, pads, and boards) [Weiser 1991] and classify displays based on their external factors into six categories: head-worn, hand-held, laptop, desktop, tabletop, and wall-sized displays.

MDE	Authors	Displays					
		Head-worn	Hand-held	Laptop	Desktop	Tabletop	Wall-sized
Colab	Stefik 1987				✓		✓
PARC Tabs, Pads & Boards	Weiser 1991		✓	✓			✓
Hybrid User Interface	Feiner 1991	✓		✓			
i-Land	Streitz 1999					✓	✓
Augmented Surfaces	Rekimoto 1999			✓		✓	✓
EMMIE	Butz 1999	✓	✓		✓		✓
StudierStube	Schmalstieg 1999	✓			✓		✓
iRoom	Johanson 2002a			✓		✓	✓
MagicMeeting	Regenbrecht 2002	✓	✓		✓		✓
Herding Sheep	MacWilliams 2003	✓	✓	✓	✓	✓	
Head-worn + Projected Wall	Low 2003	✓					✓
VITA	Benko 2004	✓		✓	✓	✓	✓
Multi-User Google Earth	Forlines 2006a			✓		✓	✓
UIC	Slay 2006	✓	✓				✓

**Table 2.1: Comparison of various MDEs with regards to the displays they incorporate. The author’s hybrid MDE system VITA is highlighted in gray.**

This categorization is primarily based on display placement, but it also implies the ordered relationship between display’s form factor, size, and mobility. For example, wall-sized (the equivalent of Weiser’s board) is usually considered the largest and the least mobile category. Tabletop is a growing display category where the table’s horizontal surface is used to display information (e.g. [Dietz 2001, Wilson 2005a]) while the desktop category includes all of the standard computer monitors that are usually placed on the work desk. Laptop in this classification is the rough equivalent of the “pad” (includes both laptop and tablet devices) and handheld is the equivalent of the “tab” category (e.g., PDAs, ultra-portable computers, and SmartPhones). The head-worn category is not part of Weiser’s classification, but it is very important for hybrid MDE as the personal near-eye display capable of completely surrounding the user in an immersive virtual environment.



The Table 2.1 shows which of these different display categories were explored in various MDE systems described previously in Section 2.1. For completeness, we have included our own hybrid MDE system called VITA in the table. The details of the VITA system are described in greater detail in Chapter 4.

## 2.3 Interacting in MDEs

We now focus on human factors issues and review the interaction techniques and input technologies that enable the users of MDE applications to take advantage of available displays. While most MDEs provide some way to interact across the displays, it is important to think about MDE interactions as a superset of regular single-display interactions. Supporting interactive MDEs requires that the designers support both the interaction within individual screens, as well as provide ways to associate those screens and transition data across them.

We examine several research areas that influenced our MDE interaction: MDE mouse pointer interactions, physical controller devices (e.g., pens, wands, panels, and PDAs), ad-hoc display association, and interactions through displays. We conclude this section by examining several studies that evaluated the differences between multiple-monitor and single-monitor use.

### 2.3.1 Interacting across Displays

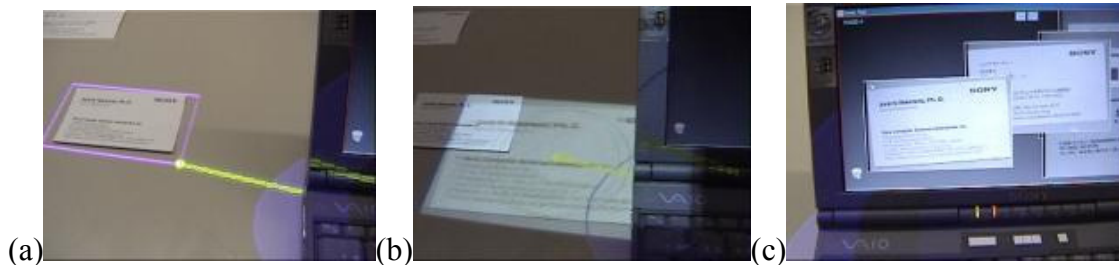
When considering interactions across displays, there are two approaches that dominate the MDE implementations: extending existing mouse-based pointer interactions and using a tracked handheld device (such as a pen, a wand, or a PDA).

#### 2.3.1.1 *Mouse Pointer Interactions*

The standard pointer behavior in all modern operating systems that support multi-monitor configurations is to allow the mouse pointer that reaches the display edge (bezel) to continue its movement on an adjoining display. This *bezel traversal* interaction allows the user to extend the reach of a standard mouse input device (e.g., mouse,

touchpad, trackball, or pointing stick) to use it across several displays. Improving target acquisition across multiple monitors has been explored in context with eliminating warping effects caused by mismatched monitor alignment and differing screen resolutions with *Mouse Ether* [Baudisch 2004], as well as avoiding the need to cross the bezels by bringing the targets closer to the current cursor location with *drag-and-pop* [Baudisch 2003a]. Baudisch et al. also proposed visual enhancements, such as *high-density cursor* [Baudisch 2003b], that increase the visibility of cursors at high speeds.

Many distributed MDE systems also implement this bezel traversal interaction. In such systems, it is usually referred to as *pointer redirection*, since the data from the local pointing input device is redirected to a different device via the network to allow for the control of the remote pointer (e.g., *Courtyard* [Tani 1994], *PointRight* in the iRoom MDE [Johanson 2002b] and *Clicky* in the Gaia MDE [Andrews 2004]). Usually, such MDEs also implement a shared clipboard, which allows the user to seamlessly transition data across devices and displays, by simply dragging it across the bezel.



**Figure 2.6: Using hyperdragging to interact with real and virtual objects [Rekimoto 1999]: (a) the user can use hyperdragging to select a physical object (business card), (b) an image of an object can be taken via the ceiling mounted camera, and (c) the user can then hyperdrag the image to their computer desktop.**

Rekimoto and Saitoh demonstrated this idea within their Augmented Surfaces MDE [Rekimoto 1999], and they called it *hyperdragging*. With hyperdragging, the user's pointer was allowed to escape the laptop computer display and continue its movement on the projected tabletop. In addition to interacting with standard virtual objects, such as pictures or files, hyperdragging also allowed the user to select a tracked physical object on the tabletop and to create a virtual link between real and virtual objects (Figure 2.6).

Booth et al. [Booth 2002] developed *Mighty Mouse*, a mouse sharing groupware tool to allow multiple users to interact on many displays using a pointer. Nacenta and colleagues present a *perspective cursor* [Nacenta 2006] that adjusts the pointer movement and transitions across all displays in perspective space. Each display in their MDE displays a perspective corrected pointer based on the 3D position tracking of the user's head.

### 2.3.1.2 Interacting with Pens, Wands, and PDAs

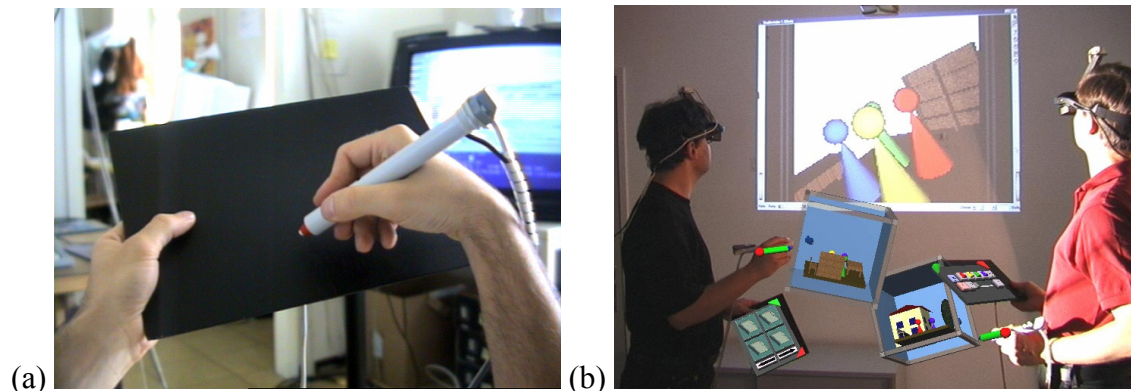
In addition to mouse pointer interactions, many MDE research prototypes use various tracked handheld devices, such as pens, wands, or PDAs, as MDE physical controller devices (PCDs). A PCD controls a display by combining the selection metaphor (usually some sort of ray-based selection) with additional capabilities, such as different control buttons (e.g., XWand [Wilson 2003b]) or a graphical user interface for a set of controls (e.g. UIC [Slay 2006]).

Most of the PCD implementations in MDEs borrow heavily from the extensive research on remote selection (i.e., selection at a distance) in the field of immersive 3D virtual environments. We provide a brief summary of the most influential work in this domain and refer the reader to a book by Bowman, Kruijff, LaViola Jr., and Poupyrev [Bowman 2004] for a more in depth review of various 3D VR selection techniques. The earliest “magic wand” interactions can be attributed to Roberts who created the 3D position-tracked *Lincoln Wand* [Roberts 1966] and Vickers who explored many ray-casting selection techniques with a 3D position-tracked wand in combination with a 6DOF tracked head-worn display [Vickers 1972]. *Flashlight* [Liang 1994] and various aperture techniques [Forsberg 1996] were examined as extensions to simple ray-casting selection, where the linear ray was replaced with a cone-based selection mechanism to account for difficulties with selection at a distance. Ray-casting selection techniques have been formally evaluated in numerous virtual reality experiments since then (e.g., [Bowman 1997, Poupyrev 1997]).

When using the PCD in an MDE, the transfer of data across displays is usually performed via a *copy-paste*-like interaction. For example, Rekimoto developed the

*Pick’N’Drop* technique [Rekimoto 1997] which allows the user to pick (copy) a virtual object on one display by selecting it with a digital pen, and drop (paste) it onto a different one by literally moving the pen to the other display. Similarly, the EMMIE MDE [Butz 1999] presented the user with a 3D cursor attached to a tracked wand which allowed the user to transfer (drag-n-drop) objects from the 3D environment into the 2D screen, by dropping them in the screen’s vicinity. In both of these implementations, the actual pen or wand does not contain any memory to store the copy of the virtual objects, but the underlying network infrastructure is used to simulate this behavior.

Wilson and Shafer developed XWand [Wilson 2003b], a device that incorporated wireless position and orientation tracking, and allowed the user to control a variety of devices in the environment by a combination of pointing and speech commands. In the WorldCursor project [Wilson 2003a], Wilson and Pham extend the XWand work, and introduced a 3DOF controlled laser pointer device that projected a cursor on the physical environment.



**Figure 2.7:** The use of PIP in StudierStube hybrid MDE [Schmalstieg 2000]: (a) the hardware of PIP is composed out of a lightweight (wooden) panel and a tracked pen, (b) multiple users each using their own PIP to interact with 3D and 2D content.

In addition to simple remote pointing, several projects experimented with creating a graphical user interface on a handheld device to enable the remote control of displays or devices. Szalavari explored the use of a tracked pen in combination with a completely passive handheld interaction panel, called *Personal Interaction Panel* (PIP) [Szalavári 1997], onto which a graphical user interface was overlaid in the user’s see-through head-worn display. PIP provided the benefit of passive haptic feedback for the user and

was integrated into several hybrid MDE projects within StudierStube framework [Schmalstieg 2000] (Figure 2.7).

Regenbrecht and colleagues used a PDA device as a PCD in MagicMeeting [Regenbrecht 2002] and used it to display different visual markers. The markers were used as anchors for 3D objects overlaid in the user's head-worn display and the exchange of data between the 2D screen and the 3D AR environment was demonstrated. Automatically generating a graphical user interface on PDAs, specifically tailored for control of a remote device, was explored by Nichols and colleagues [Nichols 2002] and more recently by Slay and Thomas [Slay 2006].

### 2.3.2 Interacting through Displays

The previous sections explored the use of independent input devices to allow for interaction across displays. In this section, we explore the use of display themselves to facilitate interaction.

To simplify interactions with a distant display, several researchers explored replicating a portion of a remote display on a local display (e.g., *Semantic Snarfing* [Myers 2001], *WinCuts* [Tan 2004], and *Frisbee* [Khan 2004]). In this approach, the local display acts as a “portal” to the remote display. For example, *Semantic Snarfing* [Myers 2001] allows the user to select the region of interest on the remote display by pointing at it with a ray-casting-based technique and then that remote region is replicated onto the user's local handheld PDA. The user is then able to modify the remote interface by simply modifying its local replica. Tan and colleagues have further extended this portal metaphor in *WinCuts* [Tan 2004] to allow multiple users to simultaneously interact on a large display, with each user having their own replicated portal on the remote display. In *WinCuts*, replicated regions of interest could even be overlapping.

In hybrid MDEs, tracked handheld displays are often been used as “magic lenses” (e.g., tangible lens on the metaDESK [Ishii 1997], *Herding Sheep* [MacWilliams 2003]) or alternatively “magic mirrors” (e.g., *Vampire Mirrors* [Butz 1998]). The magic lens interfaces assume that the user is looking through them, and by virtue of being tracked, they can present a perspectively correct visualizations overlaid on top of the environ-

ment. Although rarely used, magic mirrors reverse this concept and augment the environment in front of the display.

By default, all head-worn AR interfaces require the user to perceive the entire surrounding environment *through* a see-through display, and therefore could be considered a sort of a magic lens interface. The ability of such interfaces to directly overlap the content of multiple displays is one of the unique features of hybrid MDE configurations as mentioned in Section 1.2.

### 2.3.3 Display Association

In order to be able to interact across displays, the user needs to be aware of the association between displays. The association can be either explicitly created by the user or implicitly built-in by the designers of MDE. For example, the PointRight pointer redirection interaction [Johanson 2002b] requires the user to specify which display edge will be “shared” between two displays. Specifying this edge explicitly associates those two displays in a particular relationship. A lot of the display associations are also implicit. For example, the tracked handheld display, used as a magic lens on the metaDESK tabletop tangible user interface [Ishii 1997], has an implicit relationship with the tabletop display as it presents a 3D view embedded within the larger 2D map visualization projected on the tabletop.

Several research projects developed ways to support ad-hoc connection of displays into meaningful associations. In the ConnecTables project [Tandler 2001], Tandler and colleagues equipped the tablet displays, built into movable furniture, with proximity sensors, which allowed the users to create a shared workspace between two displays by simply moving the displays close together. They relied on pen interactions to transition objects between displays. Swindells and colleagues described a process for identifying and connecting devices through a pointing gesture using custom tags and a custom stylus called the *gesturePen* [Swindells 2002].

Holmquist et al. were the first ones to demonstrate that a connection between two handheld devices can be created by a *synchronized action*. The connection between two handheld devices was accomplished by simultaneously shaking both of them together in

Smart-Its Friends [Holmquist 2001]. Hinckley generalized this approach and proposed a new class of interaction techniques called *synchronous gestures* [Hinckley 2003]. He defined those as interactions that establish a connection between two devices by recognizing a similar pattern of user activity that occurs on both devices in time, or in a specific sequence. Hinckley demonstrated connecting two tablets by bumping them into each other [Hinckley 2003] as well as via pen gestures, called *stitching* [Hinckley 2004]. Stitching was performed by starting a pen stroke on one tablet, moving the pen to the tablet edge, and then continuing the motion on a different tablet. From this single motion both the connection and the relative position and orientation of the tablets would be inferred.

Rekimoto and his colleagues explored the use of a dedicated “Sync” button installed on various devices [Rekimoto 2004] to establish connections between multiple devices. They simplified the synchronized gesture metaphor by requiring the user to synchronously press the Sync button on two devices to connect them.

### 2.3.4 Experimental Studies of MDE Interactions

Unfortunately, there are relatively few studies that systematically examine the influence of different MDE configurations on human factors. Furthermore, we are not familiar with any study that systematically examined a hybrid MDE configuration. This is partially due to the relative immaturity of those research prototypes and their significant implementation complexity. However, studies of the use of such systems are clearly needed to identify the benefits and limitations of hybrid MDE configurations.

Several interview-based ethnographic studies explored the ways that people use multiple monitors in comparison with the single-monitor use. Grudin interviewed designers and programmers who use multi-monitor configurations in their daily work and concluded that users treat them as logical means of organizing their applications within their visible space [Grudin 2001]. Ringel investigated the use of virtual desktops to multiple monitors, and concluded that while preferences were split across her participants, the multi-monitor users found the main benefits to be the ability to view two documents

simultaneously in order to compare them and the ability to peripherally monitor some part of the system for change [Ringel 2003].

Hutchings and Stasko also interviewed users who use multi-monitor configurations to understand their space management practices and found that even when having a large available display space, the users will not always try to fill it up completely with content, but that they tend to hide windows for variety of reasons and tend to maintain some “empty space” [Hutchings 2004c]. In addition, Hutchings and colleagues suggested *window visibility time* as a potential metric for evaluating multi-monitor use [Hutchings 2004b].

Tan and colleagues studied the differences between interacting across multiple monitors, when two monitors are separated in depth, but occupy the same visual angle. They observed that (even at similar visual angles) placing information on displays that are separated in depth is more detrimental to performance than the corresponding position at similar depths [Tan 2003]. The observed effect was relatively small, but statistically significant (about 10% performance decrease). Baudisch’s Mouse Ether work [Baudisch 2004] studied the discontinuities caused by mismatched display resolutions and sizes, and concluded that such discontinuities have a significant effect on bezel traversal pointer performance.

Su and Bailey report on a study examining different multi-monitor configurations that varied in angle and distance on a contiguous vertical plane [Su 2005]. Their work presents several guidelines for positioning large displays in interactive workspaces and, in particular, they advise that separating displays more than a subtended visual angle of 45° or orienting them at an angle larger than 45° relative to each other can significantly hinder performance.

## **2.4 Freehand Interaction**

Given the large amount of research in human gesture- and touch-based interaction for control of various computer interfaces, it is rather surprising that very little effort has been put into incorporating them in MDEs. MDE implementations rely, almost exclusively, on the use of tangible objects and PCDs for interaction.



This dissertation explores this void and we present several gestural and touch-based interaction approaches [Benko 2004, Benko 2005b, Benko 2006, Benko 2007a] designed for hybrid MDEs (Chapters 4–7). Recently, Forlines and colleagues have also explored the use of a multi-user touch-sensitive tabletop display to interact with several peripheral displays [Forlines 2006a]. In their system, the user can annotate and manipulate a map-based visualization (Google Earth [Google 2006] application) on a touch-sensitive tabletop and simultaneously see different 3D views of that same map area displayed on the peripheral displays.

This dissertation is significantly influenced by previous research in the use of bare hands as direct input to control various computer systems. We review the previous work in gestural interactions with standard displays and 3D virtual environments, and discuss research on bimanual and multi-touch interactions.

#### 2.4.1.1 *Hand Gesture Theory*

Sturman [Sturman 1992] explored the applicability of hand interactions to control of computer tasks. For example, he used various degrees-of-freedom of the human hand to control the animation of several walking virtual characters. His analysis is independent of the application or sensing modality and he develops a design method that can be used to evaluate the suitability of the hand gestures for a particular task.

Quek has studied vision-based hand gesture interfaces and his work provides a useful taxonomy for classifying human movement into different gesture categories [Quek 1995]. He classifies human gestures as symbols and acts and indicates that humans typically use whole hand or finger movement, but rarely combine the two.

In addition to the human-computer interaction field, hand gestures have also been extensively researched in computer vision and pattern recognition (e.g., [Darrell 1993, Starner 1995, Pavlovic 1997]) and psychology (e.g., [Kendon 1994, Krauss 1998, Kita 2003]); however, a review of those areas is beyond the scope of this thesis.

### 2.4.1.2 *Gestural Interaction with Standard Displays*

Combining hand position and orientation tracking with speech commands was first demonstrated in Bolt's "Put-That-There" system [Bolt 1980]. His pioneering multimodal system allowed the user to interact with a map on a large display by talking and pointing at it.

Krueger's *VIDEOPLACE* [Krueger 1982] is an early example of a human-computer interface that allows the user to interact with 2D video silhouettes through multi-hand and multi-finger interaction. Krueger used video cameras to track 2D hand position and he performed image processing to detect various hand features (e.g., thumb and fingertip). These features were then automatically mapped to control points on the screen, which resulted in a rich vocabulary of simple gestures. For example, the silhouette of the fingers and the thumb on both hands were used to manipulate the control points of a Bezier curve. Vincent's Mandala system [Vincent 1993] extended this concept and used analog chromakeying to embed the image of a real person into a virtual environment. The Mandala system was used for stage performances in which the actor could interact with various virtual objects in a completely virtual environment.

Baudel and Beaudouin-Lafon [Baudel 1993] presented the *Charade* system, which allowed the user to control the visual presentation on a large screen through a gestural language composed of hand poses and hand motions. Kjeldsen explored the use of vision-based tracking to facilitate free-hand interactions with a standard on-screen mouse pointer on a traditional single display desktop system [Kjeldsen 1997].

Wilson has also explored the use of various free hand gestures to control standard interfaces (e.g., free-hand pointing to control the mouse pointer [Wilson 2005b] and pinch gestures for two-handed manipulation tasks [Wilson 2006]). Recently, Malik and colleagues demonstrated an upgraded version of Krueger's original idea, and they used vision-based hand tracking over a tabletop surface to perform various multi-finger and whole-hand gestures to interact with a remote display [Malik 2005].

### 2.4.1.3 *Gestural Interaction in 3D Virtual Environments*

Probably the best showcase of the expressiveness of gestural techniques in 3D virtual environments comes from Mapes and Moshell. Their Polyshop system [Mapes 1995] uses hand pose and hand motion gestures as the only interaction method in the environment. Interaction in Polyshop is based completely on the use of gestures for all navigation, manipulation, and control tasks.

The Responsive Workbench [Cutler 1997] provides a set of single- and two-handed gestural interactions for manipulating 3D objects on a stereoscopic, rear-projected tabletop VR display. The authors report that the combination of a glove and stylus worked better than two gloves for the majority of their interactions, and cite the significant benefits of the interactive tabletop as being able to ground the user's hands and use the support of the surface. An example of their interactions is a two-handed zoom, where the non-dominant hand sets the focus point, and the preferred hand zooms by moving towards or away from the focus point.

Numerous other projects explored the use of natural 3D gestures for interacting in 3D virtual environments (e.g., [Wexelblat 1995, Nishino 1997, MacKenzie 2001, Kaiser 2003]). Additionally, several projects explored the use of 2D gestures to select and manipulate 3D objects (e.g., *Interactive Shadows* [Herndon 1992], *SKETCH* [Zelevnik 1996], and *head-crusher* image plane selection [Pierce 1997]).

### 2.4.1.4 *Bimanual Interaction*

Much research has been performed on bimanual interaction in user interfaces. In their pioneering work, Buxton and Myers [Buxton 1986] demonstrated that users tend to parallelize interaction tasks between hands, gaining significant performance improvements. A year later, in 1987, Guiard proposed the Kinematic Chain Model [Guiard 1987], which highlighted the asymmetric nature of human hands in many bimanual interactions. This model proposes that, for most tasks, our hands function in series rather than parallel (this model is thus called a kinematic *chain*). For asymmetric bimanual tasks this means that the dominant hand tends to articulate with respect to the non-

dominant hand, i.e., that the non-dominant hand “sets the stage” for the dominant hand to interact on.

Building on top of Guiard’s model, Bier et al. [Bier 1993], in their *Toolglass and Magic Lenses* system, allowed the user to control the transparent tool palette with the non-dominant hand, while the dominant hand controlled the primary cursor with the mouse. Bier reported that this simultaneous bimanual operation improved interaction speed due to the removal of many inefficiencies usually associated with switching modes. However, the research by Kabbash et al. [Kabbash 1994] points in the opposite direction, claiming that requiring the user to coordinate actions of their hands in order to perform two-handed interactions may complicate the overall task and slow down the performance.

#### 2.4.1.5 *Multi-Touch Technology*

While the initial research in multi-touch-sensing surfaces began with the prototype capacitance-sensing tablet by Lee, Buxton, and Smith [Lee 1985] and the camera-sensing surface by Richard Greene [Greene 1985], the recent surge in multi-touch prototypes [Matsushita 1997, Dietz 2001, Westerman 2001, Rekimoto 2002, Wilson 2004, Han 2005, Jorda 2005, Wilson 2005a] has significantly increased the amount of research interest in designing interactions that exploit more than a single contact with the surface.

Significant inspiration for our own work came from numerous projects that investigated multi-touch gestural interaction on large displays. *HoloWall* [Matsushita 1997] uses back-projected infrared illumination, which is reflected by the users’ fingers or objects. This reflected light is then captured by cameras to identify the location of the fingers or objects. The brightness of detected points is used to detect hovering above the surface. Wilson uses a similar approach in his *TouchLight* prototype [Wilson 2004] (Figure 2.8). He uses a transparent holographic screen as the interaction surface and in contrast to other multi-touch devices, *TouchLight* does not track individual touch points, but uses the optical flow computation to extract the motion of the contacts on the

surface and enable motion-driven gestural interactions. Our work in Chapter 6 is based on a tabletop prototype that is similar to HoloWall and TouchLight.

*Digital Vision Touch* (DViT) [SMART 2003] is a related vision-based touch-sensing technology which also uses infrared light spectrum to do the detection, but places four cameras in the corners of the screen rather than behind the screen. This camera placement allows DViT technology to be used with many standard display technologies and the system is able to reliably sense objects hovering over the surface. The triangulation technique used allows the system to track a maximum of two independent touches.

Another recent implementation of vision-based multi-touch sensing is Han's multi-touch drafting table surface [Han 2005]. His system uses an interesting property of light called frustrated total internal reflection (FTIR) which allows him to only provide infrared illuminant at the edge of the screen, while the cameras performing the touch sensing are mounted behind the screen, similar to the HoloWall and TouchLight prototypes. Han and his colleagues demonstrated numerous applications for his tabletop that use multi-touch gestures to control navigation in maps, orient 3D objects, invoke menus, and aid in texture-mapping of 3D surfaces.



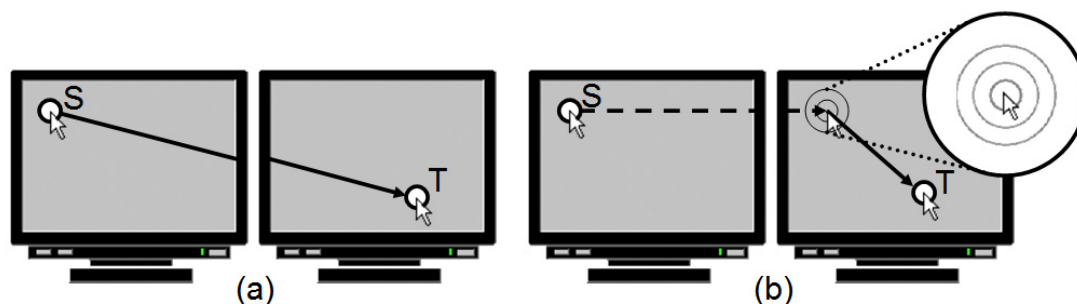
**Figure 2.8:** Using hand gestures to interact with map data on the TouchLight multi-touch prototype [Wilson 2004].

Capacitance-based sensing has also been used to track fingers on interactive surfaces. Rekimoto's *SmartSkin* project [Rekimoto 2002] presented a capacitance-sensing surface capable of detecting multiple simultaneous touches. He demonstrated several single and multi-touch interactions for manipulation of digital photographs.

The Mitsubishi Electric Research Lab DiamondTouch prototype [Dietz 2001] was primarily designed to investigate multi-user collaborative scenarios around the tabletop [Wu 2003, Ringel-Morris 2004, Shen 2004]. DiamondTouch surface requires the users to be electrostatically coupled with the tabletop (usually by standing or sitting on a conductive pad) in order for their touches to be detected. However, a serious limitation of the DiamondTouch technology is that it is only capable of reporting the largest bounding axis-aligned rectangle that encapsulates all current user touches (i.e., it does not report multiple contacts, but rather the bounding area of touches). Butler and Amant [Butler 2004] overcome this limitation via specially constructed gloves to enable precise detection of individual fingers. Our work in Chapters 4, 5, and 7 uses the DiamondTouch surface for detecting multi-touch interactions, and the work in Chapter 7 uses a solution similar to Butler and Amant's gloves to detect individual touches.

### 3 Improving Pointer Interaction in MDEs

The standard pointer behavior in all modern operating systems that support multi-monitor configurations is to allow the pointer that reaches the display edge to continue its movement on an adjoining display. This *bezel traversal* interaction is present not only in when displays are connected to a single machine, but also in many distributed MDE systems that support pointer redirection (e.g., hyperdragging [Rekimoto 1999], PointRight [Johanson 2002b], or Clicky [Andrews 2004]).



**Figure 3.1: Comparing standard bezel traversal and pointer warping in a homogeneous MDE: (a) *Bezel Traversal* – Using standard pointer movement to move between monitors from S to T; (b) *Pointer warping* – Instantaneously relocating (warping) the cursor (dashed line) to the next screen, reduces the distance traversed by conventional mouse movement. Note: “Sonar” circles are displayed to increase cursor visibility at warp destination.**

This chapter presents an alternative to bezel traversal, called *pointer warping*, which we define as “instantaneous relocation of the mouse pointer between two displays” in an MDE. Rather than requiring the user to move the pointer across monitor bezels, we argue that warping the pointer between screens, in most cases, provides a better interac-

tion metaphor for cursor migration in MDEs. The difference between pointer warping and bezel traversal is illustrated in Figure 3.1.

The notion of using pointer warping to overcome larger distances has been explored previously only within a single monitor in a combination with eye gaze (e.g., eye gaze interaction [Sibert 2000] and MAGIC pointing [Zhai 1999]) or mouse gestures (e.g., flick [Dulberg 1999]).

In the following sections, we first discuss the relevant issues concerning pointer interactions in MDEs. Then, we present the design and the implementation of a set of pointer warping interaction techniques, called *Multi-Monitor Mouse (M<sup>3</sup>)* [Benko 2005a, Benko 2007b], that facilitate pointer warping across displays, but leave the within-display mouse interactions unchanged. We conclude this chapter by discussing the results of two user studies that explore how M<sup>3</sup> affects user's performance in homogeneous and heterogeneous MDE configurations.

### **3.1 MDE Bezel Traversal Issues**

Most of the problems with pointer interactions in MDEs are caused by three interrelated factors: increased distances that the pointer needs to traverse, gaps between displays, and mismatches between resolution, size, and alignment of displays.

#### **3.1.1.1 Increased Distances**

Allowing the user to access the adjacent display via standard mouse interaction has significantly extended the amount of available desktop space. However, this increased desktop space forces the user to move the mouse cursor across monitor bezels and cover larger distances in order to access the entire display area. To make matters worse, most users of multi-monitor systems tile their displays in a horizontal manner, which results in one dimension (width) being drastically larger than the other. This tiled arrangement can cause excessive clutching when traversing multiple displays with a mouse.

To compensate for increased distances, users can increase pointer speed or acceleration, which have drawbacks pointed out by Baudisch and colleagues [Baudisch 2003b].



To correct for these problems, they propose visual enhancements, such as *high-density cursor*, that increase visibility of cursors at high speeds. An opposite approach, is to avoid the need to cross the bezels altogether by bringing the targets closer to the current cursor location with *drag-and-pop* [Baudisch 2003a]. While this approach works relatively well if the user just wants to quickly access a remote target without leaving the original display, it does not address the more general scenario of actually transitioning the mouse pointer between displays (e.g., for the purpose of switching focus between applications).

Recently, Forlines and colleagues proposed *HybridPointing* [Forlines 2006b], which lets the user switch easily between absolute and relative pointing to enable access to distant and close targets on a large display. This approach might also be promising for an MDE, but it requires a tracked remote pointing device rather than a standard mouse, which makes it less appealing to the majority of desktop MDE users. In addition, the experiments by MacKenzie and Jusoh [MacKenzie 2001] show that remote pointing interaction drastically lag in performance when compared to standard desk-based mouse interaction.

### 3.1.1.2 *Gaps between Displays*

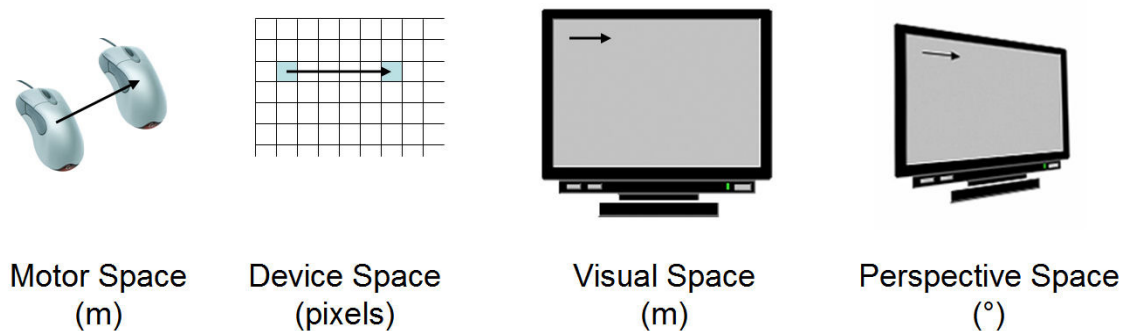
In addition to increased distances, the bezel traversal interaction is significantly affected by the physical gaps between monitors [Baudisch 2004]. Mackinlay and Heer explored the discontinuities in straight lines caused by the gaps between displays [Mackinlay 2004]. They focused their efforts on making “seam-aware” MDE application, that take into account the magnitude of the gap and they modify the visualization of data across displays to compensate for those discrepancies. Their seam-aware interface techniques effectively reduce the negative effects of visual seams (bezels).

### 3.1.1.3 *Display Mismatch*

Probably the most significant problem with bezel traversal in heterogeneous MDEs is the discontinuity caused by mismatched display size, alignment and resolution. In particular, transitioning the pointer across mismatched displays can have unpleasant ef-

fects. For example, the same amount of physical mouse movement (on the desk) can result in drastically different amounts of mouse pointer movement on two displays that have identical resolution, but are of different size. This is further exaggerated when one considers the displays from the user’s perspective, where the orientation and the distance between the displays and the user further distort the path and the velocity of mouse movement.

It is therefore important to be aware of four distinct spaces when designing interactions for heterogeneous MDE applications: motor, device, visual, and perspective (Figure 3.2).



**Figure 3.2: Comparison of four different display spaces or coordinate systems (possible units of measurement are given in parentheses).**

*Motor space:* The space in which the user operates a particular input device, such as a pen or a mouse. The motor space movement is measured by units of physical space, for example, hand movement (in cm) on the tabletop.

*Device space:* This space is determined by the pixel resolution of the frame buffer of the particular display. The movement of the pointer in device space is determined by the number of pixels traversed. This space is what the computer graphics community calls *device coordinates*.

*Visual space:* This space measures movement in physical space on the screen (e.g., cm on the screen) as defined by the physical size of the screen (i.e., width and height). For example, two displays with similar device space (same resolution), can occupy drastically different visual space if they differ in size, as their visual pixel sizes will be different.

*Perspective space:* This space measures the movement relative to the user's point of view (e.g. radians or subtended arc as seen by the user). To account for the user's perspective space deformations, the system needs to know the positions of the user's eyes and the position and orientation of the displays in the environment. This means that the displays need to be either tracked in 6DOF or at least calibrated in 6DOF, and the user's eyes need to be tracked with 3DOF. Since tracking the eyes is difficult, this process is usually approximated by tracking the user's head in 6DOF, measuring the inter-ocular distance, and estimating the exact eye positions [Deering 1992].

Most current computer interfaces consider only the first two spaces (motor and device). In fact, mouse interaction can be defined as a mapping between the motor and the device space. The ratio of pixel movement to hand movement is referred to as *control-display (C/D) ratio*, and attempts have been made to adjust it interactively (e.g., semantic pointing [Blanch 2004]). However, in heterogeneous MDEs, device, visual and perspective space can also be mismatched between displays causing substantial discontinuities in mouse interaction across displays. Some approaches have been proposed that address behavior in visual or perspective space, and we argue that pointer warping can reduce the effect of display mismatch in Section 3.4.

Mouse Ether [Baudisch 2004] attempts to eliminate the MDE display mismatch problem by adjusting the pointer speed in visual space on all monitors so that the pointer moves at a consistent visual speed irrespective of the monitor resolution. In addition, Mouse Ether allows the pointer to temporarily travel in an off-screen space to eliminate discontinuities caused by the screen borders.

The problems with bezel traversal are even more exaggerated in the case of non-static displays or non-planar display alignment. While operating systems that support multiple-monitor configurations assume that the displays are static and that a single virtual plane (virtual desktop) could be stretched across all of them, heterogeneous MDEs can include mobile devices, such as laptops or PDAs, which makes the static, single plane assumption illogical. Nacenta and colleagues present a more general approach to display position and orientation differences, called perspective cursor [Nacenta 2006], that adjusts pointer movement in perspective space. Each display in their MDE displays a perspectively corrected pointer based on the 3D position and orientation of the user's

head. In collaborative scenarios, performing perspective correction in an MDE is not advised, since visualizations only appear correct from the single user's perspective. For the additional user, the interface appears distorted and therefore presenting the content in display space is recommended. However, in hybrid MDEs, the correct perspective visualizations can be done per user in their head worn display without disturbing the views of the additional collaborators.

### **3.2 Multi-Monitor Mouse (M<sup>3</sup>)**

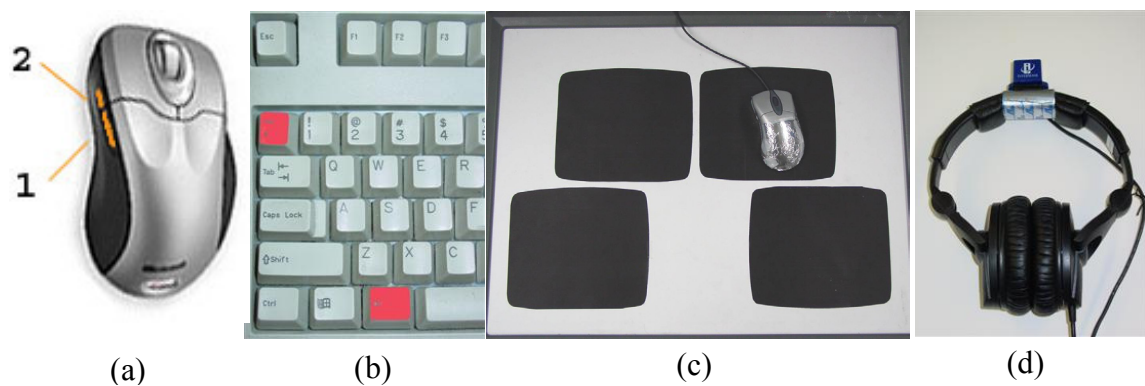
Our approach does not attempt to correct the pointer path distortions in bezel traversal, but strives to eliminate the bezel traversal altogether. The M<sup>3</sup> application (widget) runs in the background of Microsoft Windows XP, minimized to the system tray (see Appendix A). When the M<sup>3</sup> widget is launched, it reads system information about the size, number and relative location of attached screens and forms a corresponding set of virtual “frames” to represent the screens. When the user issues a frame-switch command, M<sup>3</sup> warps the mouse pointer to the new frame (display). The new location of the cursor is signaled to the user by invoking the “mouse sonar” animation around the pointer (Figure 3.1b), a built-in Windows option that enhances pointer visibility. Otherwise, pointer movement is completely unaffected by M<sup>3</sup>. (For example, the user is still free to move the pointer across screen boundaries by physically moving the mouse, but now has the option of directly warping to a different screen.)

M<sup>3</sup> segments the pointer space according to screen space divisions, thus allowing for pointer warping across screens. By warping the pointer, M<sup>3</sup> virtually simulates having one mouse pointer per monitor when using a single physical mouse device. While the techniques presented in this thesis have been formally studied for warping between displays in a multi-monitor configuration, M<sup>3</sup> also allows the same techniques, without modification, to be applied to any desktop space by dividing it into a set of virtual rectangular frames. These frames can be of arbitrary number and size, and can even overlap. For example, a large high-resolution monitor could be divided into several virtual frames, each containing the windows for one application. M<sup>3</sup> would in this case switch the mouse pointer between different applications.

There are two important decisions that need to be considered when designing pointer warping interactions: how to trigger the warp to perform a frame switch and where to place the mouse pointer in the target frame after the frame-switch has occurred.

### 3.2.1 M<sup>3</sup> Frame-Switching Alternatives

We experimented with several devices to evaluate methods for triggering the frame switch (see Figure 3.3). Two of the final five designs (keyboard and mouse button switches) require only standard computer peripherals, thus making them easy for most computer users to adopt. The other three (mouse-location, head-orientation, and head-orientation–mouse switches) support more direct switching, but require extra equipment.



**Figure 3.3:** Devices used to trigger the pointer warp: (a) side mouse buttons, (b) keyboard key combination, (c) mouse location (on different mouse-pad-shaped areas on a location sensing surface), and (d) head-orientation (3DOF orientation tracker fixed to the headphones).

#### 3.2.1.1 Mouse-Button Switch (MB)

The mouse button switch command is issued by pressing one of the two side buttons on the five-button Microsoft IntelliMouse Explorer mouse (Figure 3.3a). Since multi-monitor configurations are typically side-by-side arrangements, we decided to map the top side button to advance the frames forward (clockwise), and the bottom side button

to advance the frames backward (counterclockwise). This decision is technically arbitrary, but we believe that it has ecological validity in that it mimics the behavior of the mouse itself when those side buttons are pushed: Pushing the top button would tend to rotate the whole mouse clockwise, while pushing the bottom button would have a counterclockwise effect. The displays in our MDE form a virtual loop, making it possible to cycle through all the screens using just one of the buttons.

### 3.2.1.2 *Keyboard Switch (KB)*

The keyboard switch is modeled after a built-in Windows task-switch command (ALT+TAB). We mapped the forward frame switch to the ALT+“~” key combination (Figure 3.3b), and the backward frame switch to ALT+SHIFT+“~”. As with the mouse button switch, looping is supported. This mode, although implemented, was not evaluated in our experiments because of its functional similarity to mouse button switch.

### 3.2.1.3 *Mouse-Location Switch (ML)*

This switching method is based on the idea that every screen could have a corresponding mouse-pad. The user still manipulates only one physical mouse, but physically placing the mouse on a different pad warps the cursor to the screen corresponding to that pad. We implemented mouse-location switch using a multi-touch-sensitive surface (Mitsubishi Electric Research Lab DiamondTouch table [Dietz 2001]) on which the user can define any axis-aligned rectangle as a pad for a given screen (Figure 3.3c). To aid the user in remembering the locations of the virtual mouse pads, we provided paper mouse-pad cutouts to be placed on the surface. Since the table operates through electrostatic coupling, the mouse is wrapped in aluminum foil to allow it to be tracked by the DiamondTouch surface when held by the user.

The Keystroke-Level Model [Card 1980] can be used to predict that mouse-location switch would be slower than mouse-button or keyboard switch for a small number of monitors because the time it takes for the hand to move the mouse to a new mouse pad is greater than the time it takes to click a button a small number of times. However, we

wanted to explore whether the users would prefer the arrangement where each monitor can directly be accessed by placing the mouse onto a corresponding mouse pad.

#### 3.2.1.4 *Head-Orientation Switch (HEAD)*

By observing the work of several individuals in a multi-display environment, we noticed that a user's head position does not change much, but their head orientation changes continuously, depending upon the screen on which they are working. At a constant working distance from the user, the larger the screens, the larger the horizontal angle subtended by each screen that can be reliably measured with an absolute orientation sensor. We outfitted a pair of headphones with a 3DOF orientation sensor (InterSense InertiaCube2) and measure the user's head orientation to determine the screen at which they are looking (Figure 3.3d). When the user turns their head towards another screen, the head-orientation switch performs a frame switch. Independently of  $M^3$ , Ashdown and colleagues presented another implementation of homogeneous multi-monitor pointer warping using only head orientation switch [Ashdown 2005].

The  $M^3$  application contains a simple head-orientation calibration routine that goes sequentially through all displays, displays a visual target in the middle of each display, and then asks the user to look at the target. Once the application acquires the head-orientation values for all monitors, it determines the threshold angles between them that trigger monitor switches.

#### 3.2.1.5 *Head-Orientation–Mouse Switch (HEADMB)*

This is a hybrid technique that combines head orientation measurement (for determining the screen at which the user is currently looking) with a side mouse button trigger (to trigger a warp to that screen). The head-orientation switch (described above) suffers from the “Midas touch” problem, causing the cursor to warp across monitors even when the user just wants to glance over without switching monitor focus. To eliminate such spurious switching, we introduced a mouse button trigger, as suggested by Jacob and Sibert [Jacob 1991], which adds an additional click overhead while switching, but still reduces the number of clicks compared with regular mouse button switching. This

hybrid approach was implemented and tested in our second pointer warping experiment (Section 3.3.2) in response to the feedback received from the participants of our first experiment (Section 3.3.1.7).

### 3.2.2 M<sup>3</sup> Pointer-Placement Strategies

In addition to deciding on how to trigger the frame switch, there are several possibilities for where to warp the mouse cursor in the target frame after the frame switch has occurred. Since warping creates a visual discontinuity, it is critical to reduce the time that the user spends on visually searching for the cursor on the target screen. Three strategies were considered: *fixed-location center*, *frame-relative*, and *frame-memory*.

#### 3.2.2.1 *Fixed-Location Center Strategy (C)*

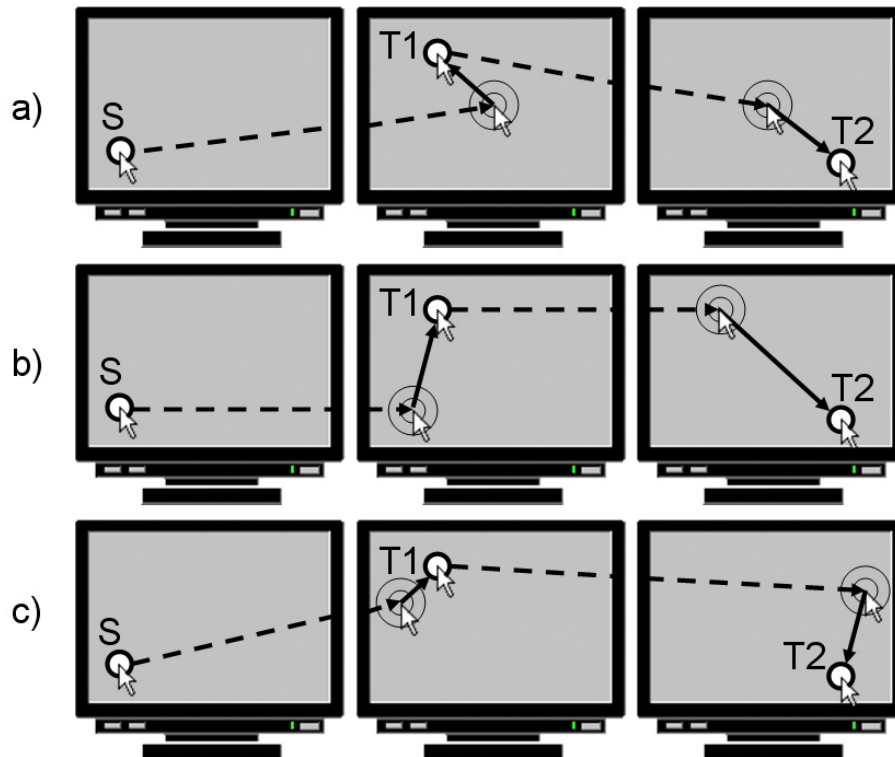
The initial implementation of M<sup>3</sup> used the single *fixed-location center* placement strategy of always warping the cursor to the center of the next frame (Figure 3.4a). While the center location is somewhat arbitrary, it does ensure that the maximum mouse traversal distance after the frame switch will always be at most half of the frame's diagonal. This can be beneficial if users distribute their tasks equally around the center, which is often the case with active working windows. However, it can be a nuisance when the target is located near an edge, which is the case for some frequent selection tasks, such as accessing a taskbar. In our current M<sup>3</sup> implementation, it is possible to select any fixed location as the warping target, as long as that location is available on all frames.

#### 3.2.2.2 *Frame-Relative Strategy (FR)*

The *frame-relative* placement strategy works by translating the pointer to the next frame at the same location relative to the new frame's upper left corner as it was relative to its old frame's upper left corner (Figure 3.4b). For example, if the user's pointer cursor is located in the bottom left corner of the screen and the warping is triggered, the pointer will warp to the bottom left corner of the next screen.



To facilitate this behavior in heterogeneous MDEs, where the displays could be of different resolutions and sizes, this strategy recorded the pointer position on a given screen as a “percentage” of that screen’s width and height. Therefore, the pixel location of the pointer (100,200) on a 1024×768 pixel display would be represented as (9.76%, 26.04%). When warping to a 1280×1024 display, the pointer would appear at the pixel location (125,267) on the new display.



**Figure 3.4: Traversing between S, T1 and T2 locations using different  $M^3$  pointer placement strategies: a) fixed-location center, b) frame-relative, c) one of many possible frame-memory scenarios. Dashed lines indicate warping; solid lines indicate conventional movement.**

This strategy essentially collapses the entire available space into one frame of mouse movement. In addition, this is the only strategy we implemented in which the effect of pointer movement prior to the frame switch will not be negated by the switch itself.

### 3.2.2.3 Frame-Memory Strategy (FM)

The *frame-memory*\* placement strategy, considers all frames as completely independent spaces. Under this strategy, the  $M^3$  widget remembers the last location of the cursor in each frame and warps the incoming cursor to that location. Thus, the last position of the cursor when the user warps out of a frame, becomes the starting location when the user eventually warps back to that frame (Figure 3.4c).

## 3.3 User Evaluations of $M^3$

$M^3$  pointer warping implementations have been evaluated in two formal user studies. We explored the behavior of pointer warping in various MDE configurations and the studies we designed aimed to answer several important questions. Is pointer warping useful in MDEs? Is it faster than bezel traversal? Which frame switch method is preferred? Which pointer strategy is the best (for a particular task)? Do people like/prefer pointer warping?

The first experiment compared several  $M^3$  triggers and strategies in a homogeneous MDE consisting of four identical tiled displays (Section 3.3.1). The second experiment evaluated the best triggers and strategies from the first experiment together with some previously untested techniques in a heterogeneous MDE consisting of three displays that differed in size, resolution, and orientation to the user (Section 3.3.2).

### 3.3.1 Experiment 1: Pointer Warping in Homogeneous MDEs

Eight right-handed participants (six male, two female, ages 23–32), all unfamiliar with the  $M^3$  techniques and with no connection to our lab, participated in an experiment that tested pointer warping in homogeneous MDEs. Participants were recruited from various departments at Columbia University, and received a small monetary compensation for their participation.

---

\* In the publication in which it was introduced [Benko 2005a], *frame-memory* placement strategy was referred to as *frame-dependent*.



**Figure 3.5: Experiment 1 setup: Homogeneous MDE test setup consisted of four monitors and four corresponding “mouse pads” used by mouse-location switch. The background is set to an inactive spreadsheet image to simulate a visually noisy working environment.**

### 3.3.1.1 Setup

The experiment was conducted on a single computer (dual Xeon 2.6GHz PC, 2GB RAM) with four identical monitors tiled in a horizontal arrangement, driven by two ATI Radeon (9000 and 9800) graphics cards. All four monitors were Samsung SyncMaster 240T (24" diagonal, 1920×1200 resolution, 60Hz refresh), which resulted in a total desktop space of 7680×1200 pixels (see Figure 3.5).

The monitors were arranged in a semicircle of 80cm radius, with 12cm horizontal separation (including bezels) between each monitor’s display, and the user was seated in the center to ensure equal distance and viewing angle to all monitors. Thus, each display occupied a 35° horizontal viewing angle with 8° separation. Head orientation was measured by an Intersense InertiaCube2 tracker mounted on a set of headphones Figure 3.3d). These were worn by participants throughout the entire experiment to eliminate the potential confound of wearing them only during head-tracking conditions. The mouse speed was set to the default Windows XP setting.

### 3.3.1.2 Method

We decided to test regular unassisted bezel traversal mouse interaction (CTRL mode) with three  $M^3$  frame-switch modes: mouse-button (MB), head-orientation

(HEAD), and mouse-location (ML). We tested each switch mode using two pointer-placement strategies: frame-relative (FR) and fixed-location (at the screen center) (C). This resulted in a total of seven different conditions. The order of presentation for these seven conditions was counterbalanced across users to eliminate the effects of ordering from the data.

The study design was within-subjects and it consisted of:

- 7 blocks (one per condition) ×
- 2 direction (left-to-right and right-to-left) ×
- 3 distances ×
- 5 trials
- = 210 trials per participant

We felt that the increased memory load associated with the frame-memory placement strategy would decrease its performance in our experimental task given our four-monitor setup so we did not include this strategy in this study. However, this strategy was later evaluated in a heterogeneous MDE and the results of that study are presented in Section 3.3.2.

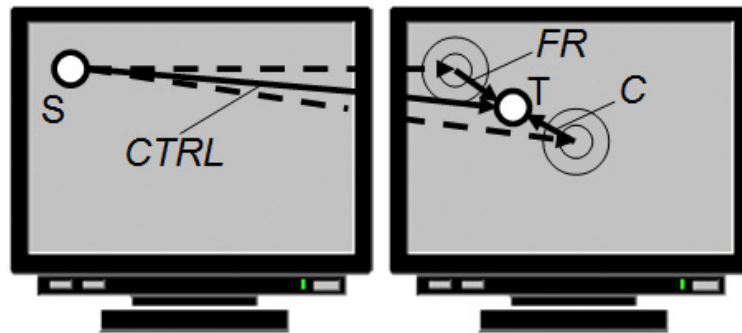
### **3.3.1.3**      *Procedure*

After greeting the participant, the experimenter gave a brief tutorial demonstrating each of the seven conditions. Each user was allowed to familiarize themselves with all mouse conditions, and performed a block of 10 practice trials before completing the block for each condition. Each block consisted of five trials for each of three different start-target distances and two directions (right and left) for a total of 30 movements per block. This resulted in 210 recorded trials (seven blocks) for the entire experiment. The total running time per session was approximately one hour. At the end of the experiment, the participant completed a satisfaction questionnaire.

### **3.3.1.4**      *Task*

The task was based on a Fitts' Law target acquisition task [MacKenzie 1992], but without any variation of start and target sizes (fixed at 30 pixels square). To eliminate target discovery overhead, we presented the participant with both start and target but-

tons at the same time, asked them to locate both before starting a trial, and recorded the time it took between clicking on the start button and clicking on the target button. We selected distances of 2134, 4047, and 5956 pixels, such that each required crossing one, two, or three sets of bezels, respectively.



**Figure 3.6: Start (S) and target (T) button layout in our user study. To eliminate strategy bias, T is located at the half-way point between the warping locations for frame-relative (FR) and center fixed-location (C) strategies. Control path (CTRL) is also shown.**

We carefully positioned the start and target buttons to eliminate strategy bias. Each target button was located at exactly the half-way point between the warping locations for FR and C strategies (i.e., at the midpoint between the center of the screen and the frame-relative location of the start button on that target screen), as shown in Figure 3.6. The direction varied between left-to-right and right-to-left.

### 3.3.1.5 Hypotheses

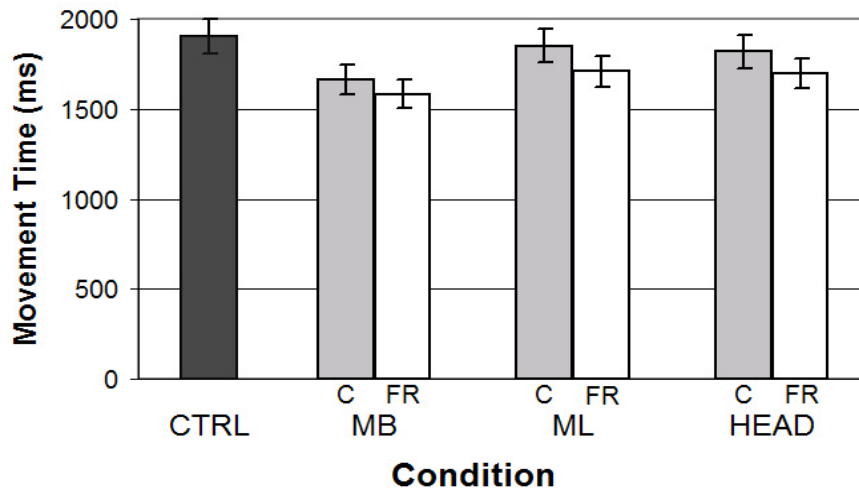
We postulated two hypotheses before beginning the experiment:

**H1:** The participants would acquire targets faster when using  $M^3$  relative to the standard bezel traversal as the number of screen bezels that needed to be crossed increased.

**H2:** In addition, we speculated that the users would be faster using the frame-relative strategy rather than the center fixed-location strategy because of the utility of movement prior to the frame switch in the former strategy.

### 3.3.1.6 Results

Movement times were first cleared by removing outliers (movement times more than two standard deviations larger than the mean for each condition), which accounted for less than 3.5% of all the trials. All data analysis was performed on a median movement time for each participant, distance, direction, and condition combination. We performed a 7 (Condition)  $\times$  2 (Direction)  $\times$  3 (Distance) repeated measures Analysis of Variance (RM-ANOVA), with our participants as a random variable. There were significant main effects for all three factors.



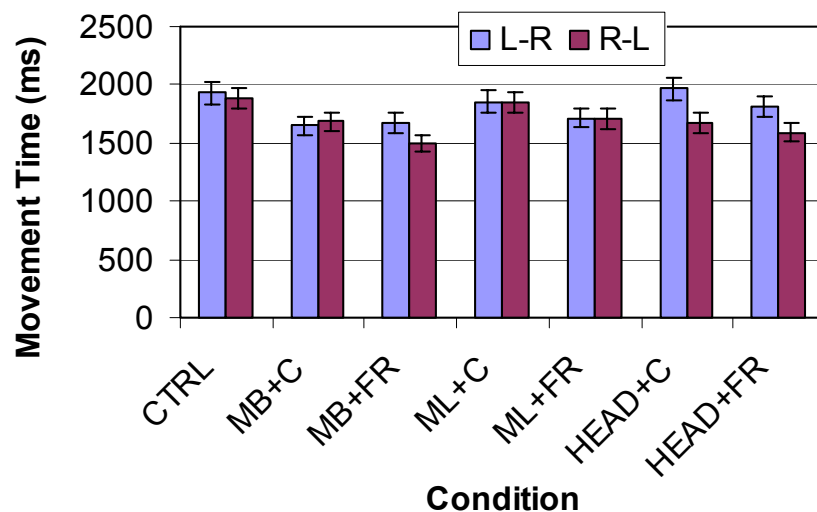
**Figure 3.7: Mean movement times (ms) for different conditions ( $F_{(6,42)}=3.88$ ,  $p<0.01$ ), shown with 95% confidence intervals.**

As expected, the Distance factor contained a significant effect,  $F_{(2,14)}=82.72$ ,  $p<0.001$ , with larger distances requiring the longer targeting time. The Condition factor effects were significant as well,  $F_{(6,42)}=3.88$ ,  $p<0.01$ , shown in Figure 3.7. In all cases the FR strategy outperformed the C strategy by an average of 0.1s (5%), which we believe is due to the C strategy discarding pointer movement prior to the frame switch. This confirmed the H2 hypothesis.

Of the  $M^3$  modes using the FR strategy, MB (1584ms) was the fastest compared with CTRL (1906ms), presenting a 17% improvement in performance ( $t_{(7)}=13.02$ ,  $p<0.001$ ). MB was followed by HEAD (1698ms) and ML (1710ms), which were both

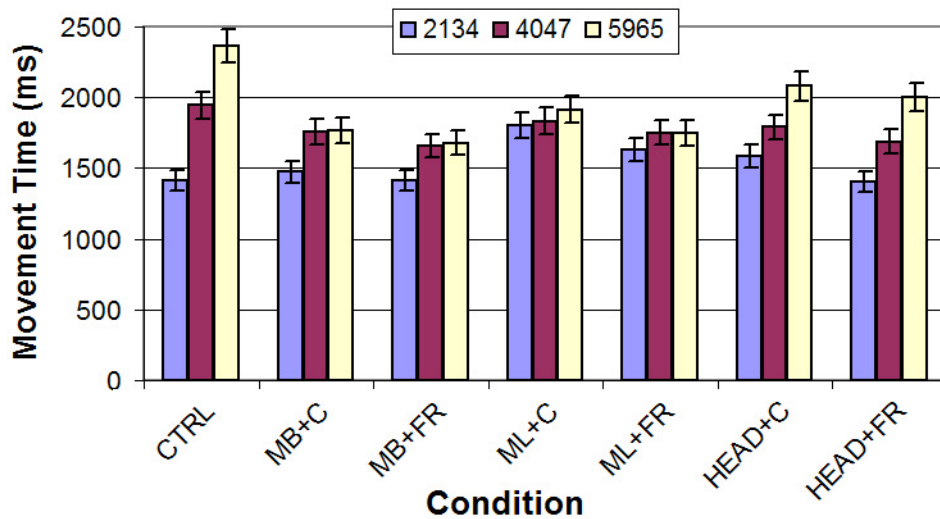
significantly faster than CTRL. The C strategy conditions (MB+C, ML+C, HEAD+C) also significantly outperformed the CTRL condition.

Surprisingly, the Direction factor also contained a significant effect,  $F_{(1,7)}=82.72$ ,  $p<0.001$ , as transitioning right-to-left was on average 0.2s (11%) faster than going left-to-right<sup>†</sup>. Our study gives us relatively limited data to make definitive conclusions; however, we speculate that this behavior might be attributed to the brain differences in visual field processing when performing reaching or aiming movements. Human motor behavior research suggests that the right brain hemisphere, that controls our left visual field, exhibits superiority for the localization and acquisition of visual targets [Carson 1996]. It is also interesting to consider the interaction of Condition and Direction ( $F_{(6,42)}= 6.1391$ ,  $p<0.001$ ), which shows that this Direction discrepancy was not present in all our Conditions. In fact, the biggest Direction difference in targeting times occurred with the HEAD mode, where right-to-left performance is about 15% faster on average than the left-to-right, while ML modes show no significant Direction differences at all (Figure 3.8). While this phenomenon seems to have little effect on the overall results of our study, we believe that it should be further explored in a future study.



**Figure 3.8: Interaction of Direction and Condition for mean movement times (ms) ( $F_{(6,42)}= 6.1391$ ,  $p<0.001$ ), shown with 95% confidence intervals.**

<sup>†</sup> Our previous publication [Benko 2005a], contained a reporting error where left-to-right was said to be faster than right-to-left.



**Figure 3.9: Interaction of Distance and Condition for mean movement times (ms)** ( $F_{(12,84)}=9.509, p<0.001$ ), with 95% confidence intervals.

The interaction of Distance and Condition (Figure 3.9) had significant effects,  $F_{(12,84)}=9.509, p<0.001$ . It is interesting to notice that for the shortest distance (2134 pixels) there were no significant differences between movement times across conditions. All of the  $M^3$  performance gains come from time saved when traversing two or more bezels, with the biggest gains (up to 29%) being present at the largest distance (5965 pixels). This confirmed H1.

### 3.3.1.7 Subjective Evaluations

The participants filled out a post-experiment questionnaire rating their experience on a 10 point Likert scale (1 being most negative and 10 being most positive). They were asked to comment on the ease of use of each condition and their subjective performance rating with each condition. The participants' subjective ratings strongly correlate with our experimental data. The MB+FR condition was rated the easiest to use and the fastest (with identical averages of  $\mu=7.9$ ), while CTRL was rated the hardest to use ( $\mu=6.1$ ) and slowest overall ( $\mu=4.9$ ).

In addition, the participants were asked to state their overall preference for a condition and a strategy. Overall, all participants strongly preferred some  $M^3$  condition to CTRL, and six out of eight users preferred the FR strategy over the C strategy. While

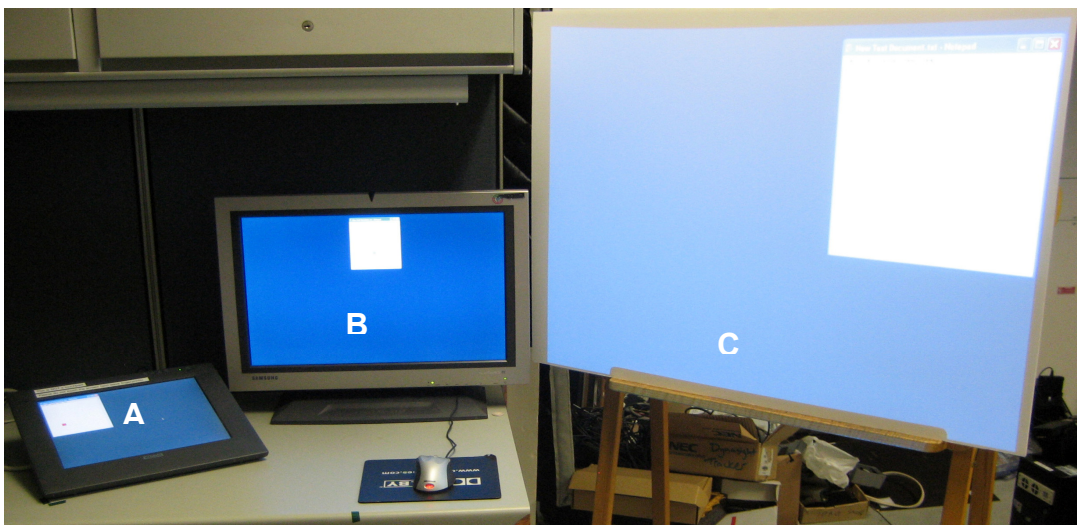


seven out of eight users preferred the MB mode over all the other modes, three mentioned that they would actually prefer a combination of HEAD and MB modes that would resolve the “Midas touch” issue of pure HEAD mode. As mentioned earlier in Section 3.2.1.5, this suggestion was implemented as the head-orientation–mouse switch (HEADMB) and tested in the follow-up experiment in heterogeneous MDEs.

### 3.3.2 Experiment 2: Pointer Warping in Heterogeneous MDEs

To evaluate the performance of the  $M^3$  pointer warping techniques in a heterogeneous MDE, we conducted a second formal user study with ten right-handed participants (seven male, three female, ages 21–27), all unfamiliar with  $M^3$  techniques and with no connection to our lab. The participants were recruited by mass email to students in our department, and received a small monetary compensation for their participation.

We wanted to evaluate whether pointer warping would yield more benefits in a heterogeneous MDE than in the homogeneous MDE case (see Section 3.3.1) possibly because the standard bezel traversal in such environments introduces significant distortions and discontinuities in the pointer paths and speeds. To test this, we created a heterogeneous MDE shown in Figure 3.10.



**Figure 3.10:** Heterogeneous MDE used in Experiment 2 consisting of a small low-resolution near-horizontal display (A), a medium high-resolution vertical display (B), and a large low-resolution vertical display (C).

### 3.3.2.1 Setup

The experiment was performed on a PC (dual Xeon 2.6GHz PC, 2GB RAM) running Windows XP Pro, with two ATI Radeon 9800 and 9000 graphics cards. The virtual desktop was extended over three displays of different orientation, resolution, and size. The displays were arranged in a semicircle (approximate radius 80cm) around the participant's seat and ordered by increasing diagonal size from the left: A (15"), B (24"), and C (48"). Table 3.1 summarizes the relevant characteristics of the displays. The NEC WT600 projector used for the right display (C) incorporates an aspheric mirror, allowing it to be placed extremely close to the screen to keep the participant out of the beam.

<b>Position</b>	<b>Left A</b>	<b>Middle B</b>	<b>Right C</b>
<b>Type</b>	Wacom Cintiq 15X LCD	Samsung SyncMaster 240T LCD	NEC WT600 Projector
<b>Size (Visual Space)</b>	12"×9" (15" diag.)	20.5"×12.75" (24" diag.)	38"×29" (48" diag.)
<b>Resolution (Device Space)</b>	1024×768	1920×1200	1024×768
<b>Visual Pixel Size</b>	0.28mm	0.24mm	1mm
<b>Orientation</b>	Near- horizontal	Vertical	Vertical

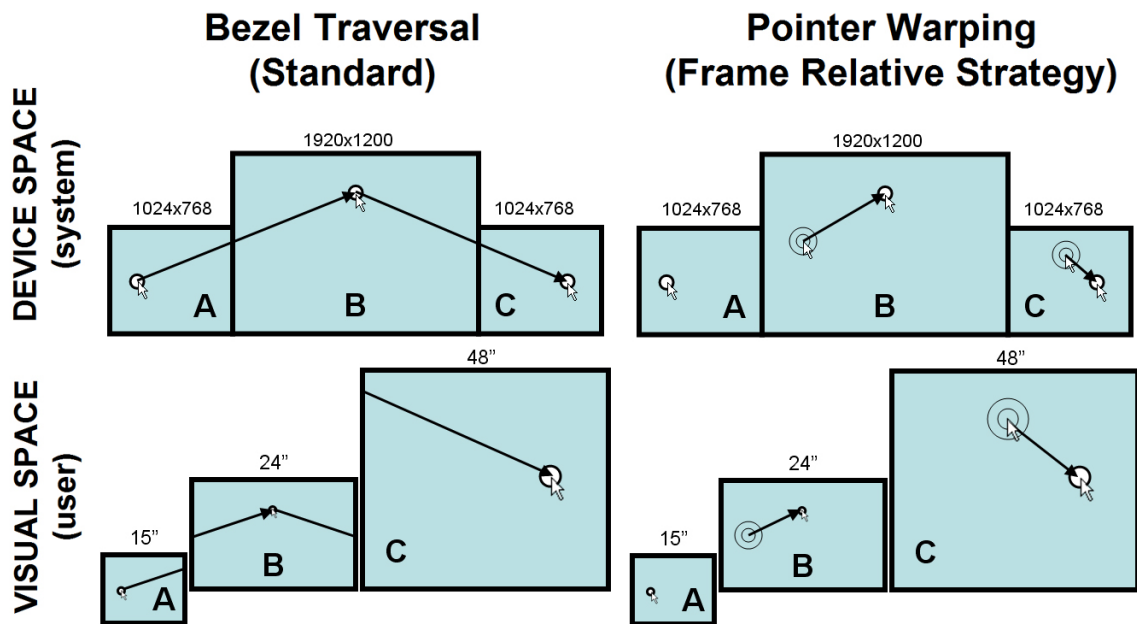
**Table 3.1: Displays used in the experiment.**

Similarly to the previous experiment, head orientation was tracked by an InterSense InertiaCube2 tracker mounted on a set of headphones worn by participants throughout the entire experiment to eliminate the potential confound of wearing them only during head-tracking conditions. Mouse pointer speed was kept at the default Windows XP setting.

We were primarily interested in examining how the mismatch between the displays' visual and device spaces influences the bezel traversal and pointer warping interactions. We use the term *device space* to describe the system's perspective of the desktop space,

where the number of pixels determines the area. In contrast, *visual space* is the user's view of the desktop space which is determined by the physical display size. For a detailed discussion of the mismatch between visual and device space please refer to Section 3.1.1.3.

Our choice of displays created an environment where displays A and C had the same device space, but different visual space. The device spaces (system's perspective) of all displays were aligned at the bottom, and Figure 3.11 shows the comparison between the device and the visual space behaviors for both standard bezel traversal and pointer warping (with frame-relative strategy). Table 3.2 summarizes the types of mismatch present when moving between monitors in our experimental environment.



**Figure 3.11:** The differences in device space (top) and visual space (bottom) representations of our experimental setup. Traversing a continuous device-space path (top left) with a standard pointer can be a cognitively demanding task in visual space (bottom left). Pointer warping aids the user by removing the need to traverse the bezels (right). Concentric “sonar” circles help increase the visibility of the cursor after the warp by highlighting the destination.

Screen Transition	A–B	A–C	B–C
Visual Space Mismatch	2.42	10.20	4.21
Visual–Device Space Mismatch	0.85	3.57	4.17
Orientation Mismatch	73°	73°	0°
Bezel Crossings	1	2	1

**Table 3.2: Display transition characteristics in our experiment.**

*Visual space mismatch* represents a ratio between the areas of the two screens and provides us with a simple metric of the visual size difference between displays. (Note that a more accurate metric would take into account the distances between the displays and the user and their orientation to determine the solid angle they subtended in the user’s visual field.)

*Visual–device mismatch* is the ratio of visual pixel sizes between displays. It is important to note that all screens were used at their native resolution, which caused the largest adjacent monitor mismatch between the visual space and the device space to occur between B and C (i.e., pixels on C are 4.2 times larger than on B).

*Orientation mismatch* is the pitch angle difference between screens. The left display (A) was mounted at a near-horizontal angle of 17° with respect to the desk surface, as per ergonomic guidelines suggested by the manufacturer (Wacom). Thus, A was offset by 73° about the horizontal, relative to the other two displays. Note that arranging all displays in a semicircle around the participant ensured equal mean distance to each display. Therefore, we do not consider differences in yaw as orientation mismatch. Su and Bailey [Su 2005] found that for the optimal performance for a stationary user, the vertical displays should not be positioned in the same plane, but positioned at an angle of up to 45° with respect to each other to ensure equal visual angles and minimal amount of distortion. Our setup follows their guidelines; however, note that Nacenta and colleagues [Nacenta 2006] take a more general approach to display position and orientation differences.

### 3.3.2.2 Method

We decided to test standard unassisted mouse movement (CTRL) and compare it to four pointer warping combinations: mouse button with frame relative (MB+FR), mouse button with frame memory (MB+FM), head-orientation–mouse with frame relative (HEADMB+FR), and head-orientation–mouse with frame memory (HEADMB+FM). This resulted in a total of five different conditions. The study design was a 5 condition  $\times$  2 direction (left-to-right and right-to-left)  $\times$  9 paths (specific paths across displays)  $\times$  4 trials within-subjects design. In our experiment, each participant performed five blocks of 72 trials for a total of 360 trials per participant. Each block tested one condition (CTRL, MB+FR, MB+FM, HEADMB+FR, or HEADMB+FM) and the order of presentation of blocks was counterbalanced across participants. Each block consisted of four identical trials for each combination of nine paths and two directions (R and L). All trials within a block were randomized to reduce ordering and learning effects.

In summary, the experiment consisted of:

- 5 blocks (one per condition)  $\times$
- 9 paths  $\times$
- 2 directions  $\times$
- 4 identical trials
- = 360 trials per participant

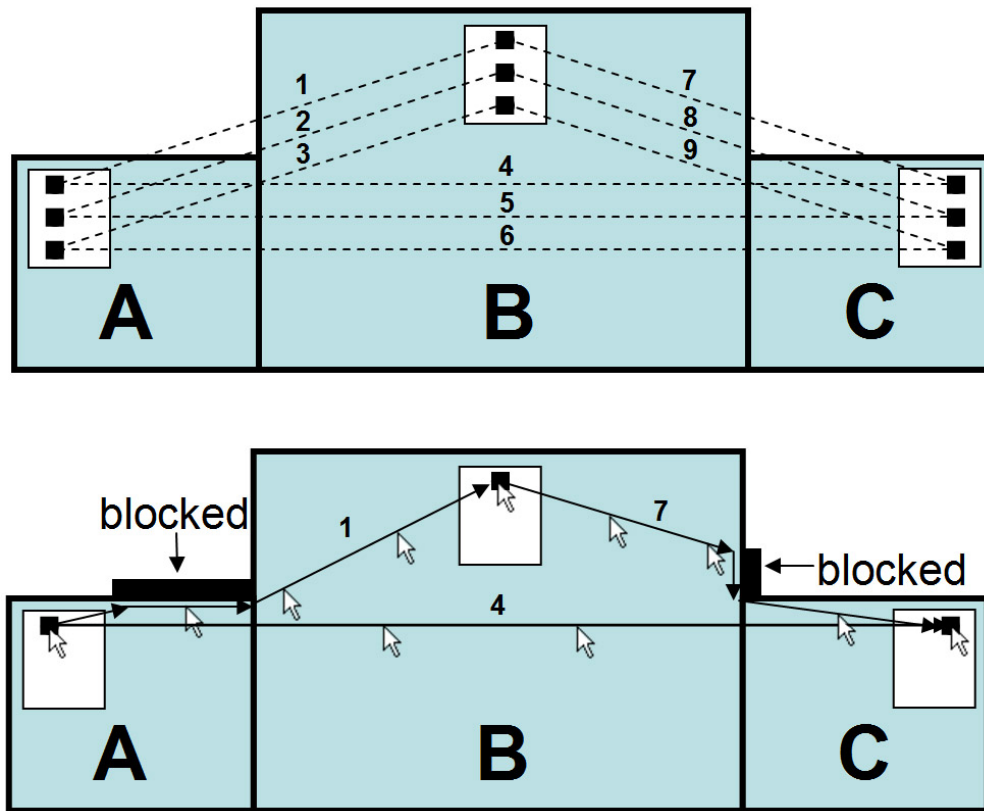
### 3.3.2.3 Procedure

After greeting the participant, the experimenter gave a brief tutorial demonstrating each of the 5 conditions. Each user was allowed to familiarize themselves with all conditions, and then performed a block of 32 practice trials before completing the block for each condition. The participant completed the entire session with the experimenter watching. Total running time per session was approximately 1 hour. At the end of the experiment, the participant completed a satisfaction questionnaire.

### 3.3.2.4 Task

The task was based on a Fitts' Law target acquisition task [MacKenzie 1992], but without the variation of start and target sizes (fixed at 25 $\times$ 25 pixels). To eliminate the

overhead of the visual search time, the participant was presented with both start and target buttons simultaneously, asked to locate both before commencing a trial, and the elapsed time between clicking on the start and target buttons was recorded as our dependent variable.



**Figure 3.12:** Our experimental task setup consists of nine paths (top), shown here as dashed lines, each connecting two of nine targets, distributed over three screens. Note that portions of paths 1, 2, 7, and 8 are blocked by the edges of the screens. The actual paths (bottom) in device space that a pointer could follow in CTRL mode on path 1 (from A to B), 4 (from A to C), and 7 (from B to C).

Three target locations were chosen on each screen. The targets on each screen were aligned, but separated by 100 vertical pixels. Connecting the corresponding targets resulted in nine conceptual paths (Figure 3.12), none of which are straight paths in visual space. In device space, paths 1, 2, and 3 are the symmetric equivalents of paths 7, 8, and 9, but in visual space these paths cross different size and resolution boundaries. Furthermore, paths 1, 2, 7, and 8 are not straight paths in device space because the edges of

the screen block standard cursor movement in off-screen space and require the participant to move along screen edges. All paths were evaluated in both left-to-right and right-to-left directions.

To increase the visual grouping of the targets, the experiment background showed an inactive Notepad window (of identical pixel size) on each screen at the target location. This was intended to reinforce the idea pointed out by Grudin [Grudin 2001] that users tend to switch among tasks (windows) when switching displays.

### 3.3.2.5 Hypotheses

Prior to our second experiment, we postulated the following four hypotheses:

**H1:** Pointer warping modes should outperform CTRL, due to the overall reduction of necessary mouse movement.

**H2:** Warping modes using the FM strategy should be the fastest for this task, since they will require the least amount of mouse movement.

**H3:** Pointer warping modes should not be as affected by the distance or visual-device space mismatch between screens as CTRL.

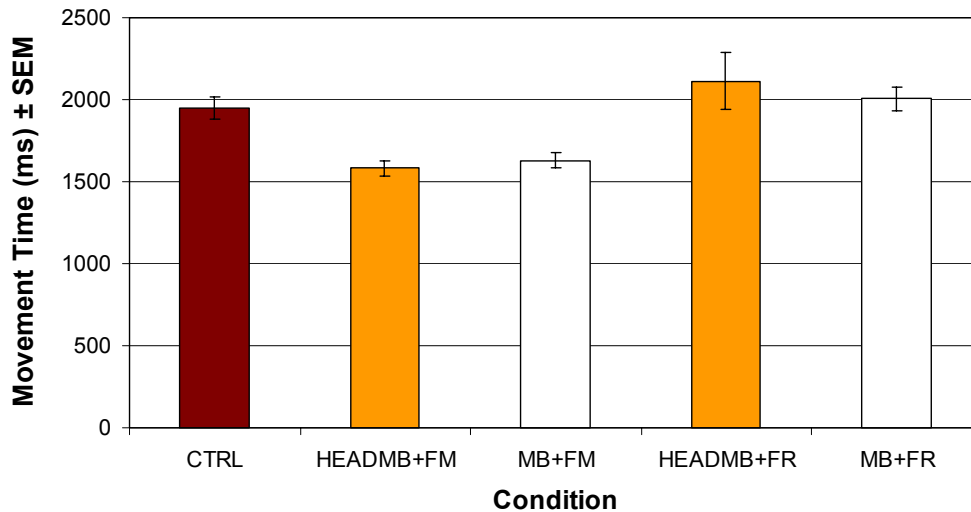
**H4:** Paths 1, 2, 7 and 8 should require longer targeting times than paths 3 and 9 in the CTRL condition, due to screen edges blocking the direct path between targets; however, this should not be the case for pointer warping.

### 3.3.2.6 Results

Movement times were first cleared by removing outliers (movement times more than two standard deviations further from the mean for each condition), which accounted for less than 1% of all trials. We performed a 5 (Condition)  $\times$  9 (Path)  $\times$  2 (Direction) repeated measures ANOVA on median movement time, with our participants as a random variable. As expected, there were significant effects for the Condition factor,  $F_{(4,36)}=11.46, p<0.001$  (Figure 3.13).

Additionally, the paired samples t-tests comparing CTRL and HEADMB+FM ( $t_{(17)}=4.758, p<0.001$ ) and CTRL and MB+FM ( $t_{(17)}=5.101, p<0.001$ ) showed that FM conditions significantly outperformed the CTRL condition. Since both FR conditions

were not found to be statistically different from CTRL in our experiment, we conclude that our H2 hypothesis was confirmed and that FM strategy was overall the fastest strategy. Our participants demonstrated an overall performance gain of 19% for pointer warping with the FM strategy compared to CTRL.



**Figure 3.13: Aggregated movement mean times (ms) for the Condition factor ( $F_{(4,36)}=11.46$ ,  $p<0.001$ ).**

H1 was not completely confirmed, since the performance of both FR conditions was not statistically different from CTRL. Although the graph in Figure 3.13 shows CTRL slightly outperforming HEADMB+FR and MB+FR, our analysis shows no statistically significant difference in performance among these three conditions (when compared via t-tests). We believe that there are two reasons why the FR conditions (HEADMB+FR and MB+FR) did not outperform CTRL. First, the previous homogeneous MDE study of pointer warping (Section 3.3.1) confirmed that the FR strategy significantly improves targeting performance only when crossing two or more monitor bezels. However, two-third of trials for the second experiment required only one bezel crossing. Second, given our particular experimental task design, the FR conditions were always required to traverse a significant distance on the target display after the warp. This might have penalized them in comparison with the FM conditions, which in our test cases always warped the cursor to a location near the target.



We made an interesting observation about the behavior of our participants in CTRL condition. During the experiment, we noticed that they appeared to adopt the following strategy for reaching the target on the next display. First, they appeared to use a fast, deliberate, but often imprecise hand movement to move the mouse pointer to the target screen. Second, once the pointer was found on the next screen, the participants appeared to perform a precise targeting task. This, in effect, is very similar behavior to that of the pointer warping strategies, where the first action warps the pointer to the target screen, followed by precise targeting afterwards. This observation is consistent with Sears and Shneiderman's analysis of touch screen pointing [Sears 1991], which separated the targeting task into a gross arm (ballistic) movement, followed by finer finger movements.

Direction contained a significant effect ( $F_{(1,9)} = 17.447$ ,  $p < 0.005$ ), with overall movement left to right (visually smaller to visually larger screens) found to be about 5% slower than right to left. This is consistent with results from the first experiment which also showed that right-to-left was faster overall. In this particular instance, with displays being arranged in order of increasing size from left to right, we speculate that visually larger displays potentially required a longer visual search time to locate the pointer after the pointer transition, which could be responsible for at least some of the increase in overall time when moving from left to right. However, our study gives us relatively limited data to make any definitive conclusions, and we believe that this phenomenon should be further explored in a separate study.

The Path factor,  $F_{(8,72)} = 7.625$ ,  $p < 0.001$ , showed significant effects, with the longest overall paths (4–6) taking the longest time (Figure 3.14). The middle paths (2, 5, 8) were the fastest paths overall for their respective screen transitions (A–B, A–C, B–C). This was probably due to the target placement requiring the least mouse movement on average. The aggregated screen transition times (Figure 3.15) show that two-bezel screen transition (A–C) required on average 10% more time than one-bezel screen transition (A–B or B–C). While this was expected due to crossing across one extra bezel, it is interesting to notice that the A–B screen transition was slightly faster than B–C, even though both consisted of transitioning identical device space distances and only a single bezel.

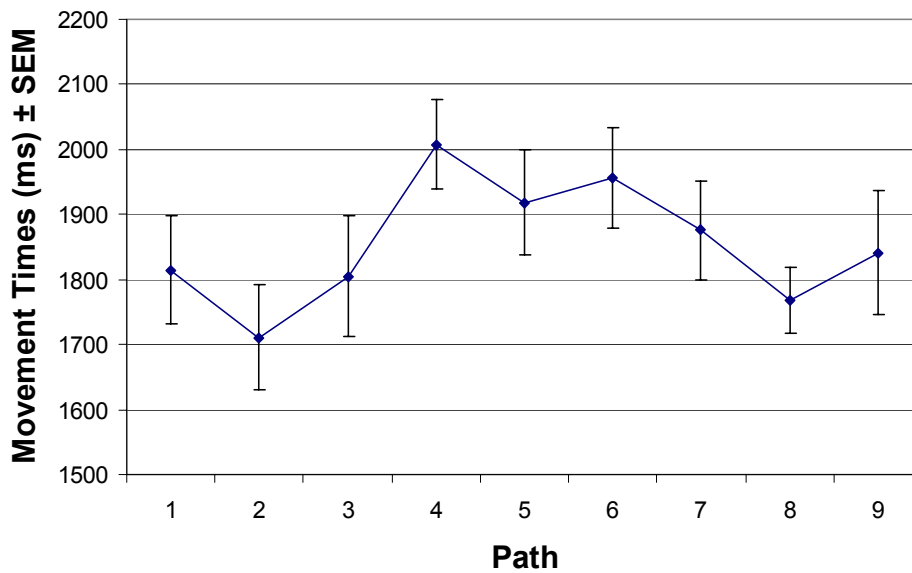


Figure 3.14: Movement mean times (ms) for the Path factor ( $F_{(8,72)}=7.625, p<0.001$ ).

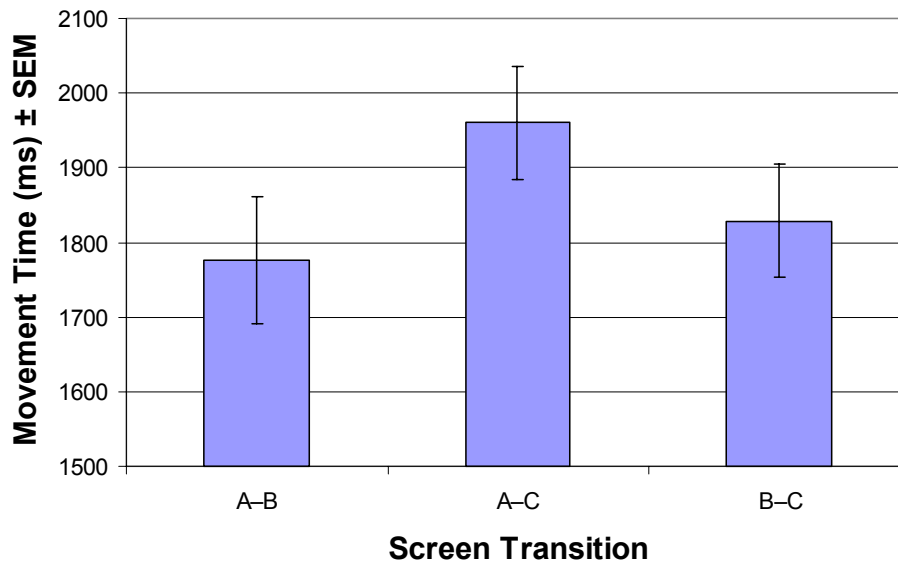
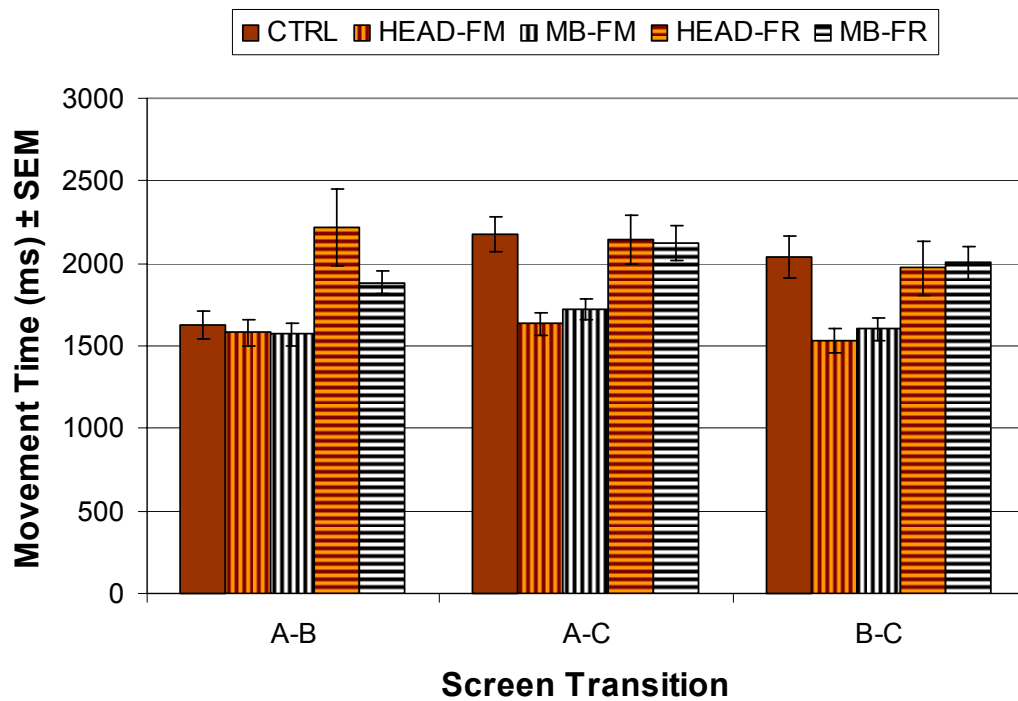


Figure 3.15: Mean screen transition times (ms), computed as aggregated path times (e.g., A–B screen transition is the aggregated times of paths 1, 2, and 3).

The explanation for this discrepancy comes from the analysis of the aggregated screen transition times (ms) for each Condition (Figure 3.16). Performance of CTRL in the A–B screen transition (paths 1–3) was significantly faster than in the A–C or B–C monitor transition. In fact, performance of CTRL seems to be inversely proportional to

the degree of mismatch between visual space and device space: higher visual–device mismatch (A–C and B–C) resulted in lower targeting performance, while lower mismatch (A–B) showed improved performance. Interestingly, the mismatch between visual space and device space (B–C) had a significantly bigger impact on CTRL performance than the mismatch in orientation and visual size alone (A–B).

In contrast, we did not observe this drastic difference between A–B and B–C in any of the pointer warping conditions, which performed almost uniformly across both the high and low mismatch transitions and across one and two bezel transitions (confirming H3). In particular, the pointer warping conditions combined with the FM strategy seemed to be the least affected when transitioning visual-device mismatched screens, requiring on average 1.6s to reach the target in our experiment. When traversing the paths of high visual–device space mismatch (B–C), pointer warping (with FM strategy) provided a performance speedup of up to 30% compared to CTRL (Figure 3.16).



**Figure 3.16: Mean screen transition times (ms) for each Condition.**

While we do note a slight tendency for paths 1, 2, 7, and 8 to require a longer targeting time than 3 and 9 in the CTRL condition, the observed differences were not consis-

tent or significant enough to confirm our last hypothesis (H4). We speculate that the difficulties from visual–device space mismatch may have overshadowed the influence of screen edges in this particular task. Overall, visually locating the pointer on the targeting screen seemed to be a fairly challenging task with or without the screen edge interruption. However, a more careful further investigation should be performed to systematically evaluate these influences.

### 3.3.2.7 *Subjective Evaluations*

The participants filled out a post-experiment questionnaire rating their experience with five techniques on a 10 point Likert scale (1 being most negative and 10 being most positive). They were asked to comment on the ease of use of each condition and performance of each condition, as well as provide their overall preference for condition and strategy. The participants rated HEADMB+FM the easiest to use ( $\mu=7.9$ ) and the fastest (8.5), followed by MB-FM (ease of use of  $\mu=6.8$  and performance of  $\mu=7.8$ ). In contrast, CTRL was rated the hardest to use ( $\mu=5.2$ ) and slowest ( $\mu=4.3$ ) overall.

All participants agreed that they would strongly prefer to use one of the pointer warping techniques over CTRL, with HEADMB+FM being the top choice for five participants, MB+FM for three participants and HEADMB+FR for two participants. Overall, seven out of ten participants preferred some form of HEADMB switching.

Several participants commented on their frustration with CTRL condition. One stated that “just using mouse is painful” and another that “it gets very difficult to move between monitors without warping if their resolution varies significantly.” Additionally, several participants commented about their lack of familiarity with the side mouse buttons on a five-button mouse, saying that they would definitely improve their performance with extended use of such mice. This is encouraging, because even though the participants were all very familiar with regular mouse use and completely unfamiliar with pointer warping and the side mouse buttons, pointer warping conditions performed at least as fast as CTRL, and outperformed CTRL with the FM strategy. Therefore, we hypothesize that extended M<sup>3</sup> use would further improve the overall pointer warping benefits.

### 3.4 Discussion

Our two studies confirmed that pointer warping offers a significant improvement over standard mouse behavior for both homogeneous and heterogeneous MDEs. The first experiment confirmed that in homogeneous multi-monitor configurations, the performance improvements from pointer warping were mostly due to gains achieved when crossing two or more bezels. However, in heterogeneous MDEs, the improvements are visible even when crossing only a single bezel. In either case, there was no significant penalty for using  $M^3$  when crossing just one bezel. Table 3.3 summarizes the conditions tested in our experiments with the winning techniques labeled as “best”.

Switch Trigger	Strategy	Experiment 1 Homogeneous MDEs	Experiment 2 Heterogeneous MDEs
CTRL	CTRL	✓	✓
MB	C	✓	
MB	FR	✓ (best)	✓
MB	FM		✓
ML	C	✓	
ML	FR	✓	
HEAD	C	✓	
HEAD	FR	✓	
HEADMB	FR		✓
HEADMB	FM		✓ (best)

**Table 3.3: Summary of two  $M^3$  experiments. Check symbol (✓) indicates comparison was tested.**

In the first experiment, the highest performance benefit for complete novice users was up to 29% when crossing three bezels. The results of the second experiment confirmed two major advantages to pointer warping compared with standard mouse behavior. First, the benefits grew in proportion to both the distance and the amount of visual-device space mismatch between monitors. Therefore, performance improvements were present even when crossing a single bezel (visible in the B–C transition in Figure 3.16).

Second, pointer warping modes performed almost uniformly across both the high and low visual-device mismatch transitions, providing a targeting speedup of up to 30% over standard mouse behavior.

In addition, the performance of pointer warping modes was largely dependent on placement strategy, and not on switching implementation. Remembering cursor locations for each screen was not considered too difficult by our study participants, which makes FM a particularly suitable strategy for pointer warping in heterogeneous environments. However, we believe that this conclusion depends largely on the task the user is performing. For our second experiment's task, where the targets are clustered on each display, FM mode appears to be preferable. For tasks in which targets are scattered across or span multiple displays, FR might be preferable. In addition, when dealing with a large tiled display (e.g., a grid of 5×5 displays), pointer warping will probably perform less well than doing a finer adjustment of pointer gain and speed control similar to HybridPointing [Forlines 2006b].

While seven out of ten of the second experiment participants preferred HEAD+MB switching, MB performed similarly. Since MB was the best and most preferred in the homogeneous case and because of its simplicity, robustness, and reliance solely on standard desktop technology, we consider MB the overall best method.

One of the important benefits of pointer warping is that it is a completely optional enhancement. Pointer warping is only invoked if the user wants to warp across screen bezels, and the behavior of the mouse pointer within any particular screen is left completely unchanged. From our own anecdotal evidence, we have noticed that we often employ a mixed approach: we use pointer warping when needing to traverse a larger distance across screens and we resort to traditional bezel crossing when trying to access nearby targets on the next screen.

However, there is an alternate benefit when bezel traversal is completely disabled: by maintaining impenetrable borders around individual displays, the user can greatly reduce the targeting time when targeting the user interface tools and widgets located on display borders (such as scrollbars, menus, and taskbars). In relation to Fitts' law, the impenetrable edge (border) effectively extends the size of the edge-adjacent target to infinity in one dimension, which allows the user to make one long ballistic movement to

access it, without the fear of overshooting the target [Walker 1990]. This basically eliminates the need for any corrective movement. By constraining the regular mouse movement to within-screen only, the user could benefit from using the screen edges to ease the targeting, and still use pointer warping to access additional monitors. This feature is implemented in the current  $M^3$  widget, but has not been evaluated yet.

We further hypothesize that it would be possible to combine pointer warping with some of the existing multi-monitor pointer techniques, such as Mouse Ether [Baudisch 2004], to achieve further benefits. Since Mouse Ether essentially attempts to reduce the display space to that of a homogeneous configuration, in which pointer warping has already proven to be of value, one would expect that a combination of Mouse Ether and pointer warping in a heterogeneous configuration would outperform either alone.

Overall, pointer warping is an easy-to-implement and completely optional enhancement that does not hinder existing mouse behavior in any way. It provides significant improvements in performance in multi-monitor configurations, as confirmed formally in the study reported here, and informally in regular use in our lab. Furthermore, our informal experience shows that extended regular use of pointer warping results in further improvements in performance.





## 4 VITA: A Hybrid MDE Implementation

In Section 1.2, we outlined the benefits of hybrid MDEs, which include: (a) the ability to present completely personalized or private experience in a shared environment, (b) the ability to present 3D views that are perspectively correct to each user, and (c) the ability of AR displays to be used as the “ether” that visually connects all other displays in the environment. Besides the AR display, other displays can offer additional benefits, such as higher resolution, wider field of view, or different input and interaction methods. We believe that combining all these displays in a hybrid MDE has the potential to enhance the overall user experience, especially when the task requires visualization of and interaction with a large amount of heterogeneous data (both 2D and 3D).

In this chapter, we describe our own implementation of a complex hybrid MDE designed for visualization and analysis of archaeological excavation data. The main contribution of this chapter is two-fold: it provides a case-study of the design, the implementation, and the evaluation of a complex collaborative hybrid MDE, and it presents the implementation of the MDE communication and interaction framework that forms the basis of our interaction implementations in Chapters 5 and 7.

We first present our motivation for design of an archaeological visualization MDE system. Then we describe our system in greater detail, focusing on the modular framework for our hybrid MDE development and various interaction capabilities available to the user. We report the results of two usability studies with archaeologists and conclude this chapter with a discussion of issues relevant to the development of hybrid MDEs.



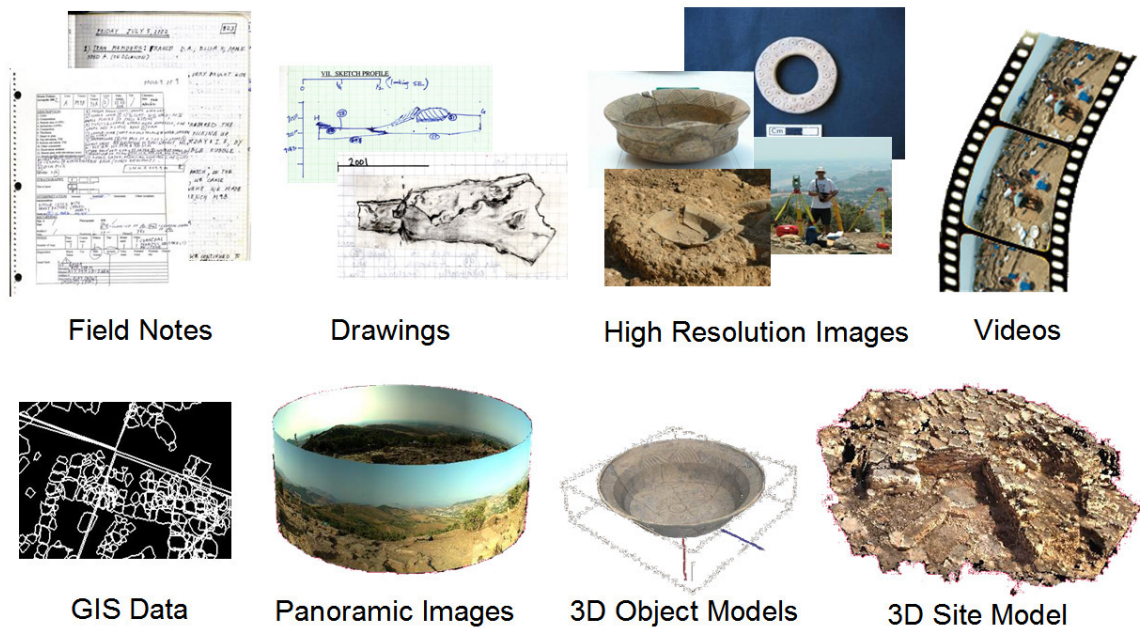
**Figure 4.1:** The author with the team of Columbia University archaeology, range-scanning, and visualization experts at the Stanford University archaeological excavations at Monte Polizzo, Sicily in July, 2003.

## 4.1 Motivation

During the summer of 2003, our research team, consisting of archaeologists, conservators, range-scanning researchers, and visualization researchers, collected a rich set of multimedia data from an ongoing archaeological excavation (Figure 4.1). The dig site was on top of Monte Polizzo in western Sicily, where a team of archaeologists from the Stanford University Archaeology Center have been excavating an Elymian acropolis built between the 6<sup>th</sup> and 4<sup>th</sup> centuries B.C. Over a span of ten days, our team used a 3D laser range scanner, a surveying system, and digital video and photo cameras to collect 2D and 3D multimedia data, including 3D point clouds and meshes of objects and the overall site, video of interesting events, panoramic images, and high-resolution photos [Allen 2004] (Figure 4.2).

The excavation of any archaeological site is by nature a destructive, and often physically unreconstructable, process. Soil layers are removed to uncover the layers beneath, small finds are taken away from the site for conservation and future analysis, and previously excavated structures are sometimes completely deconstructed to uncover even earlier structures. Detailed recording of each stage of the excavation is crucial when attempting to reconstruct the previously excavated layers. However, even though archaeologists already try to carefully record every part of an excavation, each team member

usually works on one particular section of a site, thus making it difficult to analyze, contextualize, and connect the resulting data across the entire site.



**Figure 4.2: Archaeological data collected at Monte Polizzo and incorporated into the VITA hybrid MDE. Data not typically collected at an excavation includes panoramic images, and 3D object and 3D site models.**

Archaeologists currently use various kinds of written documentation, sketches, diagrams, and photographs to document the physical state of a dig site while it is being excavated. While there are many standards and guidelines for recording the state of the dig site during an excavation (e.g., [Harris 1989, MoLAS 1994]), most of the visualization solutions rely on geographic information systems (GIS), such as the ESRI ArcGIS suite of applications. There are also some, relatively recent, attempts to use VR technology to visualize excavated sites (e.g., [Acevedo 2001, Gaitarzes 2001]). For example, Acevedo and colleagues implemented a life-size immersive visualization of the Petra Great Temple site in Jordan and used iconic representations (little pyramids) to visualize the distribution of pottery finds across the site [Acevedo 2001].

We conducted several interviews with archaeologists and they indicated that much of the typical post-excavation analysis, interpretation, report writing, and additional research they perform could benefit from the ability to visually integrate both 2D and 3D

data into a highly interactive space in which 3D terrain information is combined with sketches, images, video, and other multimedia. They envisioned using this space both for data interpretation after they have left the site, and for field planning and preparation for the next excavation season. In addition, given a large number of archaeological specialists working on any given site, they wanted to allow multiple users to simultaneously explore this data and combine their knowledge in the analysis.

Based on these interviews, we designed a collaborative hybrid MDE that extends the current archaeological practice by complementing the standard archaeological data (drawings, pictures, notes, and GIS) with a variety of newly acquired data (3D panoramic images, 3D models of objects and the site, video and ambient audio), and making it all available in one seamless collaborative hybrid MDE.

#### 4.1.1 Design Considerations

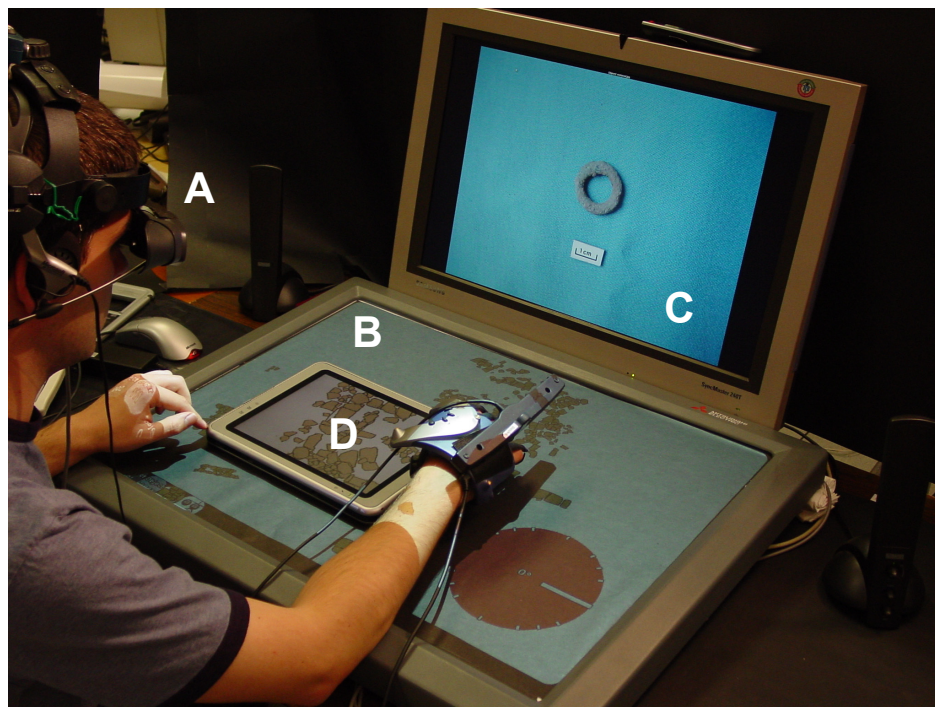
Many of our ideas and design decisions grew out of discussions with archaeologists and the focus of our development was centered on five key ideas: (a) make a collaborative visualization system, (b) provide a simpler way to explore all existing archaeological data, (c) incorporate novel 3D excavation data, (d) incorporate some widely accepted archaeological analysis paradigm (e.g., the Harris Matrix [Harris 1989]) into the environment, and (e) provide novel visualization and interaction ways to explore the excavation results. In addition, our target user was not necessarily computer knowledgeable; therefore, we tried to design our interactions to be intuitive and relatively simple.

The necessity of requiring multiple displays became apparent very early in our development, when we realized that we were dealing with a wide variety of data, each with different size, resolution and dimensionality (2D or 3D) requirements, and most of which the archaeologists wanted to be able to view simultaneously in order to easily compare and contrast them. A guiding principle was to use the most appropriate display available to present each kind of data given the data medium. For example, a textured 3D model may be best viewed in a tracked, head-worn, stereoscopic display, while high-resolution scanned field notes are best displayed on a large high-resolution moni-

tor; the video might be displayed on the large screen, while spatially varying layer information is best viewed on the tracked handheld.

## 4.2 Visual Interaction Tool for Archaeology (VITA)

The *Visual Interaction Tool for Archaeology (VITA)*\* [Benko 2004] is a highly interactive and collaborative mixed reality system for off-site visualization of an archaeological dig. The system allows multiple users to visualize and analyze excavation data in a hybrid MDE consisting of four different types of displays: tracked, see-through, head-worn display (AR); multi-user, multi-touch projected tabletop display (TABLE); high-resolution desktop display (VERTICAL); and tracked handheld display (LENS) (Figure 4.3).



**Figure 4.3: Available displays in the VITA hybrid MDE: (a) tracked head-worn (AR), (b) touch-sensitive projected tabletop (TABLE), (c) vertical high-resolution display (VERTICAL), and (d) tracked handheld (LENS).**

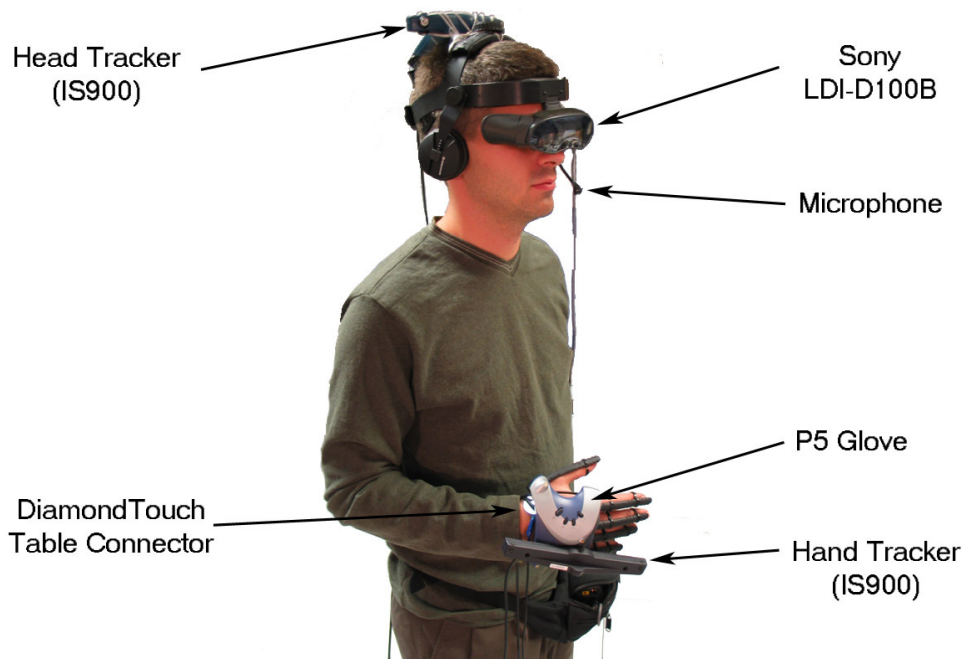
---

\* VITA was developed in collaboration with Edward Ishak. In VITA, 3D visualizations and interfaces, multimodal (speech- and gesture-based) interactions, as well as the modular framework that connects devices, displays, and users, were developed by the author. The 2D user interfaces on stationary displays were developed by Edward Ishak.



### 4.2.1 VITA Hardware Components

Each user in VITA wears a tracked, optical see-through, head-worn, color, stereo AR display (Sony LDI-D100B, 800×600 resolution). In addition to the AR display, each user also wears a tracked finger-bend-sensing glove (EssentialReality P5 glove) to facilitate 3D gesture recognition, and a microphone to allow for speech commands. We track the position and orientation of the user’s head and hand with an overhead six-degree-of-freedom tracking infrastructure (InterSense IS900). Figure 4.4 shows the components that the user wears while interacting in VITA.



**Figure 4.4: VITA user’s wearable components.**

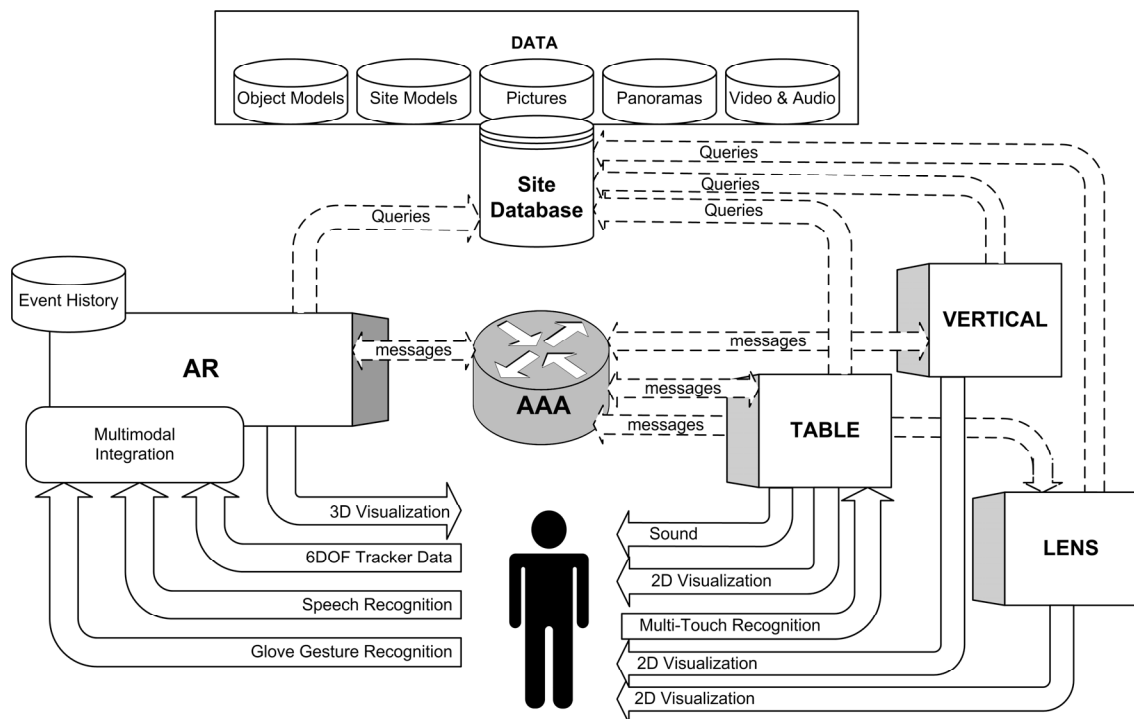
The interactive centerpiece of our environment is the MERL DiamondTouch table (TABLE display) (Figure 4.3b). The DiamondTouch touch-sensitive tabletop surface is capable of detecting multiple touches and disambiguating among multiple users, which allows for simultaneous interaction on the tabletop by multiple users. An overhead projector (InFocus Proxima x350, 1024×768 resolution) is used to display an interactive graphical user interface on the tabletop. In order to be detected by the DiamondTouch table, each user needs to be electrostatically coupled with the table, which can be achieved by either requiring the user to sit on a conductive pad, or by integrating a

small conductive DiamondTouch connector directly into the base of the user's glove (see Figure 4.4).

The VERTICAL display (Figure 4.3c) is a high-resolution 24" LCD monitor (Samsung 240T, 1920×1200 resolution) and the LENS display is a TabletPC (HP TC1100, 1024×768 resolution) (LENS). The LENS display is tracked by the DiamondTouch surface when the user is in contact with it, and it is used to provide higher resolution imagery when needed within the context of the larger projected TABLE display (Figure 4.3d).

## 4.2.2 VITA Modular Framework

To facilitate easy integration of a variety of different displays and interaction devices, we have implemented VITA using a modular framework. Each VITA display is implemented as an independent module using Java and Java3D. In addition to handling the presentation of data on the associated display, each module also handles the input and output devices that are associated with this display (for example, the TABLE module handles multi-user multi-touch sensing on the tabletop's surface).



**Figure 4.5: VITA modular architecture diagram.**

In our framework, we completely separate the data access from the communication of interaction events. Each module accesses archaeological data directly from a central database via SQL queries, while all communication between modules is conducted via message passing through a publish/subscribe blackboard system. The complete VITA architecture diagram is shown in Figure 4.5. This modular framework has been inspired by our previous research on multimodal integration for AR/VR environments [Kaiser 2003, Olwal 2003a] and serves as the base for most of the interaction explorations presented in this dissertation.

```
...
//Subscribe to messages
registerInformHandler("objectSelect(UserID, ObjID, State)",
    handler);
registerInformHandler("tableRotateObject(Username, ObjID,
    DeltaAngle)", handler);
registerInformHandler("tableRotateWIM(Username, DeltaAngle)",
    handler);
registerInformHandler("tableTranslateObject(Username, ObjID,
    Xoffset, Yoffset)", handler);
registerInformHandler("tableTranslatePinnedObject(Username,
    ObjID, Xoffset, Yoffset)", handler);
registerInformHandler("tableScaleObject(Username, ObjID, Scale)",
    handler);
...
```

**Figure 4.6: A portion of the code that illustrates how modules can subscribe to particular messages of interest.**

We use the Adaptive Agent Architecture (AAA) [Kumar 2000] to support publish/subscribe messaging between modules. Upon startup, each module informs the central AAA facilitator of its capabilities by registering a list of message types that the module can receive (Figure 4.6). Each module can then send (publish) messages directly to the facilitator, and the facilitator will act as a message router, allowing for one-to-one and one-to-many connections, and filtering out messages that are not requested by any module. Each message consists of a message type name, timestamp, as well as additional data that varies depending on a message. Usually, the modules will also indicate that some messages pertain to a particular user or particular module (display), in which case the message body will include a target user ID or a target module ID (for



example, the message *tableScaleObject* in Figure 4.6 requires the user name to be specified).

This publish/subscribe communication allows for easy distribution and implementation of the hybrid MDE, since the modules can be distributed over several machines, or run from the same device, and they do not need to be aware of the availability of other modules. Only the address of the facilitator and the location of the database need to be supplied to the modules ahead of time. In addition, a new user or a new display (i.e., new modules) can be introduced to the system without having to modify any existing modules. While the full functionality of VITA is achieved with all modules present, each one can act as an independent, self-contained unit. This is particularly useful if a hardware modification is desired. For example, we sometimes replicated the contents of our VERTICAL display to a large back-projected wall-size display for easier viewing at a distance. Adding this extra display to VITA only required that we start a separate VERTICAL module on a device connected to the wall-sized display.

To reduce the size of individual messages, all modules have direct access to the same database containing all available archaeological materials from the dig site, and only the most relevant information is communicated. For example, if an AR user selects an object, only the distinct object ID will be broadcast, allowing all other modules to retrieve all necessary information about that object directly from the database. Thus, the number of simultaneous users depends only on the number of available AR modules and is physically capped by the throughput of the facilitator and available hardware. Furthermore, users do not need to be collocated, since the modularity of our approach allows for remote collaboration; however, such remote collaboration scenarios were not explored in our work on VITA.

For each user, there is a unique *interaction integrator* component implemented as a part of the user's AR display module. The interaction integrator aggregates all interaction messages pertaining to this particular user and is therefore capable of integrating multiple recognizer streams into more complex interactions. We use this integrator component to facilitate multimodal interactions in VITA (Section 4.3.2.1) and as a base for the implementation of our additional freehand interactions (Chapters 5 and 7).

### 4.3 Interacting in VITA

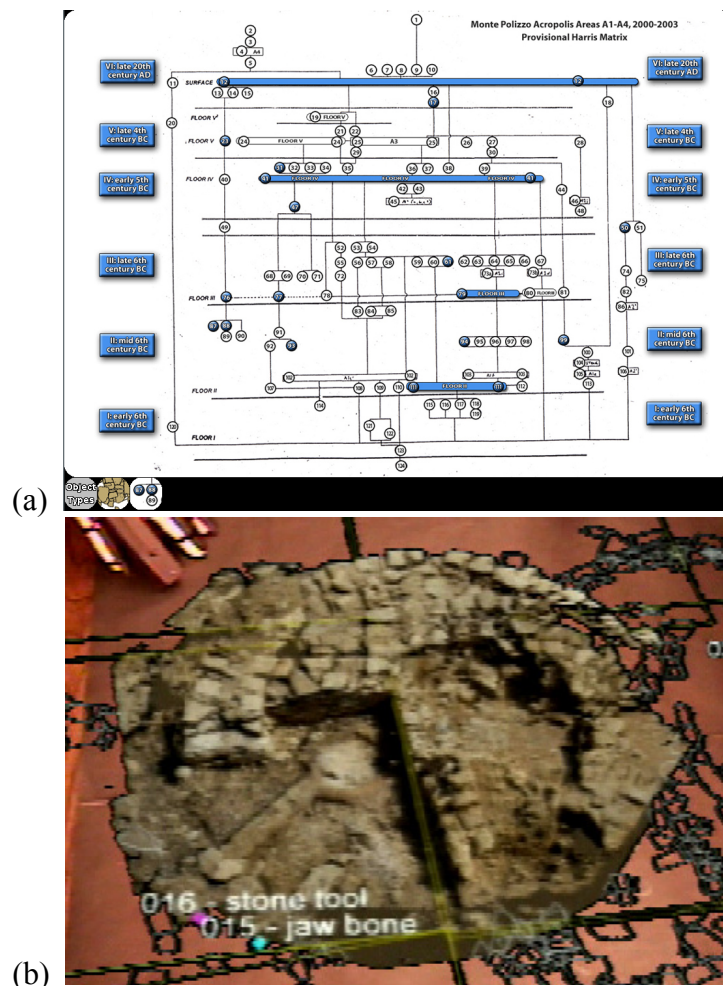
VITA supports two exploration modes: tabletop mode and life-size mode. In both modes, the base of our 3D visualization is a 3D terrain site model of Monte Polizzo, created by Peter Allen, Alejandro Troccoli and Benjamin Smith, from the Columbia University Robotics Laboratory [Allen 2004a, Allen 2004b]. We focused on facilitating simple and direct freehand interactions within VITA, such as touch-based interactions on the TABLE display, or gestural and multimodal interactions in the 3D environment. Wherever possible, we incorporated visual or audio feedback to enhance the user's comprehension of interactions.

#### 4.3.1 Tabletop Mode

The tabletop exploration mode is the primary interaction mode of VITA and it is also the mode in which users can fully utilize the benefits of the hybrid MDE (shown in Figure 4.7). This mode presents multiple users with a small-scale *world-in-miniature* (WIM) model [Stoakley 1995] of the acropolis at Monte Polizzo, fixed on the physical tabletop. Next to the WIM is the TABLE display onto which we project the 2D graphical user interface.



**Figure 4.7:** Two users simultaneously collaborate in VITA. While one user is inspecting the 3D virtual model of the ceramic vessel above the table, comparing it with the high-resolution image on the screen, the second user is looking at the 3D miniature terrain model next to the interactive tabletop.



**Figure 4.8: Temporal and spatial representations of data in the tabletop mode of VITA: (a) a Harris Matrix visualization displayed on the TABLE display (designed and implemented by Edward Ishak) represents temporal connections between layers, and (b) the 3D world-in-miniature site model places excavated finds in a spatial context.**

Edward Ishak designed the TABLE interface, through which users can access the Monte Polizzo terrain map, browse the database of objects from the excavation, as well as examine relationships between various excavated layers in an augmented *Harris Matrix* [Harris 1989]. The integration of the Harris Matrix was recommended by our archaeological experts as it is a widely used archaeological diagram tool that shows temporal relationships of all archaeological layers from the excavation site (shown in Figure 4.8a). For example, an archaeological layer could be 10 cm of soil representing the ground level of a particular time period (e.g., mid 4<sup>th</sup> century B.C.). By selecting

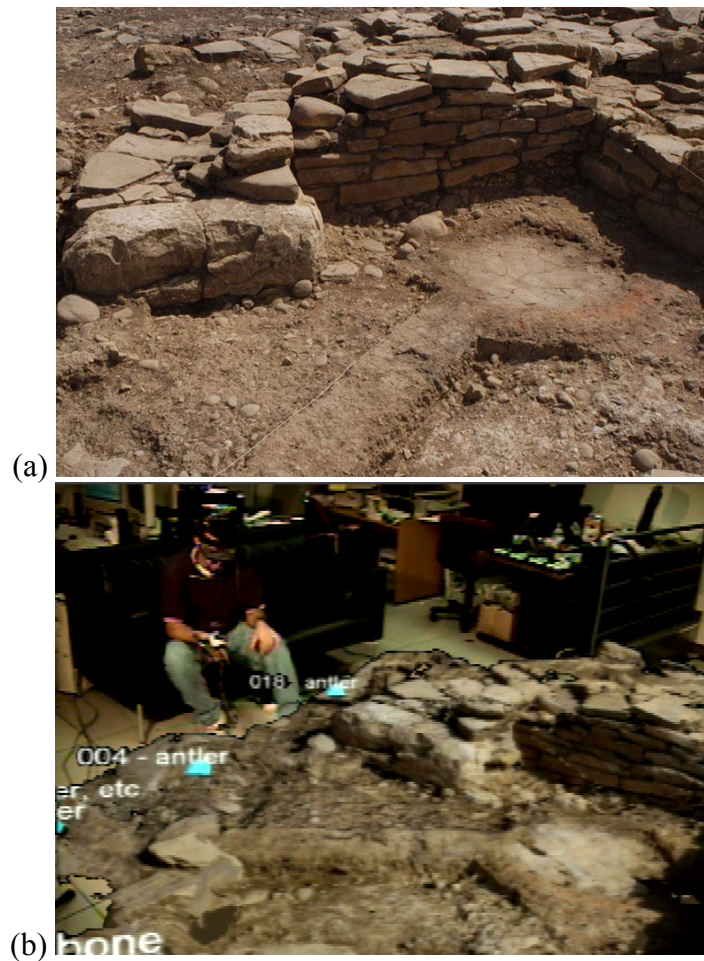
portions of the Harris Matrix, users can choose to visualize various data relevant to archaeological layers that they are currently exploring.

While the primary way to browse the site database is through the TABLE interface, all other VITA modules automatically participate in the data visualization. For example, the currently selected layers and objects appear in the scaled 3D model, making apparent the spatial relationships between them. Being able to use multiple displays to simultaneously examine temporal and spatial relationships between excavated objects was mentioned by archaeologists, during informal evaluations of the system, as one the most significant benefits of the VITA system (Section 4.4). The 3D capabilities of AR displays can also be used to view 3D models of individual objects when such models are available (e.g., a Elymian pot in Figure 4.7).

In addition, the VERTICAL module is available for browsing a variety of accompanying documents (e.g., handwritten field notes, photographs, and videos) that benefit from the high-resolution of the display, avoiding the need for repeated scaling and resizing. Furthermore, the LENS module is used as a “magic lens” [Bier 1993] to view subterranean data or additional detail within a context of a map shown on the TABLE display.

### 4.3.2 Life-Size Mode

In addition to the tabletop interactions, VITA makes it possible to view the data in life-size mode. This mode is designed to recreate the experience of being on top of Monte Polizzo. The textured 3D model of the acropolis is displayed in the user’s head-worn display, covering an area of roughly 10×10 meters, as shown in Figure 4.9b. Users wearing AR displays can walk around the site at its actual size and the system can display a surrounding panorama to provide a completely immersive VR experience. Using a tracked glove and multimodal interactions, a user can examine archaeological finds that are situated in the model at the exact locations of their discovery, each labeled with its name and description.



**Figure 4.9: Real vs. virtual: (a) Real image of a portion of the excavated structure taken at the site. (b) User exploring the same section of the site in life-size mode of VITA.**

The finds are represented in the model as colored diamonds (similar to the pyramids of [Acevedo 2001]) where color is based on the object type (for example, ceramic objects are red and wooden objects are blue). In an earlier version of the VITA system, we positioned photorealistic 3D models of these objects in the environment, but in most cases, due to their color similarity with the surrounding soil, and the drastically varying sizes of the objects, life-sized photorealistic representations were very difficult to visualize in situ.

#### 4.3.2.1 3D Multimodal Interaction

In life-size mode exploration, the users are often not sufficiently close to the TABLE display to use its 2D user interface to navigate the world. Rather than returning to the tabletop for each adjustment, we have implemented a set of multimodal interactions (combining speech and gestures) to make it possible to interact while immersed in the VITA life-size environment. In line with archaeologists' efforts to preserve the excavation site as close as possible to the state it was during excavation, modifying and manipulating the object or site model in VITA environment is not desired or allowed. Therefore, our multimodal interactions are primarily focused on selecting and inspecting objects, terrain, and multimedia data.

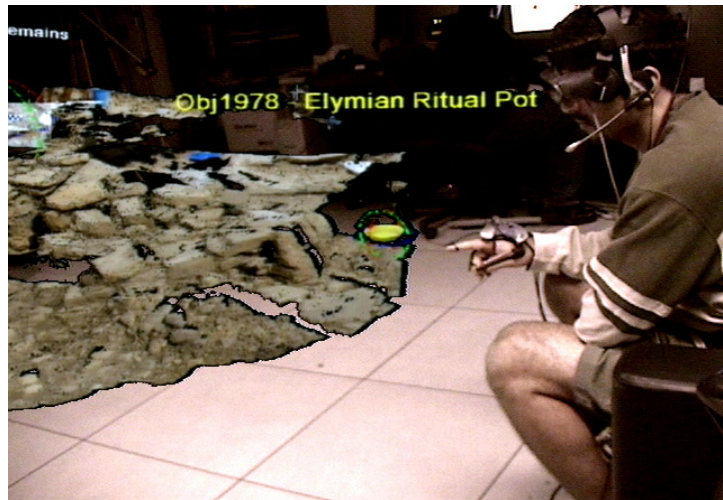
We provide several freehand interactions to accomplish 3D object selection in life-size mode. The classic way that the user can select an object of interest is to move towards an object and “grab it” by making a grab gesture pose (Figure 4.10) while intersecting the object of interest [Zimmerman 1987]. The object then becomes attached to the hand and the user can now inspect it further by adjusting its tracked hand. Releasing the object would make it automatically return to its original place.

However, the user is frequently not able to reach the object of interest due to the limited tracking area or the large distance, in which case one can point at it and issue a verbal selection command to select it remotely (Figure 4.11). This multimodal approach is inspired by our previous work on multimodal interactions in AR/VR environments [Kaiser 2003, Olwal 2003a].



**Figure 4.10: 3D poses recognized by the VITA system.**





**Figure 4.11:** The user selects an object of interest (an Elymian Ritual Pot) by pointing at it and saying “select that object.”

To facilitate the multimodal interaction, each user’s AR module contains the interaction integrator component (described in Section 4.2.2) that integrates all messages pertaining to that user. In VITA, we consider the input modalities of 3D gestures and speech. Our 3D hand gesture tracking is done by fusing a 6DOF tracker (InterSense IS900) and the EssentialReality P5 glove gesture recognizer based on five finger-bending sensors. Rather than focusing on a large vocabulary of hand gestures, the P5 glove gesture recognizer currently recognizes a small, but unambiguous set of gestures based on hand pose states such as “point”, “grab”, and “idle” (Figure 4.10); however, different hand poses (such as and “thumbs up”) can be easily introduced via a simple configuration file interface. We use an off-the-shelf commercial speech recognition system (IBM ViaVoice v.10) for grammar-driven speech interaction.

In addition to multimodal selection, simple speech commands can be used to customize and adjust different visualization parameters and issue various system commands. For example, the user can request to see all ceramic objects or to show the panoramic image.

## **4.4 Usability Evaluations of VITA**

We performed two informal evaluations with archaeologists to collect feedback and evaluate the effectiveness of the VITA hybrid MDE as a tool in archaeology curriculum. We focused on two scenarios: site exploration scenario and teacher-student scenario. These two scenarios were intended to be pilot case-studies for the follow-up formal evaluation of the VITA system. We originally planned to have VITA formally evaluated as part of a semester-long archaeology class. However, the archaeologist in charge of teaching this class left the project and we were therefore unable to perform this experiment.

The idea for the long term evaluation was motivated by the real problem that the archaeologists have with preparing and educating students about the ongoing excavation. Excavation typically last several years and, due to staffing reasons, are usually performed only during one part of the year (for example, during summer months). Numerous archaeology students participate in the excavations and are usually the main workforce at the site; however, since the majority of excavation publications and materials get compiled at the end of the excavation, lead archaeologists need to educate and prepare a new class of students each year remotely, without any of the original excavated finds, and with little written material. We were hoping to evaluate VITA as an in-class analysis tool, which could be continuously updated throughout the excavation, and could therefore be a valuable teaching tool to educate new archaeology students about the site topography, recently-excavated interesting finds, and the appearance of the site during previous excavation seasons. Ultimately, we wanted the students to use VITA in their projects for site analysis and interpretations. Unfortunately, this long term evaluation was not possible and we now briefly describe the two informal pilot studies.

### **4.4.1 Site Exploration Scenario**

We invited three expert archaeologists and three archaeology graduate students to try out the system informally. The archaeologists were extremely familiar with the Monte Polizzo site, having been active participants in previous years' excavations,



while the students, although familiar with the general practice of archaeological excavation, were completely new to this particular dig site. Each user was given an hour-long introduction to the VITA environment, in which they learned about its capabilities and constraints. They practiced exploring the site database on the TABLE interface, exploring the site in life-size mode, selecting objects via multimodal interaction, and using the LENS interface. During that hour, each user was accompanied by one of the system developers, who guided them through different aspects of the system. The current two-user tracking limitation in VITA forced us to constrain collaboration to two people at a time.

#### 4.4.1.1 Results

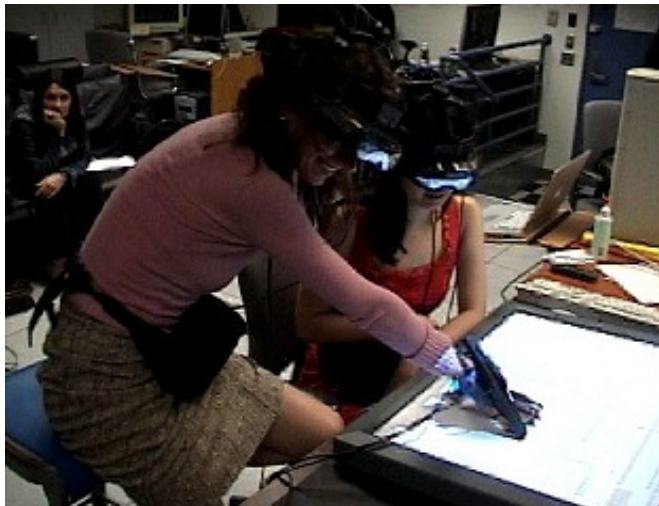
Initial user reactions were very positive. They included comments regarding the detailed 3D visualization of the excavated structure, stating it was both beneficial and complementary to the 2D notes, drawings, and diagrams that are traditionally available post-excavation. Visualizing the life-sized model was even more important than we thought, as the archaeologists considered critical to see the details of particular stones or the intersection of walls. In the same time, they appreciated the contextual overview given by the world-in-miniature model which made it very simple to visualize spatial relationships between objects across the site.

All users pointed out that they thought the major benefit of our system was the ability to use multiple displays to compare the temporal relationships of excavated objects (in the Harris Matrix) with the equivalent 3D spatial relationships (visualized in the WIM in the AR display), and while in the same time having all the additional contextual information for those objects displayed on the VERTICAL display. VITA facilitates exploring such relationship contexts in seconds, when normally, this task can take hours to complete.

Criticism of the system focused mainly on missing features that could not be implemented because of the unavailability of necessary data. In particular, users wished to see how layers were spatially connected over time through a time-lapse visualization. This problem could have been resolved during the data acquisition phase several

months earlier on Monte Polizzo, had we only been able to acquire more frequent scans over a much longer period of time.

Several comments also focused on collaboration issues. Collaborating users were frustrated when one user would not know where another was looking in the augmented environment. A major contributing factor was the inability to see the other user's eyes because of occlusion by the dark filters built into the head-worn displays; however the use of a virtual laser pointer could possibly have aided in lessening this frustration. Also, as expected, users complained about the discomfort of wearing the interaction devices for an extended period of time, saying it was too heavy and had too many wires, which often became tangled when walking around the virtual dig site.



**Figure 4.12: The VITA system being evaluated in our teacher-student scenario. The teacher (Trinity Jackman of the Stanford Archaeology Center) is Assistant Director of the Monte Polizzo excavation. She is working with a Columbia University archaeology student.**

#### 4.4.2 Teacher-Student Scenario

In addition to the general usage of VITA as a site visualization tool, we informally investigated how effective the system might be as a learning tool in a teacher-student scenario. We asked Trinity Jackman, Assistant Director for the excavations at Monte Polizzo, to use VITA as a tool to teach students a specific aspect of the excavation (Figure 4.12). The two students were recruited from the Columbia University Depart-

ment of Art History and Archaeology, and were not familiar with the site. Both the archaeologist and students were given a brief introduction to the capabilities of the VITA system, but the actual session was performed without any assistance from the system developers. The archaeologist was given a half hour to teach each student about the discoveries made during the last excavation season within a particular area of the acropolis site. The archaeologist conducted two sessions, each with a different student and we taped both sessions for further analysis.

#### **4.4.2.1 Results**

One student commented that she quickly became familiar with the excavation process at Monte Polizzo, even though she had not previously been introduced to the site. In contrast to her previous experiences, where she found it difficult to visualize and understand a dig site before actually arriving at the location, she felt that VITA gave her a fairly good understanding of the overall layout of the site and its most significant features. Both students expressed their desire to see a system like VITA used in their courses to get a full understanding of the 3D and multimedia nature of dig sites. In addition, they felt that a substantial amount of data classification and analysis time was saved when using VITA, eliminating hours of sorting through various reports and books when attempting to obtain a deep understanding of a particular site. They stated that VITA proved to be especially good at providing such contextual information.

All of the participants commented on the benefits of having several displays to simultaneously view different data which made it very easy to understand the relationships between excavated objects and instantaneously get all the contextual information they wanted. They also commented that exploring data in such a way naturally lends itself to various spontaneous “compare and contrast” discussion which they found enjoyable and useful.

The archaeologist stated that in current archaeological practice, little collocated collaboration (beyond regular meeting-style discussions) happens in analyzing excavation data, primarily because of the lack of adequate visualization and collaboration tools. However, she felt that in VITA, easy availability of data and rich visual representation

of both spatial and temporal characteristics made the environment very conducive to collaborative exploration and discussion. She also mentioned that one of the biggest problems they have in preparation for the next excavation season is how to teach the new students who will be working on the excavation, but have not been there before. Given that a typical excavation lasts several years, and that reports are not usually published until the end, very little formal material exists for the archaeologists to teach the new excavating class. All our participants agreed that VITA would be a very beneficial tool in such classrooms.

## **4.5 Discussion**

This chapter presented VITA, a prototype hybrid MDE, which allows archaeologists to collaboratively discuss and analyze a digital reconstruction of an excavation site. VITA includes a collaborative table surface, augmented with a world in miniature model of the environment and high-resolution screens, to allow for simultaneous viewing of available 2D and 3D multimedia. VITA also makes it possible for multiple users to walk around the virtual site, and explore it using multimodal interaction to inquire about interesting finds in situ.

Taking into consideration the limitations of the current cumbersome hardware, the preliminary user feedback has been overwhelmingly positive and very enthusiastic. While we were originally planning to further evaluate VITA as a tool in an archaeology classroom, those plans, unfortunately, did not materialize because of the class scheduling and staffing problems. While it was not possible to conduct such an experiment during this dissertation work, completing it in the future would provide a lot of valuable insight into the effectiveness of the hybrid MDE solutions.

During the design, development, and evaluation of VITA, we came across several issues that are relevant for design of hybrid MDEs. Those issues can be grouped into two groups: those dealing with collaboration problems and those dealing with interaction problems.

### 4.5.1 Hybrid MDE Collaboration Issues

While the rest of this dissertation focuses on solving some of the interaction issues encountered in VITA, it is important to discuss the collaboration issues we encountered while using this novel hybrid MDE.

VITA supports multiple users simultaneously interacting in the hybrid MDE; therefore, numerous situations arise where resource allocation becomes a problem. For example, when multiple users are simultaneously inspecting different objects, one might inadvertently replace some of the contextual documentation requested by the other user. We have tried to design VITA to foster social resolution of such conflicts, and we have provided subtle cues to identify what documents, objects, and actions “belong” to each user. For example, we outline each document and action with a unique color to identify them as belonging to a particular user. Even so, what is important contextual information for one user might be visual clutter that disturbs the other.

MacIntyre and Feiner suggested that hybrid MDEs adopt a more general approach to management of data (similar to the idea of operating system’s window manager), which they called *environment management* [MacIntyre 1996]. Some of that work has already been addressed by Bell in his work on *space management* [Bell 2000] and more general *view management* [Bell 2001].

We believe that the visual clutter problem could be remedied by providing completely personalized or private views in AR displays. In this approach, only the information relevant to both users should be displayed in on the shared displays, and the personal information should be overlaid in the head-worn display. Bell explored this approach in a two user hybrid MDE calendar application in his dissertation [Bell 2005].

While we did not explicitly explore personalized views, VITA has all the functionality needed to implement them and we believe that users would benefit from the ability to personalize their experience based on the level of expertise or interest. In addition to a reduction of the visual clutter, this would allow two users to visualize the same data set in completely different contexts. We discuss the implications of interacting with private data in Section 5.2.5.

### 4.5.2 Hybrid MDE Interaction Issues

A lot of the initial user feedback highlighted the pragmatic issues with interacting in VITA, such as difficulties with carrying equipment and many cables, the lack of direct eye contact due to head-worn displays, and the known problems of overloading the human speech channel with both interpersonal communication and system control.

We also noted several issues that limited the usefulness of freehand interactions in VITA, and the remainder of this dissertation addresses those issues. First, the vocabulary of freehand interaction techniques available for interacting across displays and devices in a hybrid environment, such as VITA, is very limited. In part, that is due to the hard problem of integrating different interaction devices and providing seamless interaction space across different displays. Particularly lacking are transitional interactions, which allow the user to seamlessly select and move an object from one of the stationary displays to their AR display and vice-versa. In hybrid MDE, such transitions require not only a relocation of the existing data, but often a transformation of this data between 2D and 3D. We address this issue in Chapter 5, where we describe a set of gestural interactions that facilitates data transitions, as well as data association across different displays, devices, and dimensions.

In addition to the lack of transitions, we have noticed that all freehand interactions throughout our environment suffered from a lack of precision. In particular, selecting a small target with only fingers or a whole hand was a difficult task both with touch interactions on the 2D tabletop interface and with gesture and multimodal interactions in the 3D environment. When analyzing videos recorded during VITA evaluation sessions, we noticed that our users mostly resorted to selecting objects through our tabletop interface, which was carefully designed to feature large touch targets. Using multimodal integration eliminates some of the 3D imprecision problems, but we still had to drastically increase the targets to ensure that they are easily selectable. The lack of precision was even more apparent when interacting in the WIM, where, due to the exaggerated scale, none of the embedded targets were easily selectable. However, multiple participants requested that they would like to be able to easily select various objects throughout the environment, and particularly within the 3D WIM model.

In Chapter 6, we examine the problem of precise 2D touch-based selections. Many of the current graphical user interfaces incorporate targets much smaller than the fingertip making it difficult to incorporate standard interfaces in an MDE that is based on freehand interactions. To address this, we provide several multi-touch techniques that improve the precision of touch-based selections.

In Chapter 7, we examine how one can exploit the benefits of interacting across multiple displays to address the 3D selection precision problems. We designed and evaluated a fast multi-touch gestural technique that drastically improves the selection of small scale 3D targets (such as those featured in the WIM model) without significantly impacting the selection speed.





## 5 Cross-Dimensional Gestural Interaction in Hybrid MDEs

This chapter extends our interaction work in the VITA hybrid MDE (Chapter 4), by presenting freehand interactions that facilitate two fundamental MDE tasks: transition and association across displays. We designed and implemented a set of novel freehand interactions that use synchronized gestures between the 2D touch-sensitive surface and the 3D tracked glove to provide the connection between the 2D and the 3D environments. In addition to transitions and associations, we explored ways to provide gestural modifiers to those gestures to facilitate interaction with private or personalized data. We conclude this chapter by discussing the results from an informal usability evaluation of our gestural techniques.

### 5.1 Motivation

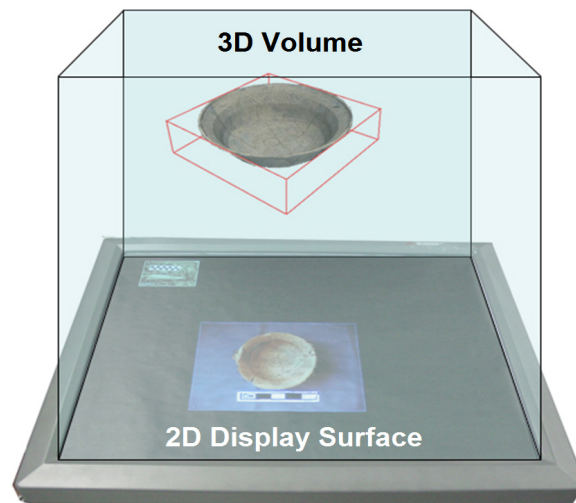
Through review of the existing literature on MDE interactions (Section 2.3), we identified that most MDE configurations support two fundamental MDE tasks: transition and association.

*Transition:* The user specifies one or more objects on one display and transfers them over to another display. In most MDEs, this is performed as a two-step action, copy and paste (e.g., Pick-N-Drop [Rekimoto 1997]) or drag-and-drop with Stitching [Hinckley 2004]).

*Association:* The user associates (connects) several displays or a combination of objects across displays into a meaningful set. Oftentimes, this is performed to extend the working area (e.g., Connectables [Tandler 2001]), synchronize views (e.g., Multi-User

Google Earth [Forlines 2006a]), provide contextual connection (e.g., leader lines in EMMIE [Butz 1999]) or provide control of the remote screen through a local screen (e.g., WinCuts [Tan 2004]).

In VITA, as explained so far, these tasks are all handled implicitly; that is, the system only allows for preconfigured transitions or associations. For example, when the user selects a thumbnail image of a certain archaeological artifact on the touch-sensitive surface, the high-resolution image automatically appears on the large vertical display in the environment. This requires an automatic method for determining the placement of data in the environment and leaves the user with limited interaction possibilities.



**Figure 5.1: The dimensionality border between a 2D surface display and a 3D volume (imaged through head-worn displays). The 2D–3D dimensionality border in our hybrid MDE can be defined as a space where the user’s tracked hands operate in 3D and simultaneously touch the 2D surface.**

## 5.2 Cross-Dimensional Gestures

We extended the VITA interaction vocabulary with a set of freehand gestural interactions, called *Cross-Dimensional Gestures* [Benko 2005b], that facilitate explicit transition and association of data across displays in a hybrid MDE. These techniques are designed specifically for hybrid MDEs that use 2D touch-sensitive displays and the 3D space around them (via 3D head-worn see-through displays). Since our gestures act

across this dimensionality border between the 2D and 3D displays (Figure 5.1), we use the term “cross-dimensional” to describe them.

The main contribution of these interactions is that they synchronize the recognition of gestures between the 2D touch-screen display and the 3D tracked glove, thus providing an explicit connection between the 2D and 3D environment. Our gestures use the multimodal interaction integrator (described in Section 4.3) to synchronize the events between the touch-based gesture recognizer\* and the glove-based gesture recognizer. We have built in part on our previous work in multimodal interaction for augmented and virtual immersive environments [Kaiser 2003, Olwal 2003a], in which our 3D gesture recognizer was originally developed. Both 2D and 3D recognizers are heuristic-based and implemented as finite state automata.

The use of synchronized actions was suggested previously [Holmquist 2001, Hinckley 2003, Hinckley 2004] as a powerful and easy way to automatically establish connection between devices and displays without requiring any additional steps. We extend the previous work, by synchronizing two different gestural modalities (2D and 3D) to provide the connection.

Although our interactions differ in the tasks they perform, they all share several high-level features: (a) they begin by selecting a source object, (b) they use a hand gesture to perform the action, (c) the end of the gesture determines a destination or a target, and (d) each gesture has an intuitive counterpart (an intuitive way to transition back to the original state). All of our gestures are inspired by informal observations of how people interact with planar objects (e.g., a piece of paper or a compact disk) and non-planar objects (e.g., a tennis ball or a pair of scissors) on a tabletop surface.











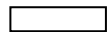

### 5.2.1 Integration of Gestural Primitives

Before we present our Cross-Dimensional Gestures, we need to describe the gesture primitives that our system recognizes independently in 2D and 3D. These gesture primitives (Table 5.1) consist of four different hand postures and are used as building blocks for more complex gestural combinations.

---

\* The 2D touch-based gesture recognizer was designed and developed by Edward Ishak.

The 2D recognizer reports the following four distinct touch states: *one-finger* (small touch area), *tap* (medium touch area, equivalent to touching the surface with a closed fist), *flat hand* (large touch area), and *chop* (elongated touch area). In 3D, we support the following hand poses (described in Section 4.3.2.1): *point* (index finger extended), *grab* (closed fist), and *idle* (all fingers unbent). In addition to pure gestural poses, we consider motion (gesture dynamics), for example a *swoop* gesture (the transition between a flat hand and a tap). While these gestures are recognized independently, both in 2D and in 3D, it is important to look at them in combination in a hybrid environment, rather than in isolation. This is especially true for hand gestures that have different meanings, depending on whether they are interpreted in 2D or in 3D (e.g., idle hand pose in 3D can be interpreted as flat hand or chop in 2D depending on its orientation).

Hand Posture	2D Touch Recognition	2D Interpretation	3D Pose Recognition	3D Interpretation
		One-finger		Point
		Tap		Grab
		Flat hand		Idle
		Chop		Idle

**Table 5.1: A set of basic hand posture primitives and their interpretations in 2D and 3D. Our 2D touch recognition is based on the largest axis-aligned rectangle that bounds all user’s touches. Our 3D pose recognition is based on finger-bend-sensing. Finger notation is similar to [Baudel 1993]:  $\blacksquare$  (finger bent),  $\!|$  (finger unbent), and  $\text{—}$  (thumb unbent).**

In our system, the 2D and 3D gestures are considered in combination (Table 5.2) and integrated together using an additional finite state automaton. By synchronizing the gestures on both sides of the dimensionality border, we can reuse the common hand poses that are normally used throughout the environment (e.g., point or grab) and give them a special meaning when interacting on the display border. In addition, by requiring

the user to touch the surface to trigger the cross-dimensional interaction, we eliminate many potential accidental invocations or ambiguities that would arise by relying on 3D gestures alone. We provide detailed descriptions of our interactions in Sections 5.2.3–5.2.5.

Cross-Dimensional Gesture	Task	Synchronized Recognition	
		2D	3D
Pull	Transition 2D → 3D	Flat hand → Tap	Idle → Grab
Push	Transition 3D → 2D	Flat hand	Idle
Pin	Associate 3D object with 2D surface	One-finger	Point
Connect	Connect 2D and 3D objects	Tap	Grab
Privacy Pull	Transition 2D → 3D private	Chop + Flat hand → Tap	Idle → Grab
Privacy Push	Transition 3D → 2D private	Chop + Flat hand	Idle

**Table 5.2: The vocabulary of Cross-Dimensional Gestures. Note: The arrow (→) indicates a transition between two hand poses.**

## 5.2.2 Design Considerations

There are three design principles we used in our implementation of Cross-Dimensional Gestures that can be useful when facilitating natural interactions in hybrid MDEs: gesture simplicity, gesture reuse, and context preservation.

*Gesture Simplicity.* Our primary considerations for selecting the gesture primitives were simplicity and relative lack of ambiguity, rather than richness and completeness. We wanted the gestures to be easy to perform by the user and easy to recognize by the system. While there are many other gestures possible in 2D and 3D, we feel that by combining gestures, rather than introducing more primitives, we can create a rich set of useful interactions that span the 2D–3D divide in a wide range of potential applications.

*Gesture Reuse.* Rather than requiring the user to memorize many hand postures or movements in order to interact with the environment (an approach taken in the *Charade* system [Baudel 1993]), we can use the contextual information as well as reuse our sim-

ple gestural primitives in various combinations to create a larger vocabulary of more complex interactions.

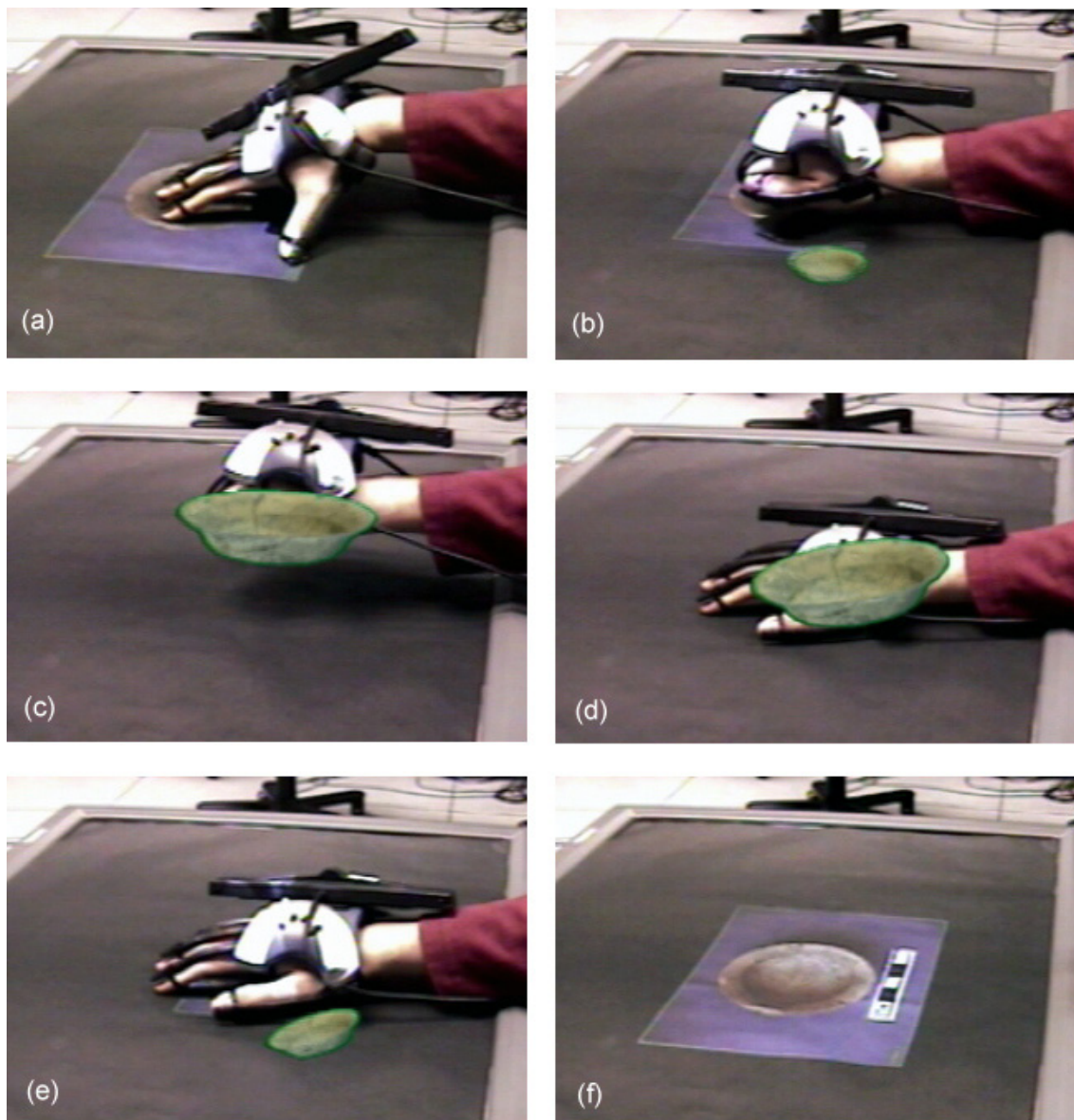
*Context Preservation.* It is important to observe and preserve the interaction context. A simple and obvious example is that the 2D interpretation should be weighted higher when the user is interacting with the 2D data on the surface. A more complex example occurs when the user wants to transition data between 2D and 3D environments. The user starts by interacting on the 2D surface, and finishes in the 3D world. The user's gesturing should follow the intended transition, from 2D to 3D. Considering contextual information also allows for gestural reuse and keeps our interactions relatively simple and intuitive.

### 5.2.3 Facilitating Transitions: Cross-Dimensional Pull and Push

There are three pieces of information required to complete a precise and successful transition across the dimensionality border: the *source object*, the *indication of transition*, and the *destination location*. We provide a way to satisfy all three requirements with one smooth and continuous gesture without ambiguities.

*Cross-Dimensional Pull* transitions an object from 2D to 3D. This gesture mimics a natural grabbing gesture and can be best described as “trying to pull an object away from the surface.” In VITA, in addition to simply performing a transition, we also transform an object to match the destination dimensionality. Therefore, in VITA, the Cross-Dimensional Pull gesture converts the 2D representation of an object (a 2D image) into a 3D representation (a 3D model) (as seen in Figure 5.2a–c).

To perform this gesture, the user places the hand flat on the table on top of the object's image representation, then moves the fingers towards the center of the object, and finally forms a 3D grab. As this occurs, the 2D image on the table scales down and disappears, while a 3D model appears and scales up, attached to the user's grabbing hand, representing the transformation from 2D to 3D. This gesture is detected as simultaneous transitions in both 2D (flat hand → tap) and 3D (idle → grab), with the hand moving away from the surface.



**Figure 5.2:** Frames from a sequence of Cross-Dimensional Pull and Push gestures. (a) Cross-Dimensional Pull is performed by selecting the 2D object and then forming a grabbing gesture. (b) The 2D object disappears, while the 3D object appears from the table. (c) Holding the 3D object. (d) Cross-Dimensional Push is performed by pressing on the table with a flat hand through the 3D object. (e) The 3D object disappears and the 2D object appears. (f) The 2D object (an image), projected on the table.

The Pull gesture exemplifies a theme common to the entire set: the preservation of interaction context. If the intention is to transfer something from 2D to 3D, the user starts the gesture on the 2D surface (indicating the source), and finishes it in the 3D

world (indicating the destination) and the transition is established by synchronizing the action on the surface.

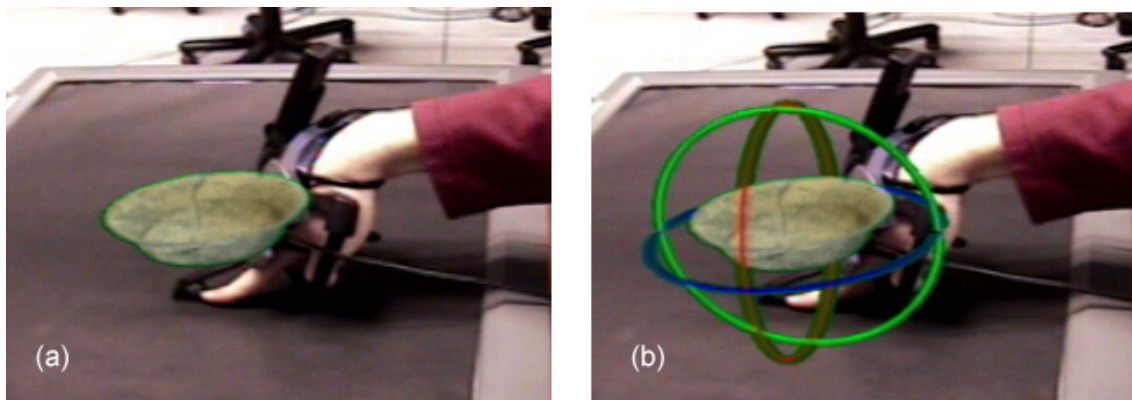
*Cross-Dimensional Push* is a complement to Cross-Dimensional Pull. This gesture transitions the object from 3D to 2D representation and is performed by pushing (pressing) through a 3D object and into surface (Figure 5.2d–f). Once the 3D object is placed above the table, a user can position a flat hand above the object and “push” downwards until her hand touches the table, mimicking an attempt to push a real physical object into the table. Unlike hand pose transitions of the Pull gesture, the Push gesture is detected in 3D as idle hand moving downwards and resulting in 2D flat hand. As this occurs, the 3D model being pushed scales down and disappears, while the 2D image appears on the table and scales up. (Alternatives to our uniform-scaling visualization include one-dimensional scaling along the axis perpendicular to the table to “flatten” a 3D object or “extrude” the 2D one, or clipping the object against a virtual plane coplanar with the table.)

Providing consistent and continuous visual feedback reinforces the metaphor of object transformation between the 2D and the 3D environment. This aspect is critical to the user’s understanding of their actions as pointed by our test users in Section 5.3.1.3.

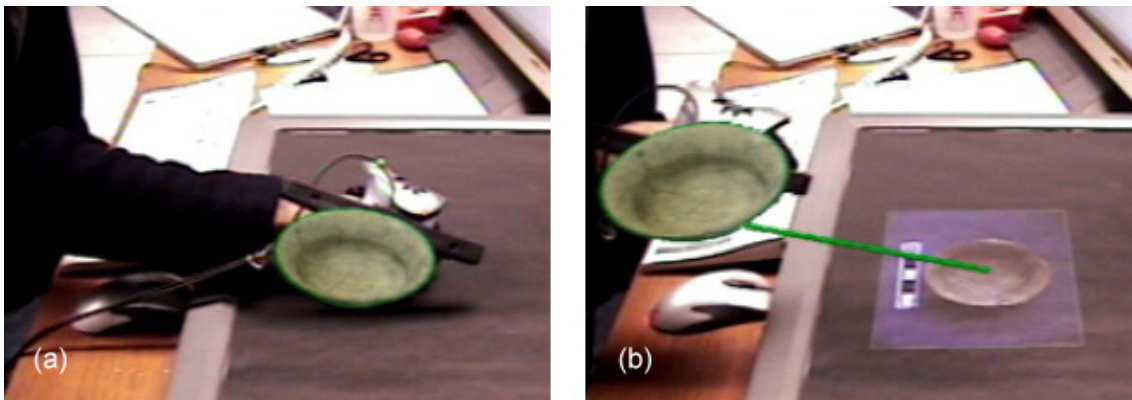
#### 5.2.4 Facilitating Associations: Cross-Dimensional Pin and Connect

Another way that the user can connect the 3D and the 2D environments is by associating a 3D object with the table. In contrast to the Cross-Dimensional Push, which transforms a 3D object into a 2D object, a *Cross-Dimensional Pin* attaches the 3D object to the 2D surface. Pinning an object is accomplished by placing the 3D object close to the table, making a 3D point gesture with the hand, and then pushing through the object until the hand touches the table (2D one-finger) (Figure 5.3a). In VITA, three colored rings, perpendicular to the object’s  $X$ ,  $Y$ , and  $Z$  axes, appear around the object to signal that the object is pinned to the table (Figure 5.3b).





**Figure 5.3: Cross-Dimensional Pin associates the 3D object with the surface: (a) Using a 3D point gesture to pin the object to the surface. (b) Rings appear once the object is pinned, providing visual feedback to the user.**



**Figure 5.4: Cross-Dimensional Connect: (a) The user performs the gesture by tapping the table while holding a 3D object. (b) The leader line connects two representations of the same object.**

To reverse this action, or *unpin* the 3D object, the user selects it (3D grab) and moves it away from the table surface. As soon as the object is grabbed in 3D, it is no longer pinned, which can be seen by the disappearance of the three colored rings around it.

A different form of association can be achieved when the user wants to establish a connection between the 2D and 3D object representations. To establish this contextual linking relationship between the two representations, the user performs a *Cross-Dimensional Connect* gesture. The gesture is performed by first selecting (3D grab) the 3D object, pulling it towards the table, and then briefly tapping (2D tap) the table any-

where while continuing to hold the object (Figure 5.4a). This results in a 2D object appearing on the table surface, while the 3D object remains visible and attached to the hand.

Once the two representations are connected, a leader line<sup>†</sup> appears between them, highlighting the connection (Figure 5.4b). This gesture resembles pulling a virtual string between one object and the other, which allows 3D objects placed anywhere in 3D space (even behind the user) to be found by first locating its 2D representation and then following its leader line to its 3D location. The same tapping gesture used to connect the two representations can also be used to break the connection (i.e., disconnect them).

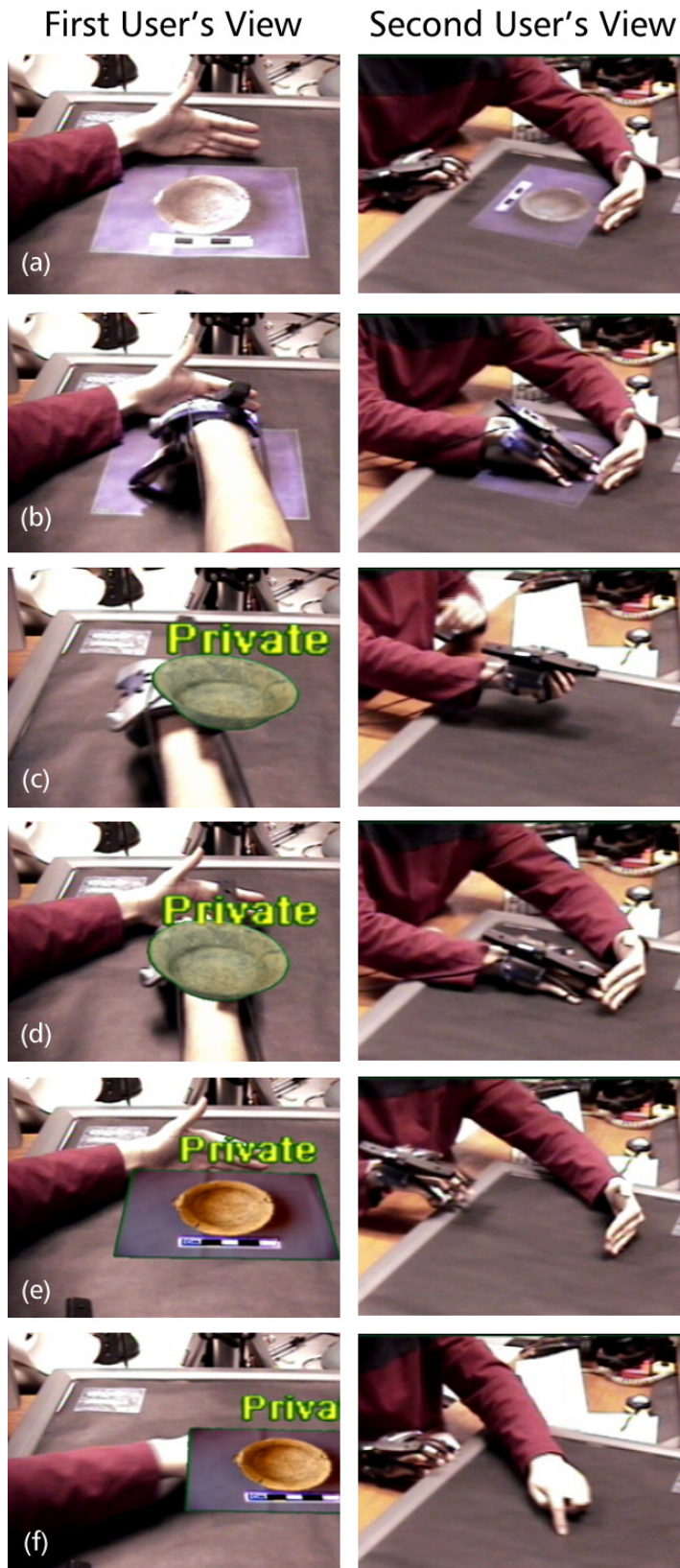
Both Cross-Dimensional Pin and Cross-Dimensional Connect gestures facilitate association of the 3D object with various 2D objects or surfaces. We believe that by doing so, they allow the user to use the affordances of the 2D surface to more easily perform manipulations on the pure virtual 3D object. For example, in addition to being able to manipulate a 3D object with a 3D tracked hand, a pinned 3D object can easily be moved or rotated around in-plane by a non-3D-tracked hand.

### 5.2.5 Gesture Modifiers: Cross-Dimensional Privacy Pull and Push

The gestures presented so far form the basic interaction set for facilitating transitions and associations in hybrid MDEs. However, it is possible to further modify such interactions to extend their functionality. For example, we implemented a privacy modifier that allows the user to change the data privacy status during transitions between displays by performing *Cross-Dimensional Privacy Pull* and *Cross-Dimensional Privacy Push* gestures (Figure 5.5). The privacy modifier is invoked by placing the hand vertically on the touch-sensitive surface (2D chop) and is inspired by the 2D shield gesture of Wu and Balakrishnan [Wu 2003]. With the 2D chop gesture, the user “shields” a Cross-Dimensional Pull or Push gesture performed with the other hand. In doing so, the selected object is tagged in transition and subsequently only visible in the head-worn display of that user and completely invisible to other users.

---

<sup>†</sup> This leader line visualization is similar to the use of leader lines in EMMIE [Butz 1999]. In EMMIE leader lines extend from a search control panel on a hand-held display to 3D virtual objects that meet the search criteria.



**Figure 5.5: Side-by-side comparison of two users' views when using Cross-Dimensional Privacy Pull and Push gestures:** (a) The user places a vertical hand on the table to indicate a private gesture. (b) A Pull gesture is performed with the 2D chop gesture above it, yielding a Cross-Dimensional Privacy Pull. (c) 3D object is tagged as private and invisible to the other user. (d) A Push with the 2D chop gesture above it results in a Cross-Dimensional Privacy Push. (e) 2D object appears as if on the table, but is invisible to the other users. (f) 2D private object behaves like all other 2D objects and can be dragged around.

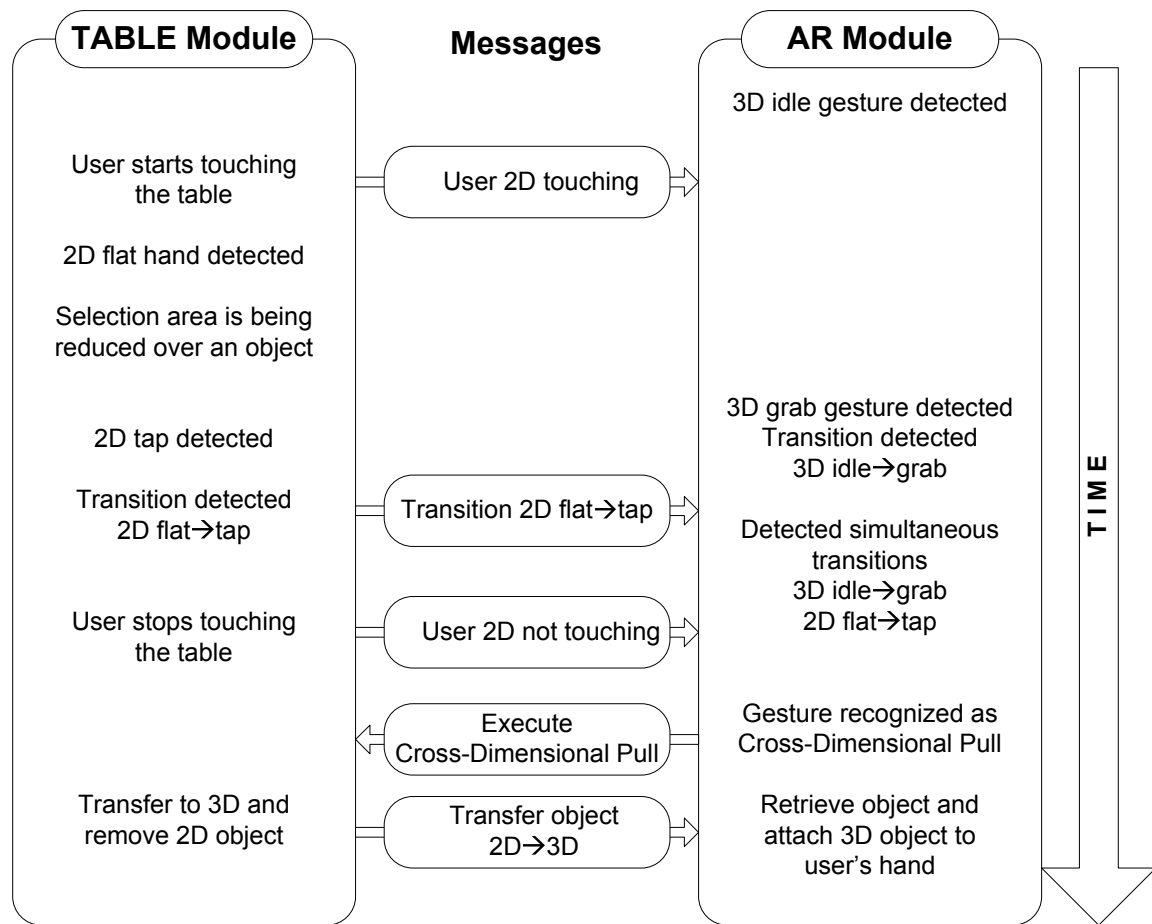
The Cross-Dimensional Privacy Pull gesture allows the user to transition 2D data into 3D and view it privately in their head-worn display (Figure 5.5a–c). A complementary Cross-Dimensional Privacy Push allows a private view of 2D information in the head-worn display, rendered in perspective as if it were projected on the tabletop surface (Figure 5.5d–f). Such 2D private objects behave just like a normal 2D object and can be manipulated using the standard 2D touch-based interactions on the surface.

There is an additional benefit to private interactions in a hybrid MDE: choosing to hide some objects, when they are not of interest to others, can drastically reduce the visual clutter in the AR displays of other collaborators. While we only experimented with privacy modifiers for Pull and Push gestures, performing private Pin and Connect, as well as using other poses to trigger modified interactions, is definitely possible and should be explored further.

### 5.2.6 Synchronization of Gesture Recognition across Devices

All cross-dimensional gesture techniques presented here are implemented on top of the VITA modular framework. The execution of cross-dimensional gestures requires that the AR module and the TABLE module communicate extensively and synchronize their gesture recognitions. Figure 5.6 illustrates the message passing and synchronization between the two modules during the recognition of a Pull gesture. In this example, all messages are assumed to pertain to the same user. However, in our multi-user system, each message is tagged with the “user name” tag to allow for disambiguation among users and a timestamp value to allow for synchronization of gestures.

For each user, there is an interaction integrator component implemented as a part of the user’s AR module. The interaction integrator aggregates all interaction messages pertaining to this particular user and performs cross-dimensional interactions when it detects that a particular combination of gestures have occurred. All gesture combinations that facilitate various cross-dimensional interactions are listed in Table 5.2.



**Figure 5.6: Sample synchronization communication between 2D and 3D modules during the recognition of Cross-Dimensional Pull.**

### 5.3 Usability Evaluation of Cross-Dimensional Gestures

To gather feedback about the ease-of-use and the intuitiveness of our Cross-Dimensional Gestures, we conducted a qualitative usability study with twelve volunteer participants. The participants were mostly male (11 males, 1 female), between the ages of 17 and 32, and all right-handed. They were recruited by email from the Columbia University student population and were all frequent computer users. Two participants had some previous virtual reality or augmented reality experience. The participants were not compensated for their time.

### 5.3.1.1 *Procedure*

Each participant was paired up with a study coordinator and spent approximately 45 minutes learning about and performing cross-dimensional gestures. Participants were given a post-test questionnaire in which they were asked to rate each gesture's ease of use and intuitiveness, and rate how stimulating and satisfying their overall experience was, each on a five-point Likert scale. The participants were also encouraged to provide free-form comments on each gesture and their overall experience.

For each participant, the study coordinator first demonstrated how to perform each of the gestures exactly once, and then afterwards asked the participant to try to perform the gestures. If at any point the participant had difficulty performing a particular gesture, the coordinator verbally coached them to completion. Once the entire set of gestures was completed and the participant was familiarized with performing the gestures, the coordinator asked the participant to perform a specific number of gestures by giving them a verbal request for each gesture. The participants then had a chance to explore the VITA system during the remainder of their time slot and fill out the questionnaire.

### 5.3.1.2 *Setup*

Our experiment was performed in the VITA environment. Both the coordinator and the participant wore head-worn displays, and were seated in front of the projected MERL DiamondTouch table. They also wore an EssentialReality P5 glove on their right hand and both their head and their hand were tracked by the overhead tracking infrastructure (InterSense IS900).

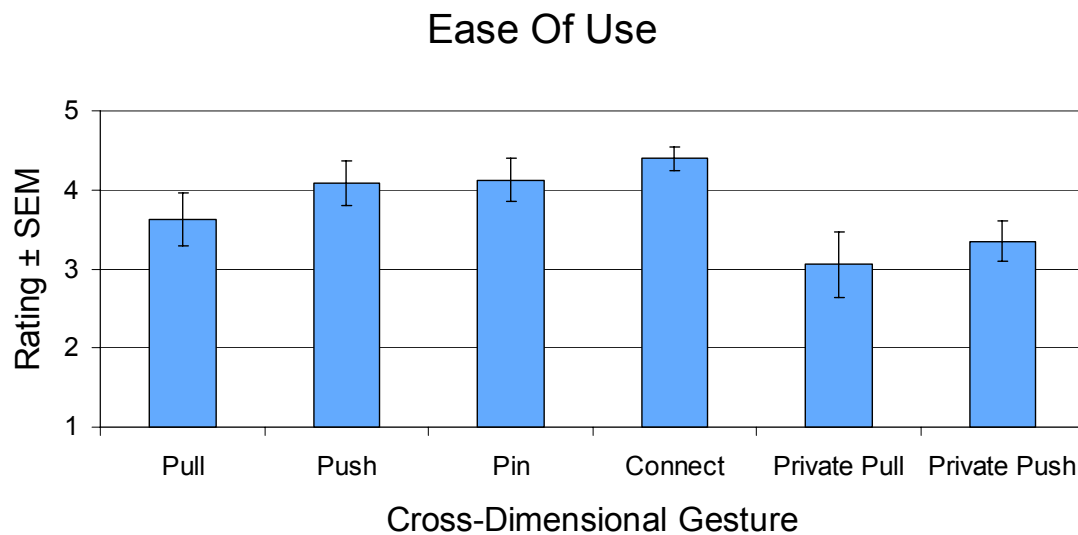
To remove all other distracting factors, only the AR and TABLE displays were used this experiment, and all other VITA displays were turned off. In addition, the tabletop interface was simplified and only consisted of a single 2D image of a sample archaeological object.

### 5.3.1.3 Results and Feedback

This non-comparative study was primarily designed to collect participant feedback and preferences used to guide future design of cross-dimensional interactions.

All twelve participants successfully completed every gesture, with most gestures performed successfully on the first attempt. Overall, we received very positive comments from all participants and their average subjective ratings for ease of use and intuitiveness of each gesture can be seen in Figure 5.7 and Figure 5.8, respectively.

Most participants found both Pull and Push to be fairly easy to use (Pull  $\mu_{\text{ease}}=3.62$ , Push  $\mu_{\text{ease}}=4.08$ ) and intuitive (Pull  $\mu_{\text{int}}=4.00$ , Push  $\mu_{\text{int}}=4.58$ ). Interestingly, the ease of use of two-handed privacy gestures was rated significantly lower than their non-private counterparts: the comparison of Pull ( $\mu_{\text{ease}}=3.62$ ) and Privacy Pull ( $\mu_{\text{ease}}=3.06$ ) reveals significantly different ratings when using the Wilcoxon Signed-Rank Test for paired ordinal data ( $W=-39$ ,  $p=0.039$ ), as does the comparison of Push ( $\mu_{\text{ease}}=4.08$ ) and Privacy Push ( $\mu_{\text{ease}}=3.35$ ) ( $W=-39$ ,  $p=0.018$ ). These differences are not present in ratings of intuitiveness.

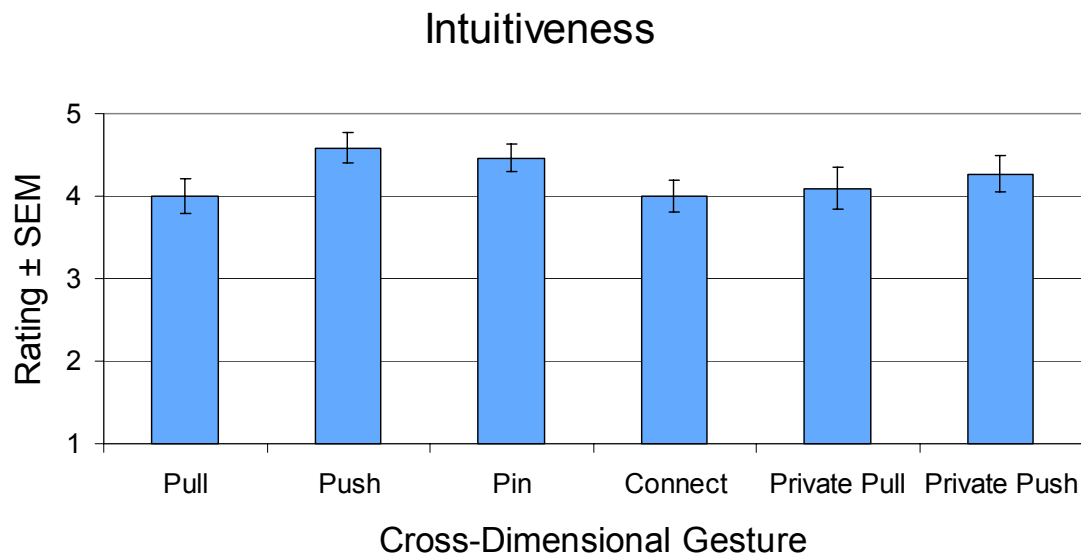


**Figure 5.7: Mean rating scores for ease of use of cross-dimensional gestures. Participants ranked each gesture on a five-point scale (1-difficult, 5-easy).**

Judging from the participant study feedback, participants appreciated the simplicity of gestures that relied only on hand poses (e.g., Push) when compared to the gestures



that also considered the pose transitions (e.g., Pull). The hand motion, such as the scooping motion of Pull (2D flat hand to 2D tap), was introduced to reinforce the natural metaphor of grabbing or scooping something from a tabletop, but it introduced many subtle timing and calibration requirements that made it a little bit more difficult to perform reliably. This effect can be seen in the difference between the ease of use score for Pull and Push.



**Figure 5.8: Mean rating scores for intuitiveness of cross-dimensional gestures. Participants ranked each gesture on a five-point scale (1-confusing, 5-intuitive).**

We believe that the constraints of our touch-sensitive surface (which only reports the largest axis-aligned rectangle containing all of the current participant’s touches) greatly contributed towards the difficulty of performing and recognizing the two-handed privacy gestures. For both Privacy Pull and Privacy Push, several participants asked for more tolerance when shielding with the left hand, stating that it was difficult to maintain a completely straight and orthogonal (to the side of the table) chop gesture. Some stated that “using two hands is annoying, it would be nice to be able to do it with one hand” and that there was a “steep learning curve for privacy gestures.” Other participants, however, stated “it was very easy to remember and use” and “the shield gesture wasn’t too difficult to perform.”



Seven participants pointed out that our short visual animations of Push, Pull, Privacy Push, and Privacy Pull, were critical in making these interactions feel like seamless transitions. Furthermore, three user study participants suggested that additional visual (and audio) feedback should be incorporated into the interactions, especially when gestures were performed incorrectly.

The Pin and Connect gestures, which were the simplest to perform, received the highest ratings for ease of use (Pin  $\mu_{\text{ease}}=4.13$ , Connect  $\mu_{\text{ease}}=4.40$ ). There were many positive reactions, which mostly conveyed participants' appreciation for the simplicity of such interactions. Three participants commented on the usefulness of the leader line visualization for providing contextual links between representations, and nine participants enjoyed being able to manipulate pinned 3D objects by touching the surface.

While the intuitiveness ratings for the Connect gesture ( $\mu_{\text{int}}=4.00$ ) do not show significantly different results from other gestures, some participants noted small frustrations when performing that gesture. One participant commented that "there was a lack of an intuitive connection between what you do and the result you get," which suggests that the some users did not find the "pulling the string" metaphor clear and appropriate. Interestingly, that participant adopted a hybrid two-handed approach to executing Connect: he would hold the 3D object in one hand, and tap the tabletop surface with the other, thus forming a connection "through his body". Interestingly, this modification of the Connect gesture did not need to be specially designed, but rather it was directly supported by our synchronized recognition approach. Since we consider two independent and synchronized modalities in the recognition of these gestures (2D touch and 3D glove), it is up to the user to decide how they will execute each modality. This leaves room for both one- or two-handed versions of our gestures, ultimately only depending on the user's personal preference.

In addition, both Pin and Connect received some negative comments addressing hardware limitations of the instrumented glove. For example, participants commented on their frustrations when a part of the glove prevented the participant's hand from directly touching the table while tapping with a closed fist. Despite this, most participants found the Pin and the Connect gestures to be fairly intuitive, commenting "it was very straight forward" and "it was very easy to do with no mistakes."

Overall, when asked about their satisfaction, participants rated their experience as very stimulating ( $\mu=4.83$ ) and fairly satisfying ( $\mu=3.84$ ). Six out of twelve participants made comments expressing their frustrations early on (during the learning phase), later followed by positive statements after having practiced the gestures several times. One participant wrote “I didn’t feel like the gestures were too tough to learn. Within one of two tries, I seemed to pick them up.” Another wrote, “I was really amazed at what was possible.”

## **5.4 Discussion**

Overwhelmingly positive comments from our usability study lead us to believe that the participants would find it both useful and enjoyable to use Cross-Dimensional Gestures to interact in a hybrid MDE like VITA. We were excited about the inherent flexibility in our approach, which automatically allowed the user to use either a single continuous gesture with one hand or a combination of gestures with two hands. Our gestures require a small learning period, after which the participants reported relatively easy and unambiguous interaction.

We designed Cross-Dimensional Gestures to require the use of both 2D and 3D modalities for three reasons. First, we believe that the use of synchronized actions is a powerful and easy way to automatically establish connection between devices and displays without requiring any additional steps. This was also suggested previously by both Holmquist and colleagues [Holmquist 2001] and Hinckley and colleagues [Hinckley 2003]. While our system currently relies on identifying who is touching the tabletop, one could use the timing of the gestures themselves as a form of virtual signature to automatically establish the connection.

Second, by considering the gestures on both sides of the dimensionality border, we can reuse the common and simple hand poses that are normally used throughout the environment (e.g., point or grab) and give them a special meaning when interacting on the dimensionality border.

Third, by requiring the user to touch the surface in order to trigger the cross-dimensional interaction, we eliminate many potential accidental invocations or ambiguities.

ties that would arise by relying on 3D gestures alone. The surface acts as a clutching mechanism that the user knowingly engages when wanting to interact. This mechanism allows the user to maintain the normal human gesticulation when interacting above the surface without the fear of accidentally triggering some interaction.

In summary, this chapter presents a set of interaction techniques for hybrid MDEs that connect 2D touch-sensitive displays and 3D AR user interfaces. Our Cross-Dimensional Gestures use synchronized 2D and 3D gestures to facilitate the seamless transition and association of data across the display and dimensional boundary. Our gestures are demonstrated using a tethered bend-sensing glove, but video-based gesture tracking above the surface could eliminate the need for a glove and allow for much more flexible un-tethered interactions.

In addition, our techniques are designed for multi-user collaborative environments, and thus address some of the need for privacy in these shared spaces. Initial user study feedback suggests that participants found the gestures very intuitive and easy to perform, after a relatively short learning period, possibly due, in many cases, to their close similarity to physical world interactions. Cross-Dimensional Gestures are presented within the VITA hybrid MDE, but we believe that they are applicable to a broad range of other hybrid environments.



## 6 Improving the Precision of Touch Screen Interaction

When relying on gestural and touch-based interactions on displays, the size of human fingers and the lack of sensing precision can make precise touch-screen interaction difficult. In VITA, the graphical user interface on the touch-sensitive tabletop was carefully designed to feature large graphical targets to allow for easy touch-based interaction. However, software developed for a normal mouse interface often features small targets that are difficult to select with a fingertip. Touch-based interaction is further complicated by noisy touch-sensing input, lower touch-tracking resolution, or the fact that fingertips can occlude small targets depriving users of visual feedback during target acquisition.



This chapter presents novel interaction techniques that improve the accuracy of selection when using fingers on a multi-touch display. We begin by discussing the issues of precise touch screen interaction in greater detail and by reviewing some previous solutions. Next, we present a set of five techniques, called *Dual Finger Selection*, that allow the user to simultaneously perform both cursor steering and selection of an assistance mode (in the form of cursor offset, an increase in scale, a reduction in speed, or a combination of these) [Benko 2006]. Lastly, we discuss a pointer stabilization technique, called *SimPress*, which reduces the amount of cursor jitter during clicking and allows us to simulate a hover state on devices unable to sense touch proximity.

The formal user study at the conclusion of this chapter examines the performance and error rate of both the SimPress and Dual Finger Selection techniques. We show that these techniques significantly improve selection accuracy on a multi-touch screen,

which makes them directly applicable towards our goal of enabling gesture-based and touch-based interaction as the main interaction metaphor in a hybrid MDE.

## 6.1 Touch-Screen Selection Issues

Five factors make precise touch-screen selection difficult: fingertip area, target occlusion, lack of a hover-state, clicking noise, and tracking noise.

Target Element	Device Space	Target Width	
		Visual Space 17" screen 1024×768	Visual Space 30" screen 1024×768
Close Button 	18 pixels	6 mm (32.8 % of finger)	10.8 mm (59.0 % of finger)
Resize Handle 	4 pixels	1.34 mm (7.3 % of finger)	2.4 mm (13.1 % of finger)

**Table 6.1: Comparison of small target widths to an average right hand index finger width at the fingertip (estimated as 18.3 mm for an adult<sup>\*</sup>). Note: Values for the 30" diagonal screen correspond to the display characteristics used in this chapter.**

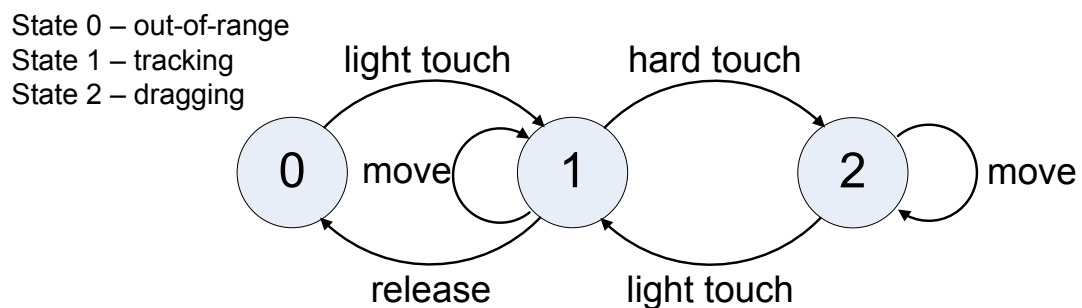
*Fingertip area:* Most of the graphical user interfaces used today are designed for indirect and pixel-precise input (e.g., using a mouse or trackball). It is a difficult task to adapt such interfaces for use with a fingertip, which covers a large number of pixels on the screen. Table 6.1 highlights the discrepancies between the average index finger width [Feeney 2002] and the widths of several standard small targets. The use of a stylus (pen) is generally preferred in many interfaces that require precise interaction. However, while a stylus has a much smaller tip, hand tremor makes selecting small targets with a stylus more difficult than with a mouse. Ren and Moriya [Ren 2000] report that

<sup>\*</sup> Feeney [Feeney 2002] reports a distal joint breadth (width) mean value of 18.3 mm for the index finger on the right hand of an adult. Distal joint is the finger articulation furthest from the hand.

the limiting size for stylus targets is about 1.8 mm, below which stylus interaction requires additional assistance.

*Target occlusion:* Placing a fingertip on top of a target can partially or completely occlude it from the user, reducing visual feedback during selection. The user’s hands and arms may also occlude the target. Depending on screen orientation, the user may be forced to either look “under hand” (with horizontally positioned screens) or “over hand” (with angled or vertically positioned screens) in order to see what they are selecting.

*Lack of a dragging state:* Buxton, Hill, and Rowley [Buxton 1985] identified that most current user interfaces require an interaction model that can reliably disambiguate between two movement states (*tracking* and *dragging*). For example, by pressing the left button on the mouse, one enters the dragging state; otherwise the mouse is in the tracking state (also known as *proximity* or *hover* state). The transition between tracking and dragging states is interpreted as a *mouse click*. However, many touch-sensitive devices can reliably sense finger location only when the finger is in direct contact with the surface, making it hard to disambiguate between dragging and tracking. A solution to this problem is to use touch pressure to disambiguate between the two movement states (states 1 and 2 in Figure 6.1).



**Figure 6.1: State diagram for simulating standard mouse interaction on a touch screen (adapted with modifications from [Buxton 1985]).**

Maintaining a relatively stable amount of pressure throughout the high-pressure dragging state is difficult and often results in spurious switching between dragging and tracking states. This is manifested in the interface as spurious mouse clicking. In addition, when devices are not pressure-sensitive, the touch area can be used to approximate

pressure (e.g., touchpads by Synaptics [MacKenzie 1998, Synaptics 2007]). Our work in this dissertation uses this touch-area-based approach.

*Clicking noise:* In touch screen interactions, the cursor is usually anchored at the center-of-mass (centroid) point of the user’s touch contact. The change in contact pressure (or area) can change the center-of-mass location and move the pointer cursor position on the surface. This is completely undesired behavior when one is trying to keep the pointer steady while clicking.

*Tracking noise:* Another source of error can be linked to touch sensing itself. This is particularly so in many multi-touch sensitive devices that rely on vision-based sensing (e.g., [Wilson 2004, Han 2005, Wilson 2005a]). The mismatch between the sensing and display resolutions, as well as video noise, often tends to make input on such devices more noisy than when using other non-video based approaches (e.g., electro-static coupling [Dietz 2001] or capacitive-based sensing [Rekimoto 2002]).

## **6.2 Previous Single-Touch Solutions**

Difficulties with precise interaction on touch-screen devices have been addressed before, initially by Potter et al. [Potter 1988]. Their *Take-Off* technique provides a cursor with a fixed offset above the tip of a finger when the user is touching the screen. Lifting the finger off the screen triggered selection (“click”). Sears and Shneiderman [Sears 1991] explored cursor stabilization improvements that effectively slowed down the cursor movement in various regions around the initial finger contact point, thus allowing for pixel-precise selection. While this method performed well for the target acquisition task, a precise steering task, such as drawing, would be hard due to varying cursor speed.

More recently, Albinsson and Zhai [Albinsson 2003] explored several on-screen widgets for increasing precision while selecting small targets on a touch screen. Their interactions were designed to be used with touch screens capable of reporting only a single contact point and therefore the users were required to execute multiple discrete steps before being able to select the target. These steps were delimited by the user lifting their finger from the screen, thus impeding the overall interaction performance. In-



terestingly, they observed that even though their baseline zooming technique (*Zoom-Pointing*) performed best out of the four techniques compared, its main drawback of losing overview or context can be a significant problem in many applications.

All these solutions were designed for single-touch interfaces and they usually require the use of an additional activation trigger (e.g., an on-screen menu item or a physical button) to invoke them. Unlike any previous work, the techniques presented in this chapter explore the benefits of multi-touch surfaces and allow the user to invoke them by simply touching the screen with two fingers. These techniques allow the user to quickly modify or switch cursor manipulation modes with the secondary (non-pointing) finger without disrupting the primary (pointing) finger.

### 6.3 Dual Finger Selection Techniques

*Dual Finger Selection* techniques are a set of five two-finger interactions that allow the user to improve touch screen targeting precision when so desired without hindering simple targeting tasks. The first two techniques (*Offset* and *Midpoint*) present simple two-finger solutions that provide a cursor offset on demand. These two techniques gave us a starting point for our later designs and they serve as baseline techniques for our later experimental evaluations (Section 6.6). The subsequent three techniques (*X-Menu*, *Slider* and *Stretch*) build on top of the baseline techniques and allow the user a finer degree of control of cursor speed or control/display ratio.

Before discussing the details of our five iteratively designed techniques, it is important to outline the device requirements that enable our interactions.

#### 6.3.1 Interaction Requirements

Dual Finger Selection techniques require a multi-touch screen that is capable of simultaneously tracking at least two independent contacts on the surface. We also assume that in addition to the location of each contact, either its contact pressure or its contact area is reported as well, in order to disambiguate between tracking and dragging states, as discussed in Section 6.1. Since our multi-touch surface was not able to sense touch

pressure, we approximate the pressure state by the contact area (a more detailed description of the prototype multi-touch device used in this chapter can be found in Section 6.5.)

### 6.3.1.1 *Multi-Finger Order-Based Labeling Approach*

Most touch screens and tabletops cannot identify which of the individual user's fingers or hands is touching the surface without specially constructed gloves (such as the gloves used by Butler and Amant in their HabillisDraw DT application [Butler 2004]). To alleviate this problem, we have assumed an order-based labeling approach, which assigns a role to each finger based on the order in which the fingers touch the surface. Therefore, we label the first contact with a surface as a *primary* finger, the second contact as a *secondary* finger, etc. The primary finger is the finger that the user normally points with and tends to be the index finger on their dominant hand. The secondary finger is a helper finger, which in Dual Finger Selection can be any other finger on the same or opposite hand. In most cases, we observed that users used the index finger on their non-dominant hand as the secondary finger. The use of primary and secondary fingers in one of our techniques can be seen in Figure 6.2.

This order-based labeling approach is inspired by the complementary notion of determining handedness from the spatial relationship of two tracked devices described by Kurtenbach and colleagues [Kurtenbach 1997]. There are several reasons why this approach increases the overall flexibility of the interaction. First, both left-handed and right-handed users have no problems adopting their preferred finger usage, since Dual Finger Selection techniques allow either hand to be the dominant positioning hand. Second, allowing the user to choose which finger/hand to use for the particular role ensures that the entire touch screen surface is easily and equally reachable. For example, it is possible to control the position of the cursor with the left hand when close to the left border, to avoid crossing the hands. Third, this approach has an added benefit, in that Dual Finger Selection techniques can be used in both two-handed and single-handed interaction.



**Figure 6.2:** This image shows the use of the Dual Finger X-Menu in selecting “slow 10×” mode: the primary finger (right) controls the cursor position, while the secondary finger (left) adjusts the cursor speed.

### 6.3.2 Dual Finger Offset

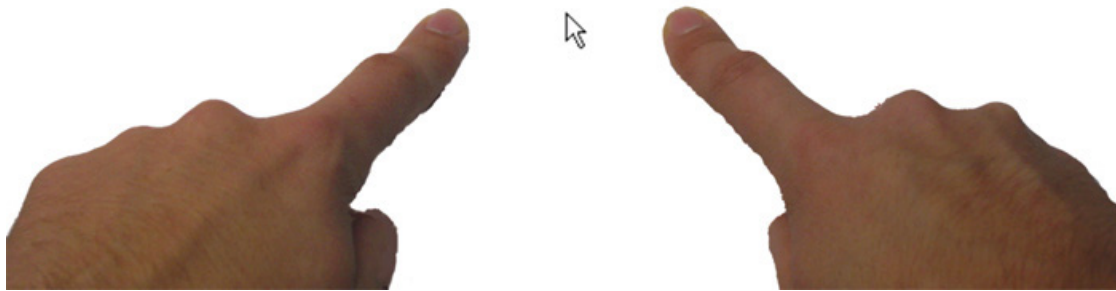
Our initial and simplest technique, called *Dual Finger Offset*, provides a user triggered cursor offset. The cursor offset is not enabled by default. However, by placing a secondary finger anywhere on the surface, the cursor is subsequently offset with respect to the primary finger by a predefined fixed amount. This offset always places the cursor above the primary finger. To accommodate both left- and right-handed users, the cursor is placed to the left or to the right of the primary finger, based on the relative position of the secondary finger. For example, by placing the secondary finger to the left of the primary, the cursor appears to the left of and above the primary finger.

### 6.3.3 Dual Finger Midpoint

In order to provide both variable offset and enable finer control of the cursor speed, we designed the *Dual Finger Midpoint* technique. This technique is triggered when the

secondary finger touches the surface. The cursor is then offset to the midpoint between the primary and the secondary finger.

While both fingers are in contact, moving either or both fingers controls the movement of the cursor. This technique allows for variable reductions in cursor speed: when both fingers are moving in the same direction and the same speed, the cursor follows with the same speed, while when only one finger is moving, the cursor moves with half the speed of that finger.



**Figure 6.3: Dual Finger Midpoint technique positions the cursor at exactly the halfway point between the two fingers, giving the user both a cursor offset as well as a variable reduction of cursor speed.**

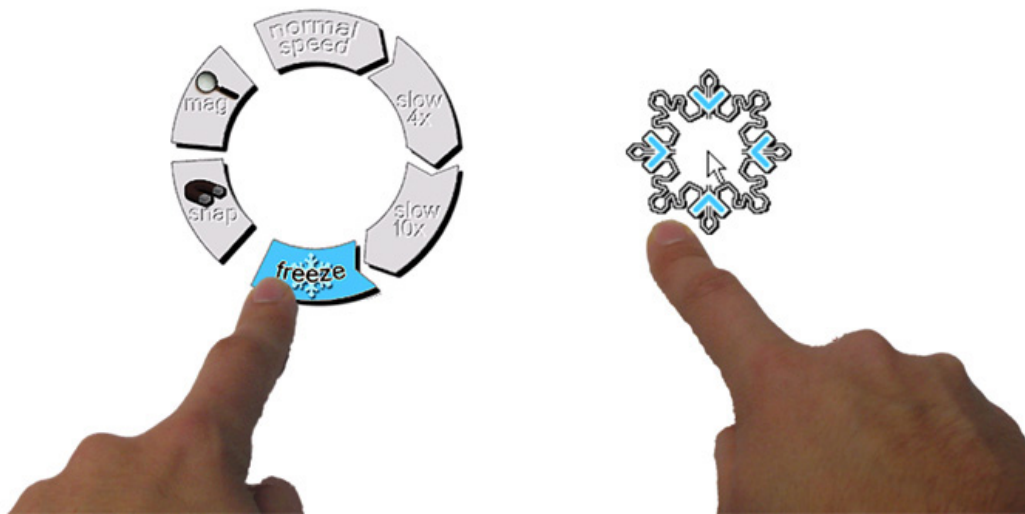
While the Dual Finger Midpoint technique was very appealing to our initial testers and very simple to master, it did not provide enough assistance for selecting the smallest targets (2 pixels or less). At best, this method reduces the finger speed by a factor of 2 which yields good results for most targets; however, it does not provide enough control for the smallest targets. An additional shortcoming of this technique is that not all locations on the screen are equally accessible. For example, screen corners are not accessible using midpoint selection. Consequently, the utility of this technique is somewhat limited by the fact that in today's user interfaces, small targets often are located in the corners of the screen.

#### 6.3.4 Dual Finger X-Menu

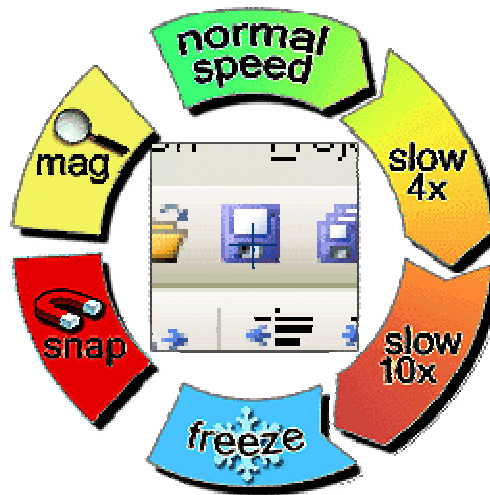
To allow users to adaptively adjust the control-display ratio, as well as obtain a cursor offset while looking at an un-zoomed user interface, we designed the *Dual Finger X-Menu* widget. This pie menu is invoked whenever the secondary finger establishes

contact with the surface. It is positioned so that the finger is located at its center. The user can select a particular assistance mode by moving the secondary finger to any of the desired regions of the menu (Figure 6.4). Removing the secondary finger from the surface causes the menu to disappear.

Once visible, the Dual Finger X-Menu is not operated by clicking on a particular area, but rather by “crossing” the finger into a particular area (hence the “X” in the name), which enables more experienced users to activate modes by simply performing quick strokes in a particular direction. The benefit of using such stroking gestures for selection of the menu item has been demonstrated by Kurtenbach and Buxton in their evaluation of marking menus [Kurtenbach 1994] and formalized later on by Accot and Zhai [Accot 2002]. In combination with the cursor notification widget, this crossing behavior allows the user to maintain their focus on the task of selection, and adjust the cursor speed in an eyes-free manner. This could eventually allow for an expert mode, much like that of marking menus, in which the menu could be completely hidden from the user.



**Figure 6.4: Dual Finger X-Menu enables the user to adjust the cursor speed by crossing over a particular area of the on-screen menu. Here, “freeze mode” is selected, making the cursor completely immobile.**



**Figure 6.5:** Dual Finger X-Menu<sup>†</sup> contains four selection areas for cursor speed control (normal, slow 4x, slow 10x and freeze), and two toggle areas (snap and magnify). Magnify mode presents an integrated magnification widget in the middle of the menu, while Snap mode removes the current cursor offset.

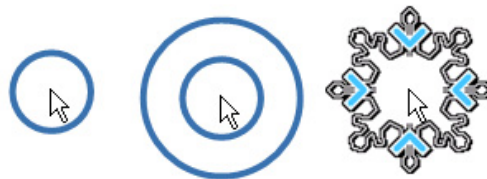
The Dual Finger X-Menu has six selection areas, as shown in Figure 6.5. Four areas control the relative speed of the cursor: *normal*, *slow 4x*, *slow 10x*, and *freeze*. Normal mode moves the cursor with the same speed as the primary finger, the two slow modes reduce the speed of the cursor by factors of 4 and 10, respectively, while freeze mode “freezes” the cursor in place, disabling any cursor movement. In preliminary experiments, we found that the ability to completely stop the cursor from moving has two benefits. First, by freezing the cursor, the user can quickly and easily establish a desired cursor offset. This is accomplished by freezing the cursor temporarily, moving the finger to achieve the desired offset, and then unfreezing the cursor again. Second, when selecting very small targets, even small amounts of noise can cause an error. Such noise can be due to device tracking errors, tremor in the user’s hand, or noise due to the clicking motion. By freezing the cursor in place, the user can ensure that the desired selection will be successful even in very noisy conditions.

The left two areas on the crossing menu invoke two helper modes: “snap” and “magnify”. When snapping is triggered, the cursor offset (if any) is removed and the

<sup>†</sup> Menu graphics were designed by Patrick Baudisch.

cursor snaps back to the current location of the primary finger. This mode is useful in repositioning the cursor in the slow movement modes because it is easy to run out of tracked screen space when using the slow cursor modes. Snapping is also accompanied by a distinct sound effect to increase the multimodal feedback to the user. Magnify mode presents a small magnification area in the middle of the crossing menu that shows the enlarged area under the cursor. The magnification factor is fixed at 2 $\times$ . This mode is particularly useful when the primary finger overlaps the cursor. In this case, the magnified image acts as a lens, showing the portion of the interface obstructed by the primary finger.

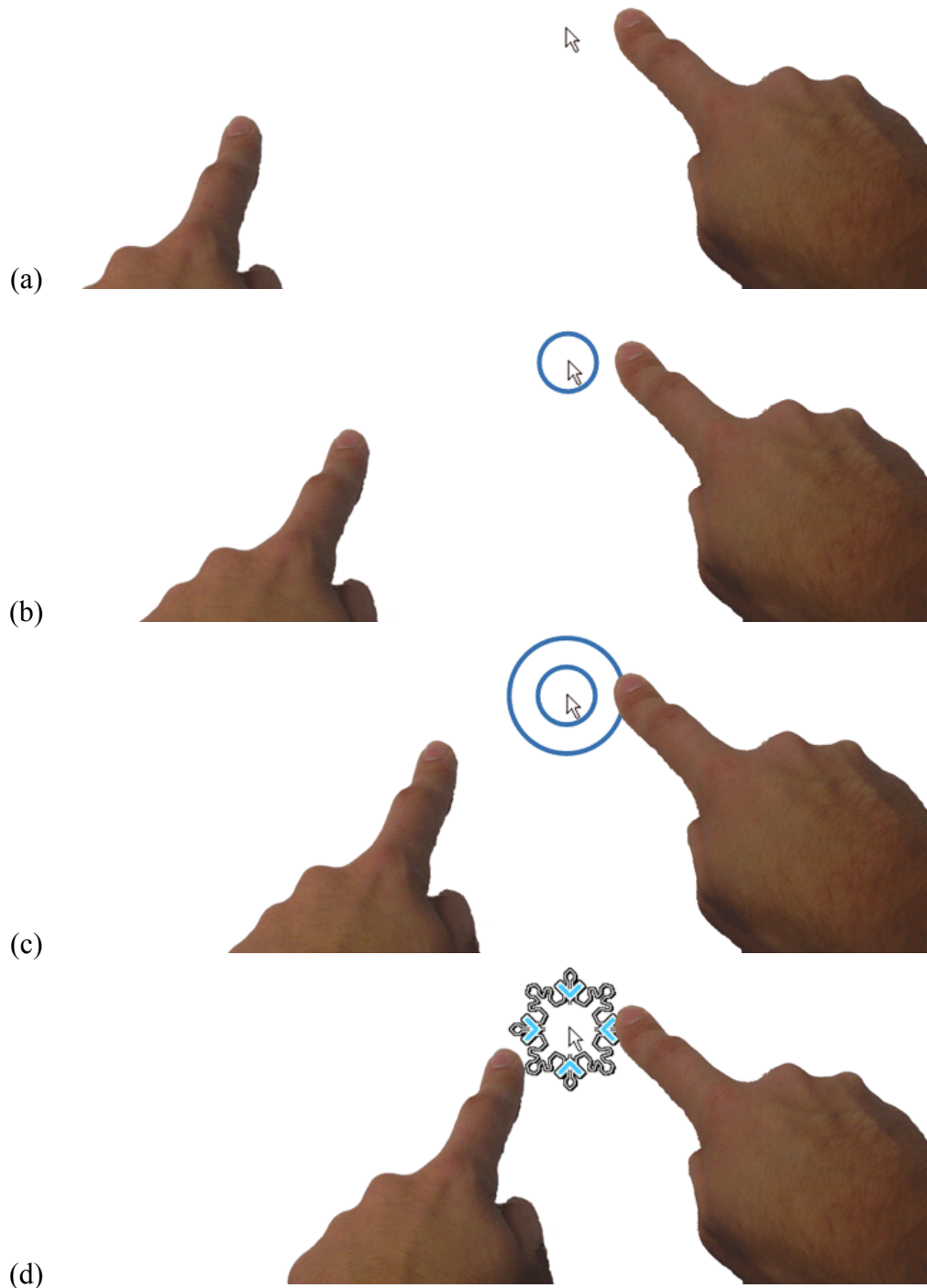
A simple cursor notification widget is used to display the currently selected cursor speed level, without requiring the user to refer back to the on-screen menu. The behavior of this notification widget can be seen in Figure 6.6.



**Figure 6.6: Cursor notification widget signals the current amount of cursor speed reduction: 4 $\times$  reduction (left), 10 $\times$  reduction (middle), and frozen cursor (right).**

### 6.3.5 Dual Finger Slider

Encouraged by the possibilities of the different interaction modes of the Dual Finger X-Menu, we developed the *Dual Finger Slider* technique, which incorporates the menu's most useful features, but simplifies and streamlines the overall interaction. Given that two-finger interactions are a very natural way of specifying distance, we have designed this interaction using the distance between fingers to switch between different cursor-speed modes. This technique does not present an on-screen graphical widget to the user and relies on the user's ability to gauge the spatial relationship between their fingers. The cursor notification widget signals the cursor speed to the user (Figure 6.6).



**Figure 6.7: Dual Finger Slider.** The right finger (primary) controls the cursor, and the left finger (secondary) is invoking the invisible slider. Speed reduction modes are achieved by moving the fingers closer together: (a) normal, (b) slow 4×, (c) slow 10×, (d) frozen cursor mode.



Moving the secondary finger towards the primary finger reduces the cursor speed in three discrete steps (Figure 6.7). This allows for the same reductions in cursor speed that are available in the Dual Finger X-Menu: *normal*, *slow 4×*, *slow 10×*, and *freeze*. Moving the secondary finger away from the primary finger increases the speed up to the normal speed.

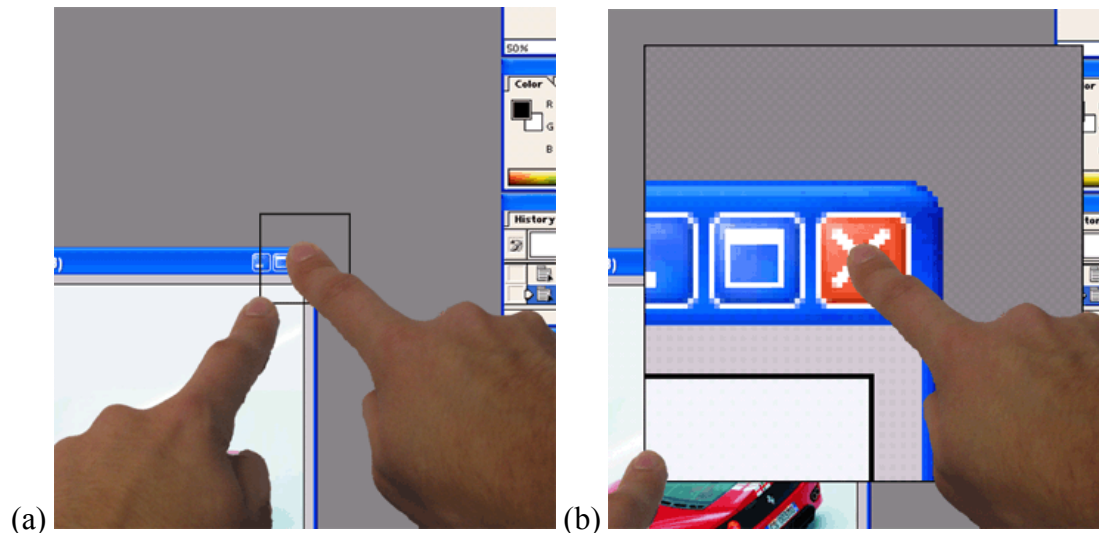
Continuing to move the fingers apart triggers a snap that warps the cursor back to the primary finger's location. Just as in the Dual Finger X-Menu, the snap is signaled by a distinct sound effect. To manipulate the cursor speed, the user moves the secondary finger along a user-approximated line that passes through both fingers. The distance that the secondary finger traverses between modes is predefined and does not depend on the distance between the fingers. The modes are remembered even after the user lifts the secondary finger, which allows for clutching in the interaction.

### 6.3.6 Dual Finger Stretch

Inspired by the strong performance of the single-touch *ZoomPointing* technique [Albinsson 2003], we designed a *Dual Finger Stretch* technique that allows the user to adaptively scale a portion of the screen with the secondary finger while the primary finger performs the selection. To allow for simultaneous “stretching” and selection, the primary finger provides the initial anchor location around which the user interface is scaled, while the secondary finger identifies the corner of the square area that will be scaled. By moving the secondary finger closer or further away from the primary finger, the square stretching area is reduced or expanded, as illustrated in Figure 6.8.

Lifting the secondary finger from the table resets the interface to its default unstretched state. Upon reset, the cursor is offset with respect to the primary finger and is placed where it was located in the stretched state. The cursor offset is reset when all fingers are removed from the table. The extent of control-display ratio manipulation depends on two physical limits: the closest perceptible distance between the user's fingers and the largest diagonal of the screen. For most common mid-screen manipulations, Dual Finger Stretch enables control-display ratios up to about 10. By allowing clutching

and repeated zooming, it may be possible to further increase this ratio, but we have not implemented this extension.



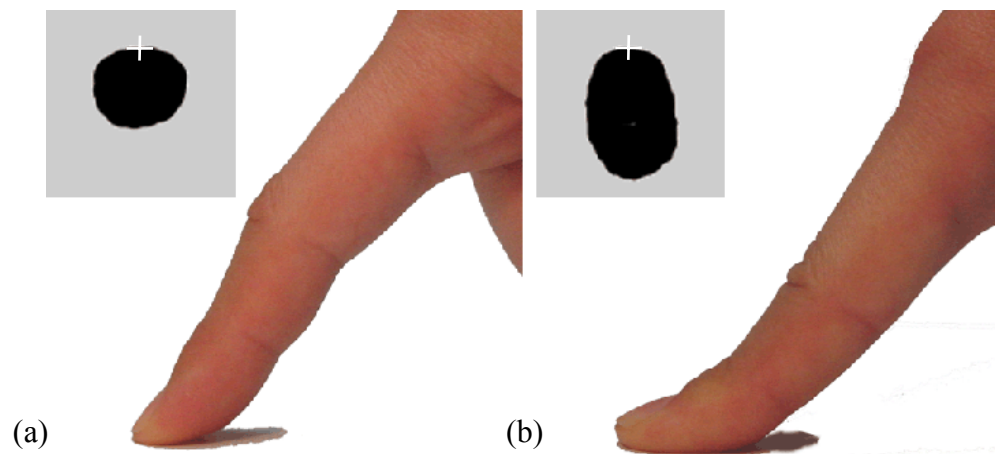
**Figure 6.8: Dual Finger Stretch technique adaptively scales the user interface. (a) The secondary finger specifies the square zooming area centered at the primary finger's location, (b) The primary finger performs precise selection while, simultaneously, the secondary finger adjusts the level of magnification.**

*Dual Finger Stretch* has several advantages over *ZoomPointing*, primarily due to the dual finger design. First, zooming and selection are not decoupled into two separate actions. Instead, they can occur concurrently, which results in a fluid interaction. Second, the interface scales in all directions from the original primary finger's location. This provides an important advantage over traditional rectangle selection, where the two points specify the diagonal corners of the zooming rectangle (also known as *bounding box zoom*). With rectangle selection, the user tends to place the primary finger off target in order to “capture” the target in the zoomed area, while with *Dual Finger Stretch*, the user places the primary finger directly on target and the interface scales underneath in all directions. Placing the finger off-target requires the user's primary finger to traverse an increased distance to perform final selection because the target will appear to move away from the finger as the zoom level increases. By encouraging placement of the primary finger as close to the target as possible, the eventual distance that this finger will need to traverse to acquire the target is minimized.

## 6.4 Clicking Stabilization Technique

As described in Section 6.1, one of the sources of errors in touch screen interaction is *clicking noise*. This is caused by the movement of the cursor during clicking, due to the slight position change of the finger on the touch screen. This clicking noise is particularly present in devices that approximate pressure via touch area (such as the touch screens used in this dissertation). On those devices, there is a need to provide significantly different areas in order to remove any spurious clicking, which in turn adds a substantial amount of cursor movement during clicking.

We introduce a simple and effective area-based gesture clicking mechanism, called *SimPress (Simulated Pressure)*, which requires the user to apply a small rocking motion with their finger in order to perform a “click”, as seen in Figure 6.9.



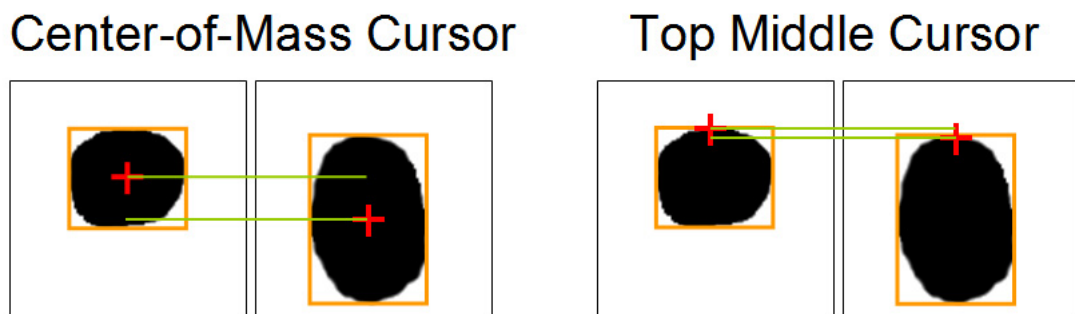
**Figure 6.9:** A small rocking motion of the user’s finger triggers the SimPress clicking technique: (a) tracking (hover) state, (b) dragging (click) state. (The inset images show the actual area of contact detected by our device as a black blob, as well as the stabilized cursor location as a white cross.)

The crucial component of this technique is the stabilization of the cursor movement during clicking, which is achieved by anchoring the cursor not to the center of mass, but rather to the top middle location (Figure 6.10). When using SimPress, the user starts pointing with their finger tip (small contact area) and then just rocks the finger to click. Consequently, the increase in touch area happens predominantly in one direction: from the tip point towards the user’s wrist. We used this fact to stabilize the cursor position

by fixing the cursor location to the top middle point of the contact area, rather than the center of mass. In our preliminary experiments, we found that this point naturally moves much less than the center point, and therefore reduces cursor noise during clicking.

The obvious constraint of SimPress is that this stabilization only works if the user always approaches the tabletop from the roughly same direction; otherwise the orientation of the hand and arm has to be taken into account. A future improvement can potentially use the orientation of the click itself to track the orientation of the user’s hand. However, given that in our experiments (Section 6.6), the orientation of the user interface was fixed, our users tended to orient themselves straight-ahead.

In addition to location stabilization, a “high-low” area threshold approach was established in order to disable spurious switching between the clicking states due to noise or hand tremor. This required users to cross the high threshold to activate the click-on state, and the low threshold to return back to the click-off state. Due to the fingerprint size differences, these high and low thresholds should be automatically recalibrated for each person. Currently, the calibration is done manually. In our Dual Finger Selection techniques, all click events are always triggered by the primary finger using SimPress.



**Figure 6.10:** The difference in the amount of cursor movement when the cursor is attached to the center of mass or to the top middle location of the user’s touch. The axis-aligned touch bounding boxes are given for reference.

## 6.5 Prototype Multi-Touch Tabletop Device

Our interaction techniques are designed for present and future multi-touch screens, and to some extent, they are device-independent. However, we have developed them on

a prototype multi-touch tabletop display to facilitate research and user studies (Figure 6.11). This prototype tabletop differs from the other touch-screen device used in this dissertation (MERL DiamondTouch). It has been designed at Microsoft Research and was used by the author during his internship there between June and August, 2005.



**Figure 6.11: The multi-touch back-projected tabletop display prototype uses an infra-red illuminant and camera to detect contacts through a diffuse surface.**

This prototype uses a diffuse screen coupled with an infrared camera and computer vision algorithm to detect contacts on the tabletop surface. The screen is back-projected with the projector integrated in the base of the table below the screen. The display uses the infra-red light spectrum for contact detection, while all projection is done in the visible spectrum. This separation allows the computer vision algorithms to ignore the projected display in order to see only surface contacts. A similar approach is used in the TouchLight [Wilson 2004] display system and elsewhere [Han 2005, Wilson 2005a].

The screen resolution of our prototype multi-touch device is  $1024 \times 768$  (pixels), which, given the screen dimensions of  $61 \times 46$  (cm), yields a pixel size of 0.6mm. A finger that is about 1.5 cm wide covers about 25 screen pixels. (The relationship between finger size and the size of some standard small interface targets is presented in Table 6.1.)

## **6.6 User Experiment**

To evaluate Dual Finger Selection techniques and the underlying SimPress technique, we conducted two user studies that challenged participants to select a range of small targets using various techniques. Additionally, we were interested in how well these techniques perform on devices of very low precision. Such devices include touch screens based on a small number of sensing elements, touch screens based on noisy computer vision processes, and many prototype research systems that do not achieve the precision of the mouse (e.g. see [Wilson 2004, Wilson 2005a]). Accordingly, in our experiments we added synthetic noise to the sensing system described above, and systematically varied its magnitude.

### **6.6.1 Participants**

Twelve paid participants (9 male and 3 female), ages 20–40, participated in the experiment. All participants were frequent computer users. They had varying experience with touch screens, ranging from “monthly” use to “several times a day” use, with the average corresponding to “weekly” use. All participants used their right hand as their dominant hand. Eleven participants identified themselves as right-handed, but even the single self-identified left-handed participant preferred using their right hand for mouse operation and chose to use their right hand as the dominant pointing hand in our experiments. The participants were all Microsoft employees, pre-screened for color blindness, and they did not receive compensation for their time.

### **6.6.2 Task**

The participants were asked to perform a simple reciprocal target selection task, with square targets of varying widths, separated by a fixed distance of 100 pixels. This task is loosely based on the Fitts’ Law target acquisition task [MacKenzie 1992], but without the variation of distance. The task involved clicking on a green square target that was surrounded by a green circle. The other (inactive) target was colored red and the targets alternated between trials. The participants were instructed to click on the cur-

rent green target as fast and as accurately as possible. We recorded both movement times and error rates, but we analyzed completion times only for successfully completed trials. We were more interested in the impact of our techniques on the reduction of error rate, than the completion times, because we had hypothesized that the smallest target sizes might not even be reliably selectable by all the techniques tested.

The experiment consisted of two parts: an evaluation of the SimPress technique and a comparative evaluation of the four Dual Finger Selection techniques under varying amounts of noise. Both used the same testing infrastructure to present targets to the participant, measure user performance and log all experimental data. In addition, the participants completed a post-experiment user preference questionnaire.

### 6.6.3 Part 1: Evaluation of SimPress Clicking

We wanted to determine the performance of the SimPress clicking technique to obtain a baseline measure of the minimal target width that is possible to select reliably without additional assistance. An additional motivation was to ensure that our participants mastered and were comfortable using SimPress, since we required them to use it throughout our later experiments. Our participants were first given an introduction to the SimPress technique and then allowed to perform one practice block before actually performing this experiment.

#### 6.6.3.1 Method

A within-subjects, repeated measures design was used, consisting of five target widths (1, 2, 4, 8, and 16 pixels). The widths were chosen to represent the range of smallest available targets in a typical GUI. For example, the smaller toolbar buttons tend to be between 15 and 20 pixels wide, while the resize handles are sometimes less than 5 pixels wide. The experiment consisted of five sets (one set per width) of five trials each. The order of the sets was randomized across participants.

In summary, this experiment consisted of:

1 technique ×  
5 target widths ×

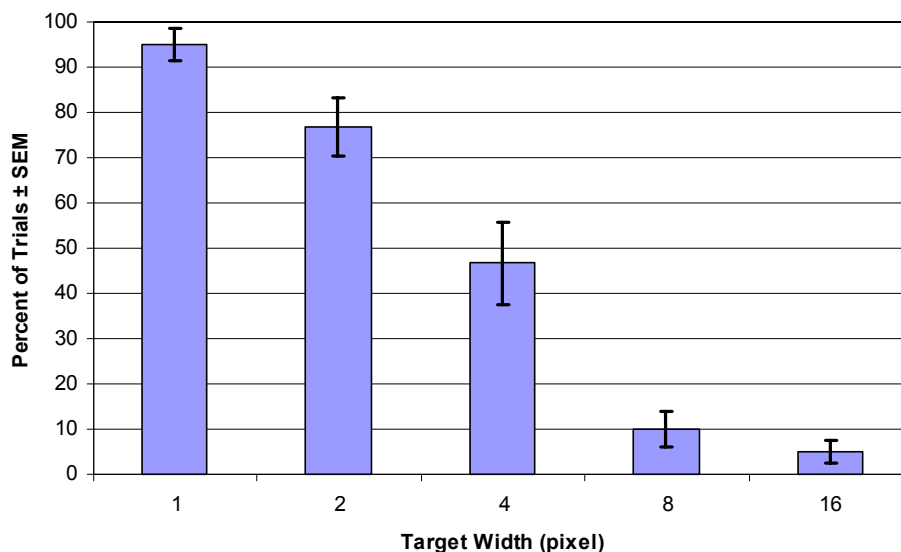
5 identical trials  
= 25 trials per participant

### 6.6.3.2 Hypothesis

Our hypothesis was that participants would be able to reliably select only the largest of our small targets (16 pixels) and that the finger's occlusion of the target and the small amount of noise still present while clicking would make the selection of other target sizes difficult.

### 6.6.3.3 Results

We performed a repeated measures ANOVA on the mean error rate data and found a significant main effect (using an  $\alpha$ -level of 5%) with the target widths ( $F_{(4,44)} = 62.598$ ,  $p < 0.001$ ). The data is summarized in Figure 6.12. The paired samples t-tests show no significant differences between the participant's performance with 8 and 16 pixel targets. A significant difference is shown between 2 and 4 pixel targets ( $t_{(11)} = 3.95$ ,  $p = 0.002$ ), 4 and 8 pixel targets ( $t_{(11)} = 4.16$ ,  $p = 0.002$ ), and 1 and 2 pixels targets ( $t_{(11)} = 2.41$ ,  $p = 0.034$ ).



**Figure 6.12: Mean error rate (%) using the SimPress technique alone without assistance ( $F_{(4,44)} = 62.598$ ,  $p < 0.001$ ).**



Contrary to our hypothesis, we found that the threshold target size, below which the SimPress technique is not reliable by itself, is around 8 pixels, which is the smallest target size with the error rate below 10%. These results show that SimPress is a viable option for use for many general selection tasks in the current user interface.

#### 6.6.4 Part 2: Comparison of Four Dual Finger Selection Techniques

For the second part of the experiment, we tested the performance of Dual Finger Selection techniques by comparing the three most versatile techniques (Dual Finger Stretch, Dual Finger X-Menu, and Dual Finger Slider) and the Dual Finger Offset technique. By providing no additional assistance other than the cursor offset, the Offset technique served as a baseline. Even though the Midpoint technique received very positive response from our initial testers, this technique was not included due to the relatively small assistance that it offered in selection (the maximum reduction of the cursor speed was a factor of two).

Additionally, we were interested in how our techniques would perform in the presence of noisy input. We note that many touch-screen devices provide noisier input than standard relative pointing devices, such as a mouse. This is particularly true of the entire class of touch-screen devices that depend on the analysis of a video signal for their touch recognition. In addition to noise in the video stream, such devices often require that the video signal is up-sampled to match the screen's resolution, which makes one video pixel cover several screen pixels. This up-sampling can produce undesired cursor jumping effects if the cursor position is not smoothly interpolated.

While the noise can be reduced with the use of a filter (the Kalman filter being the most commonly used [Kalman 1960]), this solution introduces a permanent lag in all of the interactions. We believe that there is a benefit to having interaction techniques that adaptively allow the user to reduce the noise when so desired. For example, the user could prefer to have somewhat noisy, but lag-free input for most interaction tasks, and only reduce the noise when precision is desired. By manipulating the control/display ratio, Stretch, X-Menu, and Slider implicitly allowed the reduction of the input noise as well.

In order to test how our techniques performed in noisy conditions, we added Gaussian noise to the position of each tracked finger, creating three noise levels: low (no additional noise), medium (Gaussian noise with  $\sigma=0.5$ ), and high (Gaussian noise with  $\sigma=2$ ).

#### 6.6.4.1 *Method*

Our study followed a within-subjects, repeated measures design that tested three noise levels, four techniques, and four target widths (1, 2, 4, and 8 pixels) per block. Within each block, the participant performed six trials. To eliminate the effects of switching selection strategies (e.g., deciding to use a different cursor speed reduction for a different target size), we discarded the first trial in each block. All our techniques were modified to completely reset after each click, in order to ensure the same starting state for all trials. The testing order of techniques was randomized across participants, to eliminate the effects of ordering.

In summary, this experiment consisted of:

- 4 techniques ×
- 3 noise levels ×
- 4 target widths ×
- 6 identical trials
- = 288 trials per participant

#### 6.6.4.2 *Hypotheses*

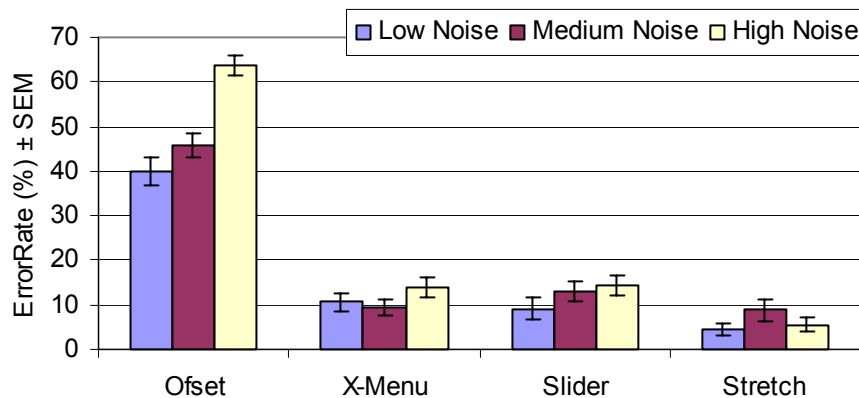
Prior to our experiment, we postulated the following two hypotheses:

**H1:** Stretch, X-Menu and Slider should be less affected by the increase in noise than the Offset technique. We hoped to show that techniques that increase the control/display ratio lessen the impact of the input noise.

**H2:** Slider would perform better than X-Menu, since Slider is controlled by the natural bi-manual way of specifying spatial extent (finger distance), rather than the independent finger actions in X-Menu.

### 6.6.4.3 Results

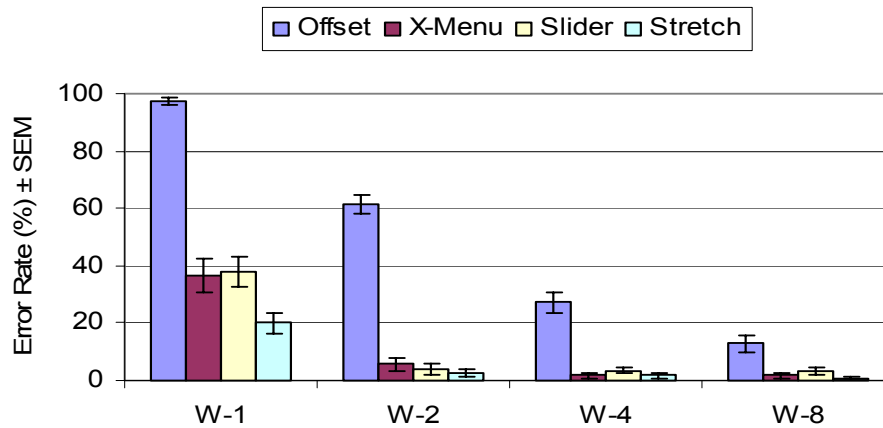
We performed a 3 (Noise)  $\times$  4 (Technique)  $\times$  4 (Width) repeated measures ANOVA on the mean error rate data and found the significant main effects across all conditions (using an  $\alpha$ -level of 5%). As expected, noise had a significant main effect on the error rate ( $F_{(2,22)}=20.24$ ,  $p<0.001$ ). This confirmed that more errors were committed in the higher noise levels. Significant main effects were also present for width ( $F_{(3,33)}=150.4$ ,  $p<0.001$ ) and technique ( $F_{(3,33)}=169.138$ ,  $p<0.001$ ). The paired samples t-tests show that the Offset technique created significantly more errors than the rest ( $t_{(11)}=14.298$ ,  $p<0.001$ ), while Stretch was better than X-Menu or Slider ( $t_{(11)}=2.864$ ,  $p=0.015$ ). We found no statistical differences in the error rate between X-Menu and Slider.



**Figure 6.13: Interaction of Noise  $\times$  Technique for error rate (%) ( $F_{(6,66)}=8.025$ ,  $p<0.001$ ).**

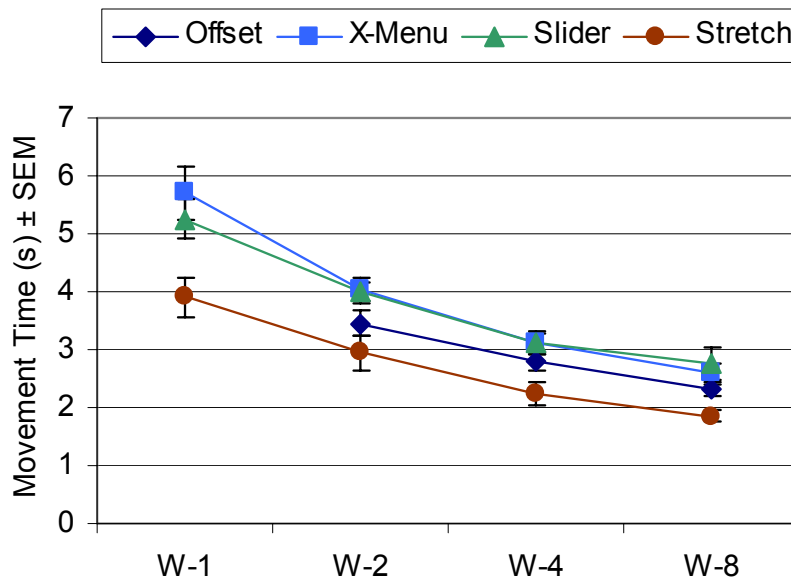
The interaction of technique and width ( $F_{(9,99)}=29.473$ ,  $p<0.001$ , Figure 6.13) is interesting, as it shows that our assistive techniques (Slider, X-Menu, and Stretch) all performed exceptionally well (less than 5% error rate) in all noise conditions for targets 2 pixels or larger (no statistical differences between techniques). For the smallest target (1 pixel), Stretch outperformed X-Menu and Slider (with borderline significance  $t_{(11)}=2.64$ ,  $p=0.023$ ). The interaction of noise and technique was also significant ( $F_{(6,66)}=8.025$ ,  $p<0.001$ , Figure 6.14). While the increase of noise greatly degraded performance of the Offset technique, the other three techniques show no statistically significant effects for the various noise levels. This confirmed our main hypothesis (H1) that users are able to

lessen the impact of noise and low precision by using techniques that allow for control-display ratio adjustments.



**Figure 6.14: Interaction of Technique × Width for error rate (%) ( $F_{(9,99)}=29.473$ ,  $p<0.001$ ).**

Due to the dramatically high number of errors committed by our participants using the Offset technique, our data contains several blocks without a single successful trial (all in the 1 pixel width condition). While this prevented us from performing a repeated measures ANOVA on movement times, we present an informal analysis of median movement time values for blocks for which we have data.

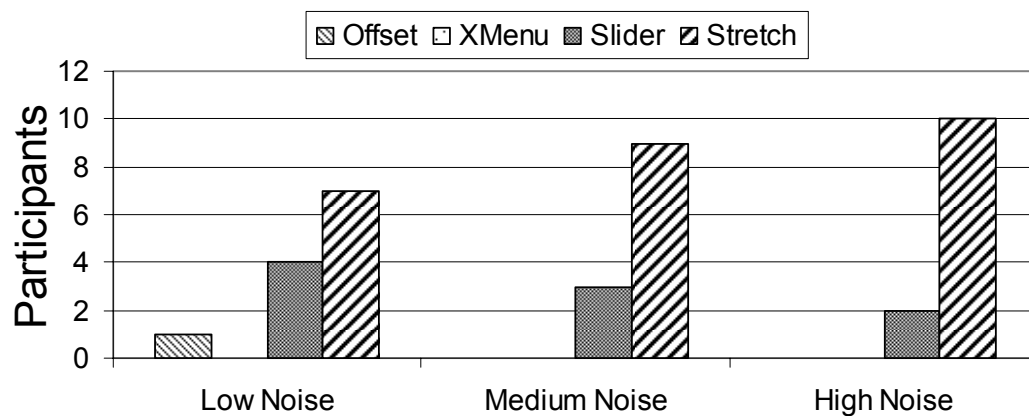


**Figure 6.15: Mean performance time (s) with respect to target widths. (Notice: time data is not shown for Offset technique at 1 pixel due to lack of successfully completed trials.)**

Median times were chosen to correct for the typical skewing due to reaction time data. This also removed the influence of any outliers in the data. Aggregating the movement times across all noise levels and all target widths, the Stretch technique is on average 1s faster than Slider ( $t_{(11)}=5.011$ ,  $p<0.001$ ). There do not appear to be statistical differences in the performance times of Slider and X-Menu. This failed to confirm our second hypothesis (H2) that Slider would outperform X-Menu. Figure 6.15 shows the performance of techniques with respect to target width. The data show a general trend of more time being spent on targeting smaller targets.

### 6.6.5 Subjective Evaluation

The users filled out a post-experiment questionnaire rating their experience with four techniques on a 5 point Likert scale (1 being most negative and 5 being most positive) They were asked to comment on the following categories: mental effort, learning time, hand fatigue, enjoyment, and performance in low, medium and high noise conditions. Overall, dual finger selection techniques received significantly different results ( $F_{(3,33)}=45.9$ ,  $p<0.001$ ). X-Menu required the most mental effort (average score of 2.88), and the longest time to learn (average score of 2.09). Data shows no significant statistical differences between techniques with respect to hand fatigue. Stretch was the most enjoyable (average score of 4.12), followed closely by the Slider technique (average score of 4.08).



**Figure 6.16: Subjective preferences for the best technique across different noise levels.**

We also asked participants to rate their overall preference for which technique they would favor for selecting targets in various noise conditions, as well as state their overall technique preference. Overall, Stretch was the most preferred (7 out of 12 participants), followed by Slider (4 out of 12 participants), while only one participant preferred X-Menu. It is interesting to note that the preference for Stretch grew with the increase of input noise: 10 out of 12 participants preferred it in the high noise condition (Figure 6.16).

## **6.7 Discussion**

Of the four techniques compared, the top performer and the most preferred technique, Stretch, was the only one that did not provide any cursor offset. This clearly demonstrated that the benefit of increased target size successfully compensated for the fingertip occlusion factor. The data from this experiment is consistent with the results from a study by Albinsson and Zhai [Albinsson 2003], which also showed that their baseline zooming technique outperformed on-screen widgets that provided cursor speed control.

We feel that Stretch is a simple and powerful interaction that utilizes the distance between fingers in a very natural way. However, in many applications, scaling may have an undesired effect of losing the overview of the interface. Therefore, we were very pleased with the strong performance of Slider and X-Menu, which provided comparable error rates with a small time penalty of about 1s when compared to Stretch. In addition, as our participants' written comments point out, those techniques strongly benefit from the ability to freeze the cursor. As one participant described it, freezing the cursor is a functional equivalent to presenting a user-controlled "are you sure?" dialogue for clicking which enables the participant to unmistakably select a particular point or go back and re-target. This was particularly useful with higher noise levels.

Experience should also substantially improve our Slider and X-Menu because zooming is a very familiar interaction for most users. As such, it might have an unfair advantage when compared with the novel cursor speed controlling techniques.

The SimPress clicking technique exceeded our performance expectations. This enables novice users to reliably click on targets that are as small as 8 pixels. We believe that with practice and more thorough calibration, this threshold could be further reduced. Some future work on stabilization is needed in order to completely remove the remaining noise from clicking.

An additional SimPress modification was implemented, but not tested, permitting the user to rapidly click on targets without requiring the rocking motion. This timer solution generates a click event if the user's touch contact was detected on the surface for less than 0.4 s. This allowed the simple selection of large targets to remain as fast as possible, while more complex interactions, such as drag and drop, can be performed using the SimPress technique.

Our study results show that Dual Finger Selection techniques present viable solutions for increasing precision and accuracy in small target selection tasks. These techniques are designed to be used on most multi-touch screens, perform well with the increase of input noise, and fully utilize the benefits of dual finger interactions. Overall, these techniques provide a palette of possible interactions, from which the user can choose, depending on the specific target application.

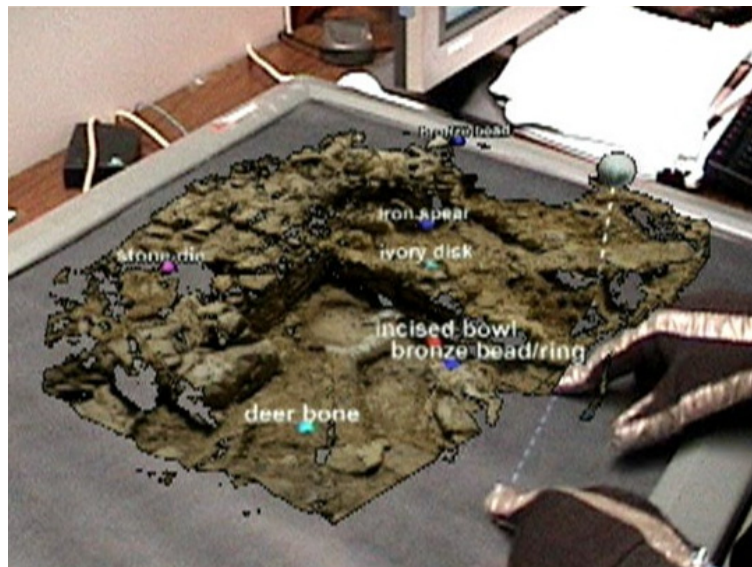
Following the initial publication of this work, several research groups extended these ideas to further improve multi-touch-based cursor control. Moscovich and Hughes developed three multi-touch techniques for cursor control [Moscovich 2006]. Their techniques focus on area cursors that are adaptively controlled by the user's touches. Recently, Iten and Lüthi presented the Digital Tweezers technique [Iten 2007], which is a close derivative of our Dual Finger Midpoint technique (Section 6.3.3).





## 7 Improving the Precision of 3D Selection in Hybrid MDEs

This chapter presents a novel hybrid MDE interaction technique, called *Balloon Selection* (Figure 7.1), which allows for precise and accurate 3D selections in a constrained within-reach 3D environment. Balloon Selection is a hybrid interaction that combines the affordances of two heterogeneous displays: a tracked stereoscopic head-worn display and a multi-touch sensitive tabletop display.



**Figure 7.1:** Balloon Selection being used to select an object of interest in the world-in-miniature model within the VITA system. Note that the objects representing the excavated finds tend to be extremely small and closely grouped together, making their selection difficult at this scale.

Balloon Selection is directly inspired by our development of *Cross-Dimensional Gestures* (Chapter 5) and *Dual Finger Selection* techniques (Chapter 6). By building on

top of the strong performance of Dual Finger Selection techniques and applying such multi-touch interactions across the dimensionality border, we have created a hybrid MDE interaction that significantly improves the precision of selections in the 3D space above the surface.

In the remainder of this chapter, we first provide an overview of issues with precise 3D selections as well as a brief review of related techniques. Then, we provide a detailed description of our technique. Balloon Selection has been evaluated in a user study that compares our technique with two well-known interaction techniques for selecting a static 3D target: a 3DOF tracked wand (chosen for its intuitive and direct mapping to the task) and keyboard cursor keys (chosen for accuracy and insensitivity to hand tremor and arm fatigue). We conclude with a discussion of our results and some potential future applications of this interaction.

## **7.1 Selection Issues in Scaled-Down 3D Models**

Balloon Selection is designed for hybrid MDE scenarios in which the user wishes to interact with a within-reach very small scale 3D model, such as a model of the city, university campus, or an archaeological dig site [Hua 2003, Benko 2004], seen from an outside-in (exocentric) view. Such models, also known as 3D maps or world-in-miniature (WIM) models [Stoakley 1995], are particularly well suited for collaborative hybrid MDE scenarios in which multiple users discuss and analyze a given virtual model over an augmented tabletop. In fact, many existing hybrid MDE implementations consider such miniaturized models the focus of the entire collaborative environment. Examples of such scaled models in hybrid MDEs include the car engine parts in MagicMeeting [Regenbrecht 2002], the sheep pasture terrain model in the Herding Sheep demo [MacWilliams 2003], and the archaeological excavation terrain model in our own VITA system (Figure 4.8b). Unfortunately, performing a selection of an object of interest in such scaled-down virtual models is difficult, because such targets tend to be small, closely grouped, and even potentially overlapping.

We designed Balloon Selection to fully exploit the benefits of a firm touch-sensitive surface to assist the user in accurate 3D selections. First, the surface is able to provide

passive haptic support for the user's hands, which would otherwise need to be held continuously in mid air. By having the user rest their hands on the surface, we improved selection accuracy by significantly reducing hand tremor and arm fatigue. Second, when the user's hands are on the surface, they tend to remain below the 3D model and thus do not obscure the 3D objects being manipulated. Third, we took advantage of the multi-touch surface by designing a technique that decomposes the 3DOF selection task into a 2DOF positioning task performed by one finger on the tabletop in an absolute 2D Cartesian coordinate system and a 1DOF positioning task performed by another finger on the tabletop in a relative 2D polar coordinate system.

### 7.1.1 Adhering to the 3D Interaction Design Guidelines

During the design of Balloon Selection, we were guided by the comprehensive guidelines compiled by Hinckley and colleagues [Hinckley 1994]. They recommend the use of spatial references (props or the user's own body), two-handed interaction, multi-sensory feedback, and physical constraints for helping users to perceive and interact with a 3D object and environment. They also strongly recommend the reduction of extraneous degrees-of-freedom (DOF) to simplify the 3D task when possible. For example, in tasks requiring translation, but not rotation, rotation should be disabled. Furthermore, they suggest that providing a clear control metaphor (e.g., *eyeball-in-hand* [Badler 1986, Brooks 1986] camera or *ray-casting* [Vickers 1972]) significantly improves the effectiveness of the interface and enhances the ability of the user to perceive the task at hand.

## 7.2 Related 3D Selection Interactions

Research on selection techniques in 3D immersive environments can be divided into two broad categories. First, the ray-based selection techniques, and their aperture-based superset, use a projected ray, either visible or not, from the user's hand or head to select targets that the ray intersects [Vickers 1972, Bolt 1980, Forsberg 1996, Mine 1997, Olwal 2003a]. Second, virtual hand techniques, or 3D cursors, provide a direct or offset

mapping between the hand position and the location of a 3D cursor [Vickers 1972, Poupyrev 1996, Mine 1997, Frees 2005]. For example, thirty-five years ago, Vickers [Vickers 1972] described the use of a hand-held 3D position-tracked wand that could be used to select vertices within a small virtual cube at its tip (3D cursor selection); buttons on the wand allowed the selection geometry to move to an otherwise out-of-reach vertex along the vector defined by the right eye and the tip of the wand (ray-based selection).

While ray-based selection has been shown to achieve faster selection times than the use of a 3D cursor for general VR tasks [Bowman 1999], it is not as well suited to selecting targets that are small, closely grouped together, or partially or completely occluded, such as the ones found in many WIM models. It is possible to enhance the WIM metaphor by showing just a relevant subset of the environment [Wingrave 2006] and then scrolling or scaling through it; however, this technique tends to lose the context provided by the complete WIM, which is one of the benefits of using a scaled model.

Several researchers have tried to address the problems posed by nearby and occluding objects in ray-based selection. Hinckley et al. [Hinckley 1994] and Bowman and Hodges [Bowman 1997] discuss methods for choosing one of multiple selections along a ray. Work by Olwal et al. on *SenseShapes* [Olwal 2003a] uses multimodal input and the history of interaction with the selection geometry to disambiguate among potential selections, while Wyss et al. [Wyss 2006] present a two-handed technique based on the intersection of two rays. Olwal and Feiner [Olwal 2003b] describe a flexible pointer that allows the user to bend the selection ray to avoid other objects. Recently, the use of flexible and adjustable rays to select 2D widgets in 3D environments has been further explored by Andujar and Argelaguet [Andujar 2006] and de Haan et al. [Haan 2006].

Improvements to 3D cursor techniques usually focus on two categories: extending the user's reach and making the user's virtual hand (or 3D cursor) more accurate. Poupyrev et al. developed *GO-GO* [Poupyrev 1996], which nonlinearly maps the distance between the user's body and their real hand to the distance that the virtual hand moves from the user. Pierce et al. developed the *Voodoo Dolls* technique [Pierce 1999], which brings scaled copies of distant objects within reach for fine manipulation. Frees and

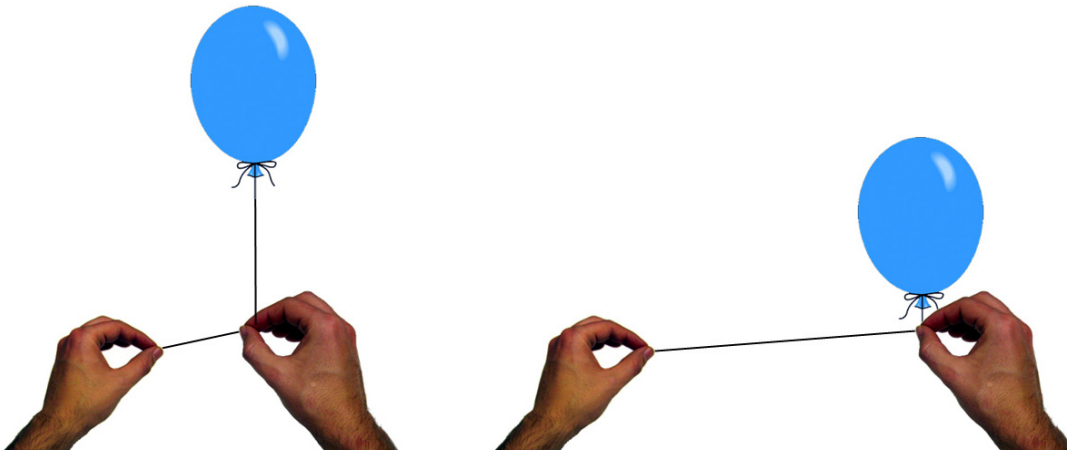
Kessler [Frees 2005] adaptively adjusted the control-display ratios of a 3D cursor to assist in precise position and rotation interactions.

A lot of researchers explored using 2D working planes to manipulate 3D objects (e.g., [Johnson 1964, Schmandt 1983, Herndon 1992]). Our manipulation of the balloon on the projection surface was inspired by Herndon and colleagues' work on *Interactive Shadows* [Herndon 1992], where they manipulate 3D objects by means of their 2D projections on principle planes.

### 7.3 Balloon Selection Technique

Balloon Selection was inspired by how a person might play with a helium balloon on a string. A tethered helium balloon floats straight up from the point where its string is being held. If one holds the string tightly with one hand, with the string passing loosely through the fingers of the other hand and moves the hands relative to each other, the balloon will change in height along a vector passing perpendicularly through the hand closest to the balloon (Figure 7.2). This simple, yet powerful, metaphor demonstrates one way in which a 3DOF positioning task can be effectively decomposed into two separable tasks [Jacob 1992]: a 2DOF positioning task in the ground plane and a 1DOF “string-pulling” task for regulating the balloon height. This basic metaphor also has the interesting feature that the user's hands only need to move in a single plane to control the 3D location of the balloon. This inspired us to use a multi-touch surface to control and select 3D objects above the surface by controlling a balloon-like a tool.

While we consider Balloon Selection to be a 3D cursor technique, it is also possible to analyze it as a modified ray-casting technique in which the ray always originates on the surface, points in a fixed direction (vertical), and has variable finite length (string length). Not being able to change the (always vertical) angle of the ray restricts the volume in which Balloon Selection can operate; however, this constraint eliminates some of the hand tremor difficulties associated with conventional ray-casting techniques in which small angular movement at the origin can result in large physical movement at a distance.



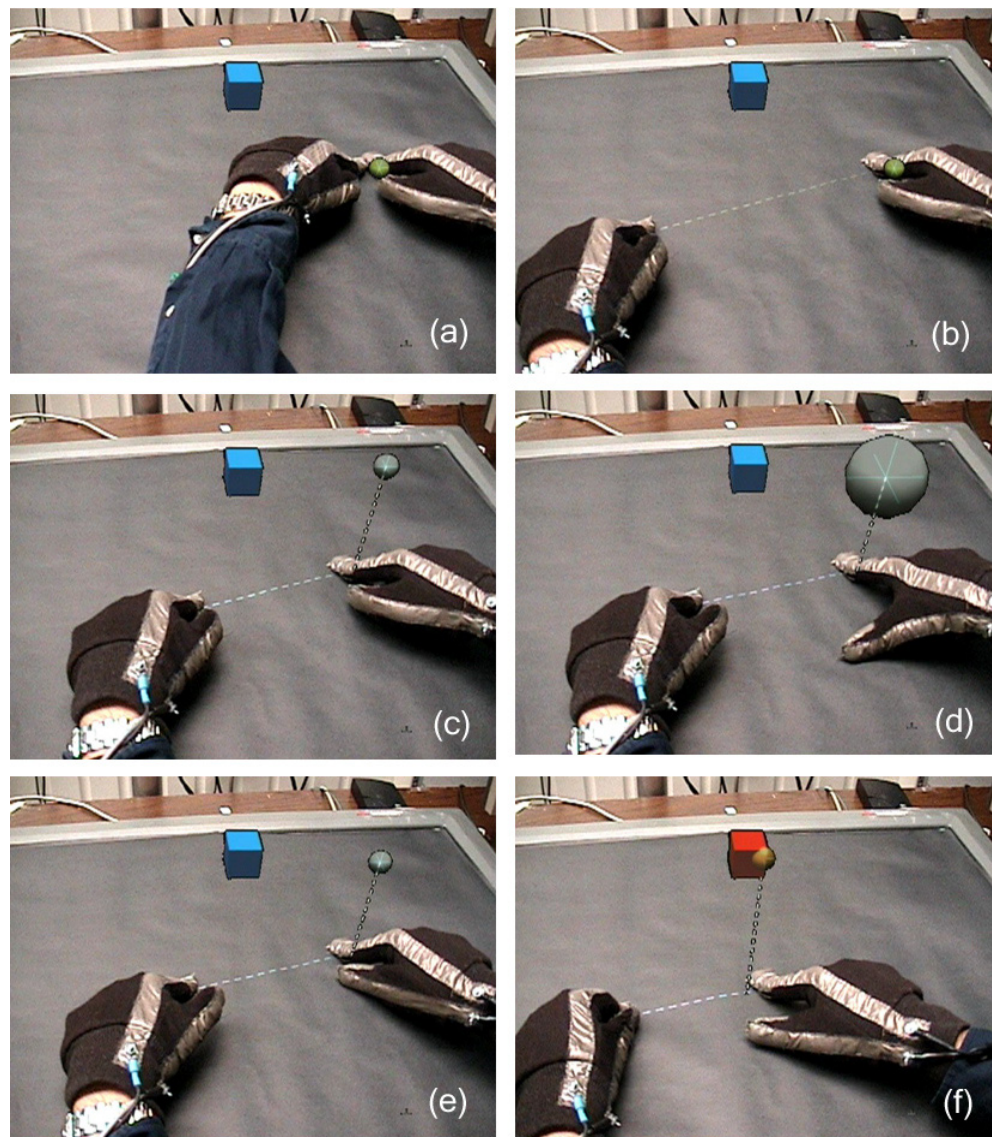
**Figure 7.2: The basic principle of Balloon Selection.**

### 7.3.1 Interaction Description

In our interaction, the physical helium balloon is replaced by a virtual balloon (3D cursor), consisting of three-axis crosshairs within a semi-transparent sphere. The physical string is replaced by a pair of dashed virtual lines. In our implementation, a horizontal line is displayed on the projected tabletop, while the cursor and a vertical line are displayed in the head-worn display; however, both lines could be displayed in the head-worn display.

To instantiate the balloon, the user touches the tabletop with one sensed finger (*anchor*) and places a second sensed finger (*stretching finger*) immediately adjacent to it. This dual-and-adjacent (or *pinch*) finger configuration triggers the creation of a 3D balloon sphere that now resides on the surface at the location of the anchor (Figure 7.3a).

By requiring the user to start the interaction with the two-finger pinch pose, we ensured that Balloon Selection can coexist in the hybrid MDE with many other multi-touch interactions present. For example, the user could use one of the Dual Finger Selection techniques to improve the precision of cursor-based interaction within the 2D screen, by placing two fingers on the surface, and seamlessly switch to using Balloon Selection by placing those same fingers together. In essence, we use the pinch pose as a simple delimiter between other multi-touch interactions and Balloon Selection.



**Figure 7.3: Balloon Selection interaction sequence. (a)** Placing two fingers (the anchor and the stretching finger) adjacent to each other on the tabletop instantiates the 3D cursor (balloon). **(b)** Moving the stretching finger away from the anchor stretches the virtual string between the two fingers. **(c)** Moving the fingers closer together raises the balloon from the surface. **(d–e)** Moving the thumb on the anchor’s hand towards or away from the anchor scales the balloon up (for easier selection) or down (for more precise selection). **(f)** Moving the anchor on the surface translates the balloon parallel to the plane of the table, while varying the distance between the anchor and the stretching finger determines the balloon’s height. Placing the thumb of the stretching finger’s hand adjacent to the stretching finger triggers the selection of the target cube.



By moving the stretching finger away from the anchor, the user “stretches” the virtual string between the fingers (Figure 7.3b). When the user reverses direction, the string’s length becomes fixed and the balloon rises from the surface (Figure 7.3c).

At this point, the user’s anchor finger controls the position of the balloon in the horizontal plane (2DOF) while the stretching finger controls its elevation from the surface (1DOF). It is important to note that only the relative distance between the anchor and the stretching finger matters, and not the absolute location of the stretching finger on the surface, thus making the specification of the height a pure 1DOF task. Moving the two fingers in parallel translates the balloon in a horizontal plane above the surface.

Additional modifications of the balloon are possible by employing a separately sensed third finger. The user can adjust the size of the balloon by changing the distance between the third finger and the anchor, which is used as a scaling factor: moving the fingers apart scales up the balloon, while moving them closer together scales it down (Figure 7.3d–e). To perform a selection, the user again uses the pinch finger configuration, this time between the stretching finger and a third finger, making the equivalent of a “clicking” action (Figure 7.3f).

### 7.3.2 String Height-Clutching

Maintaining a constant distance between the fingers while the fingers are moving can be challenging. This requirement of always needing to hold at least two fingers on the surface can be relaxed once the balloon has been created. In fact, by lifting the stretching finger off the surface, the user can *clutch*, temporarily fixing the height of the balloon. Repositioning the stretching finger on the surface automatically reacquires the string, with the height of the balloon remaining constant, while the horizontal portion of the string is lengthened or shortened to fit between the primary and secondary fingers.

Clutching has two major benefits. First, when further height adjustment is not desired, the user can eliminate this extra DOF (and potential effects of additional hand tremor) by lifting the secondary finger off the surface. Second, by clutching, the string can be extended beyond the maximum length allowed by the surface diagonal. This



makes it possible to extend the balloon's reach far above the surface, and for the total length of the balloon string to be lengthened or shortened, as desired.

While string height clutching and declutching might not be necessary in a two-handed approach, in which the distance between the primary and secondary fingers can be sufficiently great, we find it very beneficial in single-handed operation, where the physical distance between the fingers of a single hand is a serious limiting factor.

### 7.3.3 Implementation Details

We implemented Balloon Selection in a hybrid MDE consisting of a tracked head-worn display (Sony Glasstron LDI-D100B) and a multi-touch-sensitive projected table-top display (MERL DiamondTouch). To receive individual reports for each of the user's two thumbs and index fingers on the DiamondTouch surface, we created gloves with conductive fabric (Shieldex Bremen), inspired by those developed by Butler et al. for their HabilisDraw DT system [Butler 2004]. As shown in Figure 7.4, a separate strip of conductive fabric is run along each thumb and index finger, and each strip is connected by a cable to a separate DiamondTouch input. A more detailed description of our environment can be found in Section 7.4.2.



**Figure 7.4: Instrumented gloves (right hand glove shown) designed by the author that produce an independent report for each of the user's two thumbs and two index fingers on the MERL DiamondTouch surface.**

While our technique requires the use of multiple fingers, the fingers are not assigned a predefined role. Rather, we follow the multi-touch *order-based labeling approach*

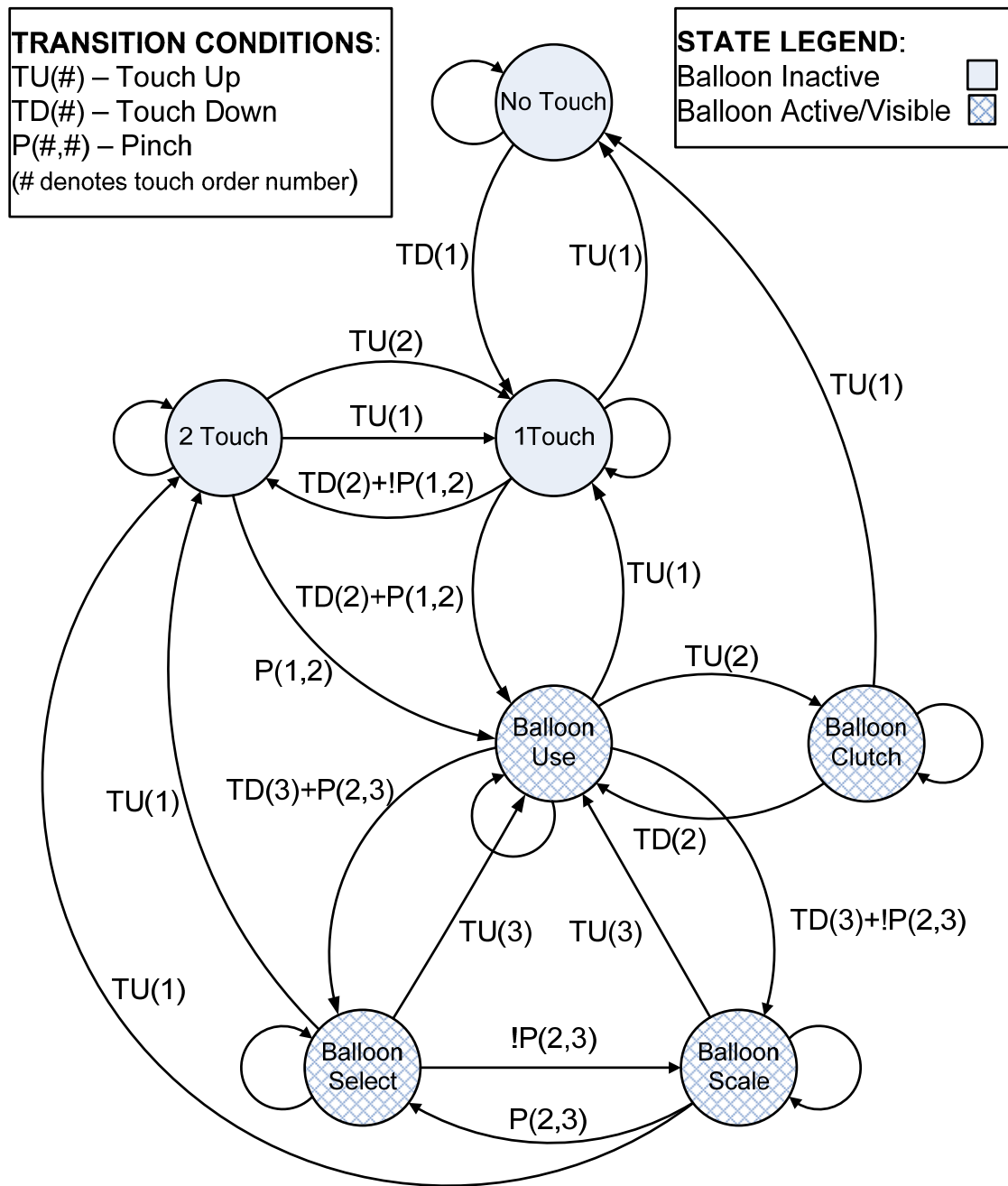
outlined in Section 6.3.1.1, and assign a role to each finger based strictly on the order in which the fingers touch the surface. Thus, rather than requiring the particular assignment of fingers shown in Figure 7.3, we generalize our technique by classifying the first contact with a tabletop surface as the *primary finger (anchor)*, the second contact as the *secondary finger (stretching finger)*, and the third contact as the *tertiary finger (scaling finger or selection finger)*. Therefore, our technique does not depend on the tabletop surface identifying the specific fingers or hands that are currently touching the surface, even though the glove-based sensing technology that we use can provide that information. This approach ensures that Balloon Selection can be used as both a two-handed and a single-handed technique; however, we note here that our current glove design supports only two touches per glove, necessitating the use of two hands/gloves to trigger the selection.

Balloon Selection is implemented using the finite state automaton depicted in Figure 7.5. Four out of seven states are used to control the balloon. The behavior of each state is described in Table 7.1.

#### 7.3.4 Visual and Audio Feedback

In initial testing, we discovered that users had difficulty understanding how to initially stretch the string, since this did not map well to a physical metaphor. Therefore, we decided to add visual and audio cues to attempt to guide the user through successful use of Balloon Selection. Currently, both the string and the balloon change color when the string is being stretched (orange during stretching, white while in use). In addition, when stretching, the user hears a rapid sequence of ratcheting clicks to enhance the physicality of the string/chain that holds the balloon.

Interestingly, despite the mixed metaphors, our test users often highlighted this feature as the key aspect in making the balloon-on-a-string metaphor believable. Additionally, we change the cursor and string color and add a different single clicking sound during selection to provide clear feedback for the triggering action.



**Figure 7.5: State transition diagram for the implementation of Balloon Selection.** Pinch is defined as placing two fingers next to each other on the surface. Therefore  $P(a,b)$  means that fingers  $a$  and  $b$  are *pinched* together, while  $!P(a,b)$  means that those same fingers are apart.

<b>State Name</b>	<b>State Description</b>
No Touch	Default state where no touches are detected on the surface. Balloon is not visible.
1 Touch	The first/only finger touching the table becomes the primary finger (anchor). Balloon is not visible.
2 Touch	The primary finger and the secondary finger are independently touching the table. Balloon is not visible.
Balloon Use	By pinching the primary and secondary fingers together, i.e., $P(1,2)$ , the balloon appears on the surface above the anchor. Moving the stretching finger away from the anchor, stretches (extends) the string. Moving those fingers closer together allows the balloon's height to be adjusted.
Balloon Clutch	Only the primary (anchor) finger remains, and the balloon enters a clutch state where its height is fixed.
Balloon Select	By pinching the secondary and the tertiary fingers together, the user can trigger selection. The selection gets released either by removing the tertiary finger or by releasing the pinch, i.e., $!P(2,3)$ .
Balloon Scale	The radius (scale) of the balloon is determined by the distance between the primary (anchor) finger and the tertiary finger.

**Table 7.1: The description of each state from Figure 7.5. In all balloon states, the position of the primary (anchor) finger determines the  $(x,y)$  position of the balloon.**

## 7.4 User Experiment

We performed a user study to determine how Balloon Selection compared with other selection techniques. Twelve paid participants (10 male and 2 female), ages 20–27, were recruited by mass email to graduate and undergraduate Computer Science students at our university. All participants were frequent computer users, but had little previous experience with VR or AR techniques or technology. Participants were screened through the standard stereoscopic depth perception test (Stereo Fly Test by Stereo Optical Company). Two participants reported that they actively play 3D computer games

(several hours per week). All participants identified themselves as right-handed. Four participants wore corrective contact lenses, while the others reported no vision impairment. While all participants reported using touch-screens regularly (mostly ATM machines), none had experience with multi-touch screens.

### 7.4.1 Baseline Comparison Techniques

To evaluate the performance of Balloon Selection (BALLOON), we decided to compare it to two existing 3D selection techniques for selecting static targets: a direct interaction technique using a 3DOF tracked wireless wand (WAND), and keyboard control of the 3D cursor (KEYBOARD). WAND has the advantage of being relatively fast and using a direct, absolute interaction device that relies on well trained 3D human motor skills. In contrast, we expected that KEYBOARD would provide the most accuracy, eliminating the effect of hand tremor and arm fatigue. All three techniques offered the user the same capabilities: 3DOF positioning of a spherical cursor, 1DOF cursor size (radius) control, and selection triggering. For each technique, all scene objects intersecting the cursor were selected when the trigger was fired.



**Figure 7.6:** Interaction devices used in our experiment (top to bottom): A 3DOF tracked three button wand (Logitech TrackMan Live), a pair of specially constructed gloves with two active fingers on each hand, and the keyboard. The devices are laid out on the DiamondTouch surface.

#### 7.4.1.1 *WAND Technique*

WAND was implemented with a wireless Logitech TrackMan Live device with its trackball removed (Figure 7.6). The 3D cursor tracked the position of the wand in space, by using an Origin Instruments DynaSight optical tracker to track a passive retroreflective target affixed to the wand. Cursor size was controlled by pressing the buttons on the wand (the left and middle buttons made the cursor incrementally smaller and larger, respectively), while selection was performed by pressing on the right button.

#### 7.4.1.2 *KEYBOARD Technique*

KEYBOARD used the number pad on a standard keyboard to provide cursor position and size control. Keys 4 and 6 moved the cursor left and right, keys 2 and 8 moved the cursor in and out, and keys 7 and 1 moved the cursor up and down. The participant could adjust the size of the cursor by pressing 3 (smaller) and 9 (larger), and could select by pressing the *space* key. Positioning was done in the absolute world-coordinate system and was therefore consistent with WAND and BALLOON. Movement was performed at a fixed speed of 4 cm / second when a key was depressed, which we found to offer a good balance between speed and accuracy in our preliminary pilot experiments. Our implementation supported moving in more than one dimension simultaneously; however, we observed that participants rarely took advantage of this.

### 7.4.2 Setup

The experiment was performed on a single computer (dual Xeon 3.0 GHz PC, 2GB RAM) running Windows XP Pro, with two NVIDIA Quadro FX 4500 graphics cards. One display output was connected to a Proxima Ultralight x350 DLP projector (1024 × 768 resolution), which projected onto a Mitsubishi Electric Research Laboratory DiamondTouch table [Dietz 2001], a multi-user, multi-touch interaction surface. The other display output was connected to a Sony LDI-D100B stereo, optical see-through, head-worn display (800 × 600 resolution). The MDE environment is based on the VITA communication framework.

Head tracking was performed by a ceiling-mounted InterSense IS900 6DOF tracker, while the wand's position was tracked by a Origin Instruments DynaSight optical tracker mounted 60 cm from the surface. This setup was deliberately chosen to maximize the measurement accuracy of the wand and therefore remove any bias against this technique. The DynaSight documentation reports that the optimal measurement resolution of the DynaSight tracker at about 40 cm is 0.05 mm in three axes. While we did not attempt to verify this numeric claim, we did determine that the rendered cursor did not move when the wand was held stationary. This sub-pixel resolution was deemed more than adequate given that the smallest target in our test was 4 mm wide and projected to multiple pixels. (Note that because we were selecting virtual, rather than real targets, we were concerned more with tracking resolution and lack of jitter than absolute accuracy.)

The DiamondTouch surface reports an interpolated touch resolution of  $2032 \times 1520$  touch pixels. The physical dimensions of the DiamondTouch surface are 64 cm  $\times$  48 cm, which translates into a measurement resolution of about 0.3 mm in each dimension. Again, this was deemed more than satisfactory for our targets of 4 mm width and up.

### 7.4.3 Method

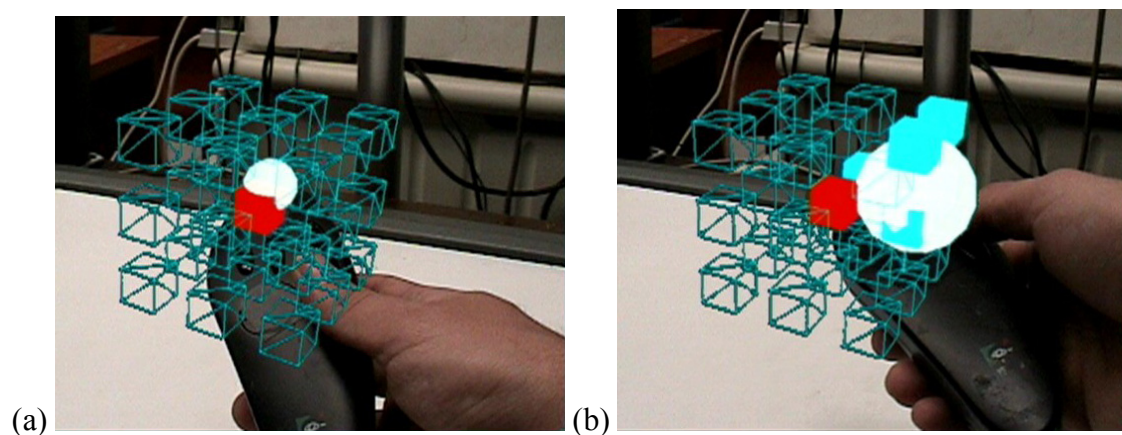
A within-subjects repeated measures design was used consisting of three techniques (WAND, BALLOON, and KEYBOARD) and four target sizes (cubes with edge lengths of 10 mm, 8 mm, 6 mm, and 4 mm). Each participant performed three blocks of 40 trials (one block per technique). Each block consisted of 10 trials for each of the four target sizes, and the order in which the blocks were presented was counterbalanced across participants. The target sizes were presented in decreasing order (starting with the largest target) within each block.

In summary, this experiment consisted of:

3 techniques  $\times$   
4 target sizes  $\times$   
10 identical trials  
= 120 trials per participant

#### 7.4.4 Task

Participants were asked to perform a target acquisition task. The experiment environment consisted of a cubic array of  $3 \times 3 \times 3$  wireframe cubes, of which only the center (red) cube was the actual goal target and the other 26 (cyan) cubes were distracter targets (Figure 7.7). Cubes were rendered solid when they intersected the balloon. To successfully complete the trial, participants needed to select just the goal target, without selecting any of the distracter targets. Not selecting the goal target or selecting the goal target and one or more distracter targets together was marked as an error trial.



**Figure 7.7:** The “cube-of-cubes” target setup being used with a WAND technique (3D cursor is a semi-transparent white sphere). (a) The successful selection of a goal target (red middle target). (b) The unsuccessful selection of more than one target by a larger 3D cursor.

Simple audio cues were used to signal to the participant whether or not their selection was successful. The sounds were chosen from the standard Windows XP sounds, with “ding.wav” signaling success and “chord.wav” signaling failure. This was done to ensure that the participants remained motivated to perform correct selections as the trials continued.

#### 7.4.5 Procedure

Our experiment tested four target sizes (cubes with edge lengths of 10 mm, 8 mm, 6 mm, and 4 mm) that approximately mapped to 25, 20, 15, and 10 pixels per edge re-



spectively in the user's head-worn display (assuming an edge was parallel to the view-plane, and assuming the average virtual distance of the edge from the user of 70 cm). The experiment lasted approximately 45 minutes, and was divided into three blocks (one per each technique), with short breaks between blocks. Each block began with a short practice period in which the participant was taught a technique and given a set of practice trials in which to learn and experiment with that technique.

When the participant felt comfortable with a technique, they proceeded to complete the test of 40 target selections (10 per target size). The targets were presented in order of decreasing size, where the target size and the distance between targets were reduced by 2 mm in each dimension after each set of ten selections. The position and orientation of targets was randomly changed for every trial, but all targets were positioned between 10 to 30 cm above the tabletop, and never more than 10 cm apart. This design resulted in 120 different recorded trials per participant.

During the experiment, we collected measurements for three dependent variables: completion time (seconds), error rate (percentage), and cursor radius (millimeters).

#### 7.4.6 Hypotheses

Prior to our experiment, we postulated the following four hypotheses:

**H1:** WAND would be the fastest technique because of its use of a direct interaction device for integrated, continuous 3D position control in an absolute 3D coordinate system, modeled on the familiar task of directly touching a 3D point.

**H2:** KEYBOARD would be the slowest technique because of its use of keys to separately control velocity separately along each of the three axes.

**H3:** BALLOON would not be as affected by hand tremor and arm fatigue as WAND, and therefore would have a lower error rate than WAND.

**H4:** KEYBOARD would have the lowest error rate, since tracking precision, hand tremor, and arm fatigue would not affect it at all.

Overall, we were hoping to show that BALLOON would perform almost as fast as WAND, but have substantially lower error rate due to reductions in hand tremor and fatigue.

## 7.4.7 Results

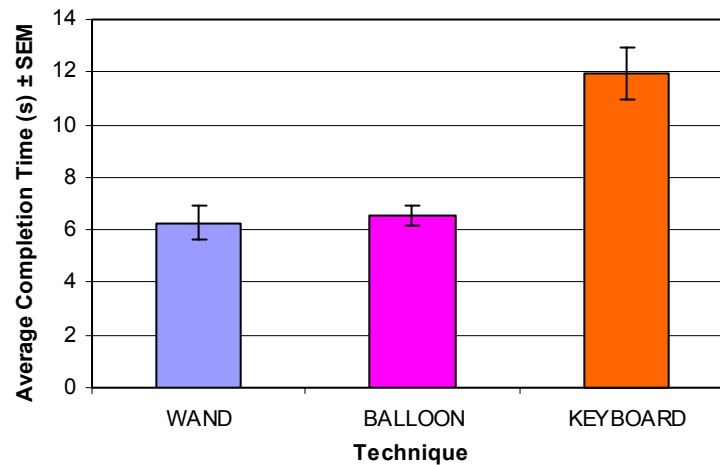
Our data was first cleaned by removing outliers, which accounted for 1.5% of all trials. The trials that ended in less than one second or more than one minute, or that resulted in a selection more than 5 cm away from the set of targets, were classified as outliers. Most of the outliers were caused by clicking mistakes when the participant inadvertently executed a selection command rather than moving or resizing the cursor. Outliers were present for all three techniques: 3 in WAND, 10 in BALLOON, and 9 in KEYBOARD, out of 480 selections for each technique (40 selections  $\times$  12 participants). We analyzed our outlier-free results according to three measured factors: completion time, error rate, and subjective ratings.

### 7.4.7.1 *Completion Time Analysis*

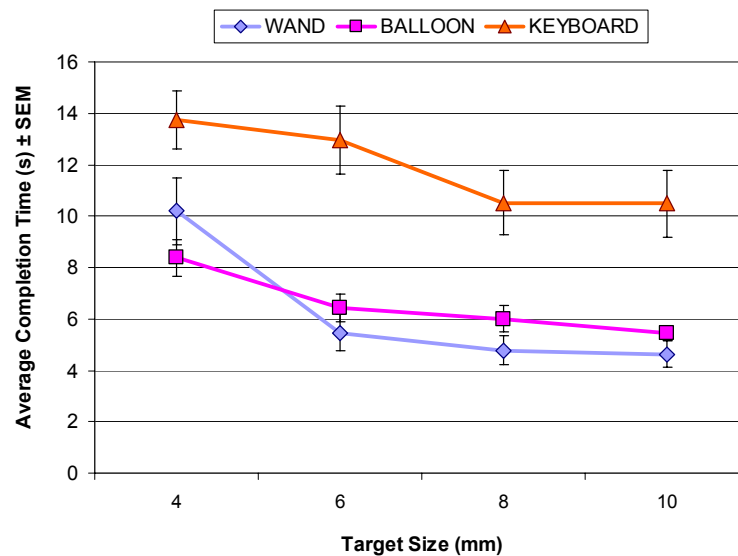
We performed a 3 (Technique)  $\times$  4 (Size) repeated measures ANOVA on mean selection times for the successfully completed trials (91% of outlier-free trials), with our participants as a random variable. We found significant main effects across all conditions (using an  $\alpha$ -level of 5%).

As expected, Technique had a significant main effect on completion times ( $F_{(2,22)}=22.334$ ,  $p<0.001$ ). KEYBOARD was on average more than 5 s slower than BALLOON ( $t_{(11)}=5.07$ ,  $p<0.001$ ), while there was no statistically significant difference in the performance of WAND and BALLOON (Figure 7.8). The average completion time difference between WAND and BALLOON techniques was very small (0.3 s). This confirmed H2, but failed to confirm H1. Size contained significant effects ( $F_{(3,33)}=20.228$ ,  $p<0.001$ ) that confirmed the expected result: selection time increased as target size decreased.

While the interaction of Technique and Size did not contain significant effects overall, it is interesting to observe that WAND was more negatively affected when selecting the smallest target (4 mm) than BALLOON (Figure 7.9). Overall, when selecting the smallest target, BALLOON was the fastest technique.



**Figure 7.8: Average completion times (s) for the three techniques ( $F_{(2,22)}=22.334$ ,  $p<0.001$ ): KEYBOARD was significantly slower than BALLOON ( $t_{(11)}=5.07$ ,  $p<0.001$ ), while WAND and BALLOON performed without statistically significant differences overall.**

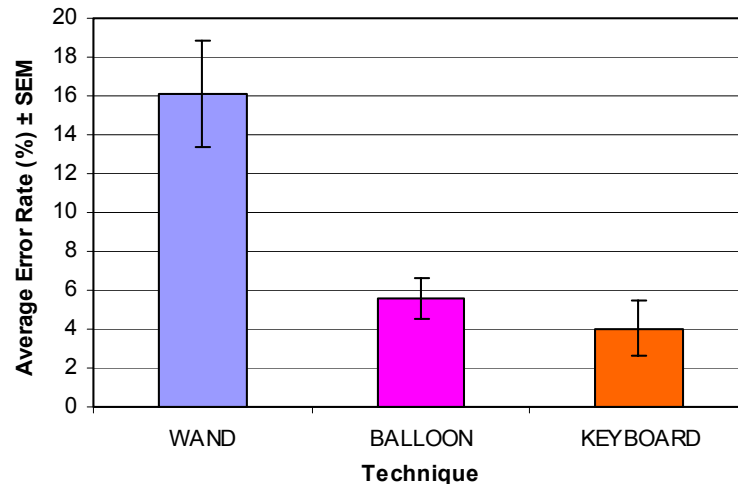


**Figure 7.9: Interaction of Technique × Size for average completion times (s).**

#### 7.4.7.2 Error Rate Analysis

To examine the performance of our techniques with regard to error rate, we performed a 3 (Technique) × 4 (Size) repeated measures ANOVA on mean error rate data, with our participants as a random variable. Significant main effects were found across all conditions (using an  $\alpha$ -level of 5%).

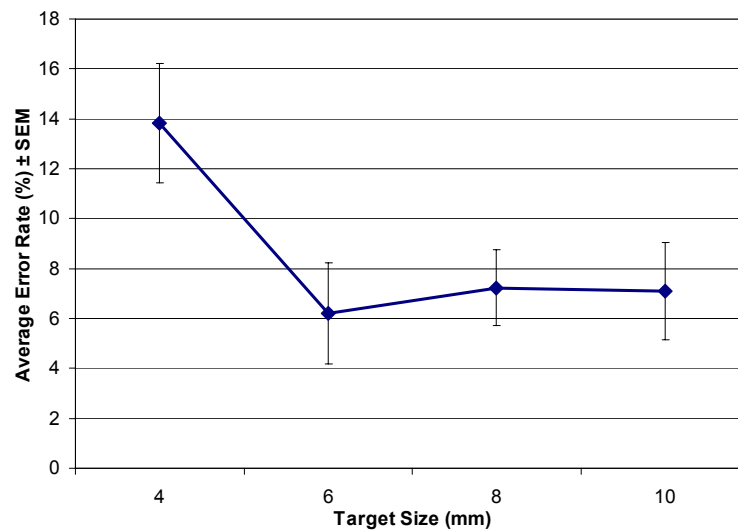
The Technique factor contained a significant main effect ( $F_{(2,22)}=18.707$ ,  $p<0.001$ ). On average, WAND was more than three times more error-prone than BALLOON ( $t_{(11)}=4.635$ ,  $p<0.001$ ), with an error rate of 16.1%, compared to 5.5% for BALLOON or 4.1% for KEYBOARD (Figure 7.10). This confirmed H3. Interestingly, while our data showed that BALLOON was slightly more error prone, this difference was not statistically significant. Therefore, H4 was not confirmed.



**Figure 7.10: Average error rate (%) of the three techniques ( $F_{(2,22)}=18.707$ ,  $p<0.001$ ). We found no statistically significant difference between the error rates of BALLOON and KEYBOARD.**

The Size factor also had a significant effect on error rate ( $F_{(3,33)}=4.672$ ,  $p<0.01$ ) (Figure 7.11). While all techniques performed somewhat consistently on 10, 8, and 6 mm targets, the smallest target (4 mm) caused significantly more errors (approximately twice as many) than the others.

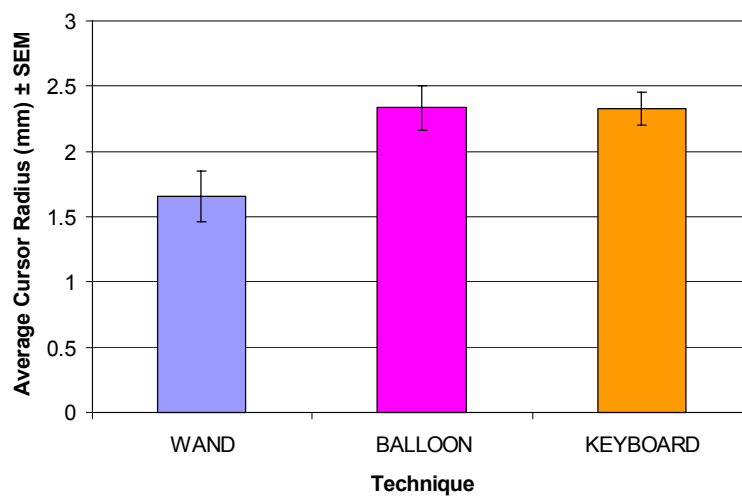
The interaction of Technique and Size was not found to be significant, with all techniques being equally affected by the target size decrease. However, we note that across all target sizes, WAND was consistently the most error-prone technique.



**Figure 7.11: Average error rate (%) across target sizes ( $F_{(3,33)}=4.672$ ,  $p<0.01$ ). On average, selecting the smallest target (4 mm) caused participants to commit twice as many errors as they did for the larger targets.**

#### 7.4.7.3 Cursor Radius Analysis

To examine the effects of the cursor radius, we performed a 3 (Technique)  $\times$  4 (Size) repeated measures ANOVA on mean radius data with our participants as a random variable. Significant main effects were found for both Technique and Size factors (using an  $\alpha$ -level of 5%).



**Figure 7.12: Average cursor radius (mm) across techniques.**

As expected, the Size factor contained significant main effects ( $F_{(3,33)}=98.538$ ,  $p<0.001$ ), which confirmed our belief that the users would reduce the cursor size when presented with smaller targets to decrease their chances of selecting any undesired targets. Interestingly, the Technique factor also contained a significant main effect ( $F_{(2,22)}=5.693$ ,  $p=0.01$ ) (Figure 7.12). The users consistently felt the need to use a significantly smaller cursor with WAND than when using BALLOON or KEYBOARD. This suggests that the users desired more precision with WAND, given that a larger cursor would have a higher chance of mistakenly selecting additional targets.

#### 7.4.7.4 *Subjective Evaluations*

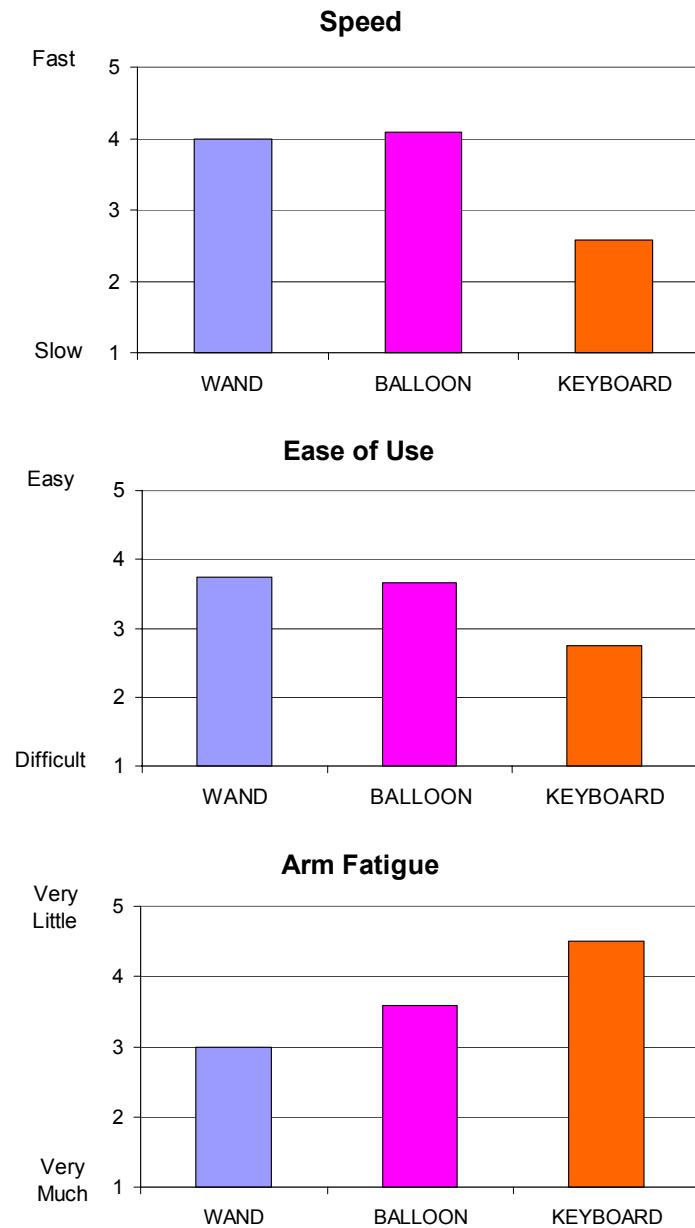
Participants filled out a post-experiment questionnaire rating their experience with the three techniques on a five-point Likert scale (1 being most negative and 5 being most positive). The participants were asked to comment on the techniques with regard to ease of use, learning time, performance speed, mental effort, hand/arm fatigue, and enjoyment.

Overall, WAND was rated best for ease of use ( $\mu=3.75$ ), learning time ( $\mu=4.41$ ), and mental effort ( $\mu=3.91$ ). In subjective evaluation of performance speed, the participants rated BALLOON ( $\mu=4.08$ ) and WAND ( $\mu=4.00$ ) as the fastest with almost identical scores, while KEYBOARD was given a much lower average score ( $\mu=2.58$ ). These results are consistent with our quantitative experimental results, which did not find a statistically significant difference between the performance speeds of BALLOON and WAND for this task.

For hand/arm fatigue, KEYBOARD was given the best rating ( $\mu=4.50$ ), with WAND rated the most fatiguing ( $\mu=3.00$ ), and BALLOON being in the middle ( $\mu=3.58$ ). Participants found BALLOON to be the most enjoyable to use ( $\mu=4.33$ ), while KEYBOARD was the most frustrating ( $\mu=2.91$ ). This might be skewed by the novelty and “coolness” factor of BALLOON, as noted by five participants.

Overall, half of our participants (6 out of 12) rated WAND as their top choice for a 3D selection task, followed by BALLOON (4 out of 12). The preference for WAND is hardly surprising, considering the directness of this technique; however, 4 out of 6 par-

Participants who preferred WAND overall, stated in their comments that for small targets they preferred BALLOON given that it “caused less fatigue and was easier to use more accurately than the wand” (Participant #8).



**Figure 7.13: Subjective user ratings for technique speed, ease of use, and arm fatigue (ratings on a 1–5 Likert scale)**

## 7.5 Discussion

While we expected that BALLOON would achieve lower error rates than WAND, we were pleasantly surprised that there was no statistically significant difference in performance speeds between BALLOON and WAND in our study. Our data suggests that Balloon Selection provides the user with fast interaction, while simultaneously reducing hand fatigue and lowering selection error rate.

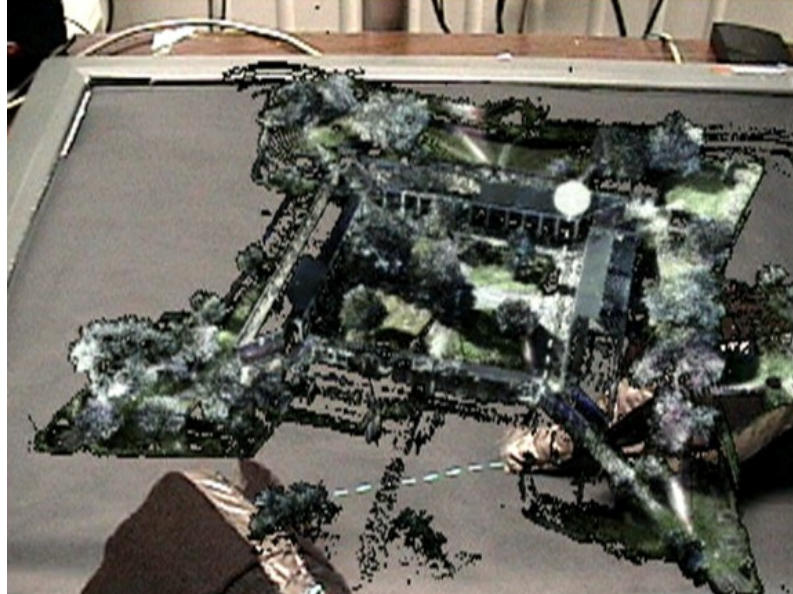
We believe that much further work can be done to optimize Balloon Selection for specific tasks. For example, it would be beneficial to investigate ways to further eliminate spurious selections. Furthermore, for environments in which a wide range of selection heights must be supported, Balloon Selection could be extended to include GOGO-like nonlinear mapping [Poupyrev 1996] of the distance between the anchor and stretching finger, or translation of the working volume. In addition, an inverted interaction could be designed that used a “weight-on-a-string” metaphor and therefore allowed the user to select objects in a 3D space within the display, rather than above it (e.g., a fish tank VR scenario [Ware 1993]).

While constraining the point of origin to the surface ensures that the user’s hands are not occluding the 3D object, it also means that the balloon’s first appearance will often be behind the virtual scene and therefore invisible to the user. One simple extension would be to support a user-settable initial offset from the surface, which would address sparse scenes with a raised ground plane. Additional visual feedback could be provided (e.g., cut-away views [Feiner 1992, Coffin 2006] or transparency) to allow the user to see the balloon upon creation or as it moves within a densely populated environment. Furthermore, being unable to see one’s hands while interacting might be disturbing to the user. This effect should be evaluated in a separate study.

In addition to using Balloon Selection in the VITA framework (Figure 7.1), Balloon Selection has been successfully deployed in a hybrid AR environment for visualizing large point cloud data sets, such as the model of the Fort Jay complex at Governors Island, NY, shown in Figure 7.14. Because of the size of the Fort Jay complex (about 22,500 m<sup>2</sup>), each of the buildings in its scaled model is only several cm tall, but still contains millions of distinct points that could be of interest. While beyond the scope of



this dissertation, it would be interesting to evaluate whether Balloon Selection significantly reduces the number of scaling operations that need to be performed on the model to be able to precisely select features of interest.



**Figure 7.14: Balloon Selection being used in a dense point cloud model of Fort Jay at Governors Island, NY. (Range scan data courtesy of Paul Blaer and Peter Allen of the Columbia University Robotics Lab.)**

We have presented a hybrid 3D selection technique that decomposes a 3DOF precise positioning task into a set of 2DOF and 1DOF positioning tasks on the tabletop. Our study did not show that users were penalized for this decoupling in terms of interaction speed and simplicity, but rather that they gained accuracy while reducing hand fatigue. Although we do not believe that Balloon Selection is the best method for 3D selection in every circumstance, we are hopeful that for many 3D tasks that operate on the volume above an interaction surface, this technique will provide a good metaphor for 3D selection. In particular, Balloon Selection appears to work well for static targets, such as the ones described in this paper, and may also prove useful for targets whose movement is perceived by the user as being appropriately decomposed into 2D and 1D components.

Our explorations of Balloon Selection, as well as Cross Dimensional Gestures, introduce a new interaction research direction of “cross-dimensional interactions” where

2D input surfaces can be used to facilitate interactions in the surrounding 3D environment and, vice versa, where 3D interactions can be used to manipulate data on 2D displays.

## 8 Conclusions and Future Work

This dissertation has explored the design and implementation of novel pointer, gesture, and multi-touch techniques for interacting within and across displays in MDEs. Each chapter thus far has provided a detailed discussion of the individual contributions presented in those sections. In this chapter, we briefly summarize our contributions and then draw from our experience in designing effective MDE interactions to present a set of guidelines for future MDE designers. We conclude by discussing potential directions for future work.

### 8.1 Summary of Contributions

The work in this dissertation has concentrated on two distinct areas. First, we explored current commercially available MDE configurations and designed new interaction metaphors that address some of the shortcomings of standard mouse-based pointer interactions across multiple displays. Then, we looked into more “exotic” hybrid MDEs that currently exist only in research laboratories. Within our implementation of a complex hybrid MDE, we focused on exploring freehand gesture- and touch-based interactions as a baseline interaction method across and within displays.

This dissertation presented the following five contributions:

- *Multi-Monitor Mouse pointer-warping techniques* (Chapter 3): These techniques improve the existing mouse pointer interaction in an MDE by allowing users to instantaneously relocate the cursor to an adjacent display instead of traversing the bezel. We contribute five alternatives to trigger the warp and three alternative strategies to locate the cursor after the warp. Our formal

- evaluations show that, in a homogeneous MDE, the benefits of pointer warping are present only when traversing two or more monitor bezels. However, in a heterogeneous MDE, the benefits grow in proportion to both the distance and the amount of visual–device space mismatch between monitors, and are present even when crossing a single monitor bezel.
- *Visual Interaction Tool for Archaeology (VITA) hybrid MDE* (Chapter 4): VITA allows for collaborative off-site analysis of archaeological excavation data by distributing the presentation of data among several head-worn, hand-held, projected tabletop, and large high-resolution displays. Within VITA, we developed a modular interaction framework that facilitated the exploration, implementation, and evaluation of hybrid MDE interactions throughout this dissertation.
  - *Cross-Dimensional Gestures* (Chapter 5): These freehand gestural techniques facilitate transitioning and associating data across devices, displays, and dimensionalities. The main contribution of these techniques is that they synchronize the recognition of gestures between a 2D multi-touch-sensitive projected display and a tracked 3D finger-bend sensor glove, and thus provide an easy and powerful way to connect the 2D and the 3D environments.
  - *Dual Finger Selection techniques* (Chapter 6): To address the precision issues with touch-based interactions on 2D screens, we developed a set of five two-finger techniques that allow for pixel-accurate selection of small targets on multi-touch-sensitive displays. The user can adaptively adjust the control-display ratio with a secondary finger, while the primary finger controls the pointer movement on the screen. A formal evaluation showed that these techniques significantly reduce the error rate when selecting small targets without introducing significant performance penalties.
  - *Balloon Selection technique* (Chapter 7): This hybrid MDE technique was inspired by our Cross-Dimensional Gestures and Dual Finger Selection techniques and it significantly improves the selection of small 3D targets viewed in a head-worn display by decoupling the 3DOF selection task into a 2DOF

task and a 1DOF task, while grounding the user's hands on a touch-sensitive tabletop display. A formal evaluation showed that users were able to reduce the number of erroneous selections by up to three times, without impacting the interaction time.

## 8.2 Design Guidelines

We now summarize our experience with designing pointer-, gesture-, and touch-based interactions in MDEs and present a set of specific guidelines that we believe are important for the future design of MDE interactions.

### 8.2.1 Hybrid MDE Implementation Guidelines

We build upon the earlier guidelines by Johanson and colleagues [Johanson 2002a] who stated that effective MDEs should support three key concepts: *heterogeneity* (integrating multiple different displays and input devices into the environment), *multiplicity* (supporting interaction of multiple users), and *dynamism* (support for easy addition and removal of devices and users in an MDE). We focus on more specific MDE interaction issues covered in this dissertation and provide examples from our work to support our claims.

*Design for independent, but connected spaces:* With the exception of attempts to present a single, large, unified display space (e.g., a large tiled display), we believe that MDE applications should exploit the difference between displays and use the display position, orientation, size, and bezels, to assist the users in grouping and organizing their tasks, as suggested by Grudin [Grudin 2001]. In addition, interactions across such displays can also benefit from adhering to the metaphor of independent, but connected spaces. For example, Multi-Monitor Mouse reduced the negative effects of moving the pointer over highly mismatched displays by treating the displays as independent spaces and allowing the user to warp the pointer instantaneously among them (Section 3.2).

*Exploit display affordances for input purposes:* In addition to using MDE displays as output devices, one should also aim to exploit the affordances of individual displays

to facilitate novel interaction capabilities. For example, the mobility, small size, and low weight of a tablet PC in VITA have allowed us to use that display as a movable lens within the context of the larger tabletop display (Section 4.3.1). This metaphor works well because of the horizontal nature of the tabletop surface, which permits the user to rest the lens display on the surface to stabilize it, rather than continuously holding it in their hands. Similarly, the same tabletop was used to support and stabilize the user's interactions above the surface in Balloon Selection (Section 7.3). Furthermore, we have used the ability of tracked head-worn see-through displays to render 3D computer graphics in correct perspective to create a seamless 3D space that connects other displays (Chapters 4, 5, and 7).

*Support simultaneous actions:* A highly collaborative system should allow multiple users to simultaneously interact in the environment, avoiding the need for serial turn-taking. In those instances where multiple people would like to access a single non-shareable resource, the system should provide enough feedback to allow the use of standard social protocols to resolve conflicts. Archaeologists who tried VITA repeatedly praised the system's ability to support simultaneous interactions throughout the environment (both across different displays and within the same display). We tagged each selected object with a unique color to identify the user who performed the selection, which allowed most conflicts to be resolved through simple verbal communication.

*Combine inputs for a richer interaction vocabulary:* By combining input from multiple sources, one can greatly reduce interaction ambiguity and significantly extend the vocabulary of interactions available from each modality. In VITA, we demonstrated this by facilitating multimodal selection with a combination of 3D gesture tracking and speech input (Section 4.3.2.1). In Cross-Dimensional Gestures, we combined 2D touch-sensing with 3D tracked glove-sensing to facilitate transitions and associations across hybrid MDE displays (Section 5.2.6).

## 8.2.2 Freehand Interaction Guidelines

A large portion of this thesis focused on freehand interactions as a baseline interaction metaphor for interacting within and across displays in a hybrid MDE. Here, we pre-

sent some specific guidelines for improving freehand interaction. While our techniques have been explored within the context of hybrid MDEs, we believe that these guidelines can be generally applied to many other gesture or multi-touch interfaces.

*Provide a clear interaction metaphor:* There is much supporting evidence on the benefits of providing clear interaction metaphors (e.g., [Brooks 1988, Hinckley 1994]), and most researchers agree that users have an easier time remembering and mastering techniques that are based on a clear metaphor. We were amazed by how differently people approached Balloon Selection when the balloon-on-a-string metaphor was explained to them (Section 7.3.4). In spite of the complexity of having many adjustable parameters, our trial users had no problems learning the technique and using it once the metaphor was clear to them. Our initial attempts to explain Balloon Selection without using the metaphor resulted in a high level of confusion and substantially longer learning time. While many of our metaphors have been based on real-world actions (e.g., pulling objects or holding the balloon), this is not necessarily required, as long as the metaphor is clear to the user. In fact, some of the most useful metaphors in desktop GUIs and VR have used supernatural magic metaphors to give the user abilities not available in the real world. A good example of such magic abilities is the use of menus in Alternate Reality Kit desktop GUI [Smith 1987], or the GOGO virtual hand technique in VR [Poupyrev 1996], which provides a non-linear extension of the human hand to manipulate objects at a distance.

*Multi-sensory feedback:* In addition to a clear metaphor, we recommend that designers provide visual and audio feedback to reinforce the metaphor and allow their users to gauge their progress with the interaction. A simple cursor notification widget (Section 6.3.4), allowed our Dual Finger X-Menu users to focus their attention on the task at hand and invoke the assistance modes without looking at the on-screen menu. Similarly, the “clicking” sound was reported to be critical for users of Balloon Selection to understand the balloon-on-a-string metaphor, even though the real string would not make such sounds (Section 7.3.4). This makes an interesting point that creating a mixed metaphor (i.e., a mix of real and magical) may well be desired and can potentially provide significant improvements on reality.

*Maintain directness and provide on-demand assistance:* The ability to directly touch an object in order to select it (without any offset or displacement) is probably the most appealing aspect of freehand interactions. Designers should support this direct manner of interaction whenever possible and require that further assistance or potential indirectness are invoked only when the user explicitly requests it. We recommend the use of a second hand or additional fingers to invoke assistance. For example, in Dual Finger Selection, we provide a cursor offset or a control-display ratio adjustment only when the user explicitly indicates the need for assistance by putting their secondary finger on the surface (Section 6.3.1). Similarly, Cross-Dimensional Privacy Pull and Push use the 2D chop gesture as an optional gesture modifier that changes the existing behavior of the transitioning interaction (Section 5.2.5).

*Support flexibility through a multi-touch, order-based, labeling approach:* To design flexible and robust multi-touch interactions, we recommend the use of the order-based approach (Section 6.3.1.1). This approach assigns roles to the user's touches based on the order in which fingers touch the surface, rather than the identity of a particular user's finger. The most significant benefits of this approach include simple accommodation of both left-handed and right-handed users, flexibility to customize finger use, easy utilization of the entire surface area, as well as support for interchangeable single-handed and bimanual operation.

*Use the distance between fingers to determine extent:* In addition to the position of a particular touch contact on the surface, the inner relationship (distance) between multiple touches is a very powerful modality that can be frequently and reliably used in multi-touch interactions. Humans naturally use the distance between fingers to specify extent and throughout this dissertation we have demonstrated multiple ways that this information can be used: triggering selection (by pinching two fingers together in Balloon Selection, Section 7.3.1), adjusting the scale or size factor (e.g., Dual Finger Stretch, Section 6.3.6), and adjusting the position or height of the balloon (Section 7.3.2).

*Consider touch-sensitive surfaces as a delimiter to engage and disengage interactions:* When interacting with 3D gestures, one should use the surface contact as a natural delimiter to interaction. Since the user cannot disengage from the 3D space in which



they live, 3D gestural interactions often have problems with delimiters (starting and stopping points). Typical solutions are to provide a starting pose or a starting location for all gestures (e.g., in VITA this was the 3D idle pose). However, one can use the surface contact to provide a clear delimiter as well, since touching the surface is usually a deliberate non-ambiguous action. Our Cross-Dimensional Gestures use surface contact to differentiate cross-dimensional interactions from the existing 3D interactions available to the user (Section 5.2). In addition, when dealing with touch-based interactions, only require the user to maintain various touch contacts if absolutely necessary. The ability to disengage from the surface without losing the interaction context is important, since that is the way that the user will be able to eliminate extra DOFs from the task. For example, by using the secondary finger to clutch/declutch the balloon string in Balloon Selection, the user can remove the height adjustment from the task and fix the balloon movement in a horizontal plane (Section 7.3.2).

*Account for spurious gestures or touches and fail gracefully:* Given the “always-on” nature of using our hands as input devices, many gestural and touch-based interfaces will inevitably suffer from a number of spurious and undesired interactions. For example, inexperienced users may have a hard time controlling more than one or two fingers on the surface, which for some users might result in several extra unwanted touches. When designing multi-touch interactions, ensure that extra touches do not end up disrupting the interaction, and provide meaningful feedback to the user about why an interaction failed. In addition, the interface should make it easy to reverse the interaction and thereby cancel the negative effect caused by the spurious action. All our observations confirm that, once alerted to the spurious touch problem, users are very capable of adjusting their behavior to ensure error free interaction execution.

### 8.3 Future Work

We close this dissertation with a brief discussion of several remaining research challenges and suggest some future work.

### 8.3.1 MDE Pointer Warping Interactions

Several issues remain to be evaluated from our pointer warping work (Chapter 3). We have demonstrated the benefits of pointer warping between individual displays; however, we have not experimented with allowing the user to setup arbitrary rectangular frames and warp among them (although the feature is implemented). Such arbitrary user-defined frames would allow the user to use pointer warping even within a single display. We believe that specifying some high usage areas of the screen as warping targets (e.g., the “Start” button on the taskbar) can significantly speed up targeting, and thus improve overall interaction time. These arbitrary frames could even be defined as application specific and could move with particular applications.

One of the Multi-Monitor Mouse users suggested using mouse gestures to specify the direction of the warp. This approach is similar to the gestures used in the Opera web browser [Opera 2007] for web page navigation, except in our case, the user could specify the direction (e.g., left, right, up, and down) and potentially the amount of the warp (e.g., one or two monitors). While this would probably not be useful in a two-monitor setup (as it would clearly introduce additional gesturing time before the warp), we speculate that it might be beneficial in an MDE setup in which the monitors are tiled in a 2D grid.

The remaining puzzling result of our pointer warping studies is that all targeting left-to-right was found to be slower than right-to-left. While we speculate that these effects might be due to the differences between brain hemispheres in processing human motor behavior [Carson 1996], a further study is needed to formally analyze these effects. This study should test targeting movements not only across displays, but also within a single display and should explore the effects of handedness, gender, and possibly even test some cultural differences (such as reading/writing direction). The results could significantly influence the future placement and organization of targets within applications, as well as distribution of applications across monitors. For example, judging from our results, the tasks that require quickest response should be placed on the left side of the screen or on the left-most display in an MDE.

### 8.3.2 Hybrid MDE Improvements

In this dissertation, we provide some insights about the usability of hybrid MDEs. However, we still lack a systematic evaluation that would explore the effects of different hybrid MDE configurations on human factors. In particular, it would be beneficial to know the effects of directly overlapping displays (e.g., a combination of a head-worn and a stationary display) on human perception and task performance. Is there a quantifiable benefit in directly combining 2D and 3D displays? What are the perceptual and usability issues when visualizing 3D content through various form factors of see-through displays (i.e., head-worn and handheld)?

We would also like to evaluate the effectiveness of the VITA hybrid MDE in extended use. We believe that during our usability evaluations, most users were too overwhelmed by the novelty of the interface to be able to seriously critique its performance and usability. Thus far, only the developers of VITA have actually spent more than two hours in the environment, and it would be important to collect the impressions and usage patterns through prolonged VITA use by its target audience. We originally planned to have VITA evaluated as part of a semester-long archaeology class; however, due to scheduling and staffing conflicts, this was not possible.

To further improve interaction in VITA, many hardware constraints will need to be eliminated. In fact, hardware-related issues received the largest number of complaints from our test users. In particular, the use of more transparent and lighter head-worn displays, completely wireless tracking, and the removal of glove-based gesture tracking would significantly improve interaction in VITA, as indicated in questionnaire feedback. In the future, we are interested in exploring camera-based gesture tracking in a constrained volume above the tabletop surface to eliminate the need for a tracked bend-sensor glove on the user's hand (similar to existing approaches that focus on interactions on pure 2D displays [Kjeldsen 1997, Wilson 2005b, Wilson 2006]).

Another significant area of improvement in VITA stems from a lack of simple ways for non-programmers to modify the environment. Currently, the behavior of applications across VITA displays is completely predetermined and hard-coded, which makes it difficult to introduce new functionality or a different display module. A beneficial and

relatively simple extension to VITA would be to incorporate a clipboard sharing mechanism across devices, similar to Synergy [Schoeneman 2007], which shares the mouse, keyboard, and clipboard across different machines. Clipboard sharing would allow VITA users to easily bring non-VITA displays and applications into the hybrid MDE. In that way, users could integrate some of the existing standard applications into the VITA analysis process. Another option is to integrate immersive authoring facilities into VITA (similar to [Sandor 2005]), which would result in a live, dynamically changeable and adaptive hybrid MDE.

### 8.3.3 Collaboration, Personalization, and Privacy

A significant benefit of VITA is that it allows multiple users to collaboratively analyze archaeological data. As mentioned in Section 4.5.1, there are numerous unexplored questions about the use of head-worn displays in collaborative hybrid MDEs, particularly with regards to private or personalized views. While our Cross-Dimensional Privacy Pull and Push gestures (Section 5.2.5) allow the user to change the privacy status of objects as they transition between displays, we have not yet formally evaluated the effects of personalized views on collaboration. We envision that personalized views (similar to [Agrawala 1997]) in VITA would be a particularly important collaboration scenario between two experts from different archaeology sub-fields (e.g., ceramics and metallurgy). Each expert might require that different data attributes be visible when discussing some portion of the excavation site.

While simply hiding the data is not difficult, due to the private nature of head-worn displays, interesting problems arise when the user is unaware that their collaborator cannot see the same data. Novel visualizations could be employed to signal the existence of hidden data, but not reveal its identity. Also, some objects could automatically be tagged as private to reduce visual clutter in the environment which would require the use of some automated environment management solution (e.g., [MacIntyre 1996, Bell 2001]).

### 8.3.4 Freehand Gestural and Multi-Touch Interaction

This dissertation has showed the possibilities of using freehand gestural interactions to interact in an MDE. Given the increase in research of multi-touch prototypes [Matsushita 1997, Dietz 2001, Wilson 2004, Han 2005, Wilson 2005a] and the recent announcement of some multi-touch commercial handheld devices (i.e., Apple iPhone [Apple 2007]), we believe that it is very reasonable to assume that the unifying input method across future displays will not be a mouse and a keyboard (since such input methods are not particularly suitable for small handheld or tablet devices), but more likely a pen or a touch. Hence we believe that it is important to continue researching touch-based gestural interactions and we believe that those techniques will significantly impact future MDE interactions.

One important direction in multi-touch interaction design is to keep designing techniques that could be executed both bimanually and single-handedly (similar to our Dual Finger Selection and Balloon Selection). Designing interactions with such flexibility will automatically make them relevant on a variety of surface sizes ranging from wall-sized displays to small handhelds.

With regard to future improvements of our own techniques, we highlight several possible directions. The feedback from the usability study of Cross-Dimensional Gestures showed slight preference towards simpler interactions (Pin and Connect). We believe that the Cross-Dimensional Pull interaction could be drastically simplified by considering only two fingers and their pinching movement on the surface. It remains to be evaluated whether such implementation would be preferred to the current gesture. Also, to maintain a consistent metaphor, the Push gesture would probably also need to be modified to require only two fingers. This two-finger implementation might also be simpler to detect and track with camera-based tracking, which would allow us to completely free the user's hands of any sensors or gloves.

An alternate approach to completing connections between 3D and 2D environments was demonstrated by one of our Cross Dimensional Gestures study participants. He held the 3D object with one hand and used the other hand to touch the tabletop and establish connection. This "through-the-body" connection metaphor opens up numerous

possibilities of designing alternate interactions to establish connections between objects and displays.

We believe that much further work remains to be done to optimize Balloon Selection. For example, we are considering using non-linear mapping (similar to GOGO [Poupyrev 1996]) to control the string length. This would allow our users to more easily reach further above the surface. Furthermore, one could use a control-display ratio adjustment (similar to Dual Finger Slider) to slow down the balloon in all three dimensions. This could potentially improve the precision of Balloon Selection, but whether the control of such a cursor would be manageable and desired by our users, remains to be evaluated. In addition, a separate study should investigate the cognitive effects of not being able to see one's hands while interacting below a 3D object.

## **8.4 Final Remarks**

The world of personal computing is experiencing a significant transition from individual, isolated desktop computers with a single attached monitor to heterogeneous computing environments with multiple connected devices and displays. Such rich environments can include large displays, laptops, tablets, tabletops, personal digital assistants, and mobile phones. While all these devices increase in power and connectedness every day, we currently lack the interaction vocabulary to use them in a synergistic fashion. In fact, most of the models and metaphors used for interacting across them, are the same ones originally developed for devices with only a single display.

We believe that a new perspective is needed, as simply taking old methods and applying them to new devices and spaces often produces mediocre results. Multi-display environments must be approached on their own merits as new holistic systems, and not simply as a mixture of known parts. Only by emphasizing system characteristics such as heterogeneity, connectivity, and physical discontinuity between devices and displays, can we start designing effective MDE interactions.

This dissertation has attempted to look at the implications of complex MDEs on human-computer interaction. Our novel interactions do not disregard the fact that they are operating over different displays, but instead use the affordances of multiple dis-

---

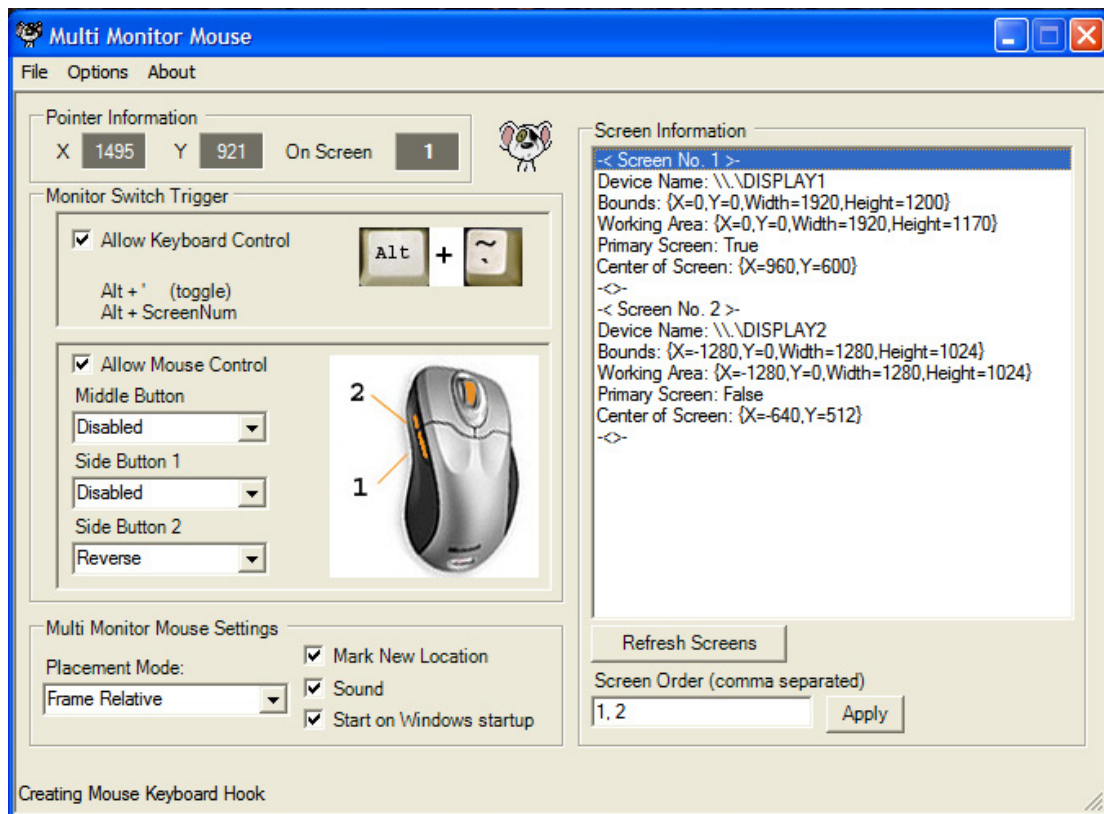
plays to help the user. While we have explored several MDE combinations, we have only scratched the surface of MDE interaction possibilities. Much work remains to be done and we are excited about the possibilities that lie ahead. We anticipate that in the future, we will stop thinking about MDE interaction as an extension of the “standard” single-display interaction space, instead inverting the model such that single-display interaction is simply considered a subset of the more general multi-display interaction space.





## Appendix A Multi-Monitor Mouse Widget

We have packaged a subset of our original Multi-Monitor Mouse widget code (described in Section 3.2) and made it available for free download. This standalone widget supports all warping strategies tested, and incorporates the two simplest frame-switching techniques (mouse button and keyboard switch). Those frame-switching techniques do not require any additional hardware and are therefore suitable for wide distribution.



**Figure A.1:** The screenshot of the control interface for the Multi-Monitor Mouse widget application.

The widget is implemented as a Windows XP system tray application and is fully customizable from a simple graphical user interface (Figure A.1). Upon startup, the widget automatically reads the system's information about the connected displays and uses that to form virtual frames between which the user can warp their pointer. In addition, the user can manually rearrange the order between displays to better suit their needs. All configuration options are automatically stored in a configuration file, and the widget provides an option to start automatically upon Windows XP startup.

While the use of a five button mouse is not required, it is highly recommended. If a five button mouse is not used, the widget can use the middle mouse button to trigger the warp.

## **A.1 Implementation Details**

The Multi-Monitor Mouse widget is implemented in C# and requires Microsoft .NET Framework 1.0 or higher. It comes as a standard installation package which automatically installs all necessary components to the user's system.

In order to provide the pointer warping functionality that is completely application independent, our widget processes low-level mouse events by placing a global hook (`WH_MOUSE_LL`) into the operating system. We have adapted the system hook code from George Mamladze's project called GlobalHook [Mamladze 2004]. This code contains no restrictions on use, attribution, or commercialization. We have packaged the GlobalHook code into a dynamically linked library called `UserActivityHook.dll` which is included in our distribution.

## **A.2 Widget Distribution**

The packaged distribution of Multi-Monitor Mouse has been available for free download from the author's website since October 11, 2006: <http://www.cs.columbia.edu/~benko/projects/m3>.

The packaged distribution can also be downloaded from the CNET download.com website since April 11, 2007: <http://www.download.com/3000-20-10657306.html>.

## References

- Accot, J. and Zhai, S. (2002). "More than Dotting the i's - Foundations for Crossing-Based Interfaces," *Proc. CHI '02*, Minneapolis, MN, USA, pp. 73–80.
- Acevedo, D., Vote, E., Laidlaw, D. H. and Joukowsky, M. S. (2001). "Archaeological Data Visualization in VR: Analysis of Lamp Finds at the Great Temple of Petra, a Case Study," *Proc. IEEE Visualization 2001*, October 2001, pp. 493–496.
- Agrawala, M., Beers, A., Fröhlich, B., Hanrahan, P., McDowall, I. and Bolas, M. (1997). "The Two-User Responsive Workbench: Support of Collaboration Through Individual Views of a Shared Space," *SIGGRAPH '97*, pp. 327–332.
- Albinsson, P. A. and Zhai, S. (2003). "High Precision Touch Screen Interaction," *Proc. CHI '03*, Ft. Lauderdale, FL, pp. 105–112.
- Allen, P., Feiner, S., Troccoli, A., Benko, H., Ishak, E. and Smith, B. (2004). "Seeing into the Past: Creating a 3D Modeling Pipeline for Archaeological Visualization," *Proc. 3D Data Processing, Visualization & Transmission (3DPVT '04)*, Thessaloniki, Greece, Sept. 6–9, pp. 751–758.
- Andrews, C. R., Sampemane, G., Weiler, A. and Campbell, R. H. (2004). "Clicky: User-centric Input for Active Spaces", *Technical Report UIUCDCS-R-2004*, University of Illinois at Urbana-Champaign.
- Andujar, C. and Argelaguet, F. (2006). "Friction Surfaces: Scaled Ray-Casting Manipulation for Interacting with 2D GUIs," *Proc. 12th Eurographics Symposium on Virtual Environments*, pp. 101–108.
- Apple Inc., (2007). *iPhone*, <http://www.apple.com/iphone/> (accessed on March 30, 2007).
- Ashdown, M., Oka, K. and Sato, Y. (2005). "Combining Head Tracking and Mouse Input for a GUI on Multiple Monitors," *Proc. Extended Abstracts CHI '05*, pp. 1188–1191.

- Badler, N. I., Manocha, K. H. and Baraff, D. (1986). "Multi-Dimensional Input Techniques and Articulated Figure Positioning by Multiple Constraints," *Proc. Workshop on Interactive 3D Graphics*, pp. 151–169.
- Baudel, T. and Beaudouin-Lafon, M. (1993). "Charade: Remote Control of Objects Using Free-Hand Gestures," *Communications of the ACM*, Vol. 36 (7), July, pp. 28–35.
- Baudisch, P., Cutrell, E., Hinckley, K. and Gruen, R. (2004). "Mouse Ether: Accelerating the Acquisition of Targets Across Multi-Monitor Displays," *Proc. Extended Abstracts CHI '04*, Vienna, Austria, April 24–29, pp. 1379–1382.
- Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B. and Zierlinger, Z. (2003a). "Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-operated Systems," *Proc. INTERACT '03*, pp. 57–64.
- Baudisch, P., Cutrell, E. and Robertson, G. (2003b). "High-Density Cursor: A Visualization Technique that Helps Users Keep Track of Fast-Moving Mouse Cursors," *Proc. INTERACT '03*, Zurich, Switzerland, August, pp. 236–243.
- Baudisch, P., Good, N. and Stewart, P. (2001). "Focus Plus Context Screens: Combining Display Technology with Visualization Techniques," *Proc. UIST '01*, Orlando, FL, November, pp. 31–40.
- Bell, B. (2005). "View Management for Distributed User Interfaces," *Ph.D. Thesis*, Department of Computer Science, Columbia University, New York, NY.
- Bell, B. and Feiner, S. (2000). "Dynamic Space Management for User Interfaces," *Proc. UIST '00*, San Diego, CA, November 5–8, pp. 239–248.
- Bell, B., Feiner, S. and Höllerer, T. (2001). "View Management for Virtual and Augmented Reality," *Proc. UIST '01*, Orlando, FL, November 11–14, pp. 101–110.
- Benko, H. and Feiner, S. (2005a). "Multi-Monitor Mouse," *Proc. Extended Abstracts CHI '05*, Portland, OR, pp. 1208–1211.
- Benko, H. and Feiner, S. (2007a). "Balloon Selection: A Multi-Finger Technique for Accurate Low-Fatigue 3D Selection," *Proc. Symposium on 3D User Interfaces (3DUI '07)*, Charlotte, NC, March 10–11, pp. 79–86.
- Benko, H. and Feiner, S. (2007b). "Pointer Warping in Heterogeneous Multi-Monitor Environments," *Proc. Graphics Interface '07*, Montreal, Canada, May 28–30.
- Benko, H., Ishak, E. and Feiner, S. (2005b). "Cross-Dimensional Gestural Interaction Techniques for Hybrid Immersive Environments," *Proc. IEEE VR '05*, Bonn, Germany, March 2005, pp. 209–216.

- Benko, H., Ishak, E. W. and Feiner, S. (2004). "Collaborative Mixed Reality Visualization of an Archaeological Excavation," *Proc. International Symposium on Mixed and Augmented Reality (ISMAR '04)*, Arlington, VA, November 2–5, pp. 132–140.
- Benko, H., Wilson, A. and Baudisch, P. (2006). "Precise Selection Techniques for Multi-Touch Screens," *Proc. CHI '06*, pp. 1263–1272.
- Bier, E. A., Stone, M. C., Pier, K., Buxton, W. and DeRose, T. (1993). "Toolglass and Magic Lenses: The See-Through Interface," *Proc. SIGGRAPH '93*, August, pp. 73–80.
- Blanch, R., Guiard, Y. and Beaudouin-Lafon, M. (2004). "Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation," *Proc. CHI '04*, Vienna, Austria, pp. 519-526.
- Bolt, R. (1980). "Put-That-There: Voice and Gesture at the Graphics Interface," *Proc. SIGGRAPH '80*, pp. 262–270.
- Booth, K. S., Fisher, B. D., Lin, C. J. R. and Argue, R. (2002). "The "Mighty Mouse" Multi-Screen Collaboration Tool," *Proc. Symposium on User Interface Software and Technology*, Paris, France, pp. 209-212.
- Bootstrap Institute, (2003). *Bootstrap Institute: History in Pix*, <http://www.bootstrap.org/chronicle/pix/pix.html> (accessed on March 30, 2007).
- Bowman, D., Johnson, D. B. and Hodges, L. F. (1999). "Testbed Evaluation of Virtual Environment Interaction," *Proc. ACM VRST '99*, pp. 26–33.
- Bowman, D. A. and Hodges, L. F. (1997). "An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments," *Proc. Symposium on Interactive 3D Graphics (I3D '97)*, Providence, RI, pp. 35–38.
- Bowman, D. A., Kruijff, E., Joseph J. LaViola, J. and Poupyrev, I. (2004). *3D User Interfaces: Theory and Practice*, Addison-Wesley.
- Brooks, F. P. (1986). "Walkthrough - A Dynamic Graphics System for Simulating Virtual Buildings," *Proc. Workshop on Interactive 3D Graphics*, pp. 9–21.
- Brooks, F. P. (1988). "Grasping Reality Through Illusion: Interactive Graphics Serving Science," *Proc. CHI '88*, Washington, D.C., United States, pp. 1–11.
- Brooks, R. A., Coen, M., Dang, D., Bonet, J. D., Kramer, J., Lozano-Perez, T., Mellor, J., Pook, P., Stauffer, C., Stein, L., Torrance, M. and Wessler, M. (1997). "The Intelligent Room Project," *Proc. The Second International Conference on Cognitive Technology (CT '97)*, pp. 271–278.

- Brumitt, B., Meyers, B., Krumm, J., Kern, A. and Shafer, S. (2000). "EasyLiving: Technologies for Intelligent Environments," *Proc. Handheld and Ubiquitous Computing*, pp. 12–29.
- Butler, C. G. and Amant, R. S. (2004). "HabilisDraw DT: A Bimanual Tool-Based Direct Manipulation Drawing Environment," *Proc. Extended Abstracts CHI '04*, Vienna, Austria, pp. 1301–1304.
- Butz, A., Beshers, C. and Feiner, S. (1998). "Of Vampire Mirrors and Privacy Lamps: Privacy Management in Multi-User Augmented Environments," *Proc. UIST '98*, San Francisco, CA, USA, pp. 171–172.
- Butz, A., Höllerer, T., Feiner, S., MacIntyre, B. and Beshers, C. (1999). "Enveloping Users and Computers in a Collaborative 3D Augmented Reality," *Proc. International Workshop on Augmented Reality (IWAR '99)*, October 20–21, pp. 35–44.
- Buxton, W. (1985). "Issues and Techniques in Touch-Sensitive Tablet Input," *Proc. ACM SIGGRAPH '85*, San Francisco, CA, July 22–26, pp. 215–224.
- Buxton, W. and Myers, B. (1986). "A Study in Two-Handed Input," *Proc. CHI '86*, pp. 321–326.
- Card, S. K., Moran, T. P. and Newell, A. (1980). "The Keystroke-Level Model for User Performance Time with Interactive Systems," *Communications of the ACM*, Vol. 23 (7), pp. 396–410.
- Carson, R. D. (1996). "Putative Right Hemisphere Contributions to the Preparation of Reaching and Aiming Movements," in D. Elliott and E. A. Roy, eds., *Manual Asymmetries in Motor Performance*, Boca Raton: CRC Press, Inc., pp. 159–172.
- Caudell, T. P. and Mizell, D. W. (1992). "Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes," *Proc. IEEE Hawaii International Conference on Systems Sciences '92*, January, pp. 659–669.
- Coffin, C. and Hollerer, T. (2006). "Interactive Perspective Cut-away Views for General 3D Scenes," *Proc. IEEE Symposium on 3D User Interfaces (3DUI '06)*, pp. 25–28.
- Cruz-Neira, C., Sandin, D., DeFanti, T., Kenyon, R. and Hart, J. (1992). "The CAVE: Audio Visual Experience Automatic Virtual Environment," *Communications of the ACM*, Vol. 35 (6), pp. 65–72.
- Cutler, L. D., Fröhlich, B. and Hanrahan, P. (1997). "Two-Handed Direct Manipulation on the Responsive Workbench," *Proc. ACM Symposium on Interactive 3D Graphics (I3D '97)*, Providence, RI, pp. 107–114.

- Czemuszenko, M., Pap, D., Sandin, D., DeFanti, T., Dawe, G. and Brown, M. (1997). "The ImmersaDesk and InfinityWall Projection-Based Virtual Reality Displays," *Computer Graphics*, Vol. 31 (2), May, 1997, pp. 46–49.
- Darrell, T. and Pentland, A. (1993). "Space-Time Gestures," *Proc. Computer Vision and Pattern Recognition*, New York, NY, USA, pp. 335–340.
- Deering, M. (1992). "High Resolution Virtual Reality," *Proc. SIGGRAPH '92*, pp. 195–202.
- Dietz, P. and Lehigh, D. (2001). "DiamondTouch: a Multi-User Touch Technology," *Proc. UIST '01*, pp. 219–226.
- Dulberg, M. S., Amant, R. S. and Zettlemoyer, L. S. (1999). "An Imprecise Mouse Gesture for the Fast Activation of Controls," *Proc. INTERACT '99*, pp. 375–382.
- Ehnes, J., Hirota, K. and Hirose, M. (2005). "Projected Augmentation II — A Scalable Architecture for Multi Projector Based AR-Systems Based on 'Projected Applications'," *Proc. IEEE/ACM ISMAR '05*, pp. 190–191
- Engelbart, D. C. and English, W. K. (1968). "A Research Center for Augmenting Human Intellect," *Proc. AFIPS Fall Joint Computer Conference*, pp. 395–410.
- Feeney, R. (2002). *Specific Anthropometric and Strength Data for People with Dexterity Disability*, Consumer and Competition Policy Directorate, Department of Trade and Industry, London, UK.
- Feiner, S. and Seligmann, D. (1992). "Cutaways and Ghosting: Satisfying Visibility Constraints in Dynamic 3D Illustrations," *The Visual Computer*, Vol. 8 (5–6), June, pp. 292–302.
- Feiner, S. and Shamash, A. (1991). "Hybrid User Interfaces: Breeding Virtually Bigger Interfaces for Physically Smaller Computers," *Proc. UIST '91*, pp. 9–17.
- Forlines, C., Esenther, A., Shen, C., Wigdor, D. and Ryall, K. (2006a). "Multi-User, Multi-Display Interaction with Single-User, Single-Display Geospatial Application," *Proc. UIST '06*, Montreux, Switzerland, pp. 273–276.
- Forlines, C., Vogel, D. and Balakrishnan, R. (2006b). "HybridPointing: Fluid Switching Between Absolute and Relative Pointing with a Direct Input Device," *Proc. UIST '06*, Montreux, Switzerland, pp. 211–220.
- Forsberg, A., Herndon, K. P. and Zeleznik, R. (1996). "Aperture Based Selection for Immersive Virtual Environments," *Proc. UIST '96*, pp. 95–96.

- Frees, S. and Kessler, G. (2005). "Precise and Rapid Interaction through Scaled Manipulation in Immersive Virtual Environments," *Proc. IEEE VR '05*, Bonn, Germany, pp. 99–106.
- Gaitarzes, A., Christopoulos, D. and Roussou, M. (2001). "Reviving the Past: Cultural Heritage Meets Virtual Reality," *Proc. Conference on Virtual Reality, Archaeology and Cultural Heritage*, pp. 103–110.
- Garlan, D., Siewiorek, D., Smailagic, A. and Steenkiste, P. (2002). "Project Aura: Toward Distraction-Free Pervasive Computing," *IEEE Pervasive Computing*, Vol. 1 (2), pp. 22–31.
- Google, Inc., (2006). *Google Earth*, <http://earth.google.com> (accessed on March 7, 2007).
- Greene, R. (1985). "The Drawing Prism: A Versatile Graphic Input Device," *Proc. SIGGRAPH '85*, San Francisco, CA, USA, pp. 103–110.
- Grudin, J. (2001). "Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use," *Proc. CHI '01*, Seattle, WA, pp. 458–465.
- Guiard, Y. (1987). "Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain Model," *Journal of Motor Behavior*, Vol. 19 (4), pp. 486–517.
- Haan, G. d., Griffith, E. J., Koutek, M. and Post, F. H. (2006). "Hybrid Interfaces in VEs: Intent and Interaction," *Proc. 12th Eurographics Symposium on Virtual Environments*, pp. 109–118.
- Han, J. Y. (2005). "Low-cost multi-touch sensing through frustrated total internal reflection," *Proc. UIST '05*, Seattle, WA, USA, pp. 115–118.
- Harris, E. C. (1989). *Principles of Archaeological Stratigraphy*, 2nd ed. Academic Press, London, New York.
- Herndon, K. P., Zeleznik, R. C., Robbins, D. C., Conner, D. B., Snibbe, S. S. and van Dam, A. (1992). "Interactive Shadows," *Proc. UIST '92*, November 15–18, pp. 1–6.
- Hinckley, K. (2003). "Synchronous Gestures for Multiple Persons and Computers," *Proc. UIST '03*, Vancouver, BC, pp. 149–158.
- Hinckley, K., Pausch, R., Goble, J. C. and Kassell, N. F. (1994). "A Survey of Design Issues in Spatial Input," *Proc. UIST '94*, pp. 213–222.



- Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P. and Smith, M. (2004). "Stitching: Pen Gestures that Span Multiple Displays," *Proc. Advanced Visual Interfaces (AVI '04)*, Gallipoli, Italy, pp. 23–31.
- Holmquist, L. E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M. and Gellersen, H.-W. (2001). "Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts," *Proc. UbiComp '01*, pp. 116–121.
- Hua, H., Brown, L. D., Gao, C. and Ahuja, N. (2003). "A new collaborative infrastructure: SCAPE," *Proc. IEEE Virtual Reality (VR'03)*, March 22–26, pp. 171–179.
- Humphreys, G. and Hanrahan, P. (1999). "A Distributed Graphics System for Large Tiled Displays," *Proc. IEEE Visualization '99*, pp. 215–223.
- Humphreys, G., Houston, M., Ng, R., Frank, R., Ahern, S., Kirchner, P. D. and Klosowski, J. T. (2002). "Chromium: A Stream-Processing Framework for Interactive Rendering on Clusters," *Proc. SIGGRAPH '02*, San Antonio, TX, pp. 693–702.
- Hutchings, D., Stasko, J. and Czerwinski, M. (2005). *Workshop on Distributed Display Environments, Conference on Human Factors in Computing Systems (CHI '05)*, Portland, OR.
- Hutchings, D. R., Czerwinski, M., Meyers, B. and Stasko, J. (2004a). "Exploring the Use and Affordances of Multiple Display Environments," *Proc. Workshop on Ubiquitous Display Environments (in conjunction with UbiComp '04)*, Nottingham, England.
- Hutchings, D. R., Smith, G., Meyers, B., Czerwinski, M. and Robertson, G. (2004b). "Display Space Usage and Window Management Operation Comparisons between Single Monitor and Multiple Monitor Users," *Proc. Advanced Visual Interfaces (AVI '04)*, pp. 32–39.
- Hutchings, D. R. and Stasko, J. (2004c). "Revisiting Display Space Management: Understanding Current Practice to Inform Next-Generation Design," *Proc. Graphics Interface '04*, London, Ontario, Canada, pp. 127–134.
- Ishii, H. and Ullmer, B. (1997). "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms," *Proc. CHI '97*, Atlanta, GA, MArch, pp. 234–241.
- Iten, C. and Lüthi, D. (2007). "The Digital Tweezers: A Precise Selection Technique for Multi-Touch Screens," *Proc. International Conference on Tangible and Embedded Interaction (TEI '07)*, (in submission).
- Jacob, R. (1991). "The use of eye movements in human-computer interaction techniques: What you look at is what you get," *ACM Transactions on Information Systems (TOIS)*, Vol. 9 (3), April, pp. 152–169.

- Jacob, R. J. K. and Sibert, L. E. (1992). "The Perceptual Structure of Multidimensional Input Device Selection," *Proc. CHI '92*, Monterey, CA, pp. 211–218.
- Jeong, B., Renambot, L., Jagodic, R., Singh, R., Aguilera, J., Johnson, A. and Leigh, J. (2006). "High-Performance Dynamic Graphics Streaming for Scalable Adaptive Graphics Environment," *Proc. SC '06*, Tampa, FL, pp. 24.
- Johanson, B., Fox, A. and Winograd, T. (2002a). "The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms," *IEEE Pervasive Computing*, Vol. 1 (2), pp. 67–74.
- Johanson, B., Hutchins, G., Winograd, T. and Stone, M. (2002b). "PointRight: Experience with Flexible Input Redirection in Interactive Workspaces," *Proc. Symposium on User Interface Software and Technology*, Paris, France, pp. 227–234.
- Johnson, T. E. (1964). "Sketchpad III a Computer Program for Drawing in Three Dimensions," *Simulation*, Vol. 2 (3), June, pp. R3–R10.
- Kabbash, P., Buxton, W. and Sellen, A. (1994). "Two-Handed Input in a Compound Task," *Proc. CHI '94*, pp. 417–423.
- Kaiser, E., Olwal, A., McGee, D., Benko, H., Corradini, A., Li, X., Cohen, P. and Feiner, S. (2003). "Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Virtual Reality," *Proc. International Conference on Multimodal Interfaces (ICMI '03)*, Vancouver, BC, November 5-7, pp. 12–19.
- Kalman, R. E. (1960). "A new Approach to Linear Filtering and Predictive Problems," *Transactions ASME, Journal of basic engineering* (82), pp. 34–45.
- Kendon, A. (1994). "Do Gestures Communicate?: A Review," *Research of Language and Social Interaction*, Vol. 27, pp. 175–200.
- Khan, A., Fitzmaurice, G., Almeida, D., Burtnyk, N. and Kurtenbach, G. (2004). "A Remote Control Interface for Large Displays," *Proc. UIST '04*, Santa Fe, NM, pp. 127–136.
- Kita, S. (2003). *Pointing: Where Language, Culture and Cognition Meet*, Lawrence Erlbaum Associates.
- Kjeldsen, F. C. M. (1997). "Visual Interpretation of Hand Gestures as a Practical Interface Modality," *Ph.D. Thesis*, Department of Computer Science, Columbia University, New York, NY.
- Krauss, R. M. (1998). "Why Do We Gesture When We Speak?," *Current Directions in Psychological Science*, Vol. 7 (2), pp. 54–59.
- Krueger, M. W. (1982). *Artificial Reality*, Addison-Wesley, Reading, MA.

- Kumar, S., Cohen, P. and Levesque, H. J. (2000). "The Adaptive Agent Architecture: Achieving Fault-Tolerance Using Persistent Broker Teams," *Proc. International Conference on Multi-Agent Systems*, July, pp. 159–166.
- Kurtenbach, G. and Buxton, W. (1994). "User Learning and Performance with Marking Menus," *Proc. CHI '94*, pp. 258–264.
- Kurtenbach, G., Fitzmaurice, G., Baudel, T. and Buxton, W. (1997). "The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency," *Proc. CHI '97*, pp. 35–42.
- Lee, S., Buxton, W. and Smith, K. C. (1985). "A Multi-Touch Three Dimensional Touch-Sensitive Tablet," *Proc. CHI '85*, San Francisco, pp. 21–25.
- Leigh, J., Renambot, L., Johnson, A., Jeong, B., Jagodic, R., Schwarz, N., Svistula, D., Singh, R., Aguilera, J., Wang, X., Vishwanatha, V., Lopez Silva, B. A., Sandin, D., Peterka, T., Girado, J., Kooima, R., Gea, J., Long, L., Verlo, A., DeFanti, T., Brown, M. and Cox, D. (2006). "The Global Lambda Visualization Facility: An International Ultra-High-Definition Wide-Area Visualization Collaboratory," *Future Generation Computer Systems*, Vol. 22 (8), pp. 964–971.
- Liang, J. and Green, M. (1994). "JDCAD: A Highly Interactive 3D Modeling System," *Computers and Graphics*, Vol. 18 (4), pp. 499-506.
- Low, K. L., Ilie, A., Welch, G. and Lastra, A. (2003). "Combining Head-Mounted and Projector-based displays for Surgical Training," *Proc. IEEE VR '03*, pp. 110–117.
- MacIntyre, B. and Feiner, S. (1996). "Future Multimedia User Interfaces," *Multimedia Systems*, Vol. 4 (5), pp. 250–268.
- MacIntyre, B., Mynatt, E. D., Volda, S., Hansen, K. M., Tullio, J. and Corso, G. M. (2001). "Support for Multitasking and Background Awareness Using Interactive Peripheral Displays," *Proc. UIST '01*, pp. 41-50.
- MacKenzie, I. S. (1992). "Fitts' Law as a Research and Design Tool in Human-Computer Interaction," *Human-Computer Interaction*, Vol. 7, pp. 91–139.
- MacKenzie, I. S. and Jusoh, S. (2001). "An Evaluation of Two Input Devices for Remote Pointing," *Proc. International Conference on Engineering for Human-Computer Interaction*, pp. 235–250.
- MacKenzie, I. S. and Oniszczak, A. (1998). "A Comparison of Three Selection Techniques for Touchpads," *Proc. CHI '98*, Los Angeles, CA, USA, pp. 336–343.
- Mackinlay, J. D. and Heer, J. (2004). "Wideband Displays: Mitigating Multiple Monitor Seams," *Proc. Extended Abstracts CHI '04*, Vienna, Austria, pp. 1521–1524.

- MacWilliams, A., Sandor, C., Wagner, M., Bauer, M., Klinker, G. and Brüegge, B. (2003). "Herding Sheep: Live System Development for Distributed Augmented Reality," *Proc. IEEE and ACM Int. Symp. on Mixed and Augmented Reality (ISMAR'03)*, October 7–12, pp. 123–132.
- Malik, S., Ranjan, A. and Balakrishnan, R. (2005). "Interacting with Large Displays from a Distance with Vision-Tracked Multi-Finger Gestural Input," *Proc. UIST '05*, pp. 43-52.
- Mamladze, George (2004). *Processing Global Mouse and Keyboard Hooks in C#*, <http://www.codeproject.com/csharp/globalhook.asp> (accessed on March 20, 2007).
- Mapes, D. and Moshell, J. (1995). "The Two-Handed Interface for Object Manipulation in Virtual Environments," *Presence: Teleoperators and Virtual Environments*, Vol. 4 (4), pp. 403–416.
- Matsushita, N. and Rekimoto, J. (1997). "HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall," *Proc. UIST '97*, pp. 209–210.
- Microsoft, (2007). *Windows SideShow*, <http://www.microsoft.com/windows/products/windowsvista/features/details/sideshow.mspx> (accessed on April 4, 2007).
- Mine, M., R., Brooks, F., P. , Jr. and Sequin, C., H. (1997). "Moving Objects in Space: Exploiting Proprioception In Virtual-Environment Interaction," *Proc. ACM SIGGRAPH '97*, Los Angeles, CA, pp. 19–26.
- MoLAS (1994). *Archaeological Site Manual*, 3rd ed. Museum of London Archaeology Service, London.
- Moscovich, T. and Hughes, J. F. (2006). "Multi-Finger Cursor Techniques," *Proc. Graphics Interface*, pp. 1–7.
- Myers, B., Peck, C. H., Nichols, J., Kong, D. and Miller, R. (2001). "Interacting at a Distance using Semantic Snarfing," *Proc. Ubiquitous Computing '01*, Atlanta, GA, pp. 305–314.
- Nacenta, M. A., Sallam, S., Champoux, B., Subramanian, S. and Gutwin, C. (2006). "Perspective Cursor: Perspective-Based Interaction for Multi-Display Environments," *Proc. CHI '06*, Montreal, Canada, pp. 289–298.
- Nichols, J., Myers, B. A., Higgins, J., Harris, T. K., Rosenfeld, R. and Pignol, M. (2002). "Generating Remote Control Interfaces for Complex Appliances," *Proc. UIST '02*, Paris, France, pp. 27–30.

- Nintendo, (2006). *Nintendo DS Lite*, <http://www.nintendo.com/channel/ds> (accessed on January 10, 2007).
- Olwal, A., Benko, H. and Feiner, S. (2003a). "SenseShapes: Using statistical geometry for object selection in a multimodal augmented reality system," *Proc. IEEE and ACM Int. Symp. on Mixed and Augmented Reality (ISMAR '03)*, Tokyo, Japan, October 7–10, pp. 300–301.
- Olwal, A. and Feiner, S. (2003b). "The Flexible Pointer: An Interaction Technique for Selection in Augmented and Virtual Reality," *Proc. Conference Supplement of UIST '03*, Vancouver, BC, Canada, November 2–5, 2003, pp. 81–82.
- Opera Software ASA., (2007). *Opera*, <http://www.opera.com/> (accessed on April 1, 2007).
- Pavlovic, V., Sharma, R. and Huang, T. (1997). "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19 (7), pp. 677–695.
- Pierce, J. S., Forsberg, A., Conway, M. J., Hong, S., Zeleznik, R. and Mine, M. R. (1997). "Image Plane Interaction Techniques in 3D Immersive Environments," *Proc. Symposium on Interactive 3D Graphics (I3D '97)*, pp. 39–43.
- Pierce, J. S., Stearns, B. and Pausch, R. (1999). "Voodoo Dolls: Seamless Interaction at the Multiple Scales in Virtual Environments," *Proc. Symposium on Interactive 3D Graphics (I3D '99)*, pp. 141–145.
- Pinhanez, C. (2001). "Using a Steerable Projector and a Camera to Transform Surfaces into Interactive Displays," *Proc. CHI '01*, pp. 369–370.
- Potter, R. L., Weldon, L. J. and Shneiderman, B. (1988). "Improving the Accuracy of Touchscreens: An Experimental Evaluation of Three Strategies," *Proc. CHI '88*, Washington, DC, pp. 27–32.
- Poupyrev, I., Billinghurst, M., Weghorst, S. and Ichikawa, T. (1996). "The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR," *Proc. UIST '96*, pp. 79–80.
- Quek, F. K. H. (1995). "Eyes in the Interface," *Image and Vision Computing* Vol. 13 (6), pp. 511–525.
- Raskar, R., Brown, M. S., Yang, R., Chen, W.-C., Welch, G., Towles, H., Scales, B. and Fuchs, H. (1999). "Multi-Projector Displays Using Camera-Based Registration," *Proc. IEEE Visualization '99*, San Francisco, CA, USA, pp. 161–522.
- Realtimesoft, (2007). *UltraMon*, <http://www.realtimesoft.com/ultramon/> (accessed on March 20, 2007).

- Regenbrecht, H. T., Wagner, M. T. and Baratoff, G. (2002). "MagicMeeting: A Collaborative Tangible Augmented Reality System," *Virtual Reality - Systems, Development and Applications*, Vol. 6 (3), pp. 151–166.
- Rekimoto, J. (1997). "Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments," *Proc. UIST '97*, pp. 31–39.
- Rekimoto, J. (2002). "SmartSkin: an Infrastructure for Free-hand Manipulation on Interactive Surfaces," *Proc. CHI '02*, pp. 113–120.
- Rekimoto, J. (2004). "SyncTap: Synchronous User Operation for Spontaneous Network Connection," *Personal Ubiquitous Computing*, Vol. 8 (2), pp. 126–134.
- Rekimoto, J. and Saitoh, M. (1999). "Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments," *Proc. CHI '99*, Pittsburgh, PA, May 15–20, pp. 378–385.
- Ren, X. and Moriya, S. (2000). "Improving Selection Performance on Pen-Based Systems: A Study of Pen-Input Interaction for Selection Tasks," *ACM Transactions on Computer Human Interaction*, Vol. 7 (3), pp. 384–416.
- Ringel-Morris, M., Ryall, K., Shen, C., Forlines, C. and Vernier, F. D. (2004). "Beyond Social Protocols: Multi-User Coordination Policies for Co-located Groupware," *Proc. CSCW '04*, pp. 262–265.
- Ringel, M. (2003). "When One Isn't Enough: An Analysis of Virtual Desktop Usage Strategies and Their Implications for Design," *Proc. Extended Abstracts CHI '03*, pp. 762–763.
- Roberts, L. G. (1966). "The Lincoln Wand," *Proc. AFIPS Fall Joint Computer Conference*, pp. 223–227.
- Roman, M. and Campbell, R. H. (2000). "Gaia: Enabling Active Spaces," *Proc. ACM SIGOPS European Workshop*, pp. 229–234.
- Sandor, C., Olwal, A., Bell, B. and Feiner, S. (2005). "Immersive Mixed-Reality Configuration of Hybrid User Interfaces," *Proc. IEEE and ACM Int. Symp. on Mixed and Augmented Reality (ISMAR '05)*, Vienna, Austria, pp. 110–113.
- Schmalstieg, D., Fuhrmann, A. and Hesina, G. (2000). "Bridging Multiple User Interface Dimensions with Augmented Reality," *Proc. International Symposium on Augmented Reality (ISAR '00)*, Munich, Germany, pp. 20–29.
- Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavari, Z., Encarnação, L. M., Gervautz, M. and Purgathofer, W. (2002). "The Studierstube Augmented Reality Project," *Presence: Teleoperators and Virtual Environments*, Vol. 11 (1), February 2002, pp. 33–54.

- Schmandt, C. (1983). "Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Work Station," *Proc. SIGGRAPH '83*, pp. 253–261.
- Chris Schoeneman, (2007). *Synergy*, <http://synergy2.sourceforge.net> (accessed on March 22, 2007).
- Sears, A. and Shneiderman, B. (1991). "High Precision Touchscreens: Design Strategies and Comparisons with a Mouse," *International Journal of Man-Machine Studies*, Vol. 43 (4), pp. 593–613.
- Shen, C., Vernier, F. D., Forlines, C. and Ringel, M. (2004). "DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction," *Proc. CHI '04*, pp. 167–174.
- Sibert, L. E. and Jacob, R. J. K. (2000). "Evaluation of Eye Gaze Interaction," *Proc. CHI '00*, The Hague, Amsterdam, pp. 281–288.
- Slay, H. and Thomas, B. (2006). "Interaction and Visualization across Multiple Displays in Ubiquitous Computing Environments," *Proc. AFRIGRAPH '06*, Cape Town, South Africa, January 25–27, pp. 75–84.
- SMART Technologies, (2003). *DViT: Digital Vision Touch Technology*, <http://www.smarttech.com/dvit> (accessed on March 25, 2007).
- Smith, R. B. (1987). "Experiences with the Alternate Reality Kit: An Example of the Tension between Literalism and Magic " *Proc. SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface '87*, Toronto, Ontario, Canada, pp. 61–67.
- Starner, T. and Pentland, A. (1995). "Visual Recognition of American Sign Language Using Hidden Markov Models," *Proc. International Workshop on Automatic Face and Gesture Recognition*, pp. 189–194.
- Stefik, M., Bobrow, D. G., Foster, G., S., L. and Tatar, D. (1987). "WYSIWIS Revisited: Early Experiences with Multiuser Interfaces," *ACM Transactions on Office Information Systems*, Vol. 5 (2), April, pp. 147–167.
- Stoakley, R., Conway, M. and Pausch, R. (1995). "Virtual Reality on a WIM: Interactive Worlds in Miniature," *Proc. CHI '95*, Denver, CO, May 7–11, pp. 265–272.
- Streitz, N., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P. and Steinmetz, R. (1999). "i-LAND: An Interactive Landscape for Creativity and Innovation," *Proc. CHI '99*, pp. 120–127.
- Streitz, N., Prante, T., Müller-Tomfelde, C., Tandler, P. and Magerkurth, C. (2002). "Roomware - The Second Generation," *Proc. UbiComp '02*, Goteborg, Sweden, pp. 77–78.

- Sturman, D. J. (1992). "Whole-hand Input," *Thesis*, Massachusetts Institute of Technology, Boston, MA.
- Su, R. and Bailey, B. P. (2005). "Put Them Where? Towards Guidelines for Positioning Large Displays in Interactive Workspaces," *Proc. Graphics Interface '05*, Victoria, British Columbia, pp. 337–349.
- Sutherland, I. (1968). "A Head-Mounted Three Dimensional Display," *Proc. Fall Joint Computer Conference '68*, Washington, DC, pp. 757–764.
- Swindells, C., Inkpen, K. M., Dill, J. C. and Tory, M. (2002). "That one there! Pointing to establish device identity," *Proc. UIST '02*, pp. 151–160.
- Synaptics Inc., (2007). *Capacitive Position Sensing*,  
<http://www.synaptics.com/technology/cps.cfm> (*accessed on March 11, 2007*).
- Szalavári, Z. and Gervautz, M. (1997). "The Personal Interaction Panel: A Two-Handed Interface for Augmented Reality," *Proc. EUROGRAPHICS '97*, Budapest, Hungary, pp. 335–346.
- Tan, D., Meyers, B. and Czerwinski, M. (2004). "WinCuts: Manipulating Arbitrary Window Regions for More Effective Use of Screen Space," *Proc. Extended Abstracts CHI '04*, Vienna, Austria, pp. 1525–1528.
- Tan, D. S. and Czerwinski, M. (2003). "Effects of Visual Separation and Physical Discontinuities when Distributing Information across Multiple Displays," *Proc. Interact '03*, pp. 252–255.
- Tandler, P., Prante, T., Müller-Tomfelde, C., Streitz, N. and Steinmetz, R. (2001). "Connectables: Dynamic Coupling of Displays for the Flexible Creation of Shared Workspaces," *Proc. UIST '01*, pp. 11–20.
- Tani, M., Horita, M., Yamaashi, K. and Tanikoshi, K. (1994). "Courtyard: Integrating Shared Overview on a Large Screen and Per-user Detail on Individual Screens," *Proc. CHI '94*, pp. 44–50.
- TechBlog, (2006). *Toshiba's Dual-Display eBook*  
<http://www.techeblog.com/index.php/tech-gadget/toshibas-dual-display-ebook>  
(*accessed on March 10, 2007*).
- Ullmer, B., Ishii, H. and Glas, D. (1998). "mediaBlocks: Physical Containers, Transports, and Controls for Online Media," *Proc. SIGGRAPH '98*, July, pp. 379–386.
- Underkoffler, J. and Ishii, H. (1999). "Urp: A Luminous-Tangible Workbench for Urban Planning and Design," *Proc. CHI '99*, Pittsburg, PA, May 15-20, pp. 386–393.



- Vickers, D. L. (1972). "Sorcerer's Apprentice: Head-Mounted Display and Wand," *Ph.D. Thesis*, Department of Electrical Engineering, University of Utah, Salt Lake City, UT.
- Vincent, V. J. (1993). "Mandala: Virtual Village," *Proc. SIGGRAPH '93*, pp. 207–208.
- Walker, N. and Smalcer, J. B. (1990). "A Comparison of Selection Times from Walking and Pull-Down Menus," *Proc. CHI '90*, Seattle, WA, USA, pp. 221–225.
- Ware, C., Arthur, K. and Booth, K. S. (1993). "Fish Tank Virtual Reality," *Proc. INTERCHI '93*, pp. 37–42.
- Weiser, M. (1991). "The Computer for the 21st Century," *Scientific American*, Vol. 3 (265), pp. 94–104.
- Weiser, M. (1993). "Some Computer Science Issues in Ubiquitous Computing," *Communications of the ACM*, Vol. 36 (7), pp. 75–84.
- Wilson, A. (2004). "TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction," *Proc. ICMI '04*, pp. 69–76.
- Wilson, A. (2005a). "PlayAnywhere: A Compact Tabletop Computer Vision System " *Proc. UIST '05*, Seattle, WA, pp. 83–92.
- Wilson, A. (2006). "Robust Vision-Based Detection of Pinching for One and Two-Handed Gesture Input," *Proc. UIST '06*, Montreux, Switzerland, pp. 255–258.
- Wilson, A. and Cutrell, E. (2005b). "FlowMouse: A Computer Vision-Based Pointing and Gesture Input Device," *Proc. Interact '05*, pp. 565–578.
- Wilson, A. and Pham, H. (2003a). "Pointing in Intelligent Environments with World-Cursor," *Proc. Interact '03*, Zurich, Switzerland, pp. 495–502.
- Wilson, A. and Shafer, S. (2003b). "XWand: UI for Intelligent Spaces," *Proc. CHI '03*, Ft. Lauderdale, FL, USA, pp. 545–552.
- Wingrave, C., Haciahetoglu, Y. and Bowman, D. (2006). "Overcoming World in Miniature Limitations by a Scaled and Scrolling WIM," *Proc. IEEE Symposium on 3D User Interfaces (3DUI '06)*, pp. 11–16.
- Wu, M. and Balakrishnan, R. (2003). "Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays," *Proc. UIST '03*, pp. 193–202.
- Wyss, H. P., Blach, R. and Bues, M. (2006). "iSith - Intersection-based Spatial Interaction for Two Hands," *Proc. IEEE Symposium on 3D User Interfaces (3DUI '06)*, Alexandria, VA, USA, pp. 59–61.

Zelevnik, R. C., Herndon, K. P. and Hughes, J. F. (1996). "SKETCH: An Interface for Sketching 3D Scenes," *Computer Graphics*, Vol. 30 (Annual Conference Series), pp. 163–170.

Zhai, S., Morimoto, C. and Ihde, S. (1999). "Manual and Gaze Input Cascaded (MAGIC) Pointing," *Proc. CHI '99*, Pittsburgh, PA, USA, pp. 246–253.

Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S. and Harvill, Y. (1987). "A Hand Gesture Interface Device," *Proc. SIGCHI/GI '87*, Toronto, Ontario, Canada, pp. 189–192.