

DESIGNS FOR SINGLE USER, UP-CLOSE INTERACTION WITH
WALL-SIZED DISPLAYS

by

Anastasia Bezerianos

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Copyright © 2007 by Anastasia Bezerianos

Abstract

Designs for single user, up-close interaction with wall-sized displays

Anastasia Bezerianos

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

2007

The decreasing cost of projectors has enabled the creation of high resolution wall-sized displays by tiling multiple projectors together. Wall displays are interesting from an interaction perspective since their extended surface and high resolution allow the presentation of large amounts of data that may be viewed and interacted upon at close proximity. These characteristics also give rise to some unique challenges that we explore in three interaction design explorations: organizing the layout of content spread across large distances can be tiresome, repeated reaching for simple selection tasks can be tedious, while tracking visual changes across the display surface may prove problematic.

In the first design exploration, layout management and context switching is achieved using alternative views of different areas on the display. We introduce a new framework, used first to instantiate examples of existing techniques, and then as a catalyst for developing three new ones. The most promising new technique was preferred to traditional approaches by the majority of users when performing a layout management, comparison and context switching task.

In the second design exploration we present and evaluate in a pilot user study a set of new techniques, which facilitate access to areas of the display that are difficult to reach without significant physical movement, by copying remote content close to the user. Through our findings we identify desirable design properties specifically for remote reaching designs, used to refine our most promising technique. The refined technique

outperformed existing approaches in multiple target selection tasks.

Our last exploration aids the identification and understanding of dynamic visual changes happening out of the user’s sight. We propose a mechanism for storing and revealing hidden changes, explore its design space, and perform an informal user feedback session on a wall display prototype. Users preferred the technique to no buffering for a game task and chose the buffered history to be presented both statically and as an animation.

Individually, each design exploration contributes significantly to the specific interaction aspect it investigates. Together, they demonstrate how careful design may lead to general-purpose techniques that can benefit any type of application performed on wall-sized displays.

to all my families

Acknowledgements

This thesis has been for years in the making and so the list of people to thank has almost reached the size of the thesis itself. But I will attempt to cover everyone by abstracting and generalizing, as a proper researcher.

I have been extremely fortunate as a student to work with a researcher such as my supervisor Ravin Balakrishnan, who always provided me with insightful comments, demonstrated how research should be done, and most importantly constantly challenged me to do better. My supervisory committee, Ron Baecker, Mark Chignell and Karan Singh, have painstakingly gone over every detail of my thesis, always made invaluable suggestions, asked the right questions over the years, and allowed me to approach my work from different points of view. I can never thank them enough, nor Khai Truong, my latest committee member, for reading my entire thesis in such short notice and giving me excellent feedback. I would also like to thank my external examiner Ken Hinckley for thoroughly going over my thesis and for all his insightful advice on how to improve it. The last person who greatly influenced my thesis work and has been a pleasure to work with is Pierre Dragicevic. It has been a true privilege to work with these brilliant researchers and exceptionally kind people.

Being a DGPier has been an amazing experience and every person in the lab has played a role in my always feeling welcome and immensely supported. But I'd like to especially thank Petros Faloutsos, Joe Laszlo and Mike Neff for helping me out when I first approached the group. Shahzad Malik and Abishek Ranjan for company on long working nights. John Hancock, Gonzalo Ramos and Noah Lockwood for always helping out even in times of pressure and last but not least, Dan Vogel for his friendship, discussions, advice and a unique view on things.

My friends from Greece, who will never read this: Katerina Koukouvaou, Eleni Kounani, Petros Raptis, Manolis Fakinos, George Papadopoulos, Manos Aligizakis, George Koletsos, Yiannis Sarris and Sofia Kousidou. They have helped me over the years to see

the humor in things and in myself (and whom I have proved wrong by graduating before my kids), and have always made me feel as if I never left home.

My dear friends in Toronto always made my life fun, interesting and are the reason I love this city: Yannis Velegrakis, Periklis Andritsos, Themis Palpanas and Stratis Ioannidis for their continuing support and friendship over the years. Rozalia Christodouloupoulou, with a heart as large as her name. Cristiana Chitic, Dana Rosu, Matei David and Raluca Cojocariu my all-time favorite Romanian gang. Theo Garefalakis for always being awake before me and Panayotis Tsaparas for always staying up later than me. George Chalkiadakis, Angeliki Maglara and Vasso Bartzorka for stimulating talks and long philosophical arguments at Future cafe. Sam Hasinoff and Paco Estrada for great justice. Apostolos Dimitromanolakis, Maher Abdulrahim, Bowen Hui, George Giakkoupis, Grace Wang, Andreas Polyrakis, Leoni Ioannidu, Michael Flouris, Vaggelis Markakis, Nora Jantschukeite, Costis Georgiou, Vassilios Tzerpos and Melanie Baljko for brightening up my days and nights in Toronto over the years. And finally my beloved reading companions around cozy cafes and personal thesis writing support group. Tasos Kementsetsidis and Aktina Stathaki who were the best of friends and made me laugh so hard I thought I would faint, and my dear white couches: Sef Stefanos Karterakis, my photographer, stylist and rock to lean on; Stavros Vassos, my gaming companion, musician and emotional balancer; and Fanis Tsantilas (the 2nd largest source of information after wikipedia, as of the writing of this thesis), my constant calmness provider and discussion partner on issues like hci research, art, life, the universe and everything.

Last in this long list are the most influential people in my life. My loving parents Nikolas and Georgia who have been my heros, friends, strongest supporters and the reason I am me. My sister Ioanna, my lifelong friend and companion. And my partner in everything, George Katsirelos, without whose unconditional love, friendship, understanding and support this thesis would not be.

Refereed publications from this thesis

Chapter 2,3: The description of the Canvas portal framework and instantiation of existing context switching techniques (Section 3.2), as well as the presentation of the newly proposed instances (Sections 3.3) are published in a slightly altered form in:

"Canvas Portals: View and Space Management on Large Displays". Anastasia Bezerianos, Ravin Balakrishnan. In IEEE Computer Graphics & Applications, Special Issue on Applications of Large Displays, vol.25, no.4, July/August 2005, pages 34-43.

The usability evaluation of ScalView portal instance (Section 3.5) and the discussion on potential applications for wall displays (Section 2.2) will be published in:

"Using alternative views for layout, comparison and context switching tasks in wall displays". Anastasia Bezerianos. In Proceedings of OzCHI 2007 - Australasian Computer-Human Interaction Conference.

The remaining sections are preliminary design explorations (Section 3.1) and further evaluation sessions of the ScaleView portal instance (Sections 3.4,3.5), not published.

As of the submission of this thesis, video of canvas portals techniques described in chapter can be found under:

<http://www.dgp.toronto.edu/~anab/videos/largeWall-several-tec.avi>

Chapter 4: The identification of desirable design goals for remote reaching techniques and the description of the redesigned vacuum (Sections 4.1,4.5), as well as its evaluation (Section 4.6) are published in a slightly altered form in:

"The Vacuum: Facilitating the manipulation of distant objects". Anastasia Bezerianos, Ravin Balakrishnan. In Proceedings of CHI 2005 - the ACM Conference on Human Factors in Computing Systems, pages 361-270.

The remaining sections are preliminary design explorations and their evaluations (Sections 4.2,4.3,4.4), not published.

As of the submission of this thesis, video of the vacuum described in chapter can be found under:

<http://www.dgp.toronto.edu/~anab/videos/chi2005/high-res-vacuum.mov> while primary designs appear in the canvas portals video (<http://www.dgp.toronto.edu/~anab/videos/largeWall-several-tec.avi>).

Chapter 5: The entire chapter (identification of issues, exploration of the Mnemonic Rendering design space and user feedback session on a prototype) are published in a slightly altered form in:

"Mnemonic Rendering: An Image-Based Approach for Exposing Hidden Changes in Dynamic Displays". Anastasia Bezerianos, Pierre Dragicevic, Ravin Balakrishnan. In Proceedings of UIST 2006 - the 19th ACM Symposium on User Interface Software and Technology, pages 159-168. Best Student Paper Award.

As of the submission of this thesis, video mnemonic rendering described in chapter can be found under <http://www.dgp.toronto.edu/~anab/videos/mnemonic.avi> and demos of a desktop prototype under <http://www.dgp.toronto.edu/~anab/mnemonic>

Contents

1	Introduction	1
1.1	Thesis contributions	5
2	Background	7
2.1	Interaction with wall displays	8
2.1.1	Remote interaction	8
2.1.2	Up-close interaction	11
2.2	Applications for interactive wall displays	12
2.3	Interaction tasks on wall displays	15
2.4	Interaction challenges	16
2.5	Designs addressing the identified challenges	19
2.5.1	Viewing and interacting with all areas of the display for layout managements and context switching	19
2.5.2	Bridging large interaction distances for remote reaching	21
2.5.3	Invisible content monitoring	26
2.6	Design goals	27
2.7	Chapter summary and closing remarks	30
2.7.1	Prototyping	31
2.7.2	Common procedures followed in statistical analysis	31

3	Accommodating Layout Management and Context Switching	33
3.1	Initial observations	35
3.2	Canvas portals framework	36
3.2.1	Instantiating existing techniques within canvas portals	42
3.3	New canvas portal instances	46
3.3.1	ScaleView portals	46
3.3.2	Window portals	50
3.3.3	Division Bands	52
3.3.4	Preliminary user feedback and design improvements	55
3.4	ScaleView portal remote area visualization study	58
	Task	58
	Visualization conditions (Figure 3.9)	58
3.4.1	Familiar portal layout session	60
	Results	61
	Discussion and user comments	62
3.4.2	Unfamiliar portal layout session	64
	Results	65
	Discussion and user comments	65
3.4.3	Design implications	66
3.5	ScaleView portal usability study	66
3.5.1	Observations	68
3.5.2	User comments and design improvements	71
3.6	Chapter summary and closing remarks	76
4	Facilitating Remote Reaching	81
4.1	Drag-and-pop/pick	82
4.2	Preliminary remote reaching designs	85
4.2.1	The vacuum tool	87

4.2.2	The edge reaching tool	92
4.2.3	The scroll ray	95
4.3	First user feedback session on designs	98
4.4	Second user feedback session on designs	100
4.4.1	Results (Figure 4.13)	102
	Discussion	104
	User comments and design guidelines	104
4.5	Redesigning the Vacuum	107
4.5.1	Visual design	109
4.5.2	Layout of Proxies	110
4.5.3	Invocation and Adjustment	111
4.5.4	Dismissal and Multiple Operations	115
4.6	Redesigned vacuum evaluation	117
4.6.1	Single target selection study	117
	Results and Discussion	120
4.6.2	Multiple target selection study	125
	Results and Discussion	129
4.6.3	Discussion	133
4.6.4	Potential refinements	136
4.7	Chapter summary and closing remarks	138
5	Presenting Invisible Changes	141
5.1	Existing approaches	142
5.2	An image based approach	145
5.2.1	Surfaces and Pixel Visibility	145
5.3	Mnemonic Rendering	148
5.4	Mnemonic rendering design space	149
5.4.1	Detecting visibility on wall displays	149

5.4.2	Restitution techniques	151
	Initial user response session	152
5.4.3	Visual Effects	155
5.4.4	Timing	157
5.4.5	Ordering	157
5.4.6	Grouping	159
5.4.7	Interactivity	161
5.5	The Mnemonic Wall	163
5.6	User feedback	163
5.6.1	Results	166
5.7	Chapter summary and closing remarks	168
6	Conclusions and future directions	170
6.1	Combining the three designs	173
6.1.1	Canvas portals and vacuum	174
6.1.2	Canvas portals, vacuum and mnemonic rendering	175
6.2	Contributions	177
6.3	Future directions	179
6.3.1	Layout management	179
6.3.2	Remote reaching	181
6.3.3	Presentation of invisible changes	182
6.3.4	Beyond these designs	184
	Bibliography	186

List of Figures

1.1	The Large Display Wall in DGP lab. The wall display is 5 x 2 m (16' x 6') with an effective resolution of 6144 x 2304 pixels.	2
2.1	(a,b) <i>Applications on physical walls</i> : (a) Director Peter Weir editing storyboards for the film "Fearless" (http://www.peterweircave.com/). (b) Design board of the Sniff-Receptor art installation (http://jan.moesen.nu/media/photos/2004/01/sniff-receptor/). (c,d,e) <i>Wall displays</i> : (c) A war room command and control center (Air Force Research Laboratory http://www.afrlhorizons.com/Briefs/Sept01/IF0012.html). (d) An electronic classroom in Georgia Tech (eClass project http://www.cc.gatech.edu/fce/c2000/pubs/). (e) A command and monitoring center of the UK Highways agency (http://www.electrosonic.com/).	14
2.2	Frisbee (a) Dragging a remote image to local space by exiting the frisbee through a transfer channel and (b) the result. (c) Dragging a local image to the remote space by entering the frisbee through a channel and (d) the result. (Image from Khan et al. [2004])	20
2.3	Bubble cursor. (a) Cursor size is such that only one target is captured at a time. (b) Cursor envelops the target when not completely contained by the main bubble. (Image from Grossman and Balakrishnan [2005])	22
2.4	The user moves content by virtually "picking" it in his PDA or stylus. (Image from Rekimoto [1998])	22

2.5	An instance of a throwing technique, drag-and-throw [Hascoët, 2003]. The user first initiates a drag. Then, the visual clues appear (take-off area, trajectory). The user adjusts her throw keeping the cursor in the take-off area, and finally ends the drag and the object is thrown on the selected target. (Image from http://www.lirmm.fr/edel/dragging/index.php)	24
2.6	Drag-and-pop [Baudisch et al., 2003a]: the user begins a drag-and-drop operation. After dragging beyond a threshold, possible targets (of compatible type, in the direction of the move) pop near the user cursor. The user terminates the drag-and-drop operation by releasing the dragged item on one of the possible targets. After the user has dropped the icon, the popped targets disappear. (Image from http://www.patrickbaudisch.com/projects/dragandpop/demo/index.htm)	25
2.7	Best and Preferred Work Zone as suggested by the Occupational Safety & Health Administration to prevent musculoskeletal disorders for Retail Grocery Stores. (Image from http://www.osha.gov/ergonomics/guidelines/retailgrocery/retailgrocery.html)	29
3.1	Canvas portal position, portal area and focal area example. The <i>position</i> of the rectangular portal is on the display, at the lower left. The <i>portal area</i> (interior of the portal) presents a view of the remote <i>focal area</i> . The position of the portal (and so the position of the <i>portal area</i>) as well as the focus of the portal (<i>focal area</i>) can be changed.	37

3.2	Canvas portal shape and attribute example. The left column shows the portal widgets (ie the portal areas of the canvas portals), which can exist on the main display; the right shows the remote focal areas on the displays. (a) Rectangular canvas portal with scale attribute modified to act as a zoom lens to the focal area. (b) Rectangular canvas portal with modified time attribute displaying a previous state of the focal areas content. (c) Circular canvas portal with semantic attribute modified to show only user interface elements, creating a tool palette.	38
3.3	Images on the left are canvas portal widgets (that can exist on the main display canvas); those on the right are remote focal areas of the portals on the display. (a) A canvas portal with a zoomed-in view of the remote location. (b) The user selects an object inside the portal. A semi-transparent rendering of the object beyond the portal indicates that the movement can be continued outside the portal. (c) The user has moved the object, that has also moved in the remote focal area on the right. (d) As the object passes the portal's boundary, it transitions to the main display canvas on the area close to the location of the portal, and disappears from the remote focal area on the left. A semi-transparent rendering of the object in the portal indicates that the user can transition the object back inside the portal and thus "transport" it to the remote focal area again.	41
3.4	Instantiations of existing techniques and newly designed ones (*).	45

3.5	A ScaleView portal is an instance of a canvas portal. The ScaleView portal is located at the bottom left corner of the display. The red pin icon on the top-left corner indicates that the ScaleView portal can be moved. In this instance the portal is not moving, thus the pin icon is "pinned". The black rectangle at the side represents a thumbnail view of the virtual display canvas, within which the yellow rectangle indicates the focal area's position on the main canvas, and the red rectangle indicates the ScaleView portal's position. An outline rectangle on the main display canvas represents the focal area.	47
3.6	Grouping of multiple ScaleView portals: (a) stacked, (b) side-by-side, and (c) cube layout. Clicking on a portal brings it to the top.	49
3.7	Window portal on a large display. An overview region on the left of the Window portal (a ScaleView portal depicting the entire display) serves as a spatial locator, a temporal locator (timeline) is on top of the widget, while the selected object is on the bottom right. Red borders around the selected object in all views help maintain context.	51
3.8	Division Bands	54
3.9	Visualization of remote focal area connections. (a) NoVis. The desired target is marked as red. The different portals are green rectangles, corresponding to 4 different remote focal areas on the display. (b) FullPortalVis. The remote focal areas of all portals and their connections are highlighted. (c) PortalVis. As the user hovers over a portal, the corresponding remote focal area and its connection is highlighted. (d) LineVis. As the user hovers over a proxy within a portal, a line connecting the proxy and the corresponding remote target is drawn. 4 portals are seen and 50% of the display is covered by distracters.	60

3.10	Mean <i>TotalTime</i> per visualization for the Familiar (Left) and Unfamiliar (Right) layout study. Time spent hovering in PortalVis and LineVis are indicated.	62
3.11	Mean <i>TotalTime</i> and <i>ErrorRate</i> per visualization for the Familiar (Left) and Unfamiliar (Right) layout study. Statistically significant measures are marked in red	62
3.12	95% of confidence interval graph of <i>ErrorRate</i> per visualization for the Familiar (Left) and Unfamiliar (Right) layout study.	63
3.13	Mean <i>TotalTime</i> per visualization for the different number of portals. For both the Familiar (Left) and Unfamiliar (Right) layout study.	63
3.14	Percentage of time spent looking inside ScaleView portals for users per task.	72
3.15	Multiple ScaleView portals. In the stacked layout (left) one minimap and one set of scroll-bars is present, that of the top portal. In the side-by-side layout each portal has its own minimap and scroll-bars.	73
3.16	Redesigning ScaleView portal transitions. As an item approaches, the border becomes transparent to highlight the crossing functionality.	75
3.17	ScaleView portals usability evaluation: Layout task setup and interaction footprint for each user labeled (i - x). Red squares indicate the users visual focus. Blue marks denote interactions with regular magnification lens(es), Green ones with ScaleView portals and Black marks are other interactions on the canvas.	78
3.18	ScaleView portals usability evaluation: Search task setup and interaction footprint for each user labeled (i - x). Red squares indicate the users visual focus. Blue marks denote interactions with regular magnification lens(es) and Green ones with ScaleView portals.	79

3.19	ScaleView portals usability evaluation: Comparison task setup and interaction footprint for each user labeled (i - x). Red squares indicate the users visual focus. Blue marks denote interactions with regular magnification lens(es) and Green ones with ScaleView portals.	80
4.1	Drag-and- <i>pop</i> sequence. (1) In regular desktop with icons, the user selects (2) a webpage icon and moves it towards a remote cluster of application icons. The user starts dragging a file icon towards a remote cluster of application icons. Compatible application icons are copied close to the user's location, in a condensed layout. (3) The user drops the moving file icon on the desired application icon (recycle bin), concluding the dragging operation (4).	83
4.2	Vacuum widget. The user can click and drag on: the pin icon to reposition the widget, the start wedge to change the start extent of the arc, and the angle control to change the arc's angle of influence.	87
4.3	Vacuum interaction. (Top) First invocation: a) After the menu selection the vacuum tool is invoked, with the angle of influence at 0° in a vertical position. b) When the user touches/clicks inside the widget, the angle of influence is opened clockwise up to the cursor position. The extension of the angle is visible, until the user lifts her pen c). (Bottom) Angle manipulation: d) When the user selects the start wedge, the extension of the angle of influence is displayed. e) The user can drag the wedge to alter the direction of the angle, and as she lifts her pen f) interaction with the tool stops and the effect of the arc of influence is no longer visible. . . .	88

- 4.4 Vacuum interaction, resetting the angle: a) After the menu selection the vacuum tool is invoked, with the angle of influence at 0° in a vertical position. b) When the user touches/clicks inside the widget, the angle of influence is opened clockwise up to the cursor position. c) The user drags the angle of influence until it reaches 360° d). When the 360° mark is crossed e), the angle resets to 0° 89
- 4.5 Top: Full screen view of stacking vacuum. Items inside the influence of the vacuum are presented as proxies stacked on top of the vacuum. Red lines show connections between items and their proxies and (for illustration only). Bottom: Close-up of the stacking vacuum. As the arc of influence is increased to include the window item (left image), a copy of the item is brought to the tool's centre (right). Motion blur in right image illustrates the animated transition that occurs during the item's movement. 90
- 4.6 Top: Full screen view of the scaling vacuum. Items inside the influence of the vacuum are presented as proxies scaled down. Blue lines show connections between items and their proxies. Bottom: Close-up of the scaling vacuum. As the arc is increased to include the window item (left image), a copy of the item is scaled down and brought to the tool's centre (right image). Motion blur in right image illustrates the animated transition that occurs during the item's movement. 91
- 4.7 Using the edge reaching tool. (a) The widget is present at the bottom left of the image. At the top right is the top-right corner of the display. (b) From the 6 selector areas of the grid on the widget, the top right one is selected. This thumbnail area corresponds to the top right area of the display. (c) The corresponding display area and proxies of all the items in the area are scaled down, surrounding the tool. 93

- 4.8 a) Full screen view of the edge reaching tool. The tool (yellow widget at center of image) is divided in 6 regions representing corresponding regions on the display (yellow regions at edges of the display). When a region is selected in the edge reaching tool, proxies of the items in the corresponding remote region on the display are "pulled" close to the user, surrounding the tool. Here two regions are selected and are highlighted both on the tool and on the display with more prominent border colors. Black lines depict the connection between proxies and remote objects and are for illustration only. b) Use of hover in the edge reaching too: when the user hovers over the bottom right grid cell of the tool, it and the corresponding display cell gets highlighted in green. 94
- 4.9 (Top) Movement of the edge reaching tool, influence remains the same. The tool moves from left (a) to right (b) on the display. As the selected thumbnail cells have not changed, the copied proxies are the same. (Bottom) Movement of the scaling vacuum, influence moves with the tool. As the tool moves from bottom (c) to top (d), the arc of influence changes position and thus different proxies are copied close to the user. Black arrows indicate the movement of the tool. Red borders highlight items that are copied close to the tool. Arrows and borders are for illustration. . . . 95
- 4.10 Scroll ray widget. a) When invoked the tool is placed in a vertical position. b) The user may alter the angle of the ray by dragging outside the length of the interactive area of the widget. All intersected items are proxy candidates and highlighted in colored borders. c) By default the furthest candidate is copied close to the user. d) Candidate items closer to the user can be reached by scrolling towards the center of the tool, while e) candidates farther from the user by scrolling towards the edges of the display. 96

4.11	Design characteristics of exploration remote reaching designs.	98
4.12	Sample tasks. (Left) Area distracters (highlighted in pink) are Many (top), None (middle) and Few (bottom), with distance Mid from start (red) to goal target (green). The density of path distracters is fixed to Few. (Right) Path distracters (blue) are Many (top), None (middle), Few (bottom). The density of area distracters is fixed to Few. The distance from start (red) to goal target (green) is Far. In the actual session the only highlighted targets were the goal (green) and starting target (red). .	102
4.13	Performance measures: Mean <i>SelectionTime</i> (sec) and <i>ErrorRate</i> per technique. No statistical significant effects were present.	103
4.14	Location of proxies in edge reaching tool. (Left) The original highlighted object is at the user's right, while its proxy is directly in front. (Right) After the tool has moved, the same object is on the user's left, while its proxy is still directly in front.	105
4.15	Angle inconsistencies in the scale vacuum. After the initial invocation the fastest way to reach items at the bottom left of the display (left) results into a different angle than that of reaching items at the bottom right (right).	106
4.16	The vacuum	110
4.17	Vacuum invocation and adjustment. Blue arrows indicate cursor movement. (a) Dragging begins in the centre of the display. (b) When the drag exceeds a threshold, the vacuum is invoked with an 20° initial arc, bringing proxies of targets 3, 5 towards its center. (c) Additional cursor movement increases the arc angle, until 120°. (d) Change in cursor direction through the center changes the direction of the arc. (e, f) Cursor moving beyond the arc's edges expands the arc.	113

4.18	Vacuum arc adjustment, maximum angle. (a, b) Moving the cursor beyond one edge moves it, expanding the arc. (c) When both edges meet, the arc is fully expanded. (d) Reversing the cursor direction reopens the arc. . .	114
4.19	Vacuum arc adjustment, minimum angle. (a, b) Moving the cursor beyond one edge moves it, expanding the arc. (c) Changing direction collapses the arc, until the minimum angle is reached. (d) Exiting the minimum angle reopens the arc.	114
4.20	Hover and dismissal. (Left) The vacuum is active and adjustable when the pen is on the surface. (Middle) When the pen is in the hover zone, the vacuum is active but not adjustable and fades out as the pen is lifted. (Right) When the pen exceeds the hover zone's threshold, the vacuum is dismissed with a popping animation.	116
4.21	Single Target experimental setup. The red square is the start target and the number 1 is the desired target. Three setups are presented: (top) Mid distance with Few path distracters, (middle) Far distance with None path distracters and (bottom) Far distance with Many path distracters. The green boxes are for illustration only.	119
4.22	Time and error related measures of the single target selection study. Significant effects are marked with red.	120
4.23	Boxplots (median,range) of TotalTime (top) and SelectionTime (bottom) broken down by technique, distance (Mid, Far), and path distracters (None, Few, Many).	121
4.24	Cost of dismissing the vacuum, by exiting the hover zone threshold. . . .	124

4.25	Multiple Target experimental setup. The red square is the start target and numbers 1-5 are the desired targets. Three setups are presented: (top) Far distance with Few layout distracters, (middle) Mid distance with None distracters and (bottom) Close distance with Many distracters. The green boxes are for illustration only.	128
4.26	Time and error related measures of the multiple target selection study. Significant effects are marked with red.	129
4.27	Total Time (top) and Adapted Total Time (bottom) broken down by technique, distance (Close, Mid, Far), and distracters (None, Few, Many). . .	132
4.28	Push-and-pop. (1) As the user drags a document, (2) the system surrounds the pointer with copies of all relevant application icons in a condensed layout. (3) The user drops the document on the Recycle Bin icon and (4) the copies disappear. (Image from Collomb et al. [2005a])	135
5.1	A pixel might be invisible to the user because: (a) of occlusion by another window, (b) it is on a virtual canvas outside the current display area, (c) its window is minimized; or it is displayed but not seen by the user due to: (d) physical occlusion, (e) interruption, (f) being outside the focus of attention.	147
5.2	Mnemonic rendering. a) A pixel is visible and rendered normally. b) If the pixel becomes invisible, its visual changes get buffered. c) When the pixel becomes visible again, the stored visual changes get restituted or reviewed to the user, d) until they reach the current visual state of the pixel. . . .	148
5.3	Mnemonic wall. A prototype of mnemonic rendering on a wall display. . .	149
5.4	All changes occurring since a pixel is hidden (1) are buffered (2). Removing the occluding element (3), results in restitution of the entire buffered history, blended (persistence) or played back (flashback) (4).	151

5.5	Two examples of using our restitution techniques. (a) Persistence, (b) Flashback. Notice how in persistence motion changes (gauge dial on the left) leave behind a motion trail, but in non-continuous changes a less informative blurring occurs (weather icons on the right).	153
5.6	Assume a moving dial that is partially hidden. (Left) Time-warping with the flashback technique: The gauge is originally partly hidden (area of red pixels). As it is slowly uncovered (from right to left), pixels are gradually revealed and start their restitution at different times (red pixel saturation indicates buffered history left for restitution- redder pixels have more history to restitute). So pixels closer to the center of the gauge begin their restitution before pixels further away. (Right) Visual break with the flashback technique: The gauge is originally partly hidden (area of red pixels). It suddenly becomes entirely visible again. The previously hidden pixels are "out of sync" with the pixels that were visible. In both cases red shading conveys pixel age.	154
5.7	Different visual effects for rendering buffered history. Restitution of a moving ball. <i>Simple</i> rendering of all buffered history at full opacity. <i>Constant blending</i> provides a constant transparent rendering of previous history states, while <i>weighted blending</i> is based on history timestamps and results in fading-in trails. Adding <i>color</i> may also indicate the areas that are currently displaying buffered information, while <i>de-saturation</i> communicates the feeling of oldness in buffered history.	156
5.8	Control dial operated remotely by the user to control restitution playback in a wall display. The visual representation of the dial is rendered in the location the user is focusing on (left), while interaction takes place close to the user (right).	162

5.9	The Mnemonic Wall with pixel persistence: visual changes outside the user's central vision leave motion trails that will fade out as the user glances at them. The user is wearing a hat with passive markers mounted. On the top of the image infrared Vicon motion capturing cameras can be seen.	164
5.10	Layout matching game on the Mnemonic Wall, using pixel persistence. The red lines and arrow are there to emphasize our approximation of the user's visual focus based on head tracking.	165

Chapter 1

Introduction

Computer input and output technology has evolved in the last decades, allowing for computer environments that range from really small mobile displays of a few centimeters (for example digital watches, cell phones and PDAs) to very large embedded displays of several meters (like digital whiteboards, public interactive displays and digital tabletops), with trends towards even smaller and larger display sizes [Fitzmaurice et al., 2003].

In particular, large sized displays have emerged as valuable tools in visualization and interaction with medical data [Hibbs et al., 2005], brainstorming and meeting facilitators [Elrod et al., 1992, Pedersen et al., 1993], awareness monitors [Greenberg and Rounding, 2001, Bardram et al., 2006a], and industrial design canvases [Balakrishnan et al., 1999] among others. As the amount of manipulated information increases, these and other application domains may benefit from larger display sizes. Prior research has identified benefits of using large display real-estate in navigation [Tan et al., 2001, Baudisch et al., 2002, Tan et al., 2003, 2004a, Ball and North, 2005a], information memorization and recall [Tan et al., 2001, Lin et al., 2002], multi-tasking [Czerwinski et al., 2003], as well as general usability benefits [Grudin, 2001, Ball and North, 2005b]. Given the desire for increased display size and advancements in input and rendering technology, interactive displays will likely reach a “wallpaper” phase in the future, with sizes that cover entire



Figure 1.1: The Large Display Wall in DGP lab. The wall display is 5 x 2 m (16' x 6') with an effective resolution of 6144 x 2304 pixels.

walls and present vast amounts of data. In the literature the term "large display" refers to either vertical board displays or horizontal tabletop displays of different scales. Although some points raised in this thesis apply to large displays of different orientations and scales, our focus is on vertical, *wall-sized displays* (approximate size of 5 x 2 m), like the one seen in Figure 1.1.

Recently, the decreasing cost of projectors and LCD displays has enabled the construction of such wall displays by tiling multiple projectors [Funkhouser and Li, 2000, Guimbretière et al., 2001, Raskar et al., 2002, Baar et al., 2003] or smaller LCD screens [Ball and North, 2005b, Johnson et al., 2006] to form a single image. These high-resolution wall displays (resolution of more than 6000 x 2000 pixels) are interesting from an interaction perspective in that they enable users to view high quality imagery even when they are up-close to the display. In contrast, single projector systems at that scale are not

suitable for up-close interaction as the image appears too pixelated.

Research on non-interactive surfaces [Lund, 1997] has demonstrated that users preferred viewing large surfaces at distances farther away than that of direct touch, in order to simultaneously see more content. Nevertheless, it has been observed that in dynamic settings users tend to move as close to the large displays as possible in order to interact with them up-close [Brignull et al., 2004, Hawkey et al., 2005]. While interacting up-close with wall displays, it is possible that the unique affordances and benefits of direct input [Sears and Shneiderman, 1991, Shneiderman, 1991] are coupled with the transfer of regular whiteboard knowledge and familiarity.

Due to recent technological advances, such touch sensitive and pen-input enabled wall displays have become a commercial reality by companies like SMART Technologies (<http://smarttech.com/>), making up-close interaction with wall displays an interesting topic worthy of further research.

Wall displays differ in scale from other direct input interactive surfaces, giving rise to unique challenges. For example some areas of the display are unreachable by the user without a substantial amount of motor effort. Interaction with these remote locations can be particularly challenging, for example in settings where the user needs to rearrange multiple objects that are placed outside arms reach; or impossible, for example when areas of the display are inaccessible physically (for instance, very high up), or socially (blocked by another user).

Moreover, as users interact at close proximity to the display and depending on their viewing angle, certain locations may be partially or completely invisible. Invisible locations may cause problems when data dynamically change without the user seeing them. Thus keeping users apprised of changes in the periphery at all times may prove challenging, as visual changes on the extended surface of wall displays can go unnoticed.

These issues become increasingly important as existing applications and interfaces designed for smaller displays are mapped onto larger surfaces (an issue identified early on

by Pedersen et al. [1993]). Attempting to address some of these limitations, we consider a set of design goals (Section 2.6): we advocate the use of application and data agnostic designs that can benefit all types of applications or interactive primitives; we discuss the need to minimize user effort, both physical in terms of movement, but also mental, in terms of recall; and finally the need for predictable and consistent designs. We follow these goals in three design explorations of techniques that facilitate layout management and context switching across large distances, aid access to remote locations, and reveal invisible changes in wall displays.

Before presenting in detail our design explorations, we provide background on large display interaction by surveying the literature and further discuss our proposed goals in the next chapter.

Our first design exploration (Chapter 3) focuses on the issue of comfortably rearranging multiple remote objects and switching focus between different remote areas. To permit layout management and context switching close to the user, we propose using alternative views of remote display locations as interaction shortcuts to these locations. We first present a framework that describes existing work on alternative views and may inform their design in the context of wall display interaction. Based on this framework we then introduce three new interaction techniques, and later refine the design and explore the use of the most interesting one through two user studies. Our results indicate that shortcuts to remote areas in the form of shortcut views are a promising alternative to walking while performing layout management and context switching tasks on wall displays.

In the second exploration (Chapter 4) we examine new designs that aid reaching across large distances, by copying interactive proxies of remote content close to the user. Based on observations from a pilot user evaluation of these designs, we refine our initial design goals and identify new ones that are appropriate for remote reaching techniques. These goals are then used to further improve the most promising of the discussed designs.

When evaluated, our refined design is shown to be more efficient than existing reaching practices in multiple target acquisition tasks on wall displays.

Revealing dynamic visual changes happening out of the user’s sight is the focus of our third design exploration (Chapter 5). We propose buffering all visual changes in locations invisible to the user and revealing them at a later time. We explore the design space of this new technique in detail, and obtain comments on its use and on different design variations through a preliminary user feedback session. Our approach provides a first step towards identifying and understanding invisible visual changes on wall displays.

Finally, Chapter 6 summarizes our conclusions, contributions, and potential future research directions.

1.1 Thesis contributions

Outlined here are the main contributions of this thesis:

Chapter 3

1. Introduction of the canvas portal framework, to aid in the classification, improve the design, and lead to the creation of new techniques that use shortcut views of remote locations.
2. Presentation of designs for context switching and layout management based on the framework, as well as the evaluation and graphical depiction of the interaction patterns using one of these designs.

Chapter 4

3. Design of a new remote reaching technique for large displays, the vacuum, that fluidly supports both complex and simple selection behavior.
4. First systematic evaluation of such a remote reaching technique in the presence of distracter targets.
5. Identification and enumeration of design guidelines for remote reaching techniques.

Chapter 5

6. Characterization of conditions where information is not seen by users.
7. Proposal of mnemonic rendering, an application agnostic approach for storing and restituting unseen visual changes.
8. Thorough exploration of the design space of mnemonic rendering and an informal user feedback session.

Chapter 2

Background

The term "large display" has been used over the years to characterize both fairly large seamless display surfaces created by tiling multiple projectors [Funkhouser and Li, 2000, Guimbretière et al., 2001, Raskar et al., 2002, Baar et al., 2003], and multi-monitor configurations [Ball and North, 2005b, Johnson et al., 2006] with seams due to their bezels. As observed by Robertson et al. [2005], the former commonly use pen or touch input as means of interaction, while the later use the familiar keyboard and mouse setting. Although our work focuses on seamless interaction surfaces, issues discussed in this thesis often affect multimon (multiple monitor) configurations.

While some research on seamless large displays examines environments integrating both digital tables and whiteboards [Geißler, 1998, Streitz et al., 1999, Forlines et al., 2006a] or multi-display environments [Tani et al., 1994, Rekimoto and Saitoh, 1999, Johanson et al., 2002b, Baudisch et al., 2004, Hinckley et al., 2004, Nacenta et al., 2006], the main body of interaction research revolves around either tabletops or vertical displays, which exhibit different characteristics and interaction challenges.

Direct touch digital tables have emerged as interaction surfaces since work by Buxton [1997]. Technological advances in direct input have enabled the creation of interactive tables ranging from two point [SmartTechnologies, 2006] to multiple point sensing [Reki-

moto, 2002, Han, 2005], and multiple user detection capabilities [Dietz and Leigh, 2001]. Research conducted in the field includes multi-finger and whole hand gestural interaction techniques for digital tabletop surfaces [Wu and Balakrishnan, 2003, Wu et al., 2006, Morris et al., 2006], techniques for multiple users on circular tables [Vernier et al., 2002, Shen et al., 2002], interaction both on top, but also under digital tabletops [Wigdor et al., 2006], as well as the investigation of territoriality issues while interacting on physical and digital tables [Scott et al., 2004, 2005, Ryall et al., 2004, 2006]. Interaction with vertical displays, such as wall displays, differs from tabletops since their vertical configuration enables data to be viewed and interacted upon not only from different viewing angles, but also from different distances.

2.1 Interaction with wall displays

This section describes work relevant to wall display interaction, often developed for smaller vertical displays (like white-board sized displays), still considered "large displays". Interaction on wall displays falls mainly under two categories: remote and up-close interaction. Although the focus of this thesis is up-close interaction, we will briefly go over work in both categories to present a broader view of current interaction research.

2.1.1 Remote interaction

Apart from the traditional keyboard and mouse input metaphor [Johnson et al., 1989, Johanson et al., 2002a,b, Izadi et al., 2003, Ball and North, 2005b], other types of input that are decoupled from the actual display have been proposed for vertical displays in general. Among the first remote interaction devices are isotonic flying mice or hand held isometric input devices [Zhai, 1998, MacKenzie and Jusoh, 2001], that do not require the device itself to move in 3D space. More recently, Soap [Baudisch et al., 2006a], a mouse that can be operated in midair, and a set of appropriate interaction techniques was used as remote input for wall displays. Finally, Malik et al. [2004, 2005] introduced

a vision based touch-pad that can be used as an input device for a remote wall display. Some of these approaches require the user to rest her arms on a desk and may thus be more comfortable to use over prolonged periods of time than direct interaction by either pointing or touching the display. But they confine the user to a single sitting and viewing position potentially wasting display space [Ball and North, 2005b]. Moreover, decoupling input and output takes away from the immediacy of direct manipulation [Sears and Shneiderman, 1991, Shneiderman, 1991] and may cause confusion over the location of the input pointer on the extended display surface [Robertson et al., 2005, Ball and North, 2005b]. Finally, remote interaction in general prevents the use of the full resolution of the wall display.

Remote pointing has also been investigated as display input, particularly direct hand pointing [Bolt, 1980, Hinckley et al., 1994, Zhai, 1998, Grossman et al., 2004, Vogel and Balakrishnan, 2005] and the use of laser pointers [Kirstein and Muller, 1998, Winograd and Guimbreti re, 1999, Olsen, Jr. and Nielsen, 2001, Myers et al., 2002, Matveyev and G bel, 2003], discussed below.

The seminal "put-that-here" work by Bolt [1980] investigated the combination of direct pointing and voice to disambiguate context, while Hinckley et al. [1994] and Zhai [1998] focused mainly on issues when using the hand for 6 DOF tasks. Direct pointing techniques have also been developed in the VR community [Pierce et al., 1997, 1999, Bowman and Hodges, 1997], with selection usually performed through the use of a button. Finally, Grossman et al. [2004] and Vogel and Balakrishnan [2005] proposed direct pointing and selection techniques that do not require buttons or other physical aids. Direct hand pointing is a very natural mechanism for referring to remote content [Kendon, 2004] and does not necessarily require any physical input device [Vogel and Balakrishnan, 2005]. But it may prove inaccurate due to hand jitter (observed repeatedly in laser pointer interaction [Myers et al., 2002]) and uncomfortable since there is no physical surface to provide hand support [Hinckley et al., 1994, Wang and MacKenzie, 2000].

Kirstein and Muller [1998] introduced the use of laser pointers as input devices for large projected surfaces, mainly focusing on technical aspects of tracking. Winograd and Guimbreti re [1999] later presented a gesture based interaction system for laser pointers that accommodates both up-close interaction (tip-positioning mode) and remote display interaction (beam-positioning mode). Similarly Parker et al. [2005] investigated the use of a stylus in tabletop interaction for both direct touch and remote pointing, by casting a virtual ray. As was generally shown [Myers et al., 2002] laser pointer interaction is inaccurate at a distance due to hand jitter, an issue Olsen, Jr. and Nielsen [2001] attempted to address through a series of cleverly designed laser pointer interaction techniques optimized to avoid hand jitter as much as possible. Apart from inaccuracy, laser pointer interaction also suffers from lack of hand support, leading to fatigue over extended use.

Less common devices have also emerged as remote input mechanisms. Wilson and Pham [2003] used a wand to control a motor actuated laser beam in an augmented environment, while VisionWand [Cao and Balakrishnan, 2003] provided inexpensive input for a wall display with a set of interaction techniques using 3D position and orientation information. Jiang et al. [2006] used cameras to interact with a wall display, while in work by Forlines et al. [2005] and Cao and Balakrishnan [2006] interactive surfaces were both projected and interacted upon using a hand-held projector. These approaches, albeit innovative, often introduce uncommon interaction paradigms for casual commonly performed tasks. Moreover they suffer from hand jitter since they operate in mid-air and are in general not well suited for detailed and prolonged manipulation. Rather they are aimed at brief coarse grained interactions. Similarly, accuracy and fatigue issues affect face tracking interaction [Nakanishi et al., 2002] on wall displays.

Thus there are tradeoffs when using remote interaction techniques. Some, such as pointing, are natural but fatiguing during prolonged use. Others, such as the mouse, are comfortable but confine the user to a single viewing position. Finally, all remote interaction techniques do not take full advantage of the resolution of the display.

2.1.2 Up-close interaction

Liveboard [Elrod et al., 1992] from Xerox was the first step toward a large vertical display integrating pen input. The Tivoli project [Pedersen et al., 1993] presented the first interface built specifically for a whiteboard display (the Liveboard) used in brainstorming settings, while Moran et al. [1997] proposed techniques for organizing and structuring material on a Liveboard for informal meetings. Although the above work presented group use scenarios, the interaction component was always performed by a single user. Work by Mynatt et al. [1999] also presented a digital whiteboard that served the function of a personal whiteboard in a semi-private office space, being clearly a single user interface. Finally, Guimbretière et al. [2001] described a wall display setting used as a brainstorming aid, and discussed techniques for fluid in-place interaction. Although most of this original work was focused on small size whiteboards, issues such as unreachable content were identified early [Elrod et al., 1992]. Swaminathan and Sato [1997] also identified problems arising due of the scale of wall displays, including pointer movement and control challenges over large distances. They proposed using a small-scale model of the display and its contents to specify pointer movement in the wall display.

Apart from general work on up-close interaction, in recent years researchers have focused on specific wall display challenges, such as comfortably moving content across accessible [Rekimoto, 1998, 1997] or inaccessible areas [Baudisch et al., 2003a, Hascoët, 2003, Collomb and Hascoët, 2004, Scott et al., 2005, Nacenta et al., 2005], or in general interacting with remote locations [Tan et al., 2004b, Khan et al., 2004]. Because these designs are relevant to the work presented in this thesis, we will discuss them in detail in the following sections.

Work by Lund [1997] has demonstrated that users preferred viewing non-interactive surfaces further back than comfortable direct touch distances, to perceive more displayed information. On the other hand, in dynamic settings users seem to move as close to large displays as possible to interact with them [Brignull et al., 2004, Hawkey et al., 2005].

Despite tradeoffs between the amount of viewed content and direct up-close interaction, most large display environments relay on direct touch or direct pen input [Robertson et al., 2005] (although some recent work [Brignall and Rogers, 2003, Vogel and Balakrishnan, 2004] discusses transitions from remote to up-close interaction).

As mentioned by Sears and Shneiderman [1991] and Shneiderman [1991], direct display input is a familiar and natural means of interaction and hand-eye coordination is easy to achieve. Nevertheless, prolonged use of a vertical display while standing up may result in fatigue. There is thus a second type of tradeoff, that of direct up-close interaction vs fatigue from prolonged use. The assumption throughout our work is that users interacting up-close with a wall display will do so in somewhat limited time intervals in order to manipulate content or view information in detail, and occasionally move further back for viewing. Although the exact timeframe for continuous use of wall displays will greatly depend on the task, we expect 1-2 hours at a time followed by sitting breaks is a reasonable estimate, based on existing practices of professionals (ex. teachers, nurses), that spend most of their day standing [Beynon and Reilly, 2001, Tissot et al., 2005].

Recent technological advances have made touch sensitive wall displays a commercial reality by companies such as SMART Technologies, making up-close interaction with wall displays an interesting research aspect of the field.

2.2 Applications for interactive wall displays

This thesis focuses on designs for up-close interaction with wall displays. Before determining the design goals that inform our design exercises, we must first characterize the types of applications that will likely make use of this emerging technology. These applications can provide insights as to some of the types of general, low level tasks that will be performed on wall displays.

A variety of direct manipulation tasks performed on physical walls and whiteboards

have been identified early on as potential applications for vertical large displays. These include content creation tasks over large canvases, as in classroom scenarios [Abowd et al., 1998] (Figure 2.1(d)), brainstorming and meeting settings [Elrod et al., 1992, Pedersen et al., 1993, Fitzmaurice et al., 2005], planning [Trimble et al., 2003], artistic creations [Balakrishnan et al., 1999, Buxton et al., 2000] or design processes [Guimbretière et al., 2001] (Figure 2.1(b)). Organizational tasks traditionally performed on large physical surfaces, such as magazine layout or story-boarding (Figure 2.1(a)), are also likely to be transferred to the digital medium as the technology becomes more readily available.

Wall displays have also been used in time critical monitoring tasks, due to their dynamic nature and scale. Emergency response (ex. [TOtal, 2006]) and traffic control centers (both air and ground [Electrosonic, 2006]) have been equipped with digital walls (Figure 2.1(e)), while security monitoring in large spaces is done through a multitude of tiled displays. Although these are essentially monitoring tasks, military command and control units (Figure 2.1(c)) have been experimenting with touch sensitive digital walls [Jedrysik, 2006], to enhance such monitoring centers with direct interaction technology.

Large dynamic data visualization in meteorology [Semeraro et al., 2004], geology [Johnson et al., 2006], network traffic [Wei et al., 2000], or medical applications [Hibbs et al., 2005], is often being done using wall displays, because of the quality and quantity of data that may be simultaneously visible. Apart from the monitoring aspect of these applications, interactions in the form of exploration, navigation, visual search, data comparison and switching between different data views are essential to the understanding and manipulation of the visualized data.

Finally, in recent years large displays have emerged as community awareness monitors and communication facilitators in hospitals [Bardram et al., 2006a,b], offices [Greenberg and Rounding, 2001, Fass et al., 2002, Ball and North, 2005b], programming environments [Biehl et al., 2007] and research settings [Russell and Gossweiler, 2001, Churchill et al., 2003, Huang and Mynatt, 2003].



(a)



(b)



(c)



(d)



(e)

Figure 2.1: (a,b) *Applications on physical walls*: (a) Director Peter Weir editing storyboards for the film "Fearless" (<http://www.peterweircave.com/>). (b) Design board of the Sniff-Receptor art installation (<http://jan.moesen.nu/media/photos/2004/01/sniff-receptor/>). (c,d,e) *Wall displays*: (c) A war room command and control center (Air Force Research Laboratory <http://www.afrlhorizons.com/Briefs/Sept01/IF0012.html>). (d) An electronic classroom in Georgia Tech (eClass project <http://www.cc.gatech.edu/fce/c2000/pubs/>). (e) A command and monitoring center of the UK Highways agency (<http://www.electrosonic.com/>).

2.3 Interaction tasks on wall displays

By carefully studying these existing and potential applications for high resolution wall displays, we identified a possible subset of frequently occurring interaction tasks that are likely to be performed on wall displays:

Content selection and movement As with all interactive environments, wall display interaction will rely on selecting and moving content. This will be especially the case in content creation scenarios, such as educational or brainstorming settings, where new content is created often and periodically moved in different locations of the board as its nature and importance evolves over time.

Layout management Several of the applications studied involve organization of material scattered across the display, identifying layout management as another interaction task to consider. Examples of such organizational and layout tasks include arranging and re-arranging story-boarding material and design mockups, creating and modifying the layout of magazine articles and images, etc.

Search and navigation Since a number of potential wall display applications involve the manipulation and exploration of high resolution dynamic imagery, navigation and searching actions are likely to occur often. Examples of such actions could be following the course of a river on a terrain map or of a particular weather front on a meteorological map, or identifying all proteins involved in a particular experiment in a protein connection graph.

Context switching Given the vast amounts of data involved in many of the applications for wall displays, switching between different views of the displayed data will likely take place in numerous data visualization tasks. For example in a protein connection graph, some aspects of the connections may need to be viewed in detail with respect to one experiment, while also viewing the context of the protein con-

nections from other experiments; in terrain maps areas might be viewed in terms of distance from the sea level or in terms of seismographical activity.

Comparison The dynamic nature and quantity of the displayed data on wall displays will likely require simultaneous views of different contexts in order to compare discrete visual elements. Examples of such comparisons may be the need to contrast different meteorological fronts, or cracks in high resolution structural photographs of construction material. Comparison tasks can be seen as a form of context switching, where both contexts need to be visible at the same time. We thus discuss comparison as part of context switching from now on.

Monitoring Finally, several applications examined include a monitoring component of dynamic data. Examples include activities such as monitoring security videos or realtime feeds from remote sources, as in command and control rooms.

Although this list may not be comprehensive, it enables us to identify some of the issues that might arise due to the unique aspects of up-close wall display interaction. We will next discuss each of the above tasks and how they might prove challenging in wall display settings, and then motivate the research directions of this thesis.

2.4 Interaction challenges

Wall display use has already been adopted in different application settings and will likely become more prominent in the future as technology becomes more readily available. It is thus important to investigate early on interaction challenges arising due to the unique aspects of wall displays, large size and high resolution. We will briefly discuss how these aspects affect interaction, especially in the context of the identified sample tasks.

In this thesis we will focus on challenges we feel are unique to the medium and we will attempt to provide general purpose solutions. Although search and navigation are

identified as probable and potentially problematic tasks in wall displays, they do not seem to uniquely affect high resolution wall displays, but any display type presenting part of a large virtual canvas. In fact, initial studies have shown that users are actually faster in search and navigation tasks in large high resolution displays [Ball and North, 2005a, Ball et al., 2007]. This is enforced by findings that indicate that while using large focus plus context screens [Baudisch et al., 2002] users are able to make use of their peripheral vision in dynamic navigation and search tasks. As this initial research has yielded positive results in search and navigation on wall displays, we decided to focus on other identified tasks that are more likely to be adversely affected by the large wall display scale.

Because large size and high resolution enable vast amounts of data to be presented over a large spatial canvas, using existing techniques for layout organization, context switching or remote location comparisons may prove problematic. For example, simple dragging of items across the length of the display to rearrange layout, using soft keyboard shortcuts to switch between applications, or moving around the display to compare data at different display locations becomes tedious. Since layout management and organization, comparison and context switching have been identified as potential wall display tasks and seem negatively affected by large display scale, further investigation is needed. In Chapter 3 we discuss these three types of tasks in conjunction, as a common approach (namely using alternative views of different areas of the display) may benefit all three.

Although users can easily access items close to their location, selecting and interacting with remote content can be challenging over large display surfaces. As selection will likely occur in wall display tasks, we further examine remote reaching issues in Chapter 4. Remote reaching does not necessarily refer to single item selection. It also characterizes any type of brief remote access, for example remote item picking, movement of a proximal item to a remote location, etc.

Due to the large area of wall displays, content is not always visible from the user's

location. Thus remembering and identifying visual changes on wall displays may be hard, especially considering their potential to simultaneously present large amounts of dynamic data. This issue affects all types of applications that encompass some form of monitoring. We further discuss invisible changes in Chapter 5.

In this thesis we discuss these interaction challenges arising in wall display settings and describe designs that attempt to accommodate layout management and context switching across large surfaces, facilitate remote reaching, and aid monitoring changes in the visual state of the display. As mentioned, we do not investigate search and navigation from the original sets of tasks. Moreover we concentrate on interaction with existing content and do not explicitly discuss content creation, as past research in large displays [Rekimoto, 1998, Mynatt et al., 1999, Guimbretière and Winograd, 2000, Guimbretière et al., 2001] indicates that the creation process will likely take place close to the user and it thus falls under research on in-place content creation. In the future it is probable that more interaction challenges may arise in wall display interaction. This thesis is an initial exploration of the field that we hope will inspire more work in the area.

The issues discussed in this section present challenges in both single and multiple user scenarios. Wall displays are particularly well suited for multi-user concurrent interaction since they can comfortably accommodate more than one users. Nevertheless, in this thesis we concentrate on single user interaction designs because, firstly many of the discussed scenarios for wall displays involve essentially one user (for example artistic endeavors). Moreover, research has shown that very often in multi-user settings, a single person drives the interaction [Russell et al., 2002], as seen in educational or meeting settings. Finally, before delving into multi-user interaction research, the single user aspects need to be looked at first, in order to provide us with insights on the use of wall displays, that may later prove relevant to multiple users settings.

2.5 Designs addressing the identified challenges

In this section we will describe in detail existing work that is either explicitly designed, or may be used to address the interaction challenges identified in the previous section, given the identified tasks.

2.5.1 Viewing and interacting with all areas of the display for layout managements and context switching

Layout management, context switching and comparison tasks are characterized by the need for simultaneous access, both visual and interactive, to different areas of the display, possibly for prolonged time periods. This may prove challenging without substantial physical effort in a large scale display. Designs that alleviate the need to constantly walk across the display include distortions of the visual and interactive display space and the presentation of multiple alternative views of areas of the display.

Space distortion: A way to access remote content is to scroll the entire virtual canvas so that remote content moves toward the user for comfortable viewing and manipulation [Bezerianos and Balakrishnan, 2004]. However, in order to restore the display space, the content must be scrolled back to its original position. Systems that distort and condense the visual space through Fisheye lenses [Furnas, 1986, Carpendale and Montagnese, 2001] or rapid scaling [Bederson and Hollan, 1994] may be used to reduce distances across the display. Space distortion techniques facilitate layout management, context switching (and comparison tasks, a form of context switching) as long as they provide sophisticated focusing mechanisms (for example by supporting multiple zooming foci). But they cause a significant amount of visual disruption that can disorient users since the display covers most of the user's visual field.

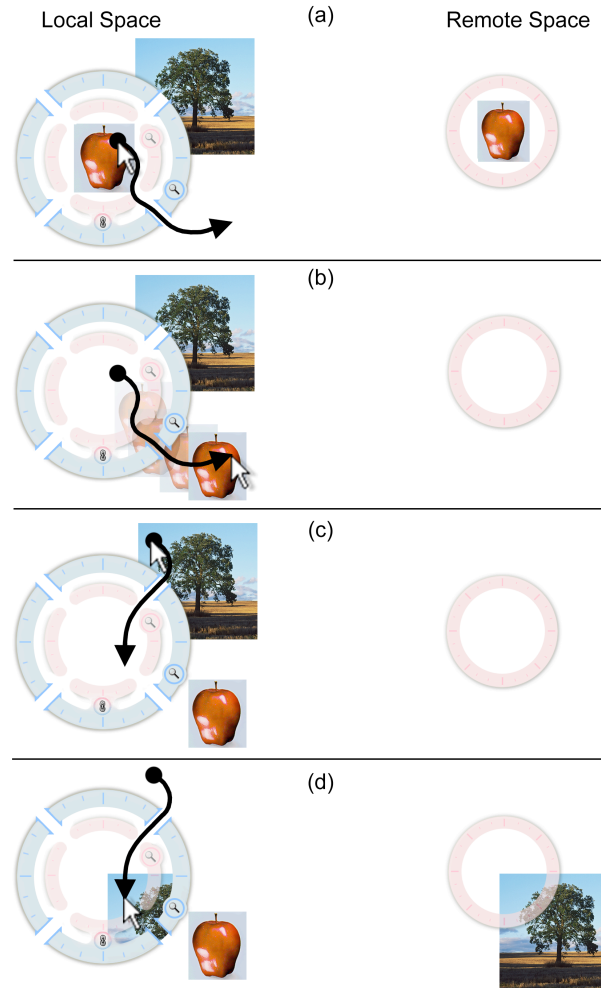


Figure 2.2: Frisbee (a) Dragging a remote image to local space by exiting the frisbee through a transfer channel and (b) the result. (c) Dragging a local image to the remote space by entering the frisbee through a channel and (d) the result. (Image from Khan et al. [2004])

Alternative views: Using alternative views of different display locations as visual and interaction shortcuts, may also facilitate remote location access, context switching and layout management. Stoakley et al. [1995] used WIM (World in Miniature) views to access content and get overviews of immersive 3D virtual environments. Similarly Swaminathan and Sato [1997] proposed early on using a small-scale view of a wall display to specify pointer movement. Tan et al. [2004b] augmented window managers by using

WinCuts, interactive views of arbitrary regions of existing application windows. Finally, Khan et al. [2004] introduced the *Frisbee*, a widget that acts as an interactive telescope to a remote area on the display (Figure 2.2). Although alternative view shortcuts aid context switching and layout management, they segment the visual display space and identifying the exact location they point to may be problematic. Alternative views will be further discussed in Chapter 3, as they are closely related to our canvas portal designs.

2.5.2 Bridging large interaction distances for remote reaching

Another major problem identified in wall display interaction is brief access to areas unreachable without substantial physical effort. Consequently much work focuses on bridging large distances. Layout management and context switching require visual and interactive access, possibly for prolonged time intervals, to different areas of the wall display. On the contrary, remote reaching actions are usually brief (for example single button press or menu selection), are focused in a particular area of the display, and often detailed viewing of the remote location is not necessary (for example icon activation). Existing approaches include ignoring virtual space empty of content, virtual picking of content, amplifying user input and copying content close to the user's location.

Ignoring virtual space empty of content: Several selection techniques have been proposed that ignore empty display space. In *semantic pointing*, the amplitude of targets was extended in motor space, allowing users to select remote targets faster. Guiard et al. [2004] introduced the idea of *object pointing*, where the cursor skipped empty space, jumping from one selectable target to another. Similarly, the *bubble cursor* [Grossman and Balakrishnan, 2005] always selected the closer target to the pointer (Figure 2.3) by adjusting the pointer's size. These techniques may aid remote target selection in sparsely populated wall displays, but in dense environments intermediate items may exist between the user and the target, so the user needs to physically move closer to the target.

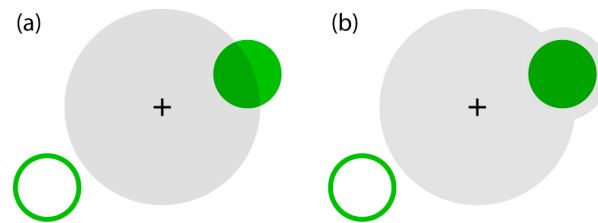


Figure 2.3: Bubble cursor. (a) Cursor size is such that only one target is captured at a time. (b) Cursor envelops the target when not completely contained by the main bubble. (Image from Grossman and Balakrishnan [2005])

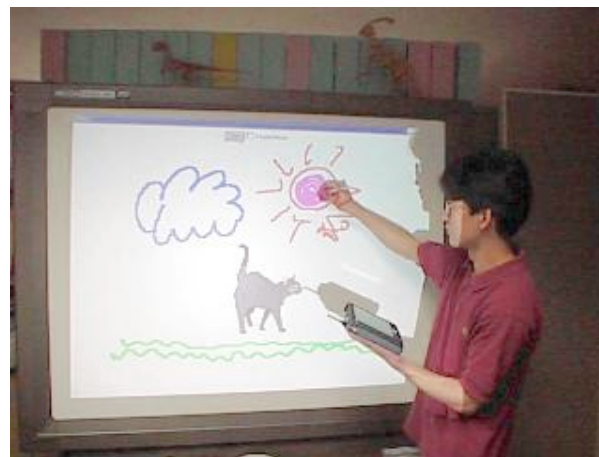


Figure 2.4: The user moves content by virtually ”picking” it in his PDA or stylus. (Image from Rekimoto [1998])

Virtually picking up content: Rekimoto [1998] distributed some functionality of a large display (text entry, widgets and pallets) to a hand held PDA. Objects from the display could also be transferred to the PDA for editing, or to be moved easily to another display location. This integration of different display spaces aids distance manipulation and layout management, since the user moves together with the content to be manipulated to another part of the display (Figure 2.4). This idea was later extended to using either a stylus [Rekimoto, 1997] or hand [Geißler, 1998] to virtually pick up and drop content. Although picking and moving items is likely more efficient than dragging

them across the display by maintaining touch, neither might be feasible (for example when content is unreachable) and users still need to walk across the length of the display. In a sense layout management and context switching is also trivially supported in these types of techniques by regular walking around the display to both organize space and switch focus.

Amplifying the user’s input: Amplification of mouse input has been proposed in the *MissleMouse* technique [Robertson et al., 2005] for distant content interaction. Users drag fast and release the mouse pointer, which continues traveling until the mouse is moved again. Forlines et al. [2006b] described *HybridPointing*, an alternative technique for transitioning between absolute direct input to relative amplified input for reaching remote locations, by quickly entering or exiting a trailing widget. Although developed for VR environments, work by Poupyrev et al. [1996], Bowman and Hodges [1997], Pierce and Pausch [2002] also amplified user input by virtually extended the reach of the user’s hand to remote locations. Finally, Rekimoto and Saitoh [1999] introduced *hyperdragging*, the idea of extending the user’s cursor outside the physical confines of one display onto other interactive surfaces.

Another set of amplification techniques, often referred to as throwing techniques, are designed to move proximal content to remote locations. Apart from throwing items at predefined distances [Geißler, 1998] or to the edges of the screen [Wu and Balakrishnan, 2003], Geißler [1998] introduced parameterized throwing: first, a small stroke is performed toward the opposite direction of the desired target and one toward the desired target. The closer the length of the two strokes, the further away the object is thrown. To improve the accuracy of throwing techniques Hascoët [2003], Collomb and Hascoët [2004] and Scott et al. [2005] experimented with different amplification functions, while providing users with visual feedback on the throw trajectory and final throw destination (Figure 2.5). In one such technique, part of the visual feedback included a *radar view*, a scaled down

abstract view of the entire canvas close to the user, similar to the alternative shortcut views discussed before. Finally, the *superflick* technique [Reetz et al., 2006] combined throwing gestures with the option to make corrections while throwing is performed, by displaying the remote destination for the thrown object.

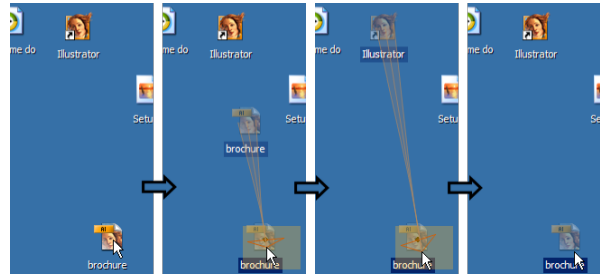


Figure 2.5: An instance of a throwing technique, drag-and-throw [Hascoët, 2003]. The user first initiates a drag. Then, the visual clues appear (take-off area, trajectory). The user adjusts her throw keeping the cursor in the take-off area, and finally ends the drag and the object is thrown on the selected target. (Image from <http://www.lirmm.fr/edel/dragging/index.php?hp>)

Evaluations of throwing models [Collomb and Hascoët, 2004, Nacenta et al., 2005] have shown them to be faster than direct drag-and-drop, but indicated that mean times and accuracy for a radar based technique was faster. This is an indication [Reetz et al., 2006] that visual feedback close to the user might be better than having it in the environment (as in the throwing techniques, or even HybridPointing), even though users have to work with a duplicate representation of the target instead of the real one. Users' comments in work by Forlines et al. [2006b] also commented that small targets were not visible in distant locations on a wall display, another indication that visual feedback close to the user might be desirable.

Copying content close to the user's location: Baudisch et al. [2003a] introduced *drag-and-pop* (Figure 2.6), an extension of drag-and-drop, where remote icons are brought close to the user's location. *Drag-and-pop* and *push-and-pop* [Collomb et al., 2005a], an

extension to *drag-and-pop* that utilizes a radar view, are further discussed in Chapter 4, since they are closely related to our research. Techniques that copy content close to the user have also been developed in the context of small display devices, such as PDAs and cellphones. For example Irani et al. [2006] provided users with proxies of off-screen content, that can be used as teleportation mechanisms to off-screen location.

Although proxy copying addresses remote reaching, it may prove challenging to arrange multiple proxies close to the user and provide sufficient visual feedback in order to aid the user to maintain the mental connection between the remote icon and the proximal copied proxy.



Figure 2.6: Drag-and-pop [Baudisch et al., 2003a]: the user begins a drag-and-drop operation. After dragging beyond a threshold, possible targets (of compatible type, in the direction of the move) pop near the user cursor. The user terminates the drag-and-drop operation by releasing the dragged item on one of the possible targets. After the user has dropped the icon, the popped targets disappear. (Image from <http://www.patrickbaudisch.com/projects/dragandpop/demo/index.htm>)

Space distortion and alternative views of remote locations presented as aids for layout management and context switching can also be used for simple remote reaching actions, by targeting a specific remote location. Nevertheless they are usually associated with parameter adjustment cost (distortion parameters and focusing), which may dominate the interaction cost for simple reaching actions. Conversely, some of the remote reaching approaches could be useful for layout management and context switching in terms of in-

teraction with remote locations. But frequent context switching between different remote areas and persistent access is not adequately supported by techniques that bring content close to the user, unless everything gets trivially copied to the user's location. Nor is visual access supported by input amplification or by ignoring space empty of content.

2.5.3 Invisible content monitoring

In current computing displays, dynamic visual changes regularly happen without the user being aware of them. Studies have shown that people are rarely able to spot visual changes when they occur during disruptions as short as eye saccades [Rensink, 2002]. Unless explicit notification or history mechanisms are provided [Freeman and Fertig, 1995, Rekimoto, 1999, Renaud, 2000], chances are that the user will miss these changes, or only become aware of them by chance or by explicitly trying to reconstruct them at a later time.

Monitoring or missing visual changes is related to visual attention [Pashler, 1998]. Managing user's attention between primary and peripheral information displays [Maglio and Campbell, 2000, Grudin, 2001, MacIntyre et al., 2001, Cadiz et al., 2002] and focusing the user's attention [Khan et al., 2005] have been examined. Although not aimed for large displays, there is a large amount of work concerning situation awareness designs for critical systems, such as air-traffic control [Endsley et al., 2003, Banbury and Tremblay, 2004], that are aimed at highlighting significant system changes. Techniques for enhancing multiple user action awareness are also relevant (see Tam and Greenberg [2006] for an overview of current techniques). Recently work [Baudisch et al., 2006b] inspired by techniques from chronophotography and comic book transitions, used animation and abstraction to indicate changes in user interface elements. This set of designs assume a high-level system knowledge of the data presented on the screen and the types of dynamic changes that are likely to occur.

Changes that go unnoticed could be potentially stored for the user, as we discuss in

Chapter 5. A large number of recording and re-visitation techniques have been proposed for various purposes including action reversibility [Edwards et al., 2000], replication [Kurlander and Feiner, 1992] and improved information search [Wexelblat and Maes, 1999, Cockburn et al., 2003, Skopik and Gutwin, 2005]. All these techniques are highly application or data dependent. A few application-independent approaches to user history storage and navigation have also been proposed [Freeman and Fertig, 1995, Rekimoto, 1999, Renaud, 2000, Apple, 2007] but they all assume a entirely new software architecture or object-oriented protocol for supporting them. These history based techniques also show changes the user has already visited in the past and relay heavily on navigation, with no special consideration for hidden or missed changes.

2.6 Design goals

In the next chapters we discuss our design explorations that attempt to addressing the unique interaction challenges discussed before. Having surveyed possible applications and tasks for wall displays, we decided on the following three design goals that inform all our design explorations.

Application/data independence. We have identified a diverse set of possible applications for wall displays and more are likely to appear as the technology becomes more widespread. These applications come from different domains and often don't share the same assumptions as to the nature of the displayed information and interactive primitives. Furthermore, until applications designed specifically for wall displays [Guimbretière et al., 2001, Alimohideen et al., 2006] prevail, existing designs and interaction paradigms based on the familiar WIMP interaction [Johnson et al., 1989] are likely to be mapped on wall displays [Baudisch et al., 2003a, Tan et al., 2004b, Collomb et al., 2005a]. It is desirable in this early phase in wall display research that new interaction techniques benefit all types of applications and data,

irrespective of the nature of the displayed information or its underlying semantics. Such techniques may be implemented on the window manager level without applications being aware of their functionality, and should work across applications and multiple use scenarios in a consistent and transparent manner.

Minimal user effort. While discussing interaction challenges and proposing solutions, we assume that although wall displays provide wide views of large amounts of data, physical interaction preferably occurs close to the user. This is particularly important given that several identified tasks involve reaching. It has been observed [Streitz et al., 1994] that reaching over large distances on wall displays can be fatiguing or socially awkward [Russell, 2002], while access to remote content may often be physically blocked. Moreover, easily accessible areas depend on the user's height, length of their arms, and whether they are left or right handed [Streitz et al., 1994]. To ensure that all areas of the display are comfortably accessible to all users and minimize fatigue, we propose that techniques are designed such that they require small movements close to the user.

Throughout our designs, the "comfort area" where users interact with the display is approximately 45-55 cm on all compass directions from the user's torso. This falls inside the "Best Work Zone" and "Preferred Work Zone" (Figure 2.7) as defined by the Occupational Safety & Health Administration [OSHA, 2007] in their guidelines for preventing musculoskeletal disorders. Work inside these areas "facilitates productivity and comfort [...] is safest when reaching is performed in these zones. Working outside these work zones results in non-neutral postures that may increase the risk of injury" [OSHA, 2007]. The design of interactions within this comfort area, apart from being physiologically comfortable, ensures that all presented information related to the technique is viewed comfortably without the need to move one's head. This within arm's reach area, also known as "personal", or "near" [Cutting and Vishton, 1995, Rizzolatti et al., 1997] space in psychology,

and "peripersonal" in neuropsychology and neurophysiology, is associated with the perception of condensed distance ("feeling close"), as well as the human ability to perceive objects fast, and form plans to interact with physical content [Holmes and Spence, 2004, Witt et al., 2005, Proffitt, 2006]. Finally, the "comfort area" falls under the "perceived personal distance" as described by the proxemics theory [Hall, 1966, Sommer, 1969], where movement feels natural and is considered socially acceptable. Thus the chosen interaction area is comfortable not only from a physiological, but also from a psychological perspective.

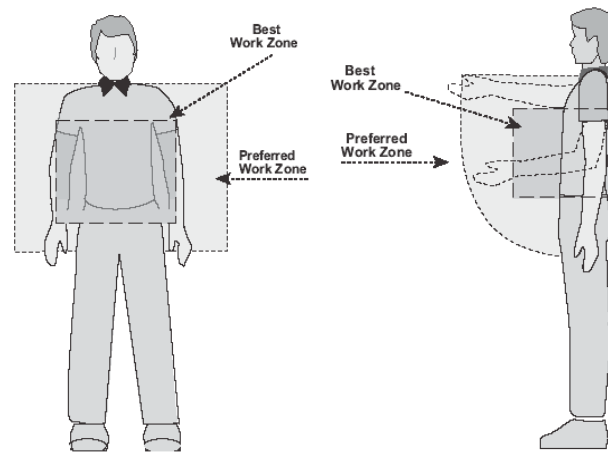


Figure 2.7: Best and Preferred Work Zone as suggested by the Occupational Safety & Health Administration to prevent musculoskeletal disorders for Retail Grocery Stores. (Image from <http://www.osha.gov/ergonomics/guidelines/retailgrocery/retailgrocery.html>)

Although direct interaction on a wall display might be limited close to the user, the visual use of the display is not. Due to human peripheral vision, the large real-estate of wall displays can provide users with large amounts of information simultaneously. Already studies have shown benefits of using large displays in navigation and searching [Baudisch et al., 2001, Tan et al., 2004a, Ball and North, 2005a], as well as an increased sense of immersion [Lund, 1997] and spatial aware-

ness [Tan et al., 2003, Bakdash et al., 2006]. While a large amount of content can be seen simultaneously on wall displays, not all content can be visible at any given time. When dynamic information at remote locations changes, say in monitoring tasks, it might go unnoticed, especially given the potential for large amounts of data drawn simultaneously. In our designs we attempt to remove the need to memorize the state of the displayed information, that may at times be large, minimizing user effort.

Predicability and consistency. With wall displays, users might need to simultaneously interact with different applications [Guimbretière et al., 2001, Baudisch et al., 2003a] and vast amounts of data [Semeraro et al., 2004, Johnson et al., 2006], while performing diverse tasks that may be of a dynamic nature [Wei et al., 2000, Jedrysik, 2006, Johnson et al., 2006]. It is thus desirable that any technique designed for wall displays does not add to the complexity of an already varied interaction vocabulary and a large visual real-estate, but displays predictable behavior that is easy to comprehend. To minimize user disorientation and improve user performance over time, apart from predictable, we require our designs to also be consistent and persistent across multiple invocations in similar situations, to enhancing learnability.

2.7 Chapter summary and closing remarks

In this chapter we briefly reviewed past and current interaction research on large displays. We then identified potential tasks that might be performed in the future on wall displays, as well as challenges that may arise when interacting at close proximity, due to their large scale and resolution. We discussed in detail existing interaction designs that focus on these specific wall display interaction challenges. Finally, based on the identified tasks and challenges, we outlined desirable characteristics of interaction designs, namely application and data independence, minimal user effort, predictability and consistency.

2.7.1 Prototyping

In the following sections we will describe new techniques, based on our design goals, that attempt to address interaction challenges of wall displays. For prototyping and testing our designs, we used a 5m wide, 2m high, back projected display. Imagery was generated by 18 projectors (1024 x 768 resolution each) in a 6 x 3 tiling, for an effective resolution of 6144 x 2304 pixels. The projectors were driven by a cluster of 18 workstations. All software was written in C++ with Chromium (chromium.sourceforge.net) providing graphics rendering over the cluster. A camera-based Vicon motion tracking system (www.vicon.com) tracked a pen's movement over the screen. Although the system could track the pen in 3D space, we used only x-y screen movements, a 10 cm hover zone, touch sensing and a single button, a setup similar to most touch sensitive interaction surfaces (Wacom tablets, tabletPC, Smartboards, etc).

2.7.2 Common procedures followed in statistical analysis

To improve readability of the sections related to user studies, we'd like to briefly describe common procedures and tests performed while analyzing user data.

Before performing Analysis of Variance (ANOVA) outliers were removed from the data. A trial was identified as an outlier when it was beyond 3 standard deviations from the mean under a given condition.

When performing ANOVA, we always test for statistically significant departure from the sphericity assumption, using the Mauchly's sphericity test. No such effect was present in any of our studies, so we will not report this again. Moreover we always check the data distribution for potential skewness. As in most of our studies reaction time data measurements were somewhat positively skewed, significance tests were performed on the logarithm of the measurements that follow the normal distribution more closely.

Finally, in our studies we often perform post-hoc pair-wise mean comparisons, in which case all p-values are adjusted using the Bonferroni criterion to protect against an inflation in the family-wise alpha error rate.

Chapter 3

Accommodating Layout

Management and Context Switching

Unlike interaction on a desktop, or even a small whiteboard-sized display, where users can reach most displayed items, data on wall displays often resides in an unreachable location. From a visualization perspective, a user might not be able to view all parts of the display with equal clarity because some of it will appear in the user's peripheral vision. As a result, existing user interfaces mapped onto displays of this scale for up close interaction would at the very least require the user to walk around the display to accomplish even simple tasks.

In any display, portions of an infinitely large virtual canvas can be invisible to the user depending on which part of the canvas is depicted on the display at any given time. On desktop sized displays, various window and desktop management schemes have been developed over the years that allow the user to effectively manage the space and layout of the available display area. With wall displays however, the space and layout management issues are somewhat different: much more data can be simultaneously displayed, but some of it will not be in the user's focal visual field, or easily accessible without substantial physical movement. The increased display space makes traditional space-multiplexing

techniques like overlapping windows less relevant, but introduces the need for appropriate techniques for managing, laying out, and fluidly accessing information in a non-spatially overlapping manner over a broader display space that spans a much larger visual field of view.

Moreover, desktop window management systems provide a variety of mechanisms for context switching between spatially overlapping applications and windows. On wall displays, it is likely that application windows will not overlap one another as much, but rather be spread out over the larger display surface, thus requiring different context switching techniques. When interacting with wall displays, a technology-free context switching mechanism is implicitly provided by simply walking towards the relevant part of the display. While this is a simple and easily understood interaction, it requires the user to physically move more than might be desirable. Furthermore, it does not support more sophisticated in-place context switching between multiple applications or different views of the same application for comparison tasks. As such, mechanisms for switching between these applications and their different views, that appropriately leverage the unique affordances of the larger display space, need to be designed.

In this chapter we examine the use of multiple interactive views of remote display locations presented close to the user, as a means of layout management, context switching and remote location comparison. We present the Canvas Portals framework, a new framework for describing these alternative views, and discuss how existing techniques are instantiated within it. Three new application independent instances of the introduced framework are described in detail: ScaleView portals, Division Bands and Window portals. The most promising of the three techniques, the ScaleView portals, is further redesigned based on user feedback and usability sessions.

3.1 Initial observations

While considering layout management and context switching challenges for wall displays, we observed that all the described tasks could be performed if different views of parts of the virtual canvas were presented close to the user. This would minimize user physical effort and increase the visibility of remote locations: rearranging content over a large area could be easily performed if the virtual canvas was presented in a scaled down view, where between-content distances are more manageable; remote locations could be visible in detail if the user had a view of the remote location presented in a proximal area; comparison or focus switching between spread out content could be achieved by presenting close to the user views of the spread out content.

Based on the above observations, we began our exploration by using a simple magnification lens, that presents a view of the virtual canvas either at higher or lower resolution than the canvas itself. The magnification factor could be adjusted through a fast slider [McGuffin et al., 2002], invoked using a marking menu [Kurtenbach and Buxton, 1993]. The lens was in fact a DragMag lens [Ware and Lewis, 1995], since it is often desirable that the depicted area does not coincide with the area underneath the lens (for example when viewing a remote location, or comparing two remote items).

We soon observed that a mechanism for determining the area of focus of the lens on the virtual canvas was needed. Simple dragging of the focus point to adjust its position [Ware and Lewis, 1995] would be tedious and require extensive physical movement from the part of the user, contrary to our design goals.

Although a simple lens provides a visual representation of a remote location, if there is a need to interact with the specific location the user still has to walk to accomplish this task. Interacting with the remote location through the lens is thus a desirable property.

Even when providing interaction in the interior of the lens as a shortcut to the main virtual canvas, there is a discontinuity and visual division between the lens and the main canvas. The user is presented with two discrete views on the display of the same content.

It might be desirable to provide a mechanism for connecting the two views and allow for content to transition from one to the other.

Finally, if such a lens is to be used on a wall display, where a large number of items is visible at any given time, it might be useful to filter particular items to reduce clutter (for example only display windows of a particular application, or only UI elements). In other words the lens could perform some semantic filtering or specialized rendering, as in magic lenses [Bier et al., 1993]. Although this is an interesting property discussed later in the chapter, it violates our application and data agnostic design goal.

This initial exploration revealed several desirable design elements for alternative view mechanisms to adequately aid context switching or layout management tasks. These elements are collected into a new framework, which we name canvas portals framework, since the techniques that can be described by it provide users with a portal to different locations, not only of the areas visible on the display, but of the entire virtual canvas.

3.2 Canvas portals framework

In the most abstract sense, canvas portals [Bezerianos and Balakrishnan, 2005b] are alternative views of different areas of the main virtual canvas. Interaction within the interior of a portal is equivalent to interacting with the depicted area. Canvas portals act as graphical and interaction shortcuts to the remote depicted areas. Since in their simplest form canvas portals present diverse views of a single virtual canvas, they may be implemented at the window manager level irrespective of the nature of applications or data representation on the wall display, satisfying our first design goal.

Canvas portals have four interactively manipulated design elements:

- The *portal area* is the graphical representation of the canvas portal within which interaction occurs (Figure 3.1). It can exist on the main wall display or on entirely separate devices such as PDAs or laptops. The portal area can have different shapes

and sizes, and users can reposition it as desired. The portal areas shape is typically closely related to that of the focal area, such that the mappings between them are simple and easily understood.

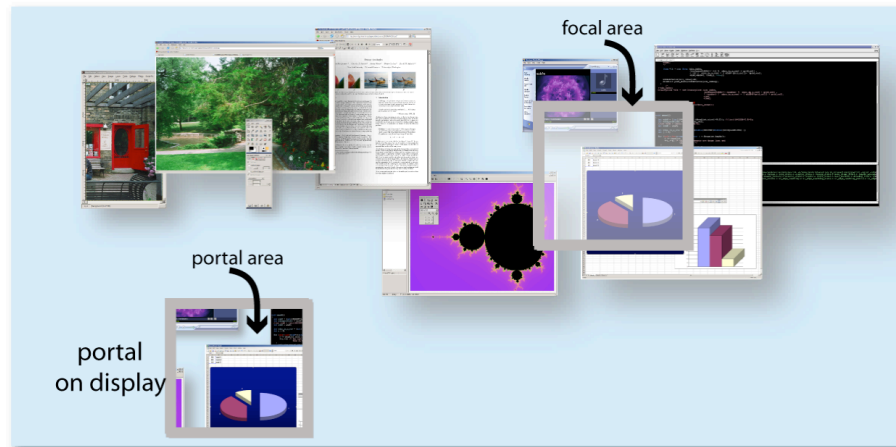


Figure 3.1: Canvas portal position, portal area and focal area example. The *position* of the rectangular portal is on the display, at the lower left. The *portal area* (interior of the portal) presents a view of the remote *focal area*. The position of the portal (and so the position of the *portal area*) as well as the focus of the portal (*focal area*) can be changed.

- The *focal area* is the remote part of the main wall display canvas on which the portal area's view is centered (Figure 3.1). Interaction with elements inside the portal area is equivalent to interaction with the actual elements within the remote focal area on the main wall display. The system should be designed to map input events occurring within the portal area to the focal area, either directly or with a transformation appropriate to the canvas portal design. This remapping can accommodate viewing and interaction with remote locations, minimizing physical movement.

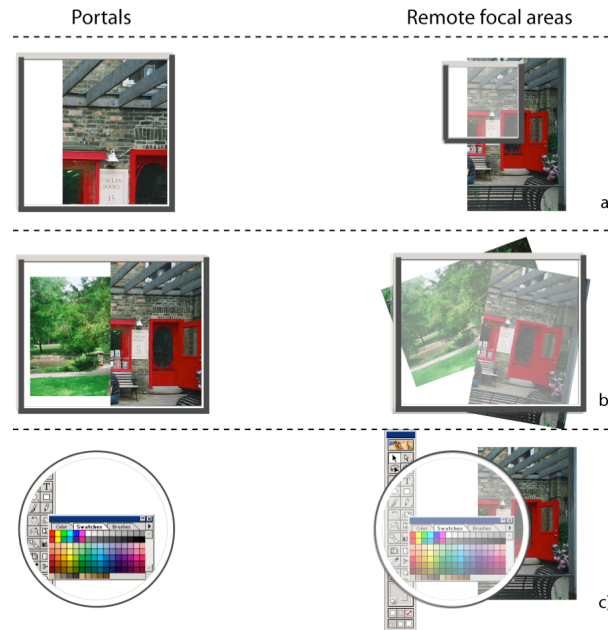


Figure 3.2: Canvas portal shape and attribute example. The left column shows the portal widgets (ie the portal areas of the canvas portals), which can exist on the main display; the right shows the remote focal areas on the displays. (a) Rectangular canvas portal with scale attribute modified to act as a zoom lens to the focal area. (b) Rectangular canvas portal with modified time attribute displaying a previous state of the focal areas content. (c) Circular canvas portal with semantic attribute modified to show only user interface elements, creating a tool palette.

- *Attributes* describe the portal area’s scale, semantics, state in time, and so on. In the simplest case, the portal area displays exactly the same content as the focal area. However, users might want to alter the portal area’s display attributes such that it displays a transformed version of the focal area’s content. By varying the portal area’s attributes, the canvas portal acts as a magic lens [Bier et al., 1993] to the display’s focal area. For example manipulating the portal area’s scale attribute, makes the portal area act like an interactive zoom lens to the focal area. Similarly, manipulating different rendering attributes lets users view, for instance, solid objects as wire frames within the portal area. Manipulating semantic

attributes can limit the portal area to depict a certain type of object. For example, the portal area might display only user interface elements, thus enabling on-the-fly tool palette creation. By manipulating time, the portal area can depict the focal area's state at a previous time instant. An attribute could even be the user to whom the portal belongs, letting different users maintain different views of the canvas layout. Figure 3.2 shows canvas portals of varying shapes, sizes, and attributes.

Some of the attributes discussed here may be implemented in an application independent manner at the window manager level (for example zoom scale, state in time, or user dependent views); others need to access application data (for example different rendering types for 3D objects, automatic filtering of user interface pallets, etc). While the framework incorporates all types of attributes, in our proposed designs we mostly discuss scale, that can be implemented in an application independent manner.

- Finally, *boundaries* define the transition zone between the portal area and the main canvas. Unlike regular magic lenses [Bier et al., 1993], canvas portals support the passing of objects between the portal area, the main display canvas, and other canvas portals. If a user is moving an object on the main canvas and the center of movement (typically the cursor) crosses a portal area's boundary, the object transitions into the portal area and continues its movement inside the portal area's coordinate system. The inverse also holds. Transition boundaries can be the portal areas discrete physical limits, a transition area beyond the portal area, or widgets in the portal area that the user interacts with to initiate the transition.

Consider, for example, a canvas portal with a zoomed-in scale attribute where the portal area's discrete border acts as the boundary (Figure 3.3). The cursor's center of movement is thus the center of translation and scaling of the objects being moved. When items cross a boundary, the system notifies the user in one of several ways.

In one implementation, when parts of a moving item cross the portal boundary, the system casts a translucent rendering of these parts on the other side of the boundary, indicating that the user can extend the movement outside the current working reference frame, which can be the main display or the portal itself. Thus, users can access remote objects by directly interacting with them within the portal, or by moving the remote object nearer to them on the main canvas by crossing the boundaries, thus providing an efficient layout management mechanism.

By supporting the seamless transition of items between a canvas portal and the main display canvas, the framework lets us quickly rearrange items in and out of the portals without always having to refocus the portal. On the contrary, if a user wants to view an item through a regular magic lens she needs to reposition the lens in a way that the item is directly underneath. Transition also enables interesting use scenarios, particularly when multiple canvas portals are active simultaneously. For example, after interacting in detail with an object in a canvas portal using a zoomed-in scale attribute, a user can push the object into a second zoomed-out canvas portal acting as a temporary space for working items without cluttering the main display area. This aided context switching is especially crucial to large display interaction because alternating between context levels by simply viewing the display from different distances can prove cumbersome. Thus both visibility and interaction with remote locations can be achieved with small user effort.

The four design elements let us instantiate canvas portals into a variety of existing and new interaction techniques. Each technique modifies design elements differently, including giving users control over some parameters while constraining others to a predefined value. For example, a Manhattan lens [Carpendale and Montagnese, 2001] (a lens whose focus is not the area under the lens) can be instantiated by fixing the portal area's position while letting users reposition the remote focal area interactively.

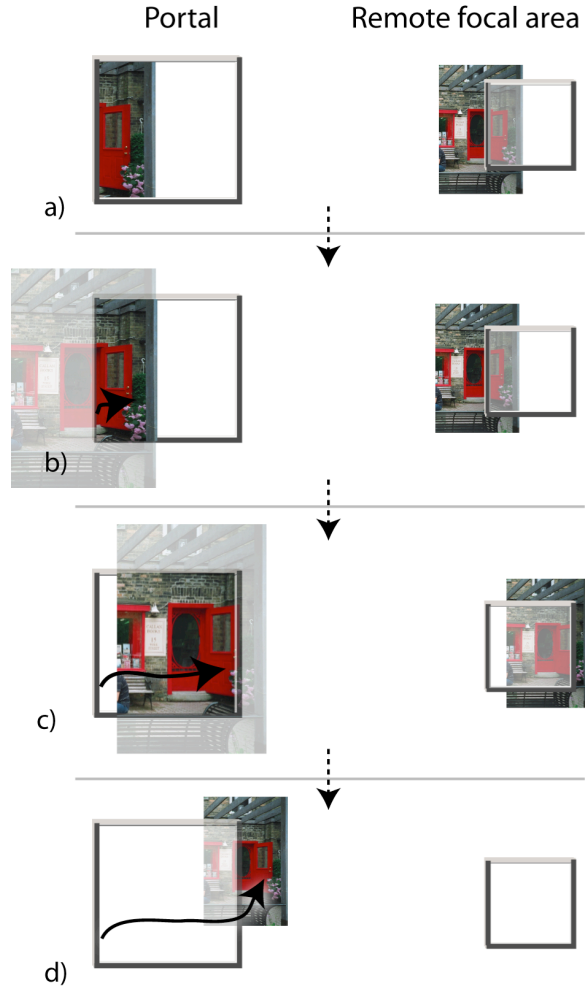


Figure 3.3: Images on the left are canvas portal widgets (that can exist on the main display canvas); those on the right are remote focal areas of the portals on the display. (a) A canvas portal with a zoomed-in view of the remote location. (b) The user selects an object inside the portal. A semi-transparent rendering of the object beyond the portal indicates that the movement can be continued outside the portal. (c) The user has moved the object, that has also moved in the remote focal area on the right. (d) As the object passes the portal's boundary, it transitions to the main display canvas on the area close to the location of the portal, and disappears from the remote focal area on the left. A semi-transparent rendering of the object in the portal indicates that the user can transition the object back inside the portal and thus "transport" it to the remote focal area again.

Canvas portals act as a context switching mechanism as well as aids for fast layout management, by providing users with multiple views of different areas of the display and allowing items to fluidly cross from one view to another. Furthermore, the passing of content between the main canvas and a canvas portal focused on a remote focal area also enables their use as remote reaching techniques.

The framework [Bezerianos and Balakrishnan, 2005b] incorporates existing layout management techniques that could be used on wall displays [Swaminathan and Sato, 1997, Guimbretière et al., 2001, Khan et al., 2004, Robertson et al., 2004, Tan et al., 2004b] and can be used to identify areas of improvement for these techniques. Moreover it enables development of new interaction techniques [Bezerianos and Balakrishnan, 2005b] that address some of the challenges outlined above in new ways.

3.2.1 Instantiating existing techniques within canvas portals

To illustrate how the framework could identify and address some limitations of current techniques, we can represent existing published work on alternative views for large displays in the canvas portal framework.

Pad++ portals. Bederson and Hollan [1994] assume the contents of information spaces are spatially spread out over an infinite canvas of different magnification levels. Interaction with the canvas is based on a panning and zooming metaphor. They discuss the use of "portals", i.e. lenses for peering into the contents of different parts of a canvas at different granularity. Interaction within the Pad++ portals is equivalent to interaction with the canvas and different display "filters" may be active (Pad++ portals may display specific content types). Pad++ portals are similar to regular canvas portals and can be implemented as such. Pad++ portals are always created at the area of the canvas they focus on, and refocusing requires panning within the portal. Thus they might prove problematic in wall displays, where areas of the canvas maybe unreachable

and distance to pan may be large. Moreover, they do not allow for seamless transition of objects between Pad++ portals, the way canvas portals do.

Dollhouse. Swaminathan and Sato [1997] identify problems that arise because of the scale of wall displays, including pointer movement and control challenges over large distances. They propose using a dollhouse metaphor: a small-scale model of the display and its contents to specify pointer movement in the wall display. We can implement their technique as a canvas portal with a modified scale attribute whose focal area encompasses the entire physical display. Movements inside the portal area cover larger distances than on the display, accelerating item selection. Although well suited for distance reaching, the technique lacks the potential for complex layout management, comparison and sophisticated context switching, because the dollhouse view uses a fixed scale factor and doesn't permit passing objects between views.

ZoomScapes. Guimbretière et al. [2001] introduce ZoomScapes, regions of the display with different zoom levels. Groups of objects crossing ZoomScapes are scaled around the center of the cursor's movement in a continuous manner. We can implement ZoomScapes as canvas portals with portal and focal areas of fixed positions, scale attributes that are modified to reflect different zoom levels, and boundaries that support scale attribute transitions as objects cross between the canvas portals corresponding to the ZoomScapes. ZoomScapes is an excellent space management technique. In its present form, however, we can't easily extend it to address distance reaching issues because it can't connect to a particular remote area of the virtual canvas.

Scalable fabric. More recently, several papers have described interesting techniques that let users define alternative work area views. Scalable fabric [Robertson et al., 2004] enhances existing window managers using an area around the user's main focus to scale down windows and create user-defined, task related groups. Selecting a task group in

this area brings it into focus, replacing the previous group in the main display area. Although designed for regular desktop computers, scalable fabric areas are like non-interactive canvas portals with varying scale attributes and interchangeable focal and portal areas. In the context of high resolution displays, scalable fabric could be enhanced by allowing users to interact with the scaled-down versions similarly to interacting with the actual windows, given that the high resolution ensures that smaller-scaled versions of objects remain appropriate for detailed interaction.

WinCuts. Tan et al. [2004b] augment window managers by letting users acquire and interact with alternative views of arbitrary regions of existing windows. Unlike WinCuts, canvas portals aren't attached to specific windows or parts of them, but on canvas locations. Thus, when a window is hidden behind other windows on the canvas, it is also hidden within the canvas portal. Nevertheless, canvas portals with an interactively modifiable time attribute can provide functionality similar to WinCuts by focusing on display layouts in which the desired window is in focus. Alternatively, imagine a canvas portal with a user-specified top window attribute that ensures a particular window or portion thereof is always in focus within the portal area. WinCuts are effective as view mechanisms of context in remote areas, but they don't accommodate layout management since transitions to and from the main canvas are not supported and only interaction with the application window seen in a WinCut is supported.

Frisbee. Khan et al. [2004] introduced the Frisbee, a widget acting as a telescope to a remote area on the display (Figure 2.2). Users manipulate remote items by interacting with their proxies within the Frisbee's main area and reposition items on the main display by moving them through specified transfer channels. Users can manipulate the remote focal area's position and the scale of the items within the Frisbee. We can instantiate the Frisbee within the canvas portals design using a single canvas portal with a modifiable scale attribute and define boundaries in specific locations around the portal area.

Technique	Attribute	Portal area	Focal area	Boundary
Pad++ portals	scale	rectangle area	custom rectangle area	no crossing
Dollhouse	scale	scaled canvas shape	entire canvas	no crossing
ZoomScapes	scale	area of canvas width	areas of scaled canvas width	smooth crossing when traversing boundary
Scalable Fabric	scale	rectangles along the display's border	areas along canvas border	smooth crossing when entering area border
WinCuts	time	custom rectangle	rectangular areas encompassing desired part of a window at a particular time	no crossing
Frisbee	scale	round shape	custom round area	smooth crossing at transfer channels
ScaleView*	scale	rectangle	rectangle	smooth crossing at edges
Window portal*	scale	rectangle	entire canvas and selection timeline	smooth crossing at edges and item in focus
Division Band*	scale	rectangle along a canvas dimension	areas of (scaled) canvas width or height	smooth crossing at edges

Figure 3.4: Instantiations of existing techniques and newly designed ones (*).

Figure 3.4 summarizes how we represent these techniques with their current functionality within the canvas portal design space. Note that we've focused on the conceptual representation of canvas portals and its various instantiations rather than the fine details that are implementation dependent but not fundamental to the underlying idea for each technique, such as interface elements for modifying parameters. The figure also incorporates newly designed techniques discussed later in the chapter.

3.3 New canvas portal instances

We developed several new techniques within the canvas portal design space and have implemented and tested them in our laboratory: ScaleView portals, Window portals and Division Bands [Bezerianos and Balakrishnan, 2004, 2005b]. Preliminary user feedback has led to ongoing refinements. Although canvas portals can have diverse rendering attributes, in our designs we decided to concentrate on scale. Given the potential for large amount of high resolution data, it is inevitable that scaling (implicit or explicit) will be a major factor in large display interaction.

Our implementation of the following techniques uses an object-oriented approach. We first created a generic canvas portal class incorporating the design elements. Each technique is an instance of the generic canvas portal class, inheriting the functional characteristics of the parent class and extending them with additional routines to enable specific interactions. In designing the diverse interactions for each of the techniques we ensured that user actions took place comfortably close to the user's location, adhering to our minimal motor effort goal. Although these techniques are prototypes, their design assumes nothing about the applications presented on the wall display or their data structure, in accordance to our application independence goal. Interactive items that are identified as such from the system are equally distinguished by the portal instances, since input events inside the portal are translated to the remote focal area. For example if on the main canvas dragging the top border of a window initiates movements, then movement is also initiated if the top border is dragged in a window depicted inside the portal. Such interactive items are able to move across portals and main canvas.

3.3.1 ScaleView portals

A ScaleView portal provides views of areas of the main canvas at different scales. The ScaleView portal, shown in Figure 3.5, appears as a window-like widget on the display and

has three user-controllable parameters: the focal area's position on the virtual canvas, the value of the portal area's scale attribute relative to that of the focal area's, and the portal area's position on the virtual canvas. Users can reposition a ScaleView portal by clicking and dragging a pin icon (see Figure 3.5). A FastSlider McGuffin et al. [2002] invoked from a marking menu Kurtenbach and Buxton [1993] attached to the portal adjusts the scale factor. Users can alter the focal area's position in two ways. The most direct and precise method is to select the "change focal area position" item from the portal's marking menu and then click the desired location on the virtual canvas. This method however, can be inconvenient if the desired focal area is on a difficult-to-reach part of the display, or impossible to achieve if the area of interest is on a part of the virtual canvas that isn't currently displayed on the display.

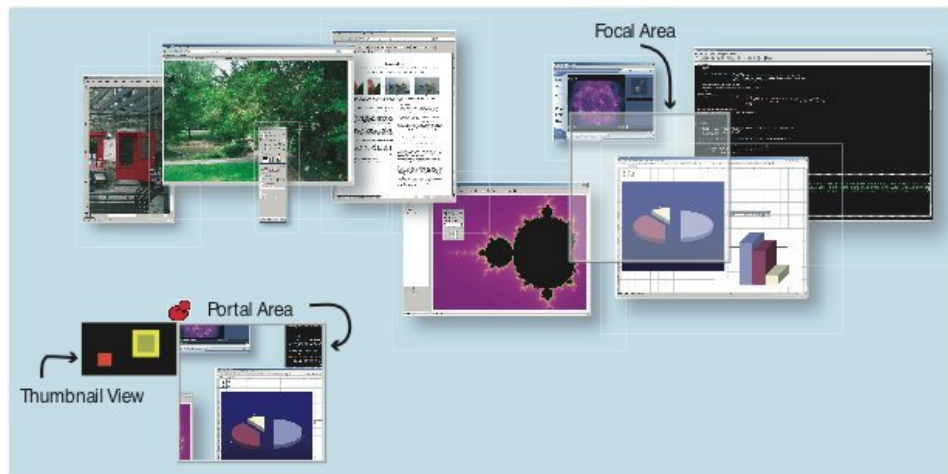


Figure 3.5: A ScaleView portal is an instance of a canvas portal. The ScaleView portal is located at the bottom left corner of the display. The red pin icon on the top-left corner indicates that the ScaleView portal can be moved. In this instance the portal is not moving, thus the pin icon is "pinned". The black rectangle at the side represents a thumbnail view of the virtual display canvas, within which the yellow rectangle indicates the focal area's position on the main canvas, and the red rectangle indicates the ScaleView portal's position. An outline rectangle on the main display canvas represents the focal area.

In the second technique the user selects the focal area at a coarser granularity using a thumbnail representation of the entire virtual canvas. This thumbnail is attached to the top left corner of the canvas portal and shows an iconic representation of the focal area as well as the portal's position on the main canvas. The user drags the icon representing the focal area to reposition it. As the user drags the icon, the system provides context by highlighting the corresponding region on the main canvas with a semi-transparent overlay. This approximate way of changing the focal area lets the user operate the ScaleView portal without having to move around the display. The user can also reach areas at the display's extremities or areas of the virtual canvas not currently visible on the display.

The ScaleView portal's boundaries are its graphical borders. If the user drags an item out of the portal, the system removes the item from the remote focal area and positions it on the main virtual canvas at the position of the user's actions. Conversely, a user can drag an item from the main canvas into the ScaleView portal, which automatically moves it to the portal's remote focal area.

ScaleView portals let users interactively adjust the scale factor between portal area and focal area, so users can easily select remote or small-sized targets using zoomed-out or zoomed-in ScaleView portals. This enables ScaleView portals to act as remote reaching mechanisms.

Our implementation lets users pan the entire virtual canvas across the display. To retain a canvas portal view when panning the virtual canvas, users can pin a ScaleView portal to a particular area of the virtual canvas, as opposed to an area of the physical display. For example, if the user focuses and pins a ScaleView portal on a group of windows, that focal area stays unchanged as the user pans the virtual canvas.

In our system all interaction occur close to the display, yet when interacting with large-scale, high-resolution displays, users might want to interact from slightly further away, perhaps using different input modalities. Although moving away from the display

gives users a better content overview, the functionality provided by ScaleView portals can still be useful. For example, they provide additional views so users can switch context less frequently or faster when moving between tasks or applications. They also provide detail and overall views of the virtual canvas simultaneously, which is useful at any scale of interaction. Finally, users can group items and move them seamlessly across different scales and views even when interacting at a distance. ScaleView portals thus enable both context switching and layout management.

To transfer a ScaleView portal’s view to the entire display, a selection from the portal’s marking menu warps the entire display view to match that of the canvas portal, centered at the portal’s location. This way, the portal lets users specify viewpoint transformations for the entire display by matching the focus and portal areas.

In our implementation, ScaleView portals are displayed on the same display as the main canvas (although they can also reside on a remote device). Large numbers of ScaleView portals are cumbersome to relocate, require considerable display space, and can lead to overlapping. To alleviate these issues, we allow users to group multiple ScaleView portals into stacked, side-by-side, or cube layouts, such as those in Figure 3.6. Users switch between portals in a stacked or cubed group by either touching a constituent portal, which brings that portal to the top with an animated transition, or by expanding the stack and cubed view into a side-by-side layout, where the user selects a new portal by clicking on it. Although we’ve only implemented three grouping techniques for ScaleView portals, many other groupings are possible.

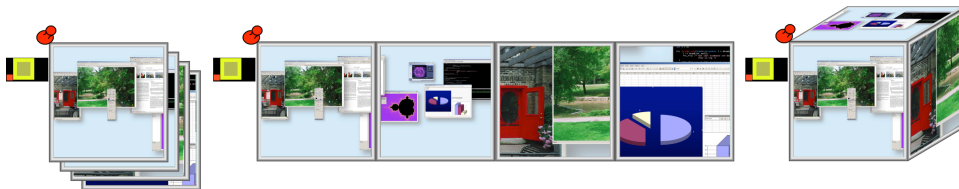


Figure 3.6: Grouping of multiple ScaleView portals: (a) stacked, (b) side-by-side, and (c) cube layout. Clicking on a portal brings it to the top.

Having multiple ScaleView portals active can make it difficult to distinguish which portal area corresponds to which focal area on the virtual canvas. To mitigate this, each time a transition between multiple portals occurs, we currently display a connecting line between the portal and focal areas. Nevertheless, as the number of portals increases, several of them might point at similar areas of the virtual canvas, making it difficult to distinguish them.

3.3.2 Window portals

Window portals, illustrated in Figure 3.7, are a variant of canvas portals providing quick access to, and switching between, application windows on a wall display. The main overview region of the Window portal is functionally equivalent to a ScaleView portal with a focal area covering the entire virtual canvas (that is, acting as a zoomed-out view of the entire canvas). When a user clicks on an item in the overview region, the system expands the item, displays it as an active object next to the overview region, and gives it a red border to indicate its special status. The user can interact in detail with the selected item or select a new one. As the user selects objects over time, a thumbnail representation of all previously selected items is displayed in a timeline over the current item. Clicking on a thumbnail turns the associated item into the active item. Thus, whereas the Window portal’s overview region acts as a spatial locator of items on the main canvas, the timeline region acts as a temporal locator for recently used items.

In addition to serving as a context-switching mechanism between an overview and detailed interaction with a specific object, Window portals provide spatial and temporal shortcuts to potentially remote objects. Unlike the overview areas of virtual desktop managers, the Window portal’s overview is fully interactive, allowing coarse actions in the overview and fine-grained actions in the detailed application window.

On most standard windows desktop, the Alt-Tab key combination lets users switch between application windows. When interacting with a large display without a conveniently

accessible keyboard, Window portals provide similar spatial and temporal switching between active windows. This functionality can also be useful in smaller displays lacking an easily accessible keyboard, such as tablet PCs in a slate configuration.

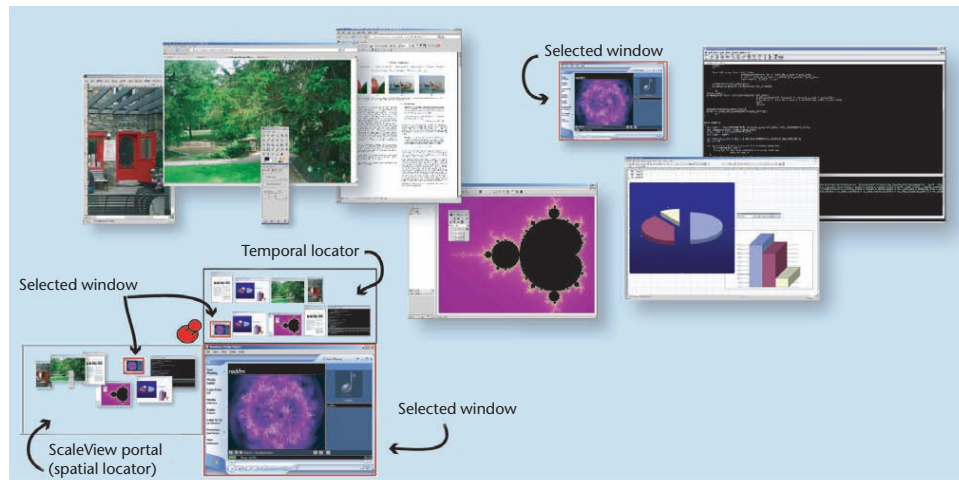


Figure 3.7: Window portal on a large display. An overview region on the left of the Window portal (a ScaleView portal depicting the entire display) serves as a spatial locator, a temporal locator (timeline) is on top of the widget, while the selected object is on the bottom right. Red borders around the selected object in all views help maintain context.

If we were to enable semantic filtering of items in the Window portal’s overview region, we could restrict the display to, for example, user interface widgets. Thus, we could create on-the-fly palettes of interface widgets that we could move around the display to operate multiple applications from a single locale. Although semantic filtering isn’t currently implemented, since it violates our application and data independence design goal, users can create palette portals themselves.

Given that the Window portal consists of a zoomed-out view of the main canvas, issues related to selecting and distinguishing between small targets arise. Although semantic filtering can limit the selection space, the potential number of items (overlapping or not) can still be large. Moreover, semantic filtering doesn’t address the issue of selecting small targets. Allowing dynamic zooming or expanding of targets is a potential solution, that

is furthermore consistent with our application independence goal.

Overlapping windows on the canvas can be problematic in Window portals as in any other multiwindow management system. Current solutions include tabs [Beaudouin-Lafon and Lassen, 2000, Beaudouin-Lafon, 2001], peeling [Dragicevic, 2004], and multi-blending [Baudisch and Gutwin, 2004]. Although we do not expect overlapping to be a major issue in wall display interaction, we’ve implemented an alternate approach that fans out a group of overlapping windows when a user clicks on any member of the group, in a manner similar to the widgets presented elsewhere [Vernier et al., 2002, McGuffin et al., 2003]. Users can select items from this fanned-out display, or collapse the group again. By default, we treat overlapping objects as a group. This enables not only the fan-out operation, but also lets users move the entire group as a whole within the Window portal. We’ve implemented this grouping feature across all our widgets.

3.3.3 Division Bands

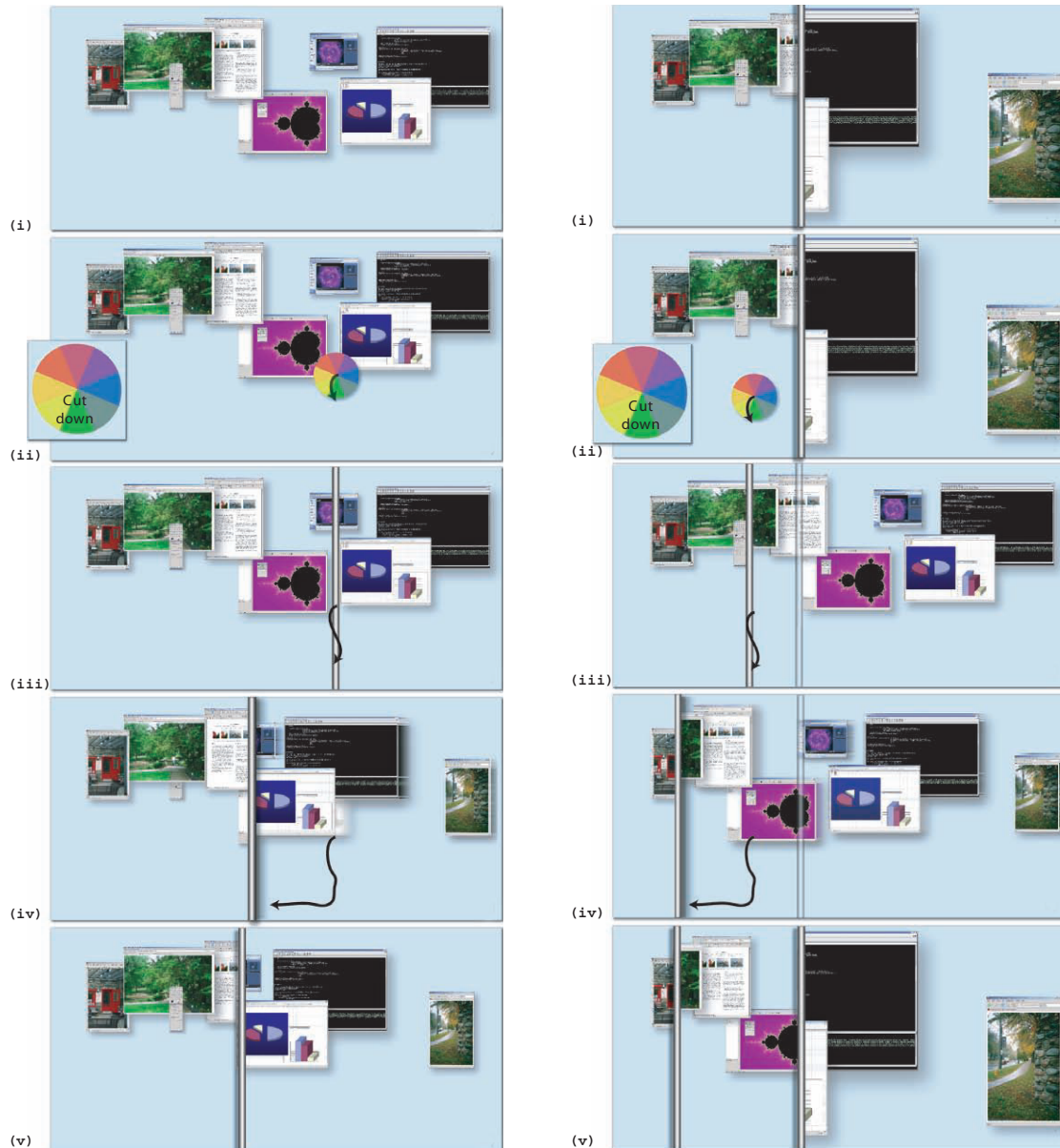
ScaleView portals and Window portals provide alternative ways to view and access data while essentially preserving the main virtual canvas’ overall view and layout. Division Bands, shown in Figure 3.8, also provide alternate views of the virtual canvas, but unlike the other techniques, they temporarily disrupt the virtual canvas’ overall view. They act much like a cutting tool that virtually slices the canvas along specified vertical and horizontal directions and lets users drag the cut portions around to quickly reveal more or less of certain parts of the virtual canvas. In essence, they are ScaleView portals with a fixed focal area and a fixed portal size equal to the display, but with boundaries that can be quickly repositioned.

Using a marking menu (Figure 3.8(a)ii), the user specifies the cut position (the menu’s invocation point) and direction (the mark used in the menu selection). We deliberately placed the cut commands in menu locations that would require selection marks in the corresponding directions to facilitate a fluid, combined specification of command and

parameter [Guimbretière and Winograd, 2000]. After the user specifies the cut position and direction, a vertical or horizontal (depending on the specified direction) Division Band widget appears on the display, and the system attaches one of the cut pieces of the canvas to it, based on the specific mark on the menu (Figure 3.8(a)iii). If the user specifies cut-down and moves to the left, for example, a vertical Division Band appears with the right side of the canvas attached to it. This Division Band is now attached to the user’s pointer and can be moved (Figure 3.8(a)iv). By dragging the Division Band left to right (or right to left) on the display, the user can shrink (or expand) the attached portion of the canvas. A quick flick-and-release motion of the cursor dismisses the Division Band. If the user releases the Division Band without the flicking motion, the Division Band remains on the display for subsequent reselection. Thus, Division Bands let users quickly drag a part of the display toward them for viewing and/or manipulation. The ability to quickly dismiss the Division Band with a flicking gesture allows for transient, quick views of remote portions of the display, much like pulling on a spring-loaded window blind.

Users can position multiple Division Bands of different orientations and directions on the display, as Figure 3.8(b) illustrates. The system presents multiple, pinned Division Bands of different orientations and directions in a fixed ordering of creation, unless one band completely occludes another (as seen in Figure 3.8(b)iv). In this case they are arranged so that both are visible (Figure 3.8(b)v).

The quick creation and dismissal feature of Division Bands makes them well suited for fast, glancing actions at remote content or parts of the virtual canvas currently not visible on the display. With the pinning option, Division Bands can also serve as persistent shortcuts to remote areas of the display or as virtual desktops, similarly to the Flatland [Mynatt et al., 1999] flip-charts. As with all canvas portals instantiations, our implementation lets users move objects between the main canvas and portions of the canvas attached to Division Bands. Division Bands thus can act as both layout management and remote reaching mechanisms.



(a) Single Division Band creation. (i) Initial state. (ii) The user invokes the marking menu. (iii) A new vertical band is created at the location of the marking menu and gets attached to the user's cursor. (iv) Pulling the band to the left reveals more of the canvas to the right. (v) Final state of the display after the user releases the band. If the user makes a quick flicking action instead, the band is discarded. (Black scribbles indicate user strokes.)

(b) Second Division Band creation. (i) One band is already present. (ii) The user invokes the marking menu and (iii) a second vertical band is created at the menu location, attached to the user's cursor. (iv) While the user moves the new band, it overlaps the existing one. After the user releases it (v), the bands are sorted so that both are visible. Here the original band is positioned on top of the newly created one. (Black scribbles indicate user strokes).

Figure 3.8: Division Bands

Newly created Division Bands have the same scale factor as the regular canvas, but we provide a menu option to enable users to change the scale, allowing for the creation of a form of ZoomScapes [Guimbretière et al., 2001]. Because bands can have their own zoom factors, users can move items back and forth between bands for different visualization effects. For example, users can store objects in zoomed-out bands to save space or bring them into zoomed-in bands for detailed inspection. Although we haven't implemented ZoomScapes' elegant scaling transition of groups of objects, we could incorporate this characteristic of the canvas portal frameworks boundary into the different techniques.

Division Bands are similar to the split-window mechanism of recent word processors in terms of splitting of the canvas, but they also provide transient and fluid ways to specify, invoke, and dismiss the splits, thus improving usability. Division Bands also enable sophisticated context switches between bands.

3.3.4 Preliminary user feedback and design improvements

Four participants (1 female) aged 23 to 30, explored our techniques for about an hour and commented on their functionality. Two were familiar with user interface research, but the other two were naïve users.

All users quickly grasped the benefits of the ScaleView and Window portals. Yet two users pointed out the lack of fine grained adjustment when selecting the remote focal area using the ScaleView minimap, and the awkwardness of moving around to directly position the remote focal area. We decided to provide an intermediate control for changing the remote focal area: vertical and horizontal scroll-bars with scroll-up and down buttons. These allow fine tuning the remote focal area position, after using the minimap for an initial course placement. Moreover their functionality is immediately obvious to the users, as it follows the common windowing system scroll-bar metaphor.

Two users were concerned that multiple ScaleView portals might be confusing. We further explored this issue by examining different ways to visualize the connection be-

tween portal and focal area and the effect of multiple portals (Section 3.4).

One user mentioned that Window portals seem most useful when switching between applications and windows, while ScaleView portals are a more general purpose tool.

Two users did not immediately grasp the purpose or operation of the Division Bands. But after a brief explanation they found the quick glancing and spring-loaded releasing action quite useful and indicated that it helped refresh their view of the state of the entire canvas. As per the comments of two users, vertical bands are seen more as temporary glancing tools, since they provide access to remote areas, and will likely be used more transiently to perform quick glancing actions. Whereas horizontal ones are seen more as space creation mechanisms, by dragging unused portions of the virtual canvas into view from the top or the bottom of the display. Horizontal bands might thus be of a more permanent nature, likely to be followed by a permanent pinning action. These comments led to the decision to place vertical Division Bands on top of horizontal ones, since the presence of the former on the display is more likely ephemeral. Bands of similar orientations are still stacked so that they are all visible.

One user got confused when new Division Bands were created, not knowing which band was currently being used. This issue was more pronounced when the band was released and existing bands were placed on top of it so as to remain visible (Figure 3.8(b)iv,v), causing sudden visual transitions. We modified our designs so that active (for example moving) or newly created bands are rendered semi-transparent and on top of existing bands, while the main canvas and existing bands remain visible underneath. When the band is no longer active, it gets stacked accordingly and becomes opaque. This serves a double function: on the one hand the semi-transparent nature of the active band uniquely identifies it and differentiates it from other Division Bands and the main canvas; on the other hand, the layout of the remaining bands is partially visible through the active band, so when the latter gets released and is placed in its final stacking order, the existing bands of the stack are already partly visible and don't visually pop-up.

From the three proposed canvas portal instances (ScaleView, Window portal and Division Band) we feel ScaleView portals are more interesting from an interaction perspective: a Window portal is a simplified version of a ScaleView portal plus a timeline and a magnified window, with similar usability issues regarding the portal component. Preliminary user comments also indicated that Window portals would be used for brief context switching tasks, with the bulk of interaction focused on the magnified selected window and not on the portal component. Similarly, based on user comments, Division Bands are perceived as lightweight and temporary access tools or storage mechanisms that are less likely to be interacted upon over extended periods of time. We decided to further investigate design and usability issues regarding ScaleView portals, as they were identified as more general purpose tools, likely remaining on the display longer.

We initially considered investigating the effectiveness of ScaleView by determining their performance in target selection tasks, as do existing wall display techniques [Baudisch et al., 2003a, Collomb and Hascoët, 2004, Collomb et al., 2005b, Forlines et al., 2006b]. Nevertheless, these techniques are mainly designed for brief access to remote locations of the display, while ScaleView portals are designed for extended use. We thus felt that it would be better to evaluate them on more complex and involved scenarios, similar to longer term tasks likely to occur on wall displays. Our goal for this evaluation session is to investigate if users understand the functionality of ScaleView portals and if they are willing to make use of it on diverse tasks, as well as overall usability issues (Section 3.5).

Before conducting this session, we decided upon investigating some design aspects of ScaleView portals further. Among several issues that could be investigated (such as visualizations of the "crossing" property of ScaleView portals, different designs for presenting the remote focal area, number of simultaneous ScaleView portals users can manage and how to visualize their grouping, possible invocation mechanisms, etc), we decided to focus on the visualization of the remote focal areas. A poor visualization design

of the connection between portal and remote focal area can clutter the wall display or render the technique unusable if users cannot make the mental connection between a portal and its remote focal area (Section 3.4).

3.4 ScaleView portal remote area visualization study

As ScaleView portals provide alternative views to different remote canvas areas, they segment the visual display space. Identifying the connection between the portals and their remote focal areas may be problematic, especially in the presence of multiple such portals. We thus investigated different focal area visualizations in the presence of multiple portals, by graphically connecting portals and their remote focal areas.

Task

The task performed was discrete target selection in the presence of distracter targets. Several ScaleView portals (stripped of menu functionality, scroll-bars and minimaps) were simultaneously presented on the display. A desired target was highlighted in a different color and participants were asked to selecting the proxy corresponding to the highlighted target, from inside one of the displayed portals. The proxy was not highlighted, so participants had to identify the correct portal to select the proxy of the desired target. All targets and their proxies were rendered as numbered squares. The goal target was assigned the number 1, while distracters were assigned random numbers from 1-3.

Visualization conditions (Figure 3.9)

NoVis: As a base case for our evaluations, no visual connection between portals and their remote focal areas was provided.

FullPortalVis: The remote focal areas of a portal is highlighted at all times and a semi-transparent band connects it to the corresponding portal.

PortalVis: As the user hovers over a portal, the remote area is highlighted and a connection between the portal and its remote focal area is drawn.

LineVis: As the user hovers over a specific item proxy in a portal, a connection between the proxy and the corresponding item at the remote focal area is drawn.

Initially we also examined another three visualizations in a pilot session: One permanently displayed connections between all proxies and remote items (*FullLinesVis*). Another permanently presented connections between all portals and remote focal areas, as well as between all proxies and remote items (*FullPortalLinesVis*). Both created a large amount of clutter on the display, even if only one or two portals were present, and were eliminated from further examination. In the third visualization (*PortalLineVis*), as the user hovered over a portal a connection to its remote focal area was presented, and if the user happened to hover over a specific proxy inside the portal, a line connection between the proxy and the corresponding remote item was also displayed. This visualization was deemed too distracting since several discrete visual elements highlighting remote focal areas and items would appear and disappear in very small time intervals.

In our tasks we tested 1 to 6 portals. We felt that managing and arranging a larger number of portals would require more sophisticated mechanisms than simple moving or stacking, and would introduce usability factors similar to those found in traditional desktop managers with multiple potentially overlapping windows.

For any visualization to be successful, it must work well in the presence of intervening objects that distract from the act of selecting the goal target. Distracter targets were placed randomly on the display to equally influence the remote focal areas of different portals. We used two distracter densities: Mid (25%) and Large (50%) display coverage.

A within-participant factorial design was used. The independent variables were visualization technique, density of distracter items and number of portals. To counterbalance for possible differences in movement direction, we positioned the starting location of trials in 3 relative positions about the centre of the display.

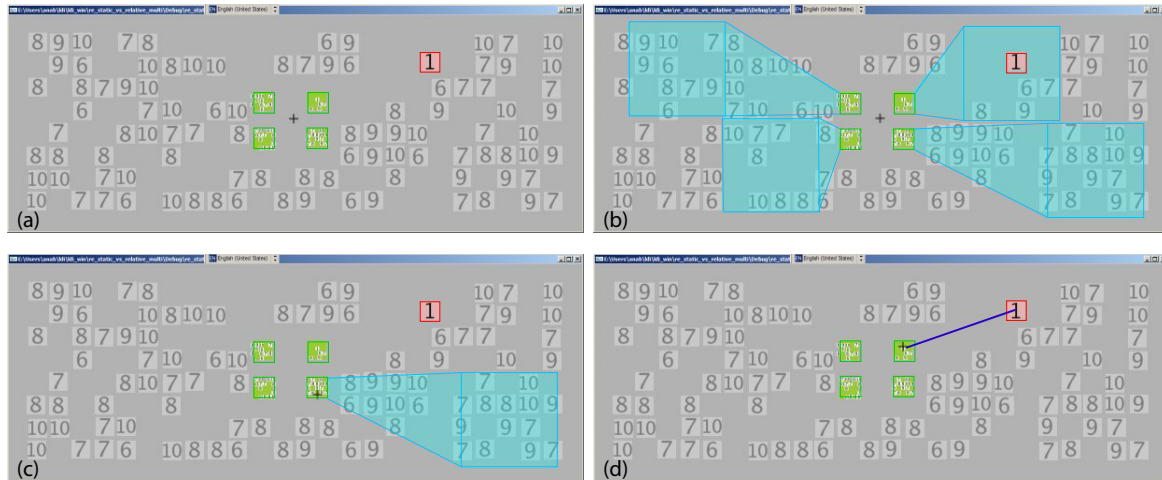


Figure 3.9: Visualization of remote focal area connections. (a) NoVis. The desired target is marked as red. The different portals are green rectangles, corresponding to 4 different remote focal areas on the display. (b) FullPortalVis. The remote focal areas of all portals and their connections are highlighted. (c) PortalVis. As the user hovers over a portal, the corresponding remote focal area and its connection is highlighted. (d) LineVis. As the user hovers over a proxy within a portal, a line connecting the proxy and the corresponding remote target is drawn. 4 portals are seen and 50% of the display is covered by distracters.

The aim of the study was to determine the best way to visually represent the connection between portals and their remote focal areas and was performed in two parts: in the first users had to differentiate between simplified ScaleView portals whose remote focal areas were highlighted throughout the study, simulating situations where users are *familiar* with the layout of the portals. In the second, remote focal areas were highlighted only as part of the different visualizations, simulating use of *unfamiliar* portal layout.

3.4.1 Familiar portal layout session

Eight volunteers (3 female), aged 20-35, participated in this session. Participants used all 4 techniques, in an ordering balanced using a Latin square. The study consisted of:

8 participants \times
 4 visualizations \times
 2 distracter densities \times
 6 number of portals \times
 3 starting positions \times
 4 repetition blocks
 = 4,608 trials

Prior to each technique participants were given a short warm-up session (10 trials) to familiarize themselves with the visualization techniques. The session lasted on average one hour for each participant, at the end of which they filled out a brief questionnaire to express their opinions.

Results

A total of 25 trials (.5%) were identified as outliers and removed from further analysis.

A fully within subjects ANOVA showed no significant effect for *visualization* on *TotalTime* (Figures 3.10.Left, 3.11.Left) and *ErrorRate* (Figures 3.11.Left,3.12.Left).

The number of portals affected users performance. There was a significant main effect of number of portals on *TotalTime* ($F_{5,15} = 94.405$, $p < .0001$). Post-hoc means comparisons showed that selecting targets in a single portal was easier (Figure 3.13.Left) and the cost of selecting the target increased with the number of portals (in all cases $p < .05$). The only non significant pair was for 5,6 portals. No other significant results were found.

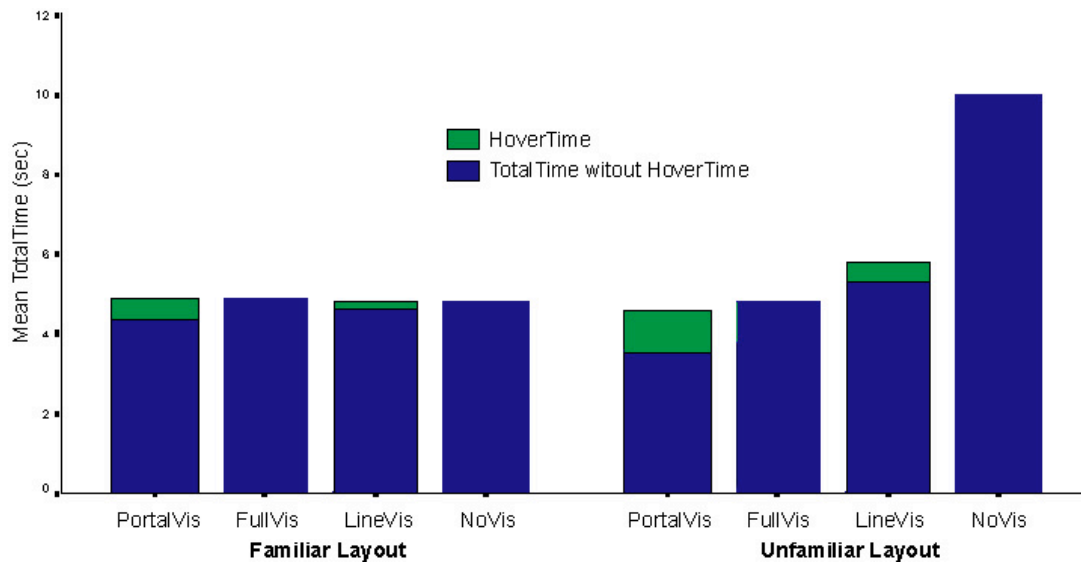


Figure 3.10: Mean *TotalTime* per visualization for the Familiar (Left) and Unfamiliar (Right) layout study. Time spent hovering in PortalVis and LineVis are indicated.

	Familiar		Unfamiliar	
	TotalTime	ErrorRate	TotalTime	ErrorRate
PortalVis	4.91 sec	2.9 %	4.60 sec	5.7 %
FullPortalVis	4.90 sec	3.3 %	4.78 sec	5.7 %
LineVis	4.84 sec	2.5 %	5.79 sec	4.3 %
NoVis	4.81 sec	3.4 %	9.99 sec	6 %

Figure 3.11: Mean *TotalTime* and *ErrorRate* per visualization for the Familiar (Left) and Unfamiliar (Right) layout study. Statistically significant measures are marked in red.

Discussion and user comments

Mean times for the different visualizations were very similar, indicating that users adopted similar strategies for all visualizations. In both FullPortalVis and NoVis most participants noted that their strategy consisted of memorizing part of the layout of the highlighted remote focal area surrounding the desired target, and then matching the memo-

rized layout with that of one of the portals. Four participants mentioned that they felt they were more accurate in the case of the PortalVis and LinesVis because they used the visualizations to check the target they picked before committing to a selection, delaying the interaction. Thus users adopted the same strategy across visualizations and hover was used as a verification mechanism.

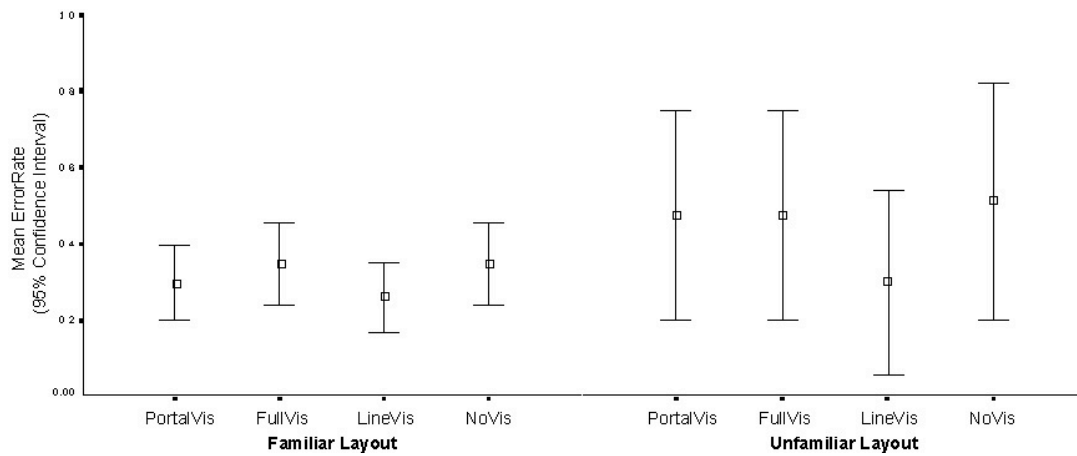


Figure 3.12: 95% of confidence interval graph of *ErrorRate* per visualization for the Familiar (Left) and Unfamiliar (Right) layout study.

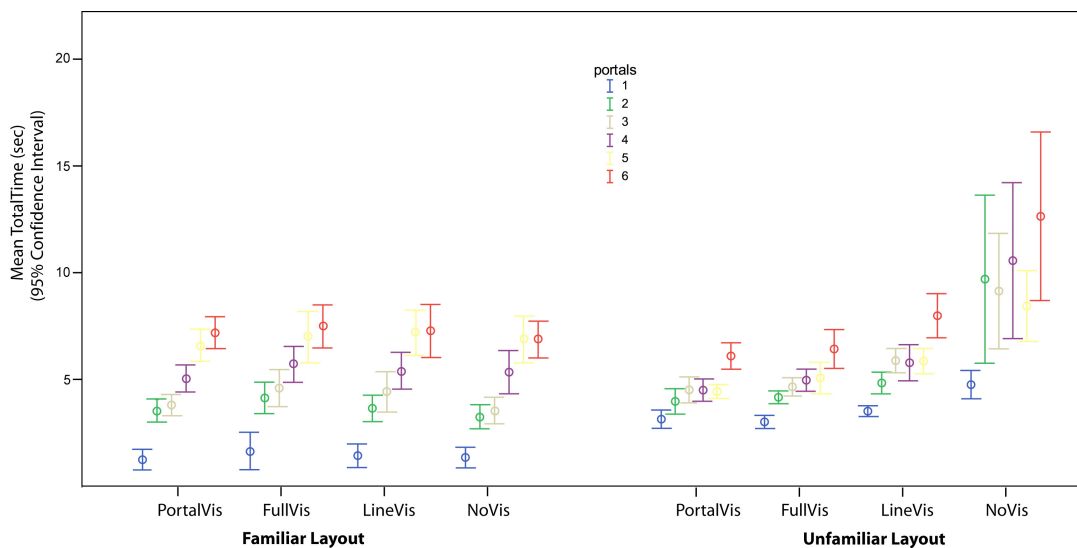


Figure 3.13: Mean *TotalTime* per visualization for the different number of portals. For both the Familiar (Left) and Unfamiliar (Right) layout study.

In terms of overall preference, 4 out of 8 users preferred LineVis, 3 the PortalVis and 1 participant the NoVis condition, indicating that hover based techniques were better received. Participants were also asked to identify the number of portals they felt comfortable identifying targets in. Five out of 8 participants felt that they could comfortably find the target only in 1 portal, while the remaining 3 felt that 4 portals or less was a comfortable number for the specific task.

3.4.2 Unfamiliar portal layout session

To examine if the results of the familiar layout session are similar to unfamiliar layouts, we performed a smaller session where we did not present the remote influence areas of portals, unless they are part of the visualization technique. We feel this task accurately represents searching targets when users are not familiar with the given portals (for example they haven't setup the portal or haven't used it in a while).

Two female and 1 male volunteers, aged 20-30, participated in the session. Data from a 4th participant was not used, since she could not complete the NoVis condition as she found it too hard. The independent variables were visualization, density of distracter targets and number of portals. Each participant used the four visualization techniques, in an ordering balanced using a Latin square. The experiment consisted of:

3 participants x
 4 visualizations x
 2 distracter densities x
 6 portals x
 3 starting positions x
 4 repetition blocks
 = 1728 trials

Participants performed 10 warm-up trials per technique and were asked to complete the trials as fast and accurately as possible, taking breaks between trials and blocks.

Results

Data from a total of 14 trials (.8%) were identified as outliers and removed from further analysis. As the distributions of the time measurements were positively skewed, significance tests were performed on the logarithm of *TotalTime*.

A fully within subjects ANOVA showed a significant effect for *visualization* on *logTotalTime* ($F_{3,6} = 15.3$, $p < .05$) (Figures 3.10.Right, 3.11.Right). Post-hoc means comparisons showed the NoVis condition to be significantly slower than all the others (all $p < .05$). The dropped data from the 4th partitioner would enforce this result, as she was unable to complete the NoVis condition because she found it too hard.

A significant main effect of number of portals on *logTotalTime* ($F_{5,10} = 62$, $p < .0001$) was also found (Figure 3.13.Right). In post-hoc means comparisons the single portal condition was found to be significantly faster than 6 portals, and the 4 portal faster than the 5 portal condition (all $p < .05$).

There was no significant effect for *visualization* on *ErrorRate* (Figures 3.11,3.12.Right).

Discussion and user comments

As the NoVis provides users with no visual indication of the portal's focal area, users had to rely solely on memorizing patterns around the target. This task was harder than in the first session, since now focal areas were not highlighted and there was no clear indication of the area around the target that had to be memorized. Thus the NoVis condition was found significantly slower than all other visualizations.

In terms of overall preference, 2 out of 3 participants thought that PortalVis was faster, and was their top preference. The last participant preferred overall LineVis. NoVis as expected was last in terms of preference for all participants and was the cause our fourth participant stopped the experiment. FullPortalVis was not preferred due to the complexity of the visualization, especially in the case of many portals. All three users agreed that it was easy to locate the target in up to 3 or 4 portals.

3.4.3 Design implications

The aim of the study was to determine the best way to present visually the connection between focal and remote portal areas of ScaleView portals and was performed in two parts, simulating situations where users are familiar or not with the layout of the portals.

User preference and mean times indicate that of the visualizations tried, the one that works best for both familiar and unfamiliar layouts is one that connects the portal area with a semi-transparent overlay that encompasses the remote focal area (PortalVis). The rendering of the remote focal area is active while the portal is being actively moved, its parameters manipulated, but also while the user hovers over its portal area.

User comments also indicated that for the tested layouts up to 3-4 portals are fairly easy to distinguish. The tested layouts included same color numbers ranging from 1 to 3 and were thus fairly uniform. In real life, it is likely that more portals could be used, if very strong color cues existed on the main canvas (for example different colored items). Color has indeed been shown as the most effective graphical mechanism for reducing display search time (followed by shape, and finally letters or digits) [Christ, 1975, 1984]. Furthermore, user labeling of the different portals and organization using tabs [Beaudouin-Lafon and Lassen, 2000] can further distinguish between multiple portals.

3.5 ScaleView portal usability study

The previous study helped determine appropriate remote focal area visualizations for ScaleView portals, and provided insights as to the ease of distinguishing different portals. Yet their overall usefulness and ease of use was not examined. To investigate the usability of ScaleView portals and gather feedback on their design a second study was conducted. Four female and 6 male participants, aged 24-35, took part in this session. Participants were asked to perform three tasks, resembling potential tasks identified in Section 2.3:

Layout task. A set of images and text from different sources (Figure 3.17Top) was present on the display, with text being readable without the need for magnification. Participants were asked to semantically group text and images as they saw fit. To ensure they didn't perform the task randomly, they were then asked to justify their grouping to the experimenter. This open ended task simulates real life layout management and organizational tasks, such as story-boarding, magazine layout etc.

Navigation task. Participants were shown two high resolution images of the western hemisphere (Figure 3.18Top) and were asked to concentrate on the left. A small red square (not visible without a magnification aid) was present on the land mass and users were asked to pinpoint it. This task resembles real life search and navigation tasks in different scales, such as examining high resolution images, dens graphs etc.

Comparison task. Participants were asked to locate four differences between the two high resolution images of the western hemisphere (Figure 3.19Top). Differences were visible with the naked eye, but more prominent when using a magnification aid. This task resembles real life navigation and comparison tasks, such as looking high resolution images for structural flaws, or examining different meteorological fronts, and then comparing them for similarities. If users did not find any difference between the two images in 10 min, hints were given. This intervention did not take away from the insights gained while users navigated the display during the first 10 min. Moreover, it doesn't negate the findings from the comparison component, since in real life comparison tasks areas of interest are often identified beforehand.

During the study, the heads of participants were tracked to approximately infer the their visual focus on the display. Participant actions were logged by the system and they were observed by one of the experimenters during their session.

The tools available for performing the tasks were ScaleView portals and simple magnification lenses, which were both explained to the participants in a brief training session. The training tasks were similar to the ones describe above, and were performed using

ScaleView portals, magnification lenses, and using no tool if possible (for example in a layout training task).

After the training session was over, users had the option of choosing whether to use no tools, regular magnification lens(es), ScaleView portal(s) or combinations of the above to perform the given tasks. This allowed us to observe the use of the tools for the different tasks on wall displays. It also provided us with insights as to how users chose between a simple, easy to use technique (like a magnification lens) and a more sophisticated, but more involved technique (as the ScaleView portal). Finally, it inspired design improvements for ScaleView portals. The following section summarizes user trends while performing the different tasks.

3.5.1 Observations

Layout task (Figure 3.17Top): Only one user, in Figure 3.17(v) chose to walk around the display and directly drag content to organize it in groups. She occasionally used a magnification lens to see content in detail. The few green marks visible are an accidental invocation of a ScaleView portal instead of the magnification lens.

ScaleView portals were used by the rest of the participants for this task. Two users, in Figures 3.17(iv),(vi), started the task using only a magnification lens, but very quickly switched to using two ScaleView portals, one zoomed-in to focus on particular items, and one zoomed-out to move items around. Nevertheless, these users did not make the connection of passing items "between portals", a functionality not explicitly shown to users in order to examine if the passing between portals felt "natural". Only passing from a portal to the main canvas was shown during the training session. These users moved the focus of the zoomed-in portal around the display using the scroll bars or the minimap instead of directly dragging items from the zoomed-out view. Another user, in Figure 3.17(x), used a single ScaleView portal that she panned, zoomed-in to focus on specific content, and zoomed-out to organize content repeatedly. When asked later about

passing content back and forth between portals, or between portal and canvas, all three users mentioned they did not recall or infer this functionality of the ScaleView portal.

One user, in Figure 3.17(i) was unique in using a combination of both a ScaleView portal and a magnifying lens. She placed the lens next to a zoomed-out ScaleView portal and would drag items from the portal under the lens to view in detail and then return them in the portal to arrange the layout. This user grasped the notion of transitioning between the portal and canvas. When asked later why she did not use a zoomed-in portal, she mentioned that she wanted the two tools to be separate and not attached, as in the current implementation. Another user, in Figure 3.17(ii) did not use a magnification tool. Her strategy was to move items from a single zoomed-out ScaleView portal to the main canvas, look at them and then organize them inside the ScaleView portal.

The remaining four users, in Figures 3.17(iii),(vii),(viii),(ix) made use of two ScaleView portals, one zoomed-out for organizing content, and one zoomed-in to view content in detail, by passed content between the two portals. This set of users took advantage of the mechanism for passing content to and from a ScaleView portal.

Three users did not perceive the passing of content from ScaleView portals to other portals or the main canvas as natural, did not recall or make use of it. Although the remaining 6 users did take advantage of the feature, stronger visual indications of the passing mechanism are desirable, to ensure that users are reminded of this option.

One question we had regarding the use of ScaleView portals, was whether users would focus visually on the portals, or use the overall display as a reference. From the 9 users that performed the task with ScaleView portals, four, in Figures 3.17(ii),(iv),(vi),(viii) often used the display as a reference (the percent of time looking at portals less than 70%), whereas five, in Figures 3.17(i),(iii),(vii),(ix),(x) performed the task by glancing little, if ever, outside the ScaleView portal area (percent of time looking at portals more than 70%). The percentage of time spent on looking inside ScaleView portals for each user can be seen in the first column of Figure 3.14.

We note that glances at the display are usually brief, whereas glances on the ScaleView portals are usually accompanied by interaction, which takes more time. Thus, if users spend more than 70% of their time on ScaleView portals we claim that they mainly focus on them and do not make heavy use of the large visual display. This decision is backed up by written comments taken by the researcher during the different sessions, where users observed to frequently glance at the main display canvas spend less than 70% of their time looking inside the ScaleView portal.

Navigation task (Figure 3.18Top): In this task, the majority of users (8 out of 10) initiated the search using a regular magnification lens, in Figure 3.17(i),(ii),(iii),(iv),(v),(vii),(viii),(x). They commented that magnification lenses were easier to use than ScaleView portals. Users interacting with the magnification lens spent time physically moving around the left part of the image in (Figure 3.18Top) (where the red square was located) and some were luckier than others in their search. 3 of these 8 users (Figures 3.18(i),(ii),(iv)) switched to using a single ScaleView portal when they could not quickly perform the task using the simple magnification lens. One of these three, in Figure 3.17(i), stated that her search was "not as systematic" with a regular magnification lens, as when using the scrolling functionality of the ScaleView.

The remaining 2 users, in Figures 3.17(vi),(ix) started and finished the task using one ScaleView portal because, as one mentioned "scroll-bars make it easy to navigate on the enlarged screen". Of these two users, one (Figures 3.17(vi)) spent time glancing at the display as a reference for the position of the focal area of the ScaleView, whereas the other (Figure 3.17(ix)) mostly focused on the displayed content of the ScaleView. In all cases where a ScaleView portal was used, visible green vertical and horizontal lines (Figure 3.17) indicate interaction with the scroll bars and small green rectangles with the minimap. The percentage of time spent on looking inside the ScaleView for each user can be seen in the second column of Figure 3.14.

Comparison task (Figure 3.19.Top): This was the most challenging task, since it

required detailed examination and visual comparisons between remote display locations.

The majority of users completed this task using ScaleView portals. One user (Figure 3.17(i)) originally invoked two magnification lenses, one for each globe. Another (Figure 3.17(viii)), invoked a single magnification lens that she attempted to move back and forth between the two globes. Both users switched to using two side-by-side ScaleView portals, each one focused on one of the two images. Another two users (Figures 3.17(ii),(iii)) originally used a magnification lens to focus on one globe and a ScaleView portal to focus on the other. The first completed the task as she started (Figure 3.17(ii)), while the other (Figure 3.17(iii)) reverted to using two side-by-side ScaleView portals. The remaining six users started and completed the task using two side-by-side ScaleView portals.

Again different users utilized the display real-estate differently. Some, seen in Figures 3.17(iv),(vii),(viii),(ix) almost completely concentrated on the views inside the ScaleView portals (percent of time looking at portal more than 70%), whereas others, seen in Figures 3.17(ii),(vi) made heavy use of the display as a reference for the location of the focal area of the portals. The percentage of time spent looking inside ScaleView portals per user can be seen in the third column of Figure 3.14. One user (Figure 3.17(iv)) had some trouble telling the portals apart and would confuse which portal was focused on which globe. As the visual representation of the remote focal area failed in this instance, labeling of portals, or more persistent visualizations, would have been beneficial.

3.5.2 User comments and design improvements

The majority of users (8 out of 10) stated that overall they preferred ScaleView portals for the given tasks. This is not surprising, given that users performed the greater part of the tasks using ScaleView portals, since, as stated, they allowed "more systematic" interaction and users "didn't have to walk around much". Nevertheless, user comments for improvement and observations led to design enhancements.

	Percent of time looking at ScaleView		
	Layout	Navigation	Comparison
(i)	77%	33%	47%
(ii)	47%	26%	50%
(iii)	75%	0%	31%
(iv)	59%	20%	85%
(v)	27%	0%	89%
(vi)	64%	54%	29%
(vii)	96%	0%	83%
(viii)	58%	0%	76%
(ix)	88%	70%	80%
(x)	77%	0%	68%

Figure 3.14: Percentage of time spent looking inside ScaleView portals for users per task.

The presentation of multiple portals invoked comments from several participants. One user expressed her desire to be able to separate different portals and move them around independently. Our initial implementation allowed only for spread out and stacked layouts (the cube variation in Figure 3.6 was deactivated). This prevented users from spatially arranging portals as they saw fit. Another two users expressed their desire to have by default ScaleView portals spread out. In order to avoid clutter we initially presented multiple portals in a stacked layout. This was in hindsight inappropriate, since users invoking new portals want to interact with them right away. Finally, one user asked for separate minimaps for each one of the portals. In the initial implementation only one portal was active at a time, thus a single minimap was needed and scroll-bars were only attached to that active portal. Two other users found this notion of "active" portal somewhat confusing, as they had to give focus to a portal (by touching its interior) before interacting with it through scroll-bars or menus.

To address the issue of active portals, we now allow all portals to be active simultaneously. Scroll-bars and a minimap are attached to each portal when not in a stacking configuration (Figure 3.15). Moreover, by default, newly created portals are presented in a spread out configuration. Finally, to address the concern of rearranging the position of individual portals, we now allow multiple separate portals, not necessarily grouped.

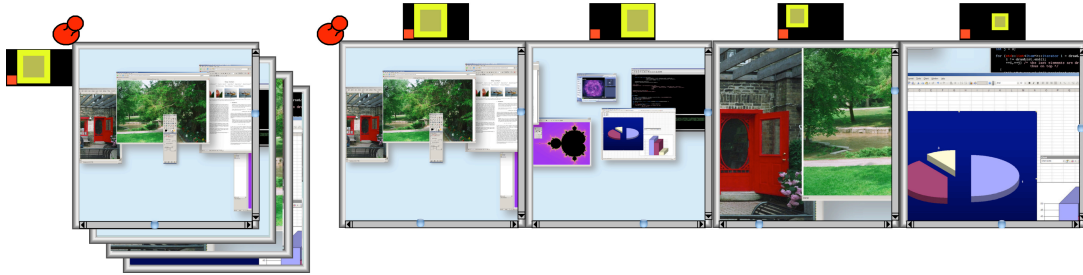


Figure 3.15: Multiple ScaleView portals. In the stacked layout (left) one minimap and one set of scroll-bars is present, that of the top portal. In the side-by-side layout each portal has its own minimap and scroll-bars.

The other major concern expressed was that of focusing portals. In terms of creating a new portal, one user asked for the option to have new portals created as "duplicates" of an existing portal. This is a reasonable feature not present in our study, in order to observe users interacting with the different ScaleView elements (the minimap, scroll-bars, zoom sliders etc). Another user asked for the option to use one portal as a means to focus another. So for example if a zoomed-out portal was active, the user could make an area selection inside it to define the focal area of a new portal.

When it came to altering an existing portal's focal area, a user proposed that while dragging on "empty" space inside the portal, the canvas depicted inside the portal should pan, thus altering the portal's focal area. Although such an approach is possible, it violates the notion that interaction within the portal is equivalent to interacting with the remote focal area, since actions specific to the "background" of the main canvas are no longer possible inside the portal (think for example the action of selecting multiple items

from the desktop in current windows interfaces). Another two users asked for the option to directly reposition the focal area of a portal at a remote location by walking. In our designs we actually provide a menu option for directly focusing the portal. This option was deactivated to elicit users comments on the alternative focusing mechanisms.

A major issue that surfaced in our usability study was the manner of defining the focal area of the ScaleView portals. Several users expressed their opinion on the matter and proposed different solutions. It became clear early on, that different users prefer different approaches. For example users that tended to concentrate their view inside the ScaleView portal, proposed ways to focus it from inside other portals. Others that made use of the entire display as a reference, suggested manners that often involved direct positioning. Thus no single solution seems to exist. We decided to keep the minimap and scroll-bars since they are intuitive, and provide another course and another detailed refocusing mechanism. Apart from "duplicating" new portals from older ones and the menu option for directly refocusing the tool, we added the following functionality:

First users may now create a new ScaleView portal from inside an existing one. The new portal is focused on the main canvas on the corresponding location depicted in the existing portal where the menu invocation took place. Thus portal focusing and creation are combined in a single action.

While the user is manipulating the remote focal area in the minimap, the representation of the focal area on the minimap is rendered semi-transparent. When the representation reaches the edges of the minimap, it is rendered outside the confines of the minimap, similar to how items are drawn when they are about to cross a border. This is an indication that users can continue dragging the representation outside the minimap. At the crossing point, the representation becomes the remote focal area center itself, which the user can now directly position on the desired location of the canvas.

Finally, two users asked for the option to synchronize two portals, such that when moving the focal area of one, the focal area of the other would follow the same transfor-

mations. Obviously this suggestion is a result of the comparison task given to the users and might not be of use in other interaction scenarios.

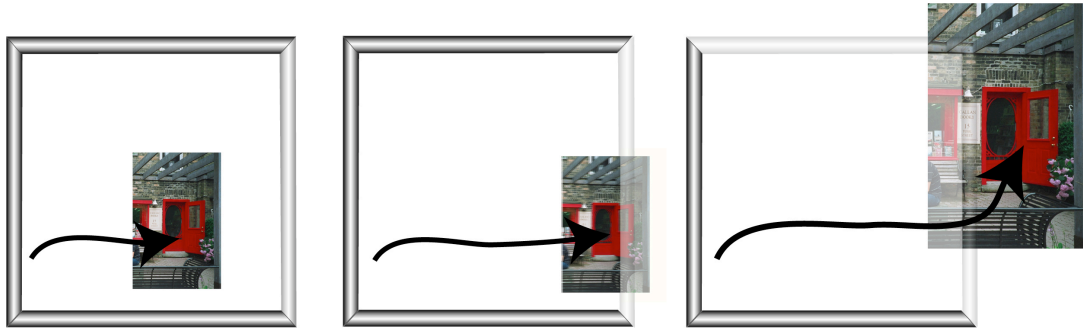


Figure 3.16: Redesigning ScaleView portal transitions. As an item approaches, the border becomes transparent to highlight the crossing functionality.

Since a number of our study participants did not recall using the "crossing" mechanism between portals and canvas, our design did not highlight this feature adequately. Originally, as a means to indicate the "crossing" option, items that were at the edge of a portal, were rendered semi-transparent outside the portal (Figure 3.3). This indication might have been too subtle, given that part of the moving items had to actually cross the borders of a portal. To enhance the crossing metaphor we added the following behavior: when a moving window comes close to a border, the border becomes progressively transparent (the more so as the item comes closer), until it disappears when the item starts crossing the border (Figure 3.16). We felt this visual indication is subtle enough to not interfere with the user's interaction, while providing a visual indication of the crossing property. Other designs considered were: "pulling" an item that is close to a border onto the border, pulling the border close to the moving item, or providing a visual connection between the item and the border (for example by coloring both in the same color). These designs we felt interfered too much with the user's interaction and were abandoned.

3.6 Chapter summary and closing remarks

In this chapter we presented designs aimed at improving layout management and context switching on wall sized displays. Early on in our design phase we developed the canvas portal framework that incorporates layout management techniques based on the idea of using alternative views as shortcuts to different canvas locations. The framework, apart from allowing users to view and interact with remote locations through the portals, supports passing of content between these views and the main canvas. These properties reduce the need for walking around the display to view or interact with content spread out over the canvas, to rearrange and manage remote data, or to acquire different perspectives of depicted data. Based on this framework, we introduce 3 new techniques.

ScaleView portals are general zoom based portals that provide arbitrary views of the main canvas. Their position, focal areas and zoom level can all be altered by the user. Multiple instances of ScaleView portals may be present at a given time on the display and transition of content between them is supported. ScaleView portals support context switching and comparison tasks between different locations on the canvas and efficient layout management.

Window portals present users with a spatial and temporal way to switch between different items on the canvas and act more as context switching mechanisms and less as layout management or comparison mediators.

Finally, Division Bands are vertical and horizontal slices of the main canvas defined by the user. Contrary to ScaleView portals that are general purpose tools for supporting layout management and switching between contexts, Division Bands act more as space creation mechanisms and as aids for fast glances or access to remote areas.

Although all of the techniques are based on the same framework, their implementation details differentiate their application on large displays. Nevertheless, they are all designed in a manner that does not make any assumptions about the underlying structure of the depicted content, other than that some elements can be interacted upon and possibly

move across the main canvas.

The use of ScaleView portals, the most general canvas portal designed, was further investigated. Apart from general design improvements made based on user comments, several other interesting issues regarding wall display interaction were observed. First of all, we found that for the tasks performed, use of complicated techniques, such as the ScaleView portal, was often preferred to simpler interactions when users felt that a simpler technique did not adequately support their task.

Some users started tasks immediately using ScaleView portals, a complex technique, indicating that they found compelling the ability to have everything concentrated close to their position and not have to move around frequently. Nevertheless, several participants used the large canvas as a peripheral reference to their tasks, by periodically glancing directly at remote canvas locations. This observation is consistent with our decision to opt for designs that minimize physical movement and in particular allow interaction close to the user's reach. Moreover, it highlights one of the benefits of using a wall sized display: users visually refer to the large, spread out content on the wall display, even if their interaction is limited close to their location.

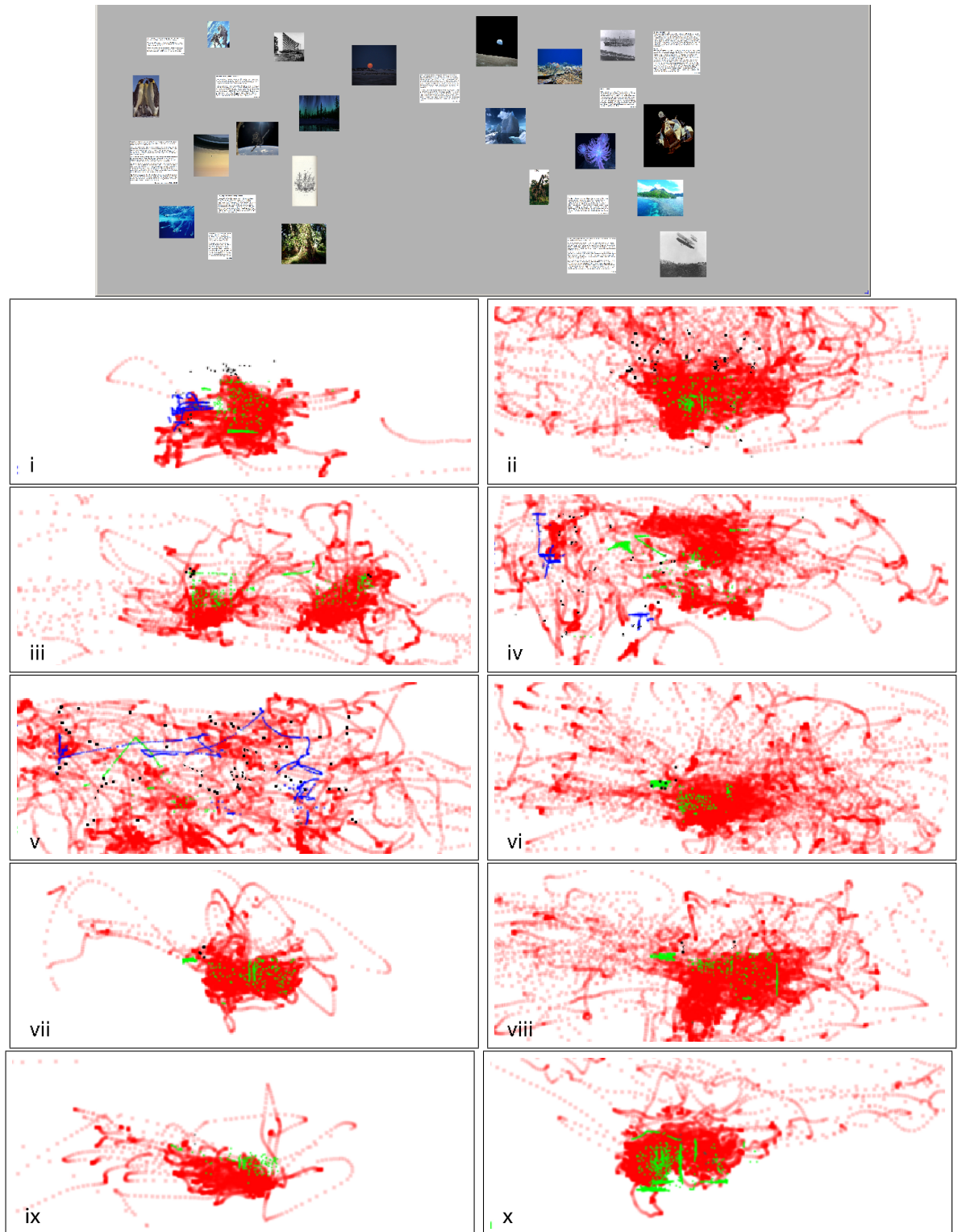


Figure 3.17: ScaleView portals usability evaluation: Layout task setup and interaction footprint for each user labeled (i - x). **Red** squares indicate the users visual focus. **Blue** marks denote interactions with regular magnification lens(es), **Green** ones with ScaleView portals and **Black** marks are other interactions on the canvas.

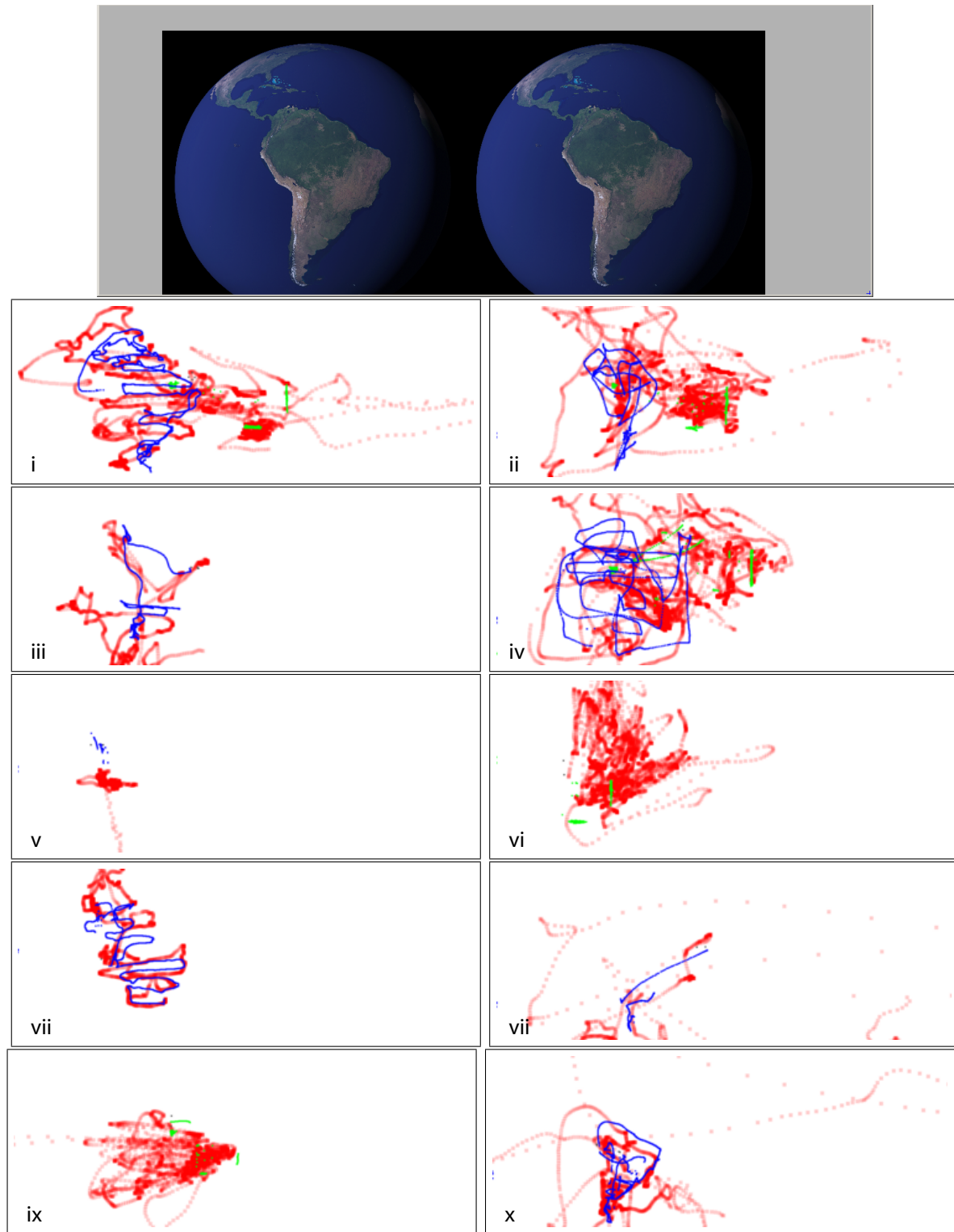


Figure 3.18: ScaleView portals usability evaluation: Search task setup and interaction footprint for each user labeled (i - x). Red squares indicate the users visual focus. Blue marks denote interactions with regular magnification lens(es) and Green ones with ScaleView portals.

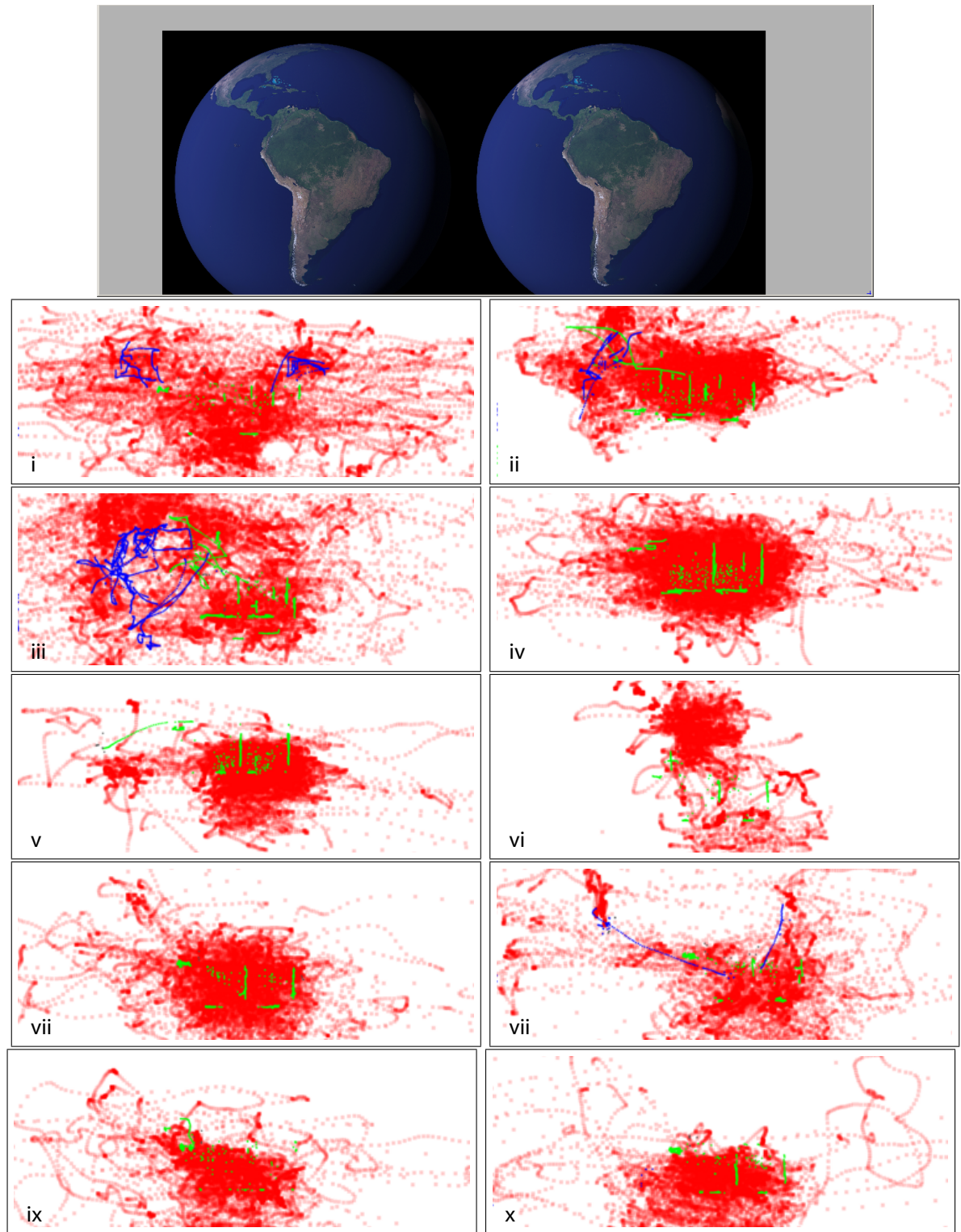


Figure 3.19: ScaleView portals usability evaluation: Comparison task setup and interaction footprint for each user labeled (i - x). **Red** squares indicate the users visual focus. **Blue** marks denote interactions with regular magnification lens(es) and **Green** ones with ScaleView portals.

Chapter 4

Facilitating Remote Reaching

Unlike interaction on a desktop or even a small white-board sized display where almost all displayed items are within arms reach of the user, data on wall displays often reside farther away, or in unreachable locations. From a visualization perspective, it can be difficult to view all parts of the display at equal clarity, since some of the display will appear in the user's peripheral vision. As a result, if existing user interfaces are mapped onto displays of this scale for up-close interaction, they would at the very least require the user to walk around the display to accomplish even simple tasks, or they may be unusable altogether when, for example, the user cannot reach the top of the display to operate an application's menu bar. Admittedly, one could operate a wall display from afar, using a mouse and a keyboard, but such an approach does not fully leverage the potential benefits that can accrue with up-close direct interaction mentioned in Chapter 2.

In attempting to facilitate remote reaching in wall displays, we need to first bridge the distance between the location of the user and a remote location, and secondly, ensure the nature and effect of the bridged interaction are obvious to the user. This section presents the design and evaluation of such a new remote reaching technique, the vacuum.

Up to date, reaching at remote locations can be achieved in several ways discussed in Section 2.5.2. We decided to further concentrate on techniques that copy content close

to the user. These may result in visual breaks between the actual and copied content, yet they allow for comfortable, proximal interaction and provide visual feedback on the performed actions close to the user, thus minimizing user physical effort. As pointed out by Reetz et al. [2006], visual feedback close to the user might be better than having it in the environment, even though users have to work with a duplicate representation of the target instead of the real one. Users in work by Forlines et al. [2006b] commented that small targets were not visible in distant locations on a wall display, also indicating that visual feedback close to the user might be desirable.

We considered several techniques for copying remote content in the form of proxies close to the user, where users may interact with the proximal copied content in lieu of the corresponding remote content. Before discussing our preliminary designs in Section 4.2, we will describe in detail drag-and-pop/pick [Baudisch et al., 2003a], an existing technique that follows this interaction paradigm and is closely related to our designs.

4.1 Drag-and-pop/pick

Drag-and-pop/pick [Baudisch et al., 2003a] allows users to quickly pick remote icons, or drag icons onto others located at far distances, by bringing proxies of the remote icons closer to the user (Figure 4.1). We would like to distinguish here between the drag-and-*pop* and drag-and-*pick* version of the technique. In the drag-and-*pop* version, as the user drags an icon across the display, icon proxies corresponding to applications compatible with the dragged icon are copied close to the user. Drag-and-*pick* is the more general version of the technique. Here as the user drags on the display, all types of remote application icons are copied close to the user.

The selection of icons to copy is done based on the user’s direction of movement. As the user’s pen moves on the display beyond a fixed distance, potential icon targets are identified within a $\pm 30^\circ$ sector centered about the pen’s movement direction. Proxies of

those targets are created close to the current cursor position and may be used in lieu of the original target icons to complete the dragging operation, significantly reducing the cursor’s travel distance. The proxy layout resembles that of the original icons (assumed to occupy a virtual grid), with empty space between them having been removed (empty rows and columns). To maintain visual persistence, rubber-band lines are drawn connecting the actual target icons to their proxies. A user study [Baudisch et al., 2003a] showed that drag-and-*pop* was up to three times faster than regular direct dragging across a large-scale display. When discussing benefits and issues of this work in the thesis we generally refer to drag-and-*pick*, since this is the most general form of this remote reaching approach.

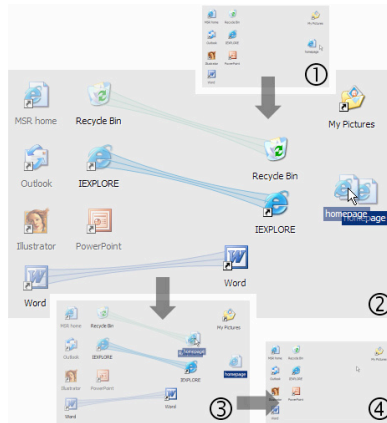


Figure 4.1: Drag-and-*pop* sequence. (1) In regular desktop with icons, the user selects (2) a webpage icon and moves it towards a remote cluster of application icons. The user starts dragging a file icon towards a remote cluster of application icons. Compatible application icons are copied close to the user's location, in a condensed layout. (3) The user drops the moving file icon on the desired application icon (recycle bin), concluding the dragging operation (4).

Drag-and-pick [Baudisch et al., 2003a] is a lightweight technique that seamlessly enhances regular dragging and picking, without the user having to learn a completely new interaction technique or set/adjust multiple parameters. Furthermore, it is a model-less and transient technique in that no tool has to be invoked or dismissed since the

entire operation is performed in a single pen-down, move, pen-up gesture. While these properties make drag-and-pick almost ideal for single target operations, there are some shortcomings with this approach, given the design goals identified in Section 2.6.

Clearly drag-and-pop/pick satisfies our design goal of minimizing user effort. Nevertheless, it relies heavily on the assumptions that the manipulated items are application icons of the same size and the only form of interaction taking place is icon selection. It thus violates our application independence design goal, as it may not accommodate more complicated interaction than single target selection, nor interaction elements of different sizes and shapes. As far as our third goal is concerned, predictability is supported in the sense that the behavior of the tool is easy to understand. Nevertheless the actual effect of the tool is not always clear. These issues are discussed in detail below, identifying specific shortcomings of the technique.

- Target identification: The sector of influence is determined by the cursor's initial movement vector. If the user starts off in an inexact direction and the wrong targets are brought closer, the only option is to abort and restart the operation. Baudisch et al. [2003a] discuss adjustments to the algorithm to favor targets in certain directions, but fundamentally the technique is limited in not allowing users to alter the sector of influence during the interaction. Thus the technique is not flexible and consistent.
- Number and size of proxies: Proxies retain the size of the original targets, and to prevent clutter or overlapping proxies around the cursor, the number of proxies is capped at a fixed value (5-10 typically). While this approach is reasonable for the iconic targets studied in the work of Baudisch et al. [2003a], it does not scale to situations of larger numbers of distracter proxies or larger sized targets (e.g., tool palettes), since the space around the cursor would quickly fill up.

- Layout of proxies: The final position of proxies around the cursor is dependent on the cursor’s initial movement vector and on the number of proxies. The algorithm also removes much of the void space around the proxies in the final layout. As such, the proxy layout does not exactly resemble the original target layout, and it can be difficult for the user to predict where exactly the proxies will end up. In particular, if there are many more proxies than the 5-10 item limit imposed in the original design [Baudisch et al., 2003a], due to space constraints it may be necessary to place some proxies on the other side of the cursor’s movement direction. Cursor backtracking would thus be required to select those proxies.
- Multiple operations: All interaction is completed within a single pen-down, drag, pen-up operation. While this contributes significantly to the transient, lightweight, operation of the techniques, it is limiting in that operations that require several pen-up/down events for a single task cannot be supported. For example, it would be difficult to interact with a slider widget at a remote corner of the display, or make multiple selections from a tool palette.

4.2 Preliminary remote reaching designs

The designs discussed here explore different ways of selecting remote content to be copied close to the user, and how this content should be presented. Users may interact with these proxies in lieu of the original remote content.

As discussed in our goals (Section 2.6), apart from minimizing physical effort in wall displays, it is important for the user to be able to keep track of state changes in a predictable manner. As such, when our techniques are dismissed (in our prototype implementation, they are invoked and dismissed through the system wide marking menu), all proxies of copied items are faded away and the actual items reactivated in the positions they held prior to the technique’s use. We do, however, allow the user to select copied

proxies to keep them around as shortcuts after the techniques are dismissed. These shortcuts are then grouped for future use. In general, the space reorganization and transformation caused by the techniques is valid only for the duration of their current invocation, unless the user performs an explicit action with the copied proxie(s).

The fact that we allow copied content to be "selected" and kept close to the user violates our application independence principle (since we assume that clicking on the proxy is an action for "selecting" the corresponding item). This is a decision we made for prototyping and evaluation purposes. If the techniques were implemented as part of an actual system, this touching action would resemble the action of giving "focus" to an interaction entity, as is the norm in current windowing systems. A more specific action would be required in order to "select" the proxy to be copied close to the user and the regular interaction vocabulary (such as clicking) would be reserved for regular interaction with the proxies. Such specific actions could for example be implemented as a system wide option, available through a system wide menu activated by a button. In general all our designs assume that the proxies are persistent on the display and the user can interact with them without any assumption as to the type of interaction taking place or as to the shape or nature of the represented content.

A stylus with a button was used as input for the prototypes. Apart from touch / lift events from the pen, a small hover range was also available. The designed techniques are invoked using a marking menu anywhere on the display (invoked through the pen button), and may be repositioned by clicking and dragging on a pin icon at their centre, which employs a pinning and "unpinning" metaphor as a consistent indicator that the widget can be repositioned. These designs are similar to drag-and-*pick* [Baudisch et al., 2003a] in the sense that they are general selection techniques of remote content. But dragging of items to remote locations (as in drag-and-*pop*) could easily be performed as well.

4.2.1 The vacuum tool

The *vacuum tool* acts as a vacuum cleaner, bringing towards its center items that reside inside an arc of influence, centred about the widget, and spanning the display. In designing the vacuum tool, we made two important design choices. First, we opted to allow the user to interactively control the angle of the arc, through its start and end extents. Thus the remote copied area is controlled by the user. Second, if the user moves the tool around, the effect of the tool is dynamically updated, with items moving into the arc being brought toward the tool, and items leaving the arc removed from the tool's control.

Widget visual design: The vacuum tool is designed as a circular knob, with three selectable parts (Figure 4.2) a pin icon that enables repositioning of the widget, and the start wedge and angle control that determine the beginning and end of the arc respectively. The arc of influence on the display, that is the extension of the angle controlled by the user, is only visible while the vacuumed proxies get updated to minimize visual clutter. We felt the visible extension of the angle/arc of influence enhances the predictability of the technique by ensuring the effect of the tool is visible.

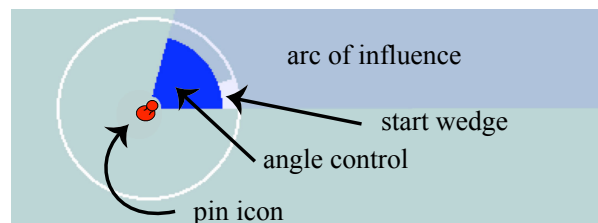


Figure 4.2: Vacuum widget. The user can click and drag on: the pin icon to reposition the widget, the start wedge to change the start extent of the arc, and the angle control to change the arc's angle of influence.

Interaction: When first invoked, the start wedge, distinguished by a white extension, is on the vertical axis and the angle of influence is at 0° (Figure 4.3.a). When the user either drags or clicks on any part of the circular widget, the angle of the arc opens

clockwise to create an angle between the start wedge, the center of the tool and the cursor's position (Figure 4.3.b,c). After the initial invocation, the user can alter the start wedge position by selecting and moving its white extension (Figure 4.3.d,e,f). The overall angle of the arc can still be changed by clicking on any point inside the circular widget. When the user performs more than a full 360° arc, the arc resets to 0° degrees, since a larger arc has no meaning (Figure 4.4). Finally the position of the vacuum on the canvas can be altered by clicking on the red pin icon at the center and dragging the tool.

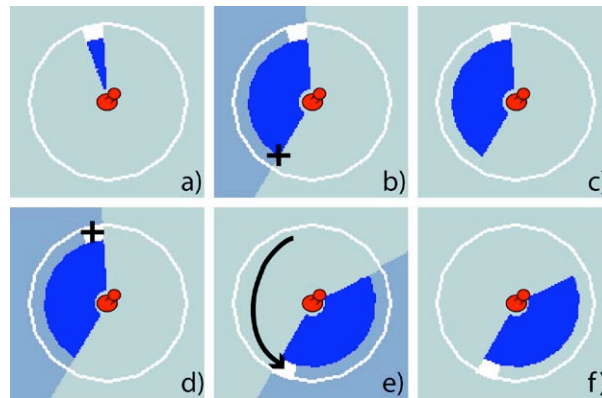


Figure 4.3: Vacuum interaction. (Top) First invocation: a) After the menu selection the vacuum tool is invoked, with the angle of influence at 0° in a vertical position. b) When the user touches/clicks inside the widget, the angle of influence is opened clockwise up to the cursor position. The extension of the angle is visible, until the user lifts her pen c). (Bottom) Angle manipulation: d) When the user selects the start wedge, the extension of the angle of influence is displayed. e) The user can drag the wedge to alter the direction of the angle, and as she lifts her pen f) interaction with the tool stops and the effect of the arc of influence is no longer visible.

Effect visual design: Apart from displaying and controlling the tool itself, we also need to represent its effect on the virtual canvas, as items entering and leaving the arc of influence get pulled toward the tool centre and vice versa. We initially experimented with simply displacing the vacuumed items entirely from their original locations on the canvas

towards the tool’s centre. However, we found this to be disorienting and instead took the approach of copying a proxy of the vacuumed items towards the tool. As an item enters the vacuum’s arc of influence we animate the movement of its proxy towards the tool’s centre, while a ghost image of the item remains at its original position (Figures 4.5, 4.6). When an item leaves the vacuum’s influence, the reverse animation takes place. This approach allows the user to maintain a sense of overall context of where everything is on the display, following our predictability design goal.

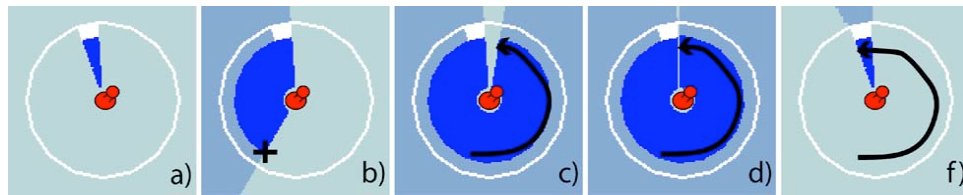


Figure 4.4: Vacuum interaction, resetting the angle: a) After the menu selection the vacuum tool is invoked, with the angle of influence at 0° in a vertical position. b) When the user touches/clicks inside the widget, the angle of influence is opened clockwise up to the cursor position. c) The user drags the angle of influence until it reaches 360° d). When the 360° mark is crossed e), the angle resets to 0° .

An important aspect of the design of the developed designs is in the representation of the copied, or in this case, vacuumed items when they are close to the tool’s centre. The vacuum tool can collect many items from a large area of the display, but must depict copies of the items within the much smaller region near the tool’s centre. For even a moderate number of items, it is impossible to display these copies at full size near the tool’s centre without significant overlap in layout. We thus explored two different ways to layout the vacuumed proxies around the widget.

Stacking vacuum. In the first layout (Figure 4.5), proxies of the vacuumed items are stacked in full size, one on top of another with a slight offset, near the widget’s centre. Items are stacked in the order in which they are vacuumed (i.e., beginning with items closest to the starting extent of the tool’s arc of influence). This layout has the

advantage of preserving the original size of the items, enabling easy perception of the proxies, similarly to drag-and-pick [Baudisch et al., 2003a]. However, manipulation of the proxies is hindered by the overlapping objects. Items have to be first selected and brought to the front, before interaction can take place. Note that we deliberately offset the items slightly in the stack in order to allow users to select items deep in the stack. Furthermore, the stack of proxies quickly fills up as numerous or large items are vacuumed.

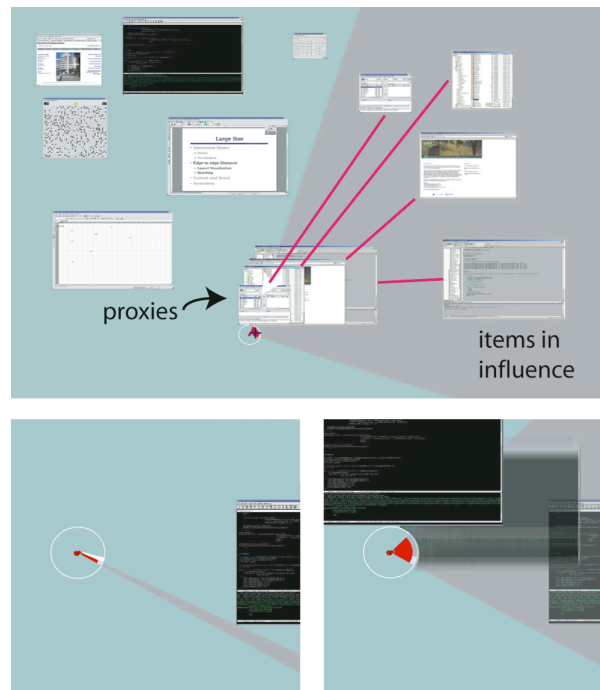


Figure 4.5: Top: Full screen view of stacking vacuum. Items inside the influence of the vacuum are presented as proxies stacked on top of the vacuum. Red lines show connections between items and their proxies and (for illustration only). Bottom: Close-up of the stacking vacuum. As the arc of influence is increased to include the window item (left image), a copy of the item is brought to the tool's centre (right). Motion blur in right image illustrates the animated transition that occurs during the item's movement.

Scaling vacuum. In the second layout (Figure 4.6), copies of the vacuumed items are scaled down in size and displayed around the tool's centre, preserving the relative spatial relationships of items to one another. A semi-transparent line connects the centers of the

copies to the centre of the original items. This line virtually passes through the center of the tool as well, so all three points are aligned. The advantage of the scaling over the stacking vacuum is that no additional overlap is introduced between vacuumed items, apart from overlap present in the original items' layout. The disadvantage is that the copies are significantly smaller, and thus harder to perceive and interact with in detail without first clicking on them to expand their size.

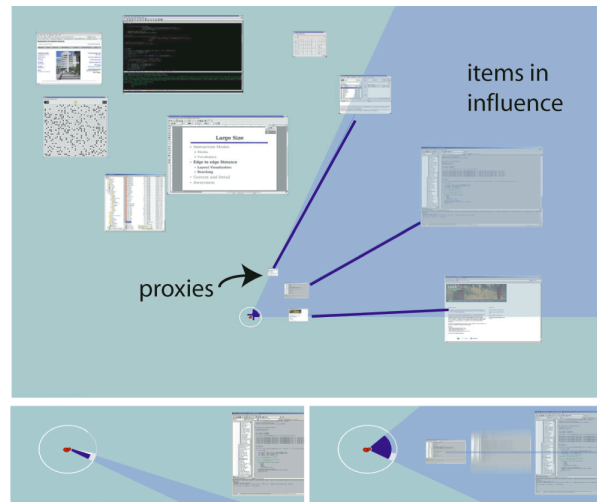


Figure 4.6: Top: Full screen view of the scaling vacuum. Items inside the influence of the vacuum are presented as proxies scaled down. Blue lines show connections between items and their proxies. Bottom: Close-up of the scaling vacuum. As the arc is increased to include the window item (left image), a copy of the item is scaled down and brought to the tool's centre (right image). Motion blur in right image illustrates the animated transition that occurs during the item's movement.

The two visual representations of the proxies provide different affordances. The stacking vacuum works well for sparse layouts, but breaks down when the number and size of the vacuumed items is large. While crowding is less severe in the scaling vacuum, it too can get difficult to navigate as the space around the tool's centre gets populated with numerous copies of items. Two refinements could alleviate these problems: A semantic sensitive vacuum that only vacuums items with certain characteristics (e.g., user interface

elements) could reduce the object space significantly, nevertheless it would violate our application independence principle. Or a "spiral" vacuum with a virtual knob could limit the outward extents of the arc of influence. As the knob is turned, the extents of the arc are changed, with the influence of the vacuum spiraling out from the tool's centre.

The idea of an arc of influence is also present in drag-and-pick [Baudisch et al., 2003a]. However, in that work the angle of influence is fixed and the technique's behavior differs from the vacuum tool in that items of interest are brought to the user as she initiates a dragging action towards a remote item. In contrast, the menu invoked vacuum tool supports persistent vacuuming.

4.2.2 The edge reaching tool

It is very likely that many existing applications will be run on large scale displays, with little change to their interface design. As noted in Pedersen et al. [1993], when applications designed for desktop scale displays are run full screen on even moderately sized large displays, such as whiteboards, the typical arrangement of user interface elements on the borders of the application window can pose major usability problems. On wall displays, the user may simply not be able to reach the top of the display to access the menu and tool bars, may not want to walk all the way to either vertical edge to access tool palettes typically found at those locations, and may find it rather inconvenient to have to bend down to reach the icons at the bottom of the display.

Given the special and extensive use of edge regions by many applications, we have designed the *edge reaching tool*, suited to accessing statically predefined areas of the display, such as the edges, as shortcuts copied close to the user.

Widget visual design: In the edge reaching tool the display gets divided into areas. For our 5 x 2 m display, we have found a 3 x 2 grid division to be appropriate in terms of the amount of content residing on each of the 6 resulting grid cells. Although not

interactive, the actual number and configuration of cells may change, for example by defining cells only at the edges of the display to create a strictly edge reaching tool. The visual representation of the tool acts as an interactive thumbnail of the entire display, with yellow bands surround the borders of both the interactive thumbnail grid and the corresponding display grid. When the user clicks on parts of the thumbnail grid, scaled down proxies of all content within the corresponding section of the display slide down to surround the tool (Figures 4.7,4.8). We could restrict selections of proxies to items within only the edge of the regions (i.e., yellow bands) mentioned before, making it a strictly edge-reaching tool. If semantic selections were enabled, only appropriate items (e.g., user interface elements) could be brought to the tool. Yet, such a semantic filtering assumes application knowledge and violates our application independence goal.



Figure 4.7: Using the edge reaching tool. (a) The widget is present at the bottom left of the image. At the top right is the top-right corner of the display. (b) From the 6 selector areas of the grid on the widget, the top right one is selected. This thumbnail area corresponds to the top right area of the display. (c) The corresponding display area and proxies of all the items in the area are scaled down, surrounding the tool.

Interaction and effect visual design: When the tool is invoked only the thumbnail grid of the widget is visible. As the user selects a cell of the thumbnail grid, proxies from the corresponding main display cell and its borders are copied around the tool (Figure 4.7). The original content of the display in the selected cell becomes semi-transparent, similarly to the vacuum, to indicate that its proxies are visible. If the user selects a cell of the thumbnail grid that is already "pulled" around the tool, the grid

is dismissed and proxies are animated to their original position. In the edge reaching tool proxies are scaled down to a predefined size and their relative position and size is preserved, similarly to the scaling vacuum. But here the user can adjust their scale by pulling at the yellow bands that represent the edges of the display (Figure 4.7) on the thumbnail grid. Finally, as the user hovers over different cells of the thumbnail grid, they and their counterparts on the display get highlighted (Figure 4.8.b).

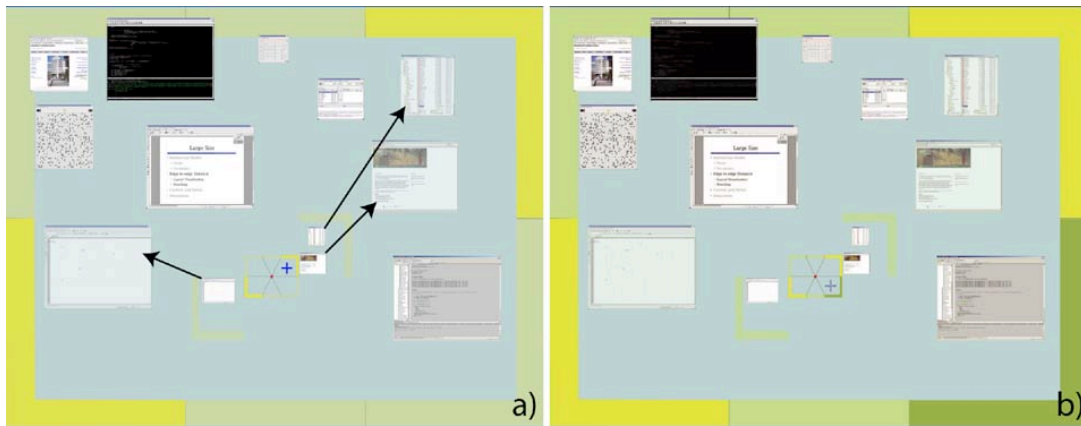


Figure 4.8: a) Full screen view of the edge reaching tool. The tool (yellow widget at center of image) is divided in 6 regions representing corresponding regions on the display (yellow regions at edges of the display). When a region is selected in the edge reaching tool, proxies of the items in the corresponding remote region on the display are "pulled" close to the user, surrounding the tool. Here two regions are selected and are highlighted both on the tool and on the display with more prominent border colors. Black lines depict the connection between proxies and remote objects and are for illustration only. b) Use of hover in the edge reaching tool: when the user hovers over the bottom right grid cell of the tool, it and the corresponding display cell gets highlighted in green.

The edge reaching tool preserves the layout of the original items in their copied proxies, allowing easy access to items in familiar display layouts. Moving the tool does not affect the copied content, as the interactive thumbnail grid cells always correspond to specific remote display cells. Contrary, the selected remote area of the vacuum is defined

by the arc of influence, which changes as the tool changes position on the display, even if the angle value remains unchanged (Figure 4.9).

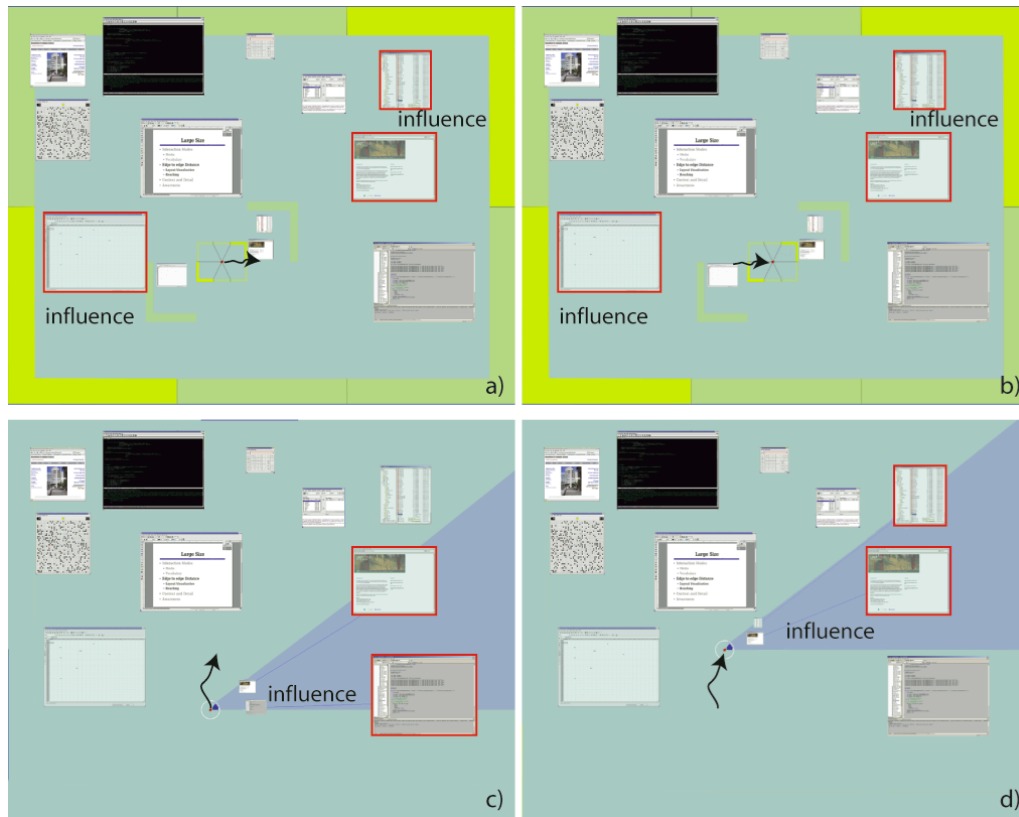


Figure 4.9: (Top) Movement of the edge reaching tool, influence remains the same. The tool moves from left (a) to right (b) on the display. As the selected thumbnail cells have not changed, the copied proxies are the same. (Bottom) Movement of the scaling vacuum, influence moves with the tool. As the tool moves from bottom (c) to top (d), the arc of influence changes position and thus different proxies are copied close to the user. Black arrows indicate the movement of the tool. Red borders highlight items that are copied close to the tool. Arrows and borders are for illustration.

4.2.3 The scroll ray

The design of the *scroll ray* ensures that the copied proxies of remote items preserve their scale, as in the stacking vacuum, while at the same time accommodating denser layouts.

Widget and effect visual design: The scroll ray (Figure 4.10) is essentially a line starting at the location of the user and spanning to the edges of the display. The tool is represented by this thin long line, and a thick one close to the user to indicate the interactive area of the widget. All remote items intersected by the ray are potential candidates for copying close to the user, and are highlighted in colored borders. One of these potential candidates is copied close to the user’s location.

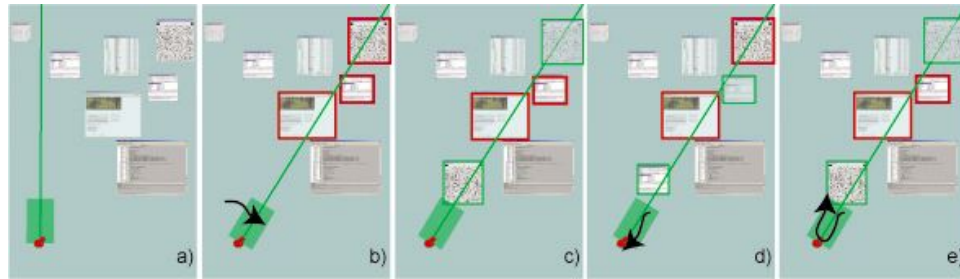


Figure 4.10: Scroll ray widget. a) When invoked the tool is placed in a vertical position. b) The user may alter the angle of the ray by dragging outside the length of the interactive area of the widget. All intersected items are proxy candidates and highlighted in colored borders. c) By default the furthest candidate is copied close to the user. d) Candidate items closer to the user can be reached by scrolling towards the center of the tool, while e) candidates farther from the user by scrolling towards the edges of the display.

Interaction: When first invoked the scroll ray is in a vertical position (Figure 4.10.a). The direction of the scroll ray may be changed by clicking on the interactive area and dragging outside its length (Figure 4.10.a,b,c). The ray may also be repositioned by dragging the pin icon, an action that dynamically alters the potential copied items.

The default copied proxy of all the intersected candidate items is the farthest one. The user may cycle through the candidates by scrolling or scrubbing across the length of the interactive area of the ray (Figure 4.10.d,e). Scrolling towards the edge of the display cycles towards more and more remote items, whereas scrolling towards the user collects items at closer proximity. To ensure that interaction with the scroll ray takes place at

a comfortable distance, users may clutch by lifting and repositioning the input device, preventing awkward stretching movements when scrolling for far away objects.

To allow for a single drag interaction (no need to lift the pen and clutch), we thought of two interaction alternatives. First, to limit the interactive area of the scroll ray to a fixed size, thus depending on the number of candidate items to be copied close to the user, they each correspond to a larger or smaller percent of the scrolling area. This fixed mapping of specific areas on the ray to specific candidate items ensures that the scrolling area is never beyond reach. On in some cases scrolling may be hard to control, for example when there is a large number of candidate items, the percentage on the scrolling area that corresponds to each one is very small. Another approach would be to always dedicate a fixed area of scrolling per item. Although this mapping would allow users to use muscle memory for selecting the closest, second closest etc item, it may result in remote objects that are hard to reach, if for example a large number of items is between them and the user. We thus felt that the clutching solution was the best.

The scroll ray preserves the scale of the copied proxies, by copying a single proxy at a time close to the user. Yet, because the user needs scroll in order to cycle through candidate items, a considerable amount of both visual and motor space search is required, whereas the other designs are mostly based on visual search of proxies. A technique similar to the scroll ray, the depth ray, has recently appeared for 3D selection tasks [Grossman and Balakrishnan, 2006].

All the described reaching techniques are similar in that they create shortcuts of items for the user to interact with in lieu of the remote items. The edge reaching tool and scaling vacuum preserve the layout of regions of the display brought close to the tool, allowing particularly easy access to items in familiar display layouts. Although both appear to be applicable in similar situations, the vacuum is more flexible in allowing fine-tuning of the area of interest. On the other hand, although the edge reaching tool statically predefines

areas of interest, multiple discontinuous areas can be active simultaneously (Figure 4.8), a feature the vacuum supports only if multiple instances of the tool are created.

In the stacking vacuum the relative position of remote items is not preserved, just their temporal relation (the order in which they entered or exited the arc of influence) and proxies are presented at full scale. Similarly, the scroll ray only copies a single full scale item close to the user and candidate items are cycled through based on their distance from the center of the tool, starting from the farthest. Both the stacking vacuum and the scroll ray preserve the actual size of the copied objects, but due to space restrictions, the stacking vacuum can only accommodate a fairly small number of items. The scrolling ray could be used for a large number of candidate items, but as the number of items increases, scrolling through all the possible selections might prove hard to control or tedious. Figure 4.11 summarizes the design characteristics of the discussed techniques.

	Stacking vacuum	Scaling vacuum	Edge reaching	Scroll ray
Remote areas	single	single	multiple	single
Remote areas defined	by user	by user	predefined	by user
Copied items	all in area	all in area	all in areas	one
Copied items Scale	full scale	scaled down	adjustable scaled down	full scale
Relative Layout	not preserved	preserved	preserved	-
Widget Movement	updates selection	updates selection	unaffected selection	updates selection
Layout Density	medium	large	large	medium

Figure 4.11: Design characteristics of exploration remote reaching designs.

4.3 First user feedback session on designs

Two male users, aged 25-30, took part in an informal user feedback session. They were observed as they performed a few single target remote reaching tasks and later they expressed their opinions on the different techniques.

This session showed us that the scroll ray was very hard to control. Users would overshoot the target when scrolling and had to constantly make corrections to retrieve the desired proxy. Moreover, when many items were intersected by the scroll ray, resulting in an increased number of potential candidates to be cycled through, selection times were very large compared to other techniques. In dense layouts, slight changes in the direction of the scroll ray could also result in very different candidate items: given the small thickness of the ray, items would be crossed and uncrossed by the ray frequently and their relative ordering over the ray changed. Overall, the scroll ray incorporated a large scrolling search component, which combined with the visual search of the candidate proxies, rendered the technique hard to use. The scroll ray thus failed to conform to our predictable and consistent design goal.

We also observed that when using the stacking vacuum users would often spend a large amount of time looking through the copied proxies. In particular, it was difficult for users to determine exactly where the proxy of a particular item was in the stack, and to identify it they would open and close the arc of influence to see the animated transition of the desired item. Thus although the behavior of the stacking vacuum was consistent across multiple invocations, it was not predictable. Given the above limitations, we removed the scroll ray and stacking vacuum from further exploration.

Both users commented that for proximal targets remote reaching techniques are not needed and may even delay and complicate the interaction. This led us to refine all our widgets by defining a within arm's reach buffer zone around the widgets, such that only items outside that zone are under their influence. This allows nearby items to be manipulated without proxies, and reduces the total number of proxies brought to the user. A similar buffer zone can be found in drag-and-pick [Baudisch et al., 2003a].

Finally, both users found the behavior of the angle of influence of the vacuum variations counterintuitive. Users occasionally crossed the 360° angle accidentally while trying to achieve large angles, and the angle would become 0° (Figure 4.4), resulting in all copied

proxies to animate back to their original location. We thus altered the behavior of the vacuum so that when the angle reaches 360° it locks to that value, until the user manipulates the angle wedge again to open it, or clicks on another area inside the tool.

4.4 Second user feedback session on designs

Although the previous feedback session pinpointed usability issues in some designs, it did not clearly identify the benefits of using either design or identify the design characteristics that seem to work best. The goal of this session was thus twofold: first, to examine how the techniques performed against simple walking and reaching to select an item, and an existing remote reaching technique, drag-and-pick [Baudisch et al., 2003a] (discussed in Section 4.1); and second, to elicit user comments on their design and behavior. Because we wanted a well defined task for users to better identify and discuss issues with the techniques, as well as measure their performance, we opted for a simple single target selection task. This type of selection task is similar to the ones used in the evaluation of diverse remote reaching techniques for large displays [Baudisch et al., 2003a, Collomb and Hascoët, 2004, Collomb et al., 2005b, Forlines et al., 2006b]. Four users (1 female), aged 25-35, took part in this session.

Task Users were shown multiple randomly numbered targets, with a goal target item uniquely numbered "1" and highlighted in green. After selecting a red starting target to initiate the trial, the widget of a given technique would appear and the user had to use it to select the copied proxy of the goal target. One of the tested techniques was simple selection. Each trial would end when the user successfully selected the goal target or its proxy, depending on the technique used. All targets were visible at the start of the trial, so users could plan their actions ahead of time, simulating a familiar display layout.

Design The techniques tested were: the edge reaching tool (Section 4.2.2), the scaling vacuum (Section 4.2.1), drag-and-pick [Baudisch et al., 2003a] and none (direct selection with no remote reaching mediator). Targets were of 2 *sizes* (Small: 7 cm and Large: 15 cm), placed at 2 *distances* from the start target location (Mid: 176 cm , Far: 264 cm), with Mid being approximately the display’s height, and Far $3/2$ its height.

In real interfaces, the path towards a target is rarely void of interfering items. As such, for any distance reaching technique to be successful, it must work well in the presence of items that distract the user from selecting a target. To examine how the different techniques faired in the presence of multiple items, we identified two distracters types: *area distracters*, items surrounding the goal target, and *path distracters*, items occupying space along the path from start to the goal target, to evaluate the effect of a cluttered layout in the line of sight of a target. We decided on a density based measure (rather than specific number of distracters) to account for cluttered vs. sparse layouts. The density of *area distracters* was determined based on the percent of distracters covering a square area surrounding the target, three times its size. For *path distracters* the density was determined by the percent of area covered in a rectangular area between the start and goal targets. For both types of distracters 3 densities were tested: no distracters, Few (40% of area filled with distracters), and Many (80%).

Primary investigation lead us to believe that the cost of the interaction will be likely dominated by *path distracters*. *Area distracters* are placed in a limited area around the target, thus the time to identify the proxy target between area distracters is most likely small compared to that of identifying the target between path distracters. We decided to remove the factor of *area distracter* density from consideration, but in all trials we used a fixed Mid density of *area distracters* around the target, to prevent users from going to the end of the cluster immediately to look for the target.

Start and goal targets were positioned in 6 relative compass *directions* (E, NE, NW, W, SW, SE) about the centre of the display, to counterbalance for any possible differences

in movement direction. We did not use the N and S directions as they would have involved starting uncomfortably from the bottom or top of the display for the Mid distance, and would have been impossible for the Far distance. Task samples are seen in Figure 4.12.

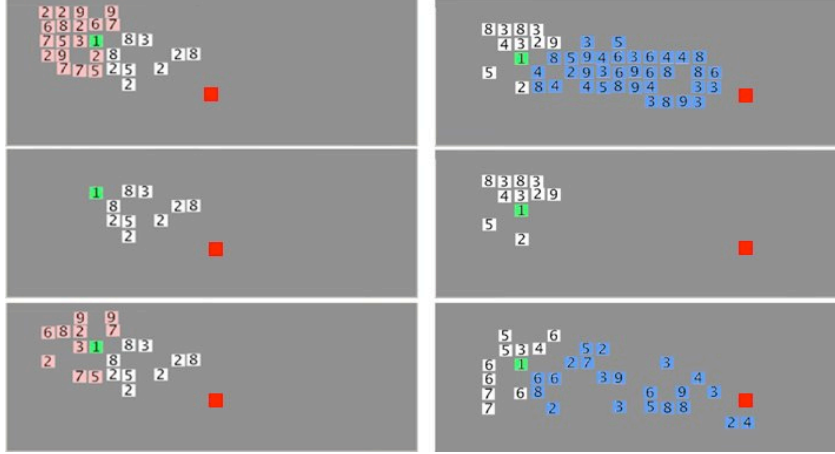


Figure 4.12: Sample tasks. **(Left)** Area distractors (highlighted in pink) are Many (top), None (middle) and Few (bottom), with distance Mid from start (red) to goal target (green). The density of path distractors is fixed to Few. **(Right)** Path distractors (blue) are Many (top), None (middle), Few (bottom). The density of area distractors is fixed to Few. The distance from start (red) to goal target (green) is Far. In the actual session the only highlighted targets were the goal (green) and starting target (red).

4.4.1 Results (Figure 4.13)

Of all trials 1.5% (52 trials) were considered outliers and were removed from further consideration. Figures 4.13(a) and 4.13(c) show *ErrorRates* and mean *SelectionTime* for each of the conditions.

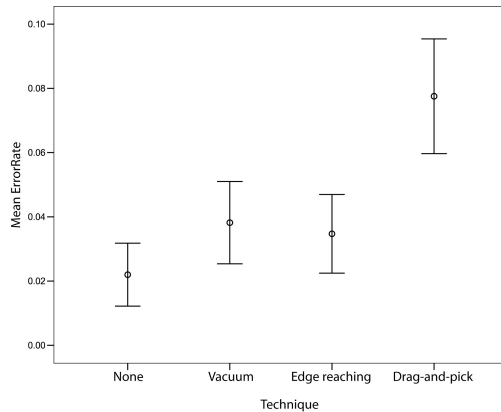
As the distributions of the time measurements were positively skewed, significance tests were performed on the logarithm of the *SelectionTime*. A fully within subjects ANOVA showed no significant effect of *technique* on *logSelectionTime*. There was a significant effect of *path distracter* density on *logSelectionTime* ($F_{2,6} = 86.9, p < .0001$)

and a significant *path distracter* x *technique* interaction ($F_{6,18} = 23.7$, $p < .0001$). A significant effect of *distance* on *logSelectionTime* ($F_{1,3} = 408.1$, $p < .0001$), and *distance* x *technique* interaction ($F_{3,9} = 40.1$, $p < .0001$) were also present. Finally, although there was a significant effect of *size* on *logSelectionTime* ($F_{1,3} = 120$, $p < .05$), there was no significant *size* x *technique* interaction.

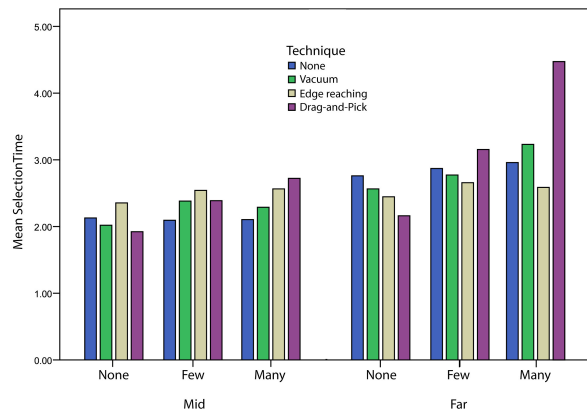
There was a borderline significant effect of *technique* on *ErrorRate* ($F_{3,9} = 3.78$, $p = .053$). It seems that with more participants drag-and-pick might prove significantly more error prone (Figures 4.13(a),4.13(b)).

	SelectionTime	ErrorRate
no mediator	2.16 sec	2.7 %
vacuum	2.41 sec	3.7 %
edge reaching	2.85 sec	3.4 %
drag & pick	2.91 sec	7.5 %

(a) Mean *SelectionTime* (sec) and *ErrorRate*.



(b) Mean *ErrorRate* 95% confidence interval for each one of the techniques.



(c) Mean *SelectionTime*(sec) per *technique* by *path distracters*(None,Few,Many) and *distance*(Mid,Far).

Figure 4.13: Performance measures: Mean *SelectionTime* (sec) and *ErrorRate* per technique. No statistical significant effects were present.

Discussion

As seen in Figure 4.13(c), the no mediator condition was affected less by the density of distracters, as participants saw the layout ahead of time and had the opportunity to identify the target, effectively ignoring the distracters. To the contrary, the existence of path distracters affected more the performance of drag-and-pick, as the number of distracter items between the start position and the location of the target determines the distance of the target proxy in the cluster of proxies brought close to the user. The more path distracters are present, the farther away the target proxy is in the cluster. The vacuum and edge reaching tool were affected by the presence of distracters, only as much as target re-identification inside the proxy cluster was concerned.

As there was no *size* \times *SelectionTime* interaction, it seems that although techniques were affected by target size, this was not more prominent in any of the techniques. Thus, to reduce the number of experimental conditions, in the next study (Section 4.6), we used only one target size of 15 cm, which we feel is representative of icons on a large display of this size.

Overall, this pilot session indicated that in terms of performance none of the techniques stood out by being statistically significant, with the exception of drag-and-pick being somewhat more error prone and the fact that drag-and-pick seems to be affected more by the presence of distracter objects.

User comments and design guidelines

Users in general preferred the proxies of the copied objects to be placed in a direction similar to that of the originals. One user commented on how it was easier to determine where to look for a proxy in the case of the vacuum than the edge reaching tool, due to its absolute nature. For example in some cases, although the goal item was to the left of the user, the corresponding area in the edge reaching tool and the proxy of the goal item was to her right (Figure 4.14). So when revisiting our design, we opted for a tool

that the final location of proxies have the same relevant orientation with respect to the user as the original objects (such as the scaling vacuum).



Figure 4.14: Location of proxies in edge reaching tool. (Left) The original highlighted object is at the user’s right, while its proxy is directly in front. (Right) After the tool has moved, the same object is on the user’s left, while its proxy is still directly in front.

Three out of 4 users ranked drag-and-pick fairly low on overall preference. They mentioned that finding the final location of an item’s proxy in the proxy cluster was hard, especially in the presence of many distracters, and that the elastic bands were not useful in disambiguating between proxies in dense layouts. On the other hand, the final location of the goal target proxy was easy to predict in both the edge reaching tool and the vacuum, indicating that here the location of proxies was predictable.

Moreover, two users commented that drag-and-pick did not provide adequate feedback as to the area of the display affected by the technique. Users spent time before each trial started to correctly select the region where the goal target was in. If they did not pay attention, they would miss the target and would not realize it until they searched through the proxy cluster. In the scaling vacuum, the arc of influence is briefly visible at the beginning of the interaction, and the final position of the proxies is consistent irrespective of the number of distracters and easy to determine. Thus users did not comment on lack of feedback in the vacuum, even if the effect of the tool is not always visible. Yet, as commented by one user, lack of visual feedback affected the edge reaching tool, when the goal target was in a position that is not clear in what part of the divided

cells it belongs to. Clearly the effect of the tool should be visible to as much as possible. These comments enforce the need for predictable and consistent techniques.

One user felt the control of the vacuum was inconsistent based on the direction of the target. Thus when the target was on the user's left, she had to move down to create an arc that was from 0° to 180° degrees, while if the target was on the left, she had to move up and right, to create an arc of 180° to 360° degrees (Figure 4.15). She mentioned this was never the case with drag-and-pick, since the dragging motion towards the target was always consistent. When redesigning our technique, we opted for consistent invocation, irrespective of the direction of the target items.

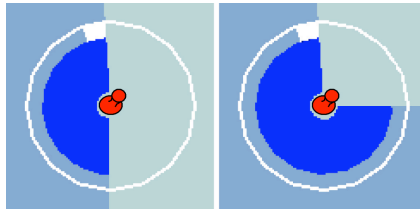


Figure 4.15: Angle inconsistencies in the scale vacuum. After the initial invocation the fastest way to reach items at the bottom left of the display (left) results into a different angle than that of reaching items at the bottom right (right).

In both the vacuum and edge reaching, the assumption is that the tools have been invoked through a pop-up menu, a cost not counted in this session, whereas the dragging motion of invoking drag-and-pick was accounted for and thus deteriorated its performance. Yet, while observing users in their tasks it became obvious that the dragging motion for invoking drag-and-pick feels very natural, as it is similar to the action performed for reaching a remote target without a mediator. Moreover it may be easily incorporated in any form of general interaction, whereas menu invocation disturbs the natural flow of other interaction tasks. Finally, given how users felt about inconsistencies in the invocation of the tool, dragging towards the direction of the target guarantees that the manner of invoking the technique is always consistent.

Nevertheless, the single drag interaction to invoke and dismiss drag-and-pick makes the technique somewhat inflexible. For example when an error in direction is made, the tool needs to be dismissed and re-invoked. Moreover, it does not support more complicated interactions than single target selection, such as adjusting the parameters of the tool, as is the case in the vacuum and edge reaching tool.

The edge reaching tool performance was consistent across conditions and deemed as faster by all users, with the vacuum generally ranking second. This might be due to the predefined divided areas on the display that alleviate the need to continuously adjust the tool parameters, as is the case with the vacuum. On the other hand, the vacuum provides the flexibility of selecting arbitrary sized regions, which may dramatically reduce the number of copied proxies in dense layouts. Thus a tradeoff exists between parameter flexibility and ease of parameter adjustment. In redesigning our techniques we considered the above tradeoff and looked for a reasonable solution that provides modification of the techniques parameters with minimal overhead.

4.5 Redesigning the Vacuum

Based on the above observations we identified the following desirable design guidelines for remote reaching techniques.

Minimal physical movement: This is one of our initial goals for large display designs and is essential for designs that attempt to bridge large distances. The technique should keep to a minimum the amount of traveling the user has to do in physical space to achieve any task.

Predictability and consistency: The technique should have a visual layout and behavior that is persistent over multiple invocations in similar situations, another of our initial design goals that was further reinforced by user comments.

Transparency: The technique should integrate seamlessly with the real tasks that users want to do, rather than present them with a whole new way of doing things. Our initial decision to utilize menus to invoke our remote reaching designs did not feel as natural as the idea of dragging towards the desired target [Baudisch et al., 2003a], an action that emulates what users do in order to reach the target without a mediator.

Low overhead: The technique should have a low, ideally zero, invocation and dismissal cost. For example, regular pointing and clicking on a target has little overhead since there is no tool to invoke or dismiss. In contrast, a modal selection mechanism such as menu invocation is not as transient. Given that the our techniques performed similarly in terms of time with the no mediator and drag-and-pick condition without taking into account the menu invocation cost, it is essential to minimize this cost.

Flexibility: It should be possible to smodify the technique’s parameters online with minimal overhead, and complex functionality should be supported when needed, but should not get in the way of basic functionality.

Trying to meet our design guidelines and address the limitations of drag-and-pick, we decided to redesign one of our initial exploration techniques. The edge reaching tool performed well and was liked by users, but has several limitations. Occasionally due to the representation of the display layout within the tool, a target might be at a different direction with respect to the user than its proxy (Figure 4.14). Furthermore, although fast for single target selection, when attempting to interact with several distinct remote locations on the canvas, the user would have to progressively select and dismiss the predefined divided areas. Finally, the edge reaching tool has no flexibility in adjusting parameters or redefining divided areas.

Based on the above observations, we decided to redesign the *scaling vacuum*, referred simply as the *vacuum* henceforth. The newly designed vacuum [Bezerianos and Balakr-

ishnan, 2005a] enables quick access to items on areas of the display that are difficult to reach by bringing them closer to the user for viewing and manipulation, thus minimizing user effort. In its simplest form, the technique brings toward it items that reside inside an arc of influence centered at its point of invocation and spanning the entire display. The redesign of the vacuum was driven by, and satisfies, the design principles outlined in this section. The end result is an interaction technique that retains the excellent properties of drag-and-pick [Baudisch et al., 2003a] including lightweight interaction and low overhead, while resolving the shortcomings of drag-and-pick identified above and in Section 4.1.

4.5.1 Visual design

The redesigned vacuum is still represented as a circular widget, with an arc of influence with a user controllable angle that extends from the centre along a user defined line, to the limits of the display (Figure 4.16). Proxies of all objects within the arc’s influence are brought close to the centre of the widget, within easy reach of the user, thus conforming to our minimal physical movement design principle. As with drag-and-pick and based on user comments (Section 4.3), we define a within-arm’s-reach buffer zone around the widget such that only objects outside that zone are under the influence of the vacuum. This allows nearby objects to be manipulated without proxies, and reduces the total number of proxies that have to be brought to the user. We also added an inner bull’s-eye centre to the vacuum that will be discussed later in the interaction section.

As mentioned, when using drag-and-pick and other designs in our early explorations and user feedback sessions, we observed that users were not always aware of the area of influence of the techniques (Section 4.4). Accordingly, in designing the vacuum, we ensured that appropriate visual feedback regarding the extents of the technique is always present in the form of the displayed arc edges and a faint semi-transparent overlay indicating the vacuum’s area of influence (Figure 4.16). Also, proxies are dynamically updated in a smooth animated fashion as the vacuum’s extents are changed, as discussed

in our initial exploration techniques. This enables the user to easily comprehend the vacuum's actions, conforming to our predictable and consistent design principle.

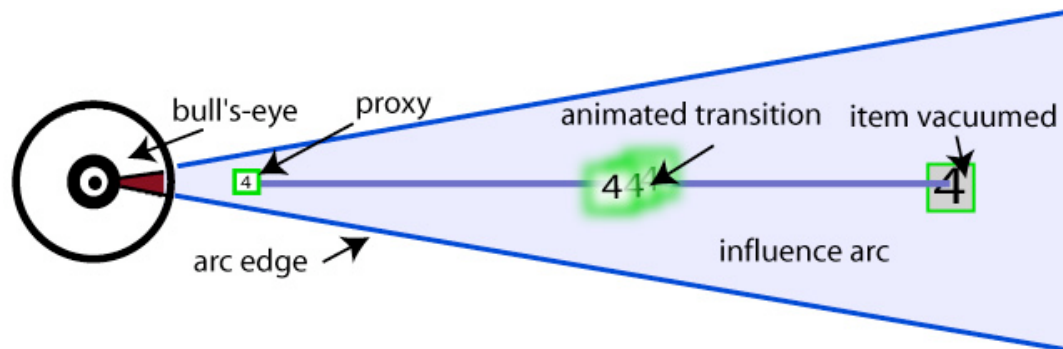


Figure 4.16: The vacuum

4.5.2 Layout of Proxies

As discussed, an important aspect of the design of the vacuum is in the representation and layout of the proxy items. The vacuum collects many items from a large area of the display, but must display copies of the items within the much smaller region near the tool's centre. As seen in our pilots, for even a moderate number of items, it is impossible to display these copies at full size near the centre without significant overlap in layout. This is why the layout decided upon is that of the scaling vacuum.

Proxies of the vacuumed items are scaled down in size and displayed around the tool's centre, preserving the relative spatial relationships of items to one another. As items enter the vacuum's influence, we animate the movement of a copy toward the tool's centre, while a ghost image of the item remains at its original position. The decision to display the semi-transparent ghost of the vacuumed item is discussed in Section 4.2.1. When an item leaves the vacuum's influence, the reverse animation takes place. This approach allows the user to maintain a sense of overall context of where everything is on the display. A semi-transparent line connects the centers of the proxies to the centre of the original items. This line passes virtually through the center of the tool as well, so

all three points are aligned. This property of the scaling vacuum allows for the copied items to be selected with cursor movements that are identical in direction, but smaller in magnitude, to the movement required for selecting the original item. Unlike drag-and-pick, the relative position of the proxies to the user is identical to that of the original objects, so if the user reinvokes the vacuum from the same position proxies will retain their position even though the user's direction might change. These properties are consistent with our predictable and consistent design principle, and alleviate the layout of proxies and number and size of proxies problems identified in drag-and-pick in Section 4.1. The disadvantage is that the proxies are significantly smaller than the original objects, and thus harder to perceive and interact with in detail without first selecting them.

4.5.3 Invocation and Adjustment

Based on our transient use design principle, the vacuum is invoked by simply dragging on the display above a fixed threshold distance (Figure 4.17.a), similarly to drag-and-pick. Due to user comments on inconsistencies to the adjustment of the arc in the scaling vacuum (Section 4.4), in the redesigned version, the vacuum angle opens towards the direction of the user's movement. Thus the invocation of the vacuum and its behavior is consistent irrespective of the direction of movement (Figure 4.17.a,b).

Initially, when redesigning the vacuum, the arc angle would open up directly to 60° towards the direction of the user's movement, as in drag-and-pick, which resulted in proxies being animated fairly fast towards the tool. This transition could be miss and users would have to spend time to re-identify the copied proxies. Given that the user drags her cursor to invoke the vacuum anyway, we decided to gradually open the angle of the arc during dragging. This decision serves several purposes. First, the effect of the tool is easier to understand through the opening of the angle, as it provides visual transitions that play an important role in understanding visual changes [Woods and Watts, 1997, Chang et al., 1998, Mertz et al., 2000]. More specifically, it explains the use of the arc

of influence, namely that as its edges cross items on the display, their proxies get copied close to the tool. Finally, it indicates that the arc of influence is fully interactive and user controllable since user actions affect its value. Thus in our final design the vacuum starts off with a fairly small arc angle (20°) that extends along the movement axis of the cursor. If the user continues dragging along the centerline of the arc away from the start position, the arc widens to a maximum of 120° (Figure 4.17.b,c). Dragging back along the centerline towards the start position narrows the arc.

The "narrowing" of the arc when moving close to the center of the tool eventually results in a very small angle, that essentially nullifies the effect of the tool. This we felt provides an opportunity to reset the effect of the tool completely. So we dedicated part of the tool's center for changing the direction of the arc. Visually we provide a bull's eye graphic to indicate this reset zone. When the user drags back through the start position (where the bull's eye graphic resides), and continues moving along another direction, the arc smoothly transitions to extending along the appropriate new direction (Figure 4.17.d). This allows for easy readjusting the direction of the angle, without having to manipulate the arc, as in the scaling vacuum, or lifting the pen, as in drag-and-pick. Thus flexibility is provided without affecting the transient use of the vacuum.

In its simplest form, the vacuum is active only during a single pen-down, drag towards the target, pen-up gesture. However, this simple drag operation precludes more involved adjustments, for example opening the arc more than 120° . In our original explorations users would have to lift their pen to adjust the angle of the scaling vacuum. To ensure that users can further adjust the vacuum's arc angle within the simple pen-down and drag actions, we provided a crossing [Apitz and Guimbretière, 2004] type manipulation. Moving off the centerline towards either edge of the arc, allows that edge to be dragged to any desired angle (Figure 4.17.e,f). Thus, if the user initially started off in an incorrect direction, a simple adjustment allows for the desired targets to be captured.

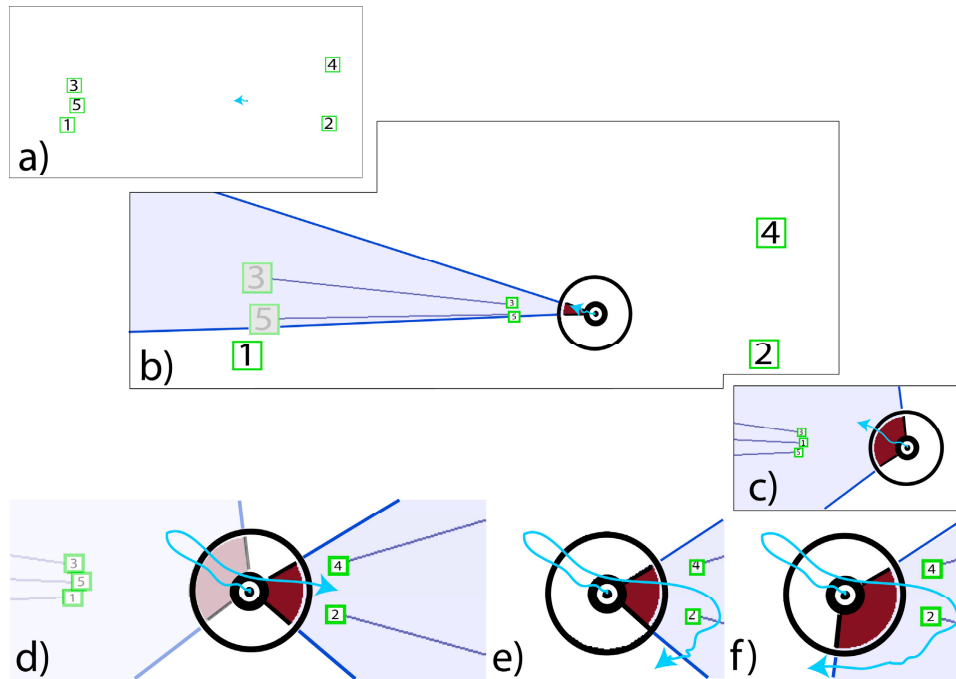


Figure 4.17: Vacuum invocation and adjustment. Blue arrows indicate cursor movement. (a) Dragging begins in the centre of the display. (b) When the drag exceeds a threshold, the vacuum is invoked with an 20° initial arc, bringing proxies of targets 3, 5 towards its center. (c) Additional cursor movement increases the arc angle, until 120° . (d) Change in cursor direction through the center changes the direction of the arc. (e, f) Cursor moving beyond the arc's edges expands the arc.

The angle of the vacuum's arc can be adjusted until it reaches a full 360° (Figure 4.18). At this point the angle "locks" until the user reverses the direction of movement to re-open the arc. This decision is based on observations of the original scaling vacuum use, where crossing the 360° angle would reset it to 0° and cause user disorientation (Section 4.3). In the new design the user crosses the edge of the arc to start moving it, implying that the edge is "pushed" by the cursor. When it reaches the other edge of the arc it "bangs" on it and stops moving, as long as the direction of cursor movement does not change. If the user crosses the edge in the opposite direction, the edge gets "pushed" the other way and starts moving, opening up the arc.

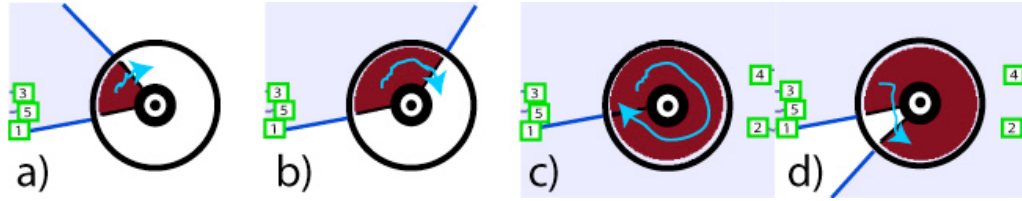


Figure 4.18: Vacuum arc adjustment, maximum angle. (a, b) Moving the cursor beyond one edge moves it, expanding the arc. (c) When both edges meet, the arc is fully expanded. (d) Reversing the cursor direction reopens the arc.

If the edge of the arc reaches the 20° that we have chosen as the minimum angle, the arc cannot be "compressed" anymore and the edge stops moving. When the users crosses in either direction after that, the corresponding edge gets attached ("pushed") and the arc starts "expanding" in that direction (Figure 4.19).

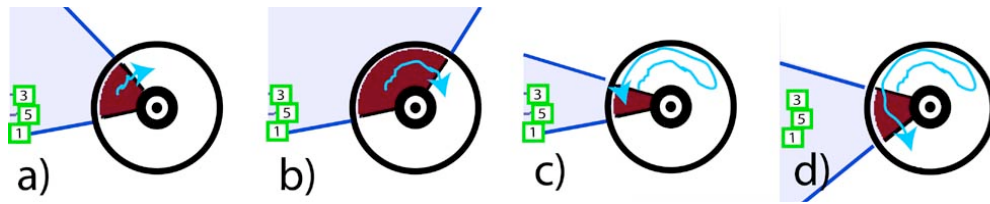


Figure 4.19: Vacuum arc adjustment, minimum angle. (a, b) Moving the cursor beyond one edge moves it, expanding the arc. (c) Changing direction collapses the arc, until the minimum angle is reached. (d) Exiting the minimum angle reopens the arc.

While the invocation of the vacuum is similar to drag-and-pick, it differs in its operation, as the direction and extents of the arc of influence are fluidly adjustable by the user simply by dragging in the appropriate directions. This ameliorates the target identification problem inherent in drag-and-pick, and conforms to our flexible design principle. Like drag-and-pick, the vacuum becomes active when dragging beyond a fixed threshold, leaving other dragging operations unavailable (ex. application defined manipulation). Nonetheless, if the invocation threshold is appropriately determined based on user preference, this should not occur accidentally.

4.5.4 Dismissal and Multiple Operations

The single pen-down, drag, pen-up gesture, with the tool being seamlessly dismissed at the pen-up event, prevents operations that require multiple pen-up/down events, like interactive manipulation of slider widgets, or multiple selections from tool palettes without re-invoking the vacuum multiple times, an issue also present in drag-and-pick. An easy way to make the vacuum persistent would be to make it a modal tool with an explicit dismiss operator, similarly to our exploration designs, which would detract from our transient design principle and reduce the vacuum's extremely fluid operation in the simple single operation situation.

Our solution was to leverage the hover state [Newman, 1968, Buxton, 1990] that is detectable in most pen-based systems. When the pen is lifted off the sensing surface, a hover zone (1-10 cm depending on the hardware) continues to track the position of the pen. As usual, a pen-down event occurs when the pen touches the surface, and a pen-up event when it leaves the surface, allowing pen-down, drag, pen-up operations to occur as expected. However, a pen-out-of-range event occurs when the pen leaves the hover zone, which we use to dismiss the vacuum. When the pen is off the surface but in the hover zone, the vacuum remains visible and active, but none of its parameters are adjustable. However, the user can now perform multiple operations with the proxies by multiple pen-up/down and drag events within the persistent vacuum. For example, multiple proxies can be selected by simply tapping on them, or a complex interface widget can be operated by sequential tapping or pen-down and drag actions.

While inside the persistent hover zone the user can pen-down on the vacuum widget and make it active again. Its parameters become adjustable, as discussed in the previous section, and the crossing metaphors as well as the bull's eye center functionality are accessed normally.

To prevent abrupt or accidental dismissal of the vacuum while inside the hover zone, we visually render the vacuum with increasing transparency as the pen's distance off the

surface increases, as feedback to the user to indicate that the hover threshold is about to be reached. When the pen reaches the hover threshold, a simple popping animation indicates dismissal of the vacuum (Figure 4.20). This implicit method of dismissing the vacuum allows for multiple operations to be fluidly supported while simultaneously retaining the valuable transient nature of the drag-and-pick technique. This conforms to our transient and flexible design principles, and also solves the multiple operations problem identified in drag-and-pick. We note that other recent techniques have also used hover in interesting ways [Grossman et al., 2006, Forlines et al., 2006b], an indication that hover might eventually become a standard part of our interaction vocabulary.

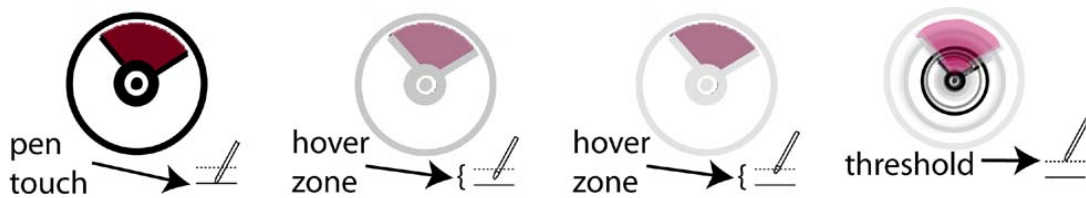


Figure 4.20: Hover and dismissal. (Left) The vacuum is active and adjustable when the pen is on the surface. (Middle) When the pen is in the hover zone, the vacuum is active but not adjustable and fades out as the pen is lifted. (Right) When the pen exceeds the hover zone’s threshold, the vacuum is dismissed with a popping animation.

By remaining active within the hover zone, the vacuum is essentially a combination (phrasing) of multiple simpler operations (chunks) [Buxton, 1986], all connected into a single interaction dialog. The dialog is the multiple target selection task, whereas the chunks are the simpler tasks of invoking the tool (drag), adjusting its angle of influence (cross edges + drag), selecting multiple objects (pen-down and up events) and dismissing the tool (exiting the hover zone). In the single target selection task, these chunks get collapsed to a single drag operation. The phrasing or combination of simpler operations using pen as input has been explored recently [Hinckley et al., 2006], using buttons clicks to combine simpler operations into a more complex interaction vocabulary.

4.6 Redesigned vacuum evaluation

After redesigning the vacuum, we evaluated its performance in two user studies.

4.6.1 Single target selection study

This study compares the performance of the vacuum to drag-and-pick and direct (unaided) picking for a single remote selection on a wall display. We expect the dismissal cost of the vacuum to affect its performance. The effect of distance from the target and density of distracter objects is also investigated. Six volunteers (2 female), aged 22-31, participated in the study. All participants had experience using a mouse, but none had previous experience with wall display interaction.

Task The task was discrete target selection in the presence of distracter targets, where users had to first click on a start target and then proceed to select a goal target. The goal target had to be successfully selected before the trial would end, even if it required multiple clicks. This removed the possibility of participants trying to race through the experiment by clicking anywhere. Targets were rendered as squares, with the start target in red, and the goal and distracter targets as numbered outlined squares. The goal target was numbered "1", and the distracters were assigned other random numbers. We felt it was necessary to test a worst case scenario (such as very similar proxies), which presents a bigger problem for the vacuum than the other techniques, due to smaller proxies. The numbering mechanism ensured that no target stood out in a prominent way. Targets were visible at the start of the trial, so users could plan their actions before beginning, simulating the situation where users are familiar with the general display layout.

Design A within-participant factorial design was used. The independent variables were selection *technique* (direct, vacuum, drag-and-pick), *distance* between start and goal targets (Mid, Far), and density of *path distracter* targets along the path from start to the goal target (None, Few, Many).

As we felt varying the size of targets affects techniques similarly (Section 4.4), in this formal study we used only one target size of 15 cm, which we feel is representative of icons on a large display of this size. Given that open windows and other applications will likely be larger, this size can be viewed as a lower-bound.

Similarly to our previous studies, we used two distances between the start and goal targets: Mid = 176 cm, roughly the display height, and Far = 264 cm, roughly $3/2$ the display height. For each of the distances, we positioned start and goal targets in 6 relative compass directions (E, NE, NW, W, SW, SE) about the centre of the display, to counterbalance for any possible differences in movement direction. We did not use the N and S directions because they would have involved starting uncomfortably from the bottom or top of the display and moving to the other edge for the Mid distance, and would have been impossible for the Far distance.

In real interfaces, the path towards a goal is rarely void of any interfering objects. As in our pilot session (Section 4.4), we focused on the effect of path distracters that were placed along the path from start to goal targets. For path distracters, we investigate three densities: None: no distracters, Few: 40% of a rectangular area between the start and goal targets is filled with distracters, and Many: 80% of this area is covered. Task samples can be seen in Figure 4.21.

Participants were randomly assigned to 3 groups of 2. In each group participants used all 3 techniques, in an ordering balanced using a Latin square. In summary the study consisted of:

6 participants \times
 3 techniques \times
 3 repetition blocks per technique \times
 2 distances \times
 6 directions \times
 3 path distracter densities = 1944 trials



Figure 4.21: Single Target experimental setup. The red square is the start target and the number 1 is the desired target. Three setups are presented: (top) Mid distance with Few path distractors, (middle) Far distance with None path distractors and (bottom) Far distance with Many path distractors. The green boxes are for illustration only.

Prior to each technique participants were given a short warm-up session to familiarize themselves with the techniques. The session lasted on average one hour for each participant, after which participants filled out a brief questionnaire to elicit their opinions.

Performance Measures To evaluate the three techniques we examined time and error measures. *TotalTime* is the time to fully conclude a trial successfully and in the case of the vacuum it includes the cost of dismissing the vacuum by moving the pen beyond the hover threshold. *SelectionTime* is the time to successfully select the target, but not including the time to dismiss the vacuum. For the direct and drag-and-pick techniques *SelectionTime* is equivalent to *TotalTime*. *ErrorRate* is the percentage of trials where the target was not selected on the first click.

Results and Discussion

Data from a total of 27 trials (1.4%) were identified as outliers for *TotalTime* (20 trials or 1.1% in the case of *SelectionTime*) and were removed from further analysis.

	TotalTime	SelectionTime	ErrorRate
direct picking	2.58 sec	2.49 sec	2 %
vacuum	3.03 sec	2.53 sec	3 %
drag-and-pick	2.49 sec	2.58 sec	5 %

Figure 4.22: Time and error related measures of the single target selection study. Significant effects are marked with red.

Time analysis (Figures 4.22, 4.23) The time measurements' distributions were positively skewed, so significance tests were performed on the logarithm of the measurements.

A fully within subjects ANOVA showed a significant effect for *technique* on *logTotalTime* ($F_{2,10} = 9.6$, $p < .05$). Post-hoc pairwise means comparisons showed the vacuum to be significantly slower than drag-and-pick ($p < .05$). Although there was a significant effect for *distance* on *logTotalTime* ($F_{1,5} = 132$, $p < .0001$), no *distance* x *technique* interaction was present. Finally, there was both a significant effect for *path distracters* on *logTotalTime* ($F_{2,10} = 34.8$, $p < .0001$) and a significant *path distracters* x *technique* interaction ($F_{4,20} = 9.9$, $p < .0001$). Post-hoc pairwise mean comparisons showed the vacuum to be significantly slower than drag-and-pick in the case of None and Few *path distracters*, and the no mediator condition to be significantly slower than drag-and-pick in the case of None *distracters* (all $p < .05$). There was no significant difference between the techniques when *logSelectionTime* was used as the dependent variable.

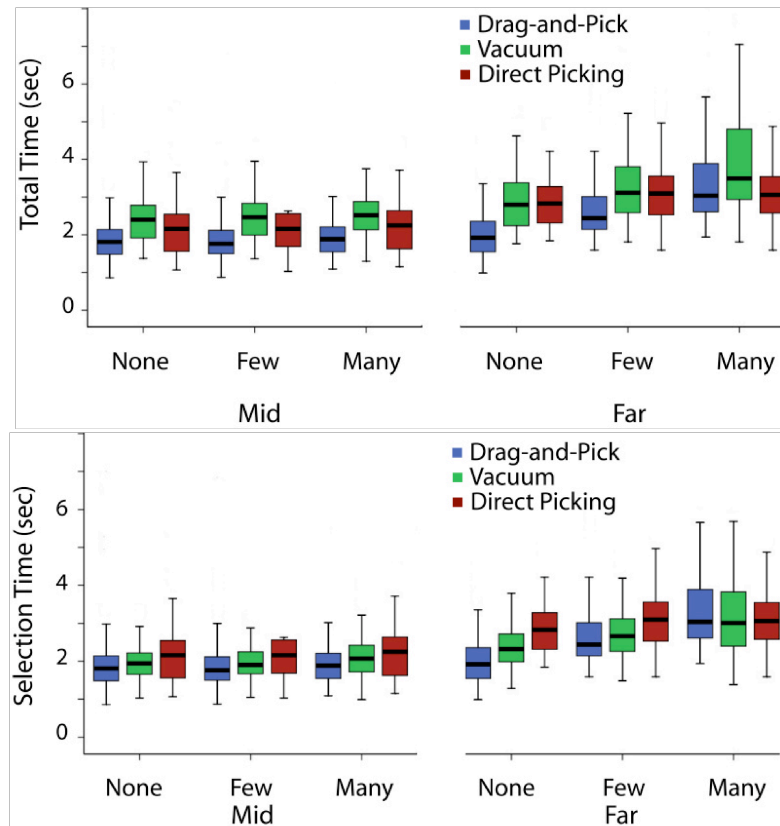


Figure 4.23: Boxplots (median,range) of *TotalTime* (top) and *SelectionTime* (bottom) broken down by technique, distance (Mid, Far), and path distracters (None, Few, Many).

Time analysis discussion As we mentioned in the case of the vacuum, the *TotalTime* is always larger than *SelectionTime*, an overhead due to the time required to dismiss the widget. Although the use of hover as an implicit dismissal mechanism was a design choice aimed at reducing this cost, it is clear that the cost is not completely eliminated (Figure 4.24(a)). Nevertheless, users did not perceive this cost as a delay and some of them commented they found the action of lifting the pen after completing the task very natural. Moreover, we believe that this dismissal cost can be further reduced for experienced users by reducing the hover zone.

As expected from Fitts' law, *distance* significantly affected *TotalTime*, but there was no significant *technique* x *distance* interaction indicating that all techniques were similarly

affected by changes in distance. This result can be explained by the following analysis of the techniques:

The scaling property of the vacuum results in the size of its proxy targets being significantly smaller than the proxies in drag-and-pick, or the original targets in direct picking. The distance between the start position and the proxy target also shrinks, and since the scaling is uniform, the resulting Fitts' law index of difficulty (ID) of the proxies in the vacuum are similar to that of the targets in direct picking. In contrast, the proxies in drag-and-pick are unchanged in size but are moved closer to the start position, resulting in a smaller Fitts' ID compared to the vacuum or direct picking. Thus one might expect vacuum and direct picking to be similarly affected by changes in distance, whereas drag-and-pick should not be affected. The fact that drag-and-pick is affected by distance in our experiment is due to the presence of *path distracters*. Since *path distracters* were controlled based on density (40% or 80% of a rectangular area between start and target), the resulting area separating the target from the start position in the Mid distance has less distracter items and is thus smaller than that in the Far distance. Thus, changes in distance affect drag-and-pick, but only when distracters are present.

Our post-hoc means comparisons showed the vacuum to be significantly slower than drag-and-pick for None and Few *path distracters*, and the no mediator (direct picking) condition to be significantly slower than drag-and-pick in the case of None *distracters*.

The small mean times for drag-and-pick in the None and Few cases is due to the fact that the technique is designed for relatively sparse layouts. In the case of None and Few distracters the cluster of potential targets is very small and the desired item is easy to pinpoint. Moreover, as mentioned, the distance to the desired target in drag-and-pick is determined by the number of intervening distracter items. So less distracters result in targets very close to the user. As seen Figure 4.23, in the Far distances drag-and-pick seems more affected by distracters than the Mid cases, as in the Mid case the absolute

number of distracters is always fairly small (0 for None and on average 8 for Few, and 15 for Many) than in the Far case (0 for None and on average 15 for Few and 30 for Far). Moreover the existence of the buffer zone surrounding the technique further reduces the number of effective path distracters in all cases (for example the maximum number of effective distracters is never bigger than 20). The buffer zone affects the number of distracters more prominently in the Mid distance, since almost half of that distance is inside the buffer zone, reducing the number of effective distracters in both densities.

As seen in Figure 4.23 direct picking is not affected by distracters since users know the location of the goal target before the start of the trial and thus the acquisition time is only affected by distance. In contrast, drag-and-pick and vacuum incur an extra cost of identifying the proxy after a set of targets are selected, and performance degrades as the density of distracters increases.

We note that the significant advantages of drag-and-*pop* over direct dragging observed by Baudisch et al. [2003a] are not seen in our study comparing drag-and-*pick* to direct picking. We believe this is because dragging [Baudisch et al., 2003a] requires users to maintain contact with the display surface, which is difficult to do over long distances. In contrast, direct picking is accomplished by tapping on the start target, lifting the pen, walking quickly over to and tapping the goal target (like in the pick-and-drop [Rekimoto, 1997] and take-it [Geißler, 1998] technique). The pen does not have to maintain display contact and so it is not all that much worse than using drag-and-pick or the vacuum. Users simply trade off walking for the complexity of drag-and-pick or the vacuum.

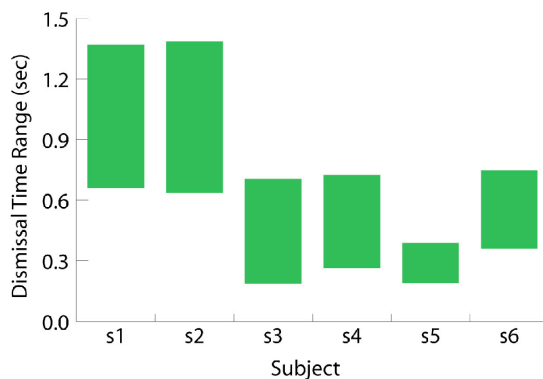
Error Analysis A fully within subjects ANOVA showed no significant effect for *technique*, *distance*, or *path distracters* on *ErrorRate* (Figure 4.22). 67% of all errors in drag-and-pick were users invoking the technique with an angle that missed the target. Errors of this nature only accounted for 5% of all vacuum errors, whereas 52% were due to accidental dismissal of the widget and 43% due to simply missing the target.

Subjective Comments from Participants After the experiment users were asked to rank the techniques based on speed, accuracy, and ease of understanding. They were also asked to state if they had a preference for any of the techniques.

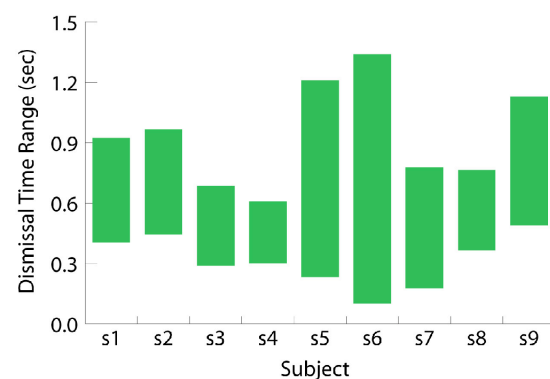
3 participants ranked vacuum and drag-and-pick as equally fast, 2 ranked the vacuum as fastest and 1 ranked drag-and-pick as fastest. The two who ranked vacuum higher mentioned that they could identify the target faster than with drag-and-pick: they would start moving towards the rough position of the target and quickly re-identify it amongst the proxies. On the contrary, they felt they spent more time deciding on the angle of approach when using drag-and-pick, time not captured for this task.

4 participants ranked direct picking as the least error prone, while the other 2 participants each ranked vacuum and drag-and-pick as the least error prone respectively.

All participants ranked direct picking as the easiest to understand. For the given task, 4 participants preferred the vacuum and 2 drag-and-pick, even though the vacuum was slower. They all preferred using a mediator instead of walking to the targets. This indicates that while direct picking was not significantly different, users are willing to use a remote reaching technique to avoid walking across a display of this size.



(a) Dismissal Time range for participants in single target experiment.



(b) Dismissal Time range for participants in multiple target experiment.

Figure 4.24: Cost of dismissing the vacuum, by exiting the hover zone threshold.

4.6.2 Multiple target selection study

The goals of this experiment were similar to the single target one, except that we now use the three techniques in the context of selecting multiple targets in sequence. We examine how between target distance affects the different techniques, as well as the presence of distracter targets on the display. Since the vacuum is designed for such a usage scenario we expect it to outperform the other two techniques. We evaluated the vacuum against the current drag-and-pick and direct picking implementation in order to see the relative performance of the core techniques (the differences can potentially be small). We decided to not evaluate an extension to drag-and-pick, since our goal was not to extend the existing technique but provide an alternative one. Nine volunteers (3 female), aged 25-35, participated in the study. All had previous experience using a mouse but none had previous experience with large display interaction.

Task The task was consecutive selection of five targets, in the presence of distracters. Participants started a trial by first selecting a red start target, and proceeded to select each of the five goal targets in sequence. If the vacuum was accidentally dismissed before all targets were selected, the user had to re-invoke it from the starting position. For drag-and-pick, each invocation started from the red start target, in order to prevent users from simply reverting to direct picking once they reached a cluster of targets. We adjust for this additional overhead in our movement time analysis.

Targets were rendered in a similar manner to experiment 1. The five goal targets were numbered 1 to 5 and distracters were assigned other numbers (Figure 4.25). In terms of multiple file/icon selection it seemed to us that 5 was a good upper bound to test trends. Moreover, we felt that an existing complicated remote interaction is close to 5 selections (for example go to a menu, select the file submenu, select the open command, press browse button, select a file).

To prevent participants from racing through the experiment by clicking anywhere,

they had to successfully select each target in the correct sequence. All targets were visible at the start of the trial, so users could plan their actions before beginning, simulating situations where users are familiar with the interface layout.

As each target of the sequence was selected it changed color from white to green to help the participant keep track of where they were in the sequence. In direct pointing and drag-and-pick, the trial ended when the fifth target was successfully selected. For the vacuum, when the fifth target was selected, all targets turned red indicating it was time to dismiss the vacuum by pulling away from the hover zone.

Design A within-participant factorial design was used. The independent variables were selection *technique* (direct, vacuum, drag-and-pick), between-targets *distance* (Close, Mid, Far), and number of *distracter* targets that were randomly scattered about the display (None, Few, Many).

The three between-target distances used were: Close = 43 cm, Mid = 86 cm, and Far = 130 cm. These represent roughly 1/4, 1/2, and 3/4 of the height of our display respectively. In the Close and Mid distances we cluster the targets as close as possible (rather than stringing them along in a continuous sequence, for example) to simulate real situations where users group windows or icons in clusters. The result was that in the Close distance, the targets were approximately within a quadrant of the display, in the Mid distance targets were in half of the display, and in the Far distance targets spanned the entire display. For each of the distances we used two orientations, one in which the first target was on the left and another on the right of the starting position (Figure 4.25).

Since goal targets were located on different areas of the display we decided to place the distracter targets randomly, rather than the more specific placements of area and path distracters in the single target study. We used layout distracters instead of path distracters so as not to bias the results against drag-and-pick. As discussed, in drag-and-pick path distracters affect the distance that the proxy is going to end up from the user.

If we had used path distracters between the start position and the targets they would affect drag-and-pick but not the vacuum (since the movement there is essentially between targets). Respectively, we did not use between targets distracters so as not bias against the vacuum. Thus we decided to arrange distracters randomly around the display.

We used three densities of distracters: None, Few: 15% display coverage, and Many: 20% coverage (Figure 4.25) so as to visually get a layout that looked reasonably crowded (less than that it was too sparse around the targets). Here we should note that because the distracters can be placed anywhere we imposed certain restrictions, for example they were no distracters at the top and bottom of the display. Moreover, due to the observation of the first experiment that the effective distracters were only outside the buffer zone, distracters we not placed in the buffer zone. So although the density reported is indeed the overall density, there are empty areas on the display and thus the concentration of the distracters is bigger around the targets. We used a percentage of Many 20% which resulted in approximately the same number of distracters on the display as in the path distracter experiment. We felt that the layout was sufficiently cluttered (given that we had already seen the effects of a big number of distracters in the single target experiment).

Participants were randomly assigned to 3 groups of 3. In each group participants used all 3 techniques, in an ordering balanced using a Latin square. For each technique participants completed 3 blocks of 18 trials. In summary the experiment consisted of:

$$\begin{aligned}
 &9 \quad \text{participants} \times \\
 &3 \quad \text{techniques} \times \\
 &3 \quad \text{repetition blocks per technique} \times \\
 &3 \quad \text{distances between targets} \times \\
 &3 \quad \text{distracter layout densities} \times \\
 &2 \quad \text{orientations} \\
 &= 1458 \text{ trials}
 \end{aligned}$$

Participants performed 10 warm-up trials per technique and were asked to complete

the trials as fast and accurately as possible, taking breaks between trials and blocks.

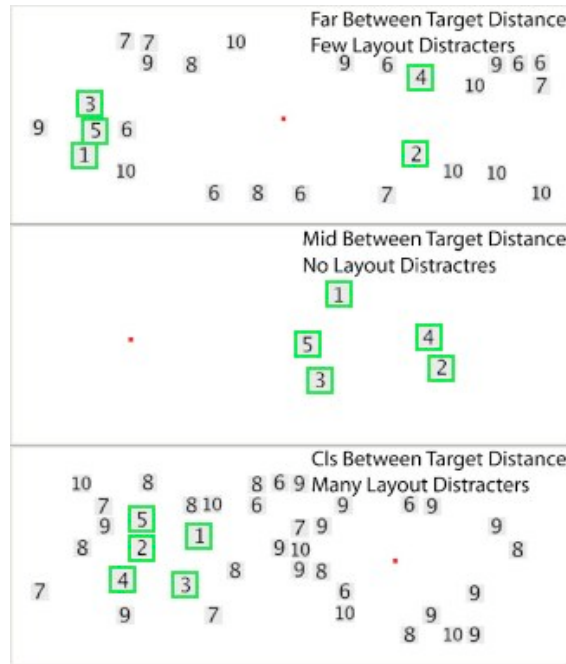


Figure 4.25: Multiple Target experimental setup. The red square is the start target and numbers 1-5 are the desired targets. Three setups are presented: (top) Far distance with Few layout distracters, (middle) Mid distance with None distracters and (bottom) Close distance with Many distracters. The green boxes are for illustration only.

Performance Measures To evaluate the three techniques we examined time and error measures. The first measure is the *TotalTime* it took users to complete a task. This includes the vacuum dismissal cost. As discussed earlier, drag-and-pick is a single target selection technique. In order to simulate realistic use of drag-and-pick for multiple target selection, our experiment design required users to return to the red starting target between each selection of the five targets. In contrast, the vacuum and direct picking easily handled multiple target selection in sequence and there was no need to return to the start position each time (except when the vacuum was accidentally dismissed).

To avoid biasing against drag-and-pick, for all drag-and-pick data we computed an additional measure called *AdaptedTotalTime* where we removed the time between select-

ing a target and returning to the start position to start the process of selecting the next target. In a sense, this measures a best case scenario for drag-and-pick. Similarly, we also removed the time to return to the start position in the few cases of accidental dismissal of the vacuum. Since the final dismissal of the vacuum is crucial for this type of task, we take it into account everywhere in our analysis. Finally, we examined the *ErrorRate*, the percentage of trials where targets were not selected correctly on the first attempt.

Results and Discussion

Data from a total of 20 trials (1.1%) were identified as outliers for *TotalTime* (22 from *AdaptedTotalTime*) and removed from further analysis.

	TotalTime	AdaptedTime	ErrorRate
direct picking	8.54 sec	8.52 sec	6.6 %
vacuum	8.11 sec	7.64 sec	16%
drag-and-pick	16.35 sec	10.67 sec	39 %

Figure 4.26: Time and error related measures of the multiple target selection study. Significant effects are marked with red.

Time Analysis (Figures 4.26, 4.27) As the distributions of the time measurements were positively skewed, significance tests were performed on the logarithm of *TotalTime* and *AdaptedTotalTime* measurement.

A fully within subjects ANOVA showed a significant effect for *technique* on *logTotalTime* ($F_{2,16} = 150.7, p < .0001$). Similar results were found for *logAdaptedTotalTime* ($F_{2,16} = 27.6, p < .0001$). Post-hoc pairwise means comparisons showed that drag-and-pick was significantly slower than vacuum and direct picking ($p < .0001$). Mean times can be seen in Figure 4.26.

Between target *distance* had a significant effect on *logTotalTime* ($F_{2,16} = 437.4, p < .0001$). A significant *technique* x *distance* interaction ($F_{4,32} = 120, p < .0001$) was

also present. When examining *TotalTime*, post-hoc pairwise means comparisons showed the vacuum to be significantly faster than drag-and-pick ($p < .01$) for all examined distances, while drag-and-pick was significantly slower than direct pointing as well for Close and Mid distances ($p < .0001$), but not for the Far distance. Between target *distance* also had a significant effect on *logAdaptedTotalTime* ($F_{2,16} = 271.8, p < .0001$). There was a significant *technique* x *distance* interaction ($F_{4,32} = 125.3, p < .0001$). Post-hoc means comparisons for the *AdaptedTotalTime* showed the vacuum and drag-and-pick significantly faster than direct picking in Far distances ($p < .05$), but drag-and-pick significantly slower than both vacuum and direct picking in Close and Mid distances (all $p < .0001$).

The *distracter* density had a significant main effect on *logTotalTime* ($F_{2,16} = 216.9, p < .0001$). Also, a significant *technique* x *distracter* interaction was present ($F_{4,32} = 37.2, p < .0001$). Post-hoc means comparisons showed drag-and-pick to be significantly slower than vacuum and direct picking in all distracter cases (all $p < .0001$). The *distracter* density also had a significant effect on *logAdaptedTotalTime* ($F_{2,16} = 205.8, p < .0001$). A significant *technique* x *distracter* interaction ($F_{4,32} = 35.3, p < .0001$) was also present. Post-hoc means comparisons showed the vacuum to be significantly faster than drag-and-pick in all distracter cases ($p < .01$) and drag-and-pick significantly slower than direct picking for Few ($p < .05$) and Many ($p < .01$).

Time analysis discussion Despite the large reduction in time for drag-and-pick when comparing *AdaptedTotalTime* to *TotalTime*, drag-and-pick was still slower than vacuum in all cases and direct picking in most cases (for both time measures). The vacuum performed very well, especially in the Far distances where it was shown faster than both direct picking and drag-and-pick.

Although we expected the performance of drag-and-pick to be similar across distances, it seems to perform better in the Far than in the Mid distance, as seen in the post-hoc

means comparisons. This may be a result of using proximal targets as guides to plan the next movement. In the Far case, although two consecutive targets are far away, there are two very tight clusters of targets at each side of the user. For instance, by selecting target 2 the user sees that target 4 is also going to end up very close to that position. In the Mid distance, as the user selects a target, the remaining desired target group might or might not end up in the resulting set of proxies, making it harder to plan ahead.

Drag-and-pick was significantly slower than vacuum and direct picking in all distracter cases for *TotalTime*. This indicates that in the presence of distracters, users exploit the persistent layout of vacuum and direct picking to identifying multiple targets. The effect is also present in the Few and Many case of *AdaptedTotalTime* due to the persistent layout benefits of direct picking being more prominent in the Few and Many cases.

As discussed earlier, drag-and-pick is a single target selection technique and our experiment design required users to return to the starting target between each selection of the five targets. To not bias against drag-and-pick we used the *AdaptedTotalTime* measure. This metric is favorable for drag-and-pick since by subtracting the return to start time, we also remove the time the users needed to plan their movement in order to include all targets in the arc of influence of drag-and-pick. Nevertheless, the only major difference occurred in the Far between targets distance.

Error Analysis A fully within subjects ANOVA showed a significant effect for *technique* on *ErrorRate* ($F_{2,16} = 36.5$, $p < .0001$), with drag-and-pick being the most error prone technique (Figure 4.26). As in the single target experiment, the majority of drag-and-pick errors occurred when users missed the target in their initial choice of movement angle (32% of all errors) and when they selected the wrong target in the group of proxies (29% of all errors) which indicates that users found it hard to distinguish the target from other proxies. In the vacuum, the majority of errors (87%) were missing targets due to their small sizes, and 13% were accidental dismissal errors.

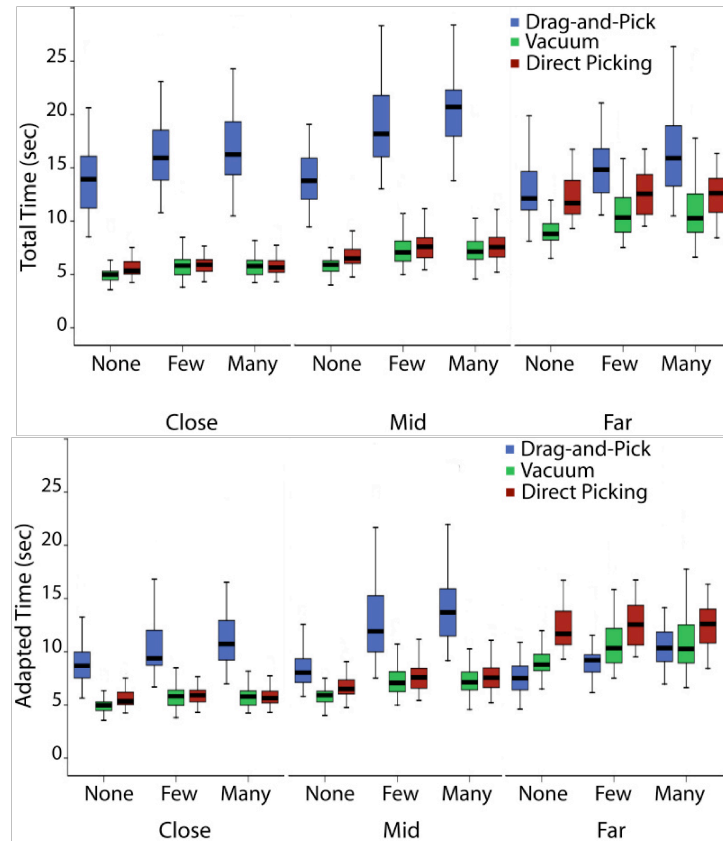


Figure 4.27: Total Time (top) and Adapted Total Time (bottom) broken down by technique, distance (Close, Mid, Far), and distracters (None, Few, Many).

Subjective Comments from Participants After the study users were asked to rank the techniques based on speed, accuracy, and ease of understanding. They were also asked if they had a preference for any of the techniques. All 9 participants ranked the vacuum as the fastest technique. All ranked drag-and-pick as the most error prone. 5 participants ranked vacuum as the least error prone, 3 direct picking, and 1 ranked both vacuum and direct picking as equivalent in terms of accuracy.

All 9 participants ranked direct picking as the easiest to understand, with vacuum generally ranking second over drag-and-pick. 4 participants mentioned that although the idea behind drag-and-pick was easy to understand, predicting the final position of the proxies was not. They added that they had to spend a lot of time to re-identify the appropriate proxy and that sometimes they missed the target completely with the

selection arc, or failed to identify it in the group of proxies. One participant made it more concrete by stating it was hard to judge due to distance where the arc was in effect. For the vacuum, they all mentioned that when targets were at a single direction the technique was extremely easy to use, but that they had some problems getting used to the angle adjustment for items scattered in opposite directions.

In terms of overall preference for the given task, all participants ranked the vacuum highest, 5 ranked drag-and-pick second, 2 ranked direct picking second, and 2 ranked drag-and-pick and direct pointing as equivalent.

4.6.3 Discussion

The vacuum offers several advantages over drag-and-pop/pick. Its visible influence arc clarifies the effects of the technique. Users can adjust the direction and angle of the influence arc dynamically during an operation, and the addition of a hover region above the surface allows multiple operations to be combined in a single invocation of the tool. Shrunk proxies allow more proxies to be represented and permit the proxies to maintain the relative spatial arrangement of the original items. Our studies' results indicate that the vacuum performs similarly to direct picking and drag-and-pick in single target selection tasks, except when the targets are located very far away with distracters along the path. However, the vacuum scales nicely to multiple target selection, and performs significantly better in this scenario than existing techniques.

The main drawback of the vacuum relative to drag-and-pick as a visualization technique is the scale of the proxies. This will be especially prominent if the need to differentiate between similar icon sized targets arises. Imagine for example looking for a particular item whose exact location is not known. With drag-and-pick repeated invocations with different directions would be necessary to locate the item. On the other hand the vacuum may act as an exploration tool when invoked with a 360° arc of influence. Nevertheless, if the vacuumed proxies are too small, the desired item might not be dis-

tinguishable. We believe the high resolution of our wall display alleviates some of the scale issues since proxies are relatively decipherable. However, lower resolution displays may make the proxies unreadable. In this instance a combination of vacuum and canvas portals might be prove efficient. A discussion on combining the two designs can be found in Section 6.1.1.

On the other hand the main drawback of drag-and-pick is its unpredictability. This can be resolved in two ways: by providing a better visualization of the sector of influence, and by positioning the items based more on the starting position of the user and less on the direction of movement. Thus although drag-and-pick could be enhanced with hover to support multiple selections (although the drag/lift paradigm changes), major alterations are needed to provide a predictable and performance. Moreover, the lack of on the fly adjustment of the tool parameters remains.

It is important to note that while direct picking performed reasonably well in our studies, in real use it may actually not be possible to directly point to all parts of a wall display. Moreover, in extended use, user comments indicated a preference for techniques that minimize physical travel, like the vacuum, even if it incurs a slight overhead.

More generally, our work demonstrates that tasks requiring infrequent remote access typically benefit from low dismissal cost techniques like drag-and-pick or a vacuum with a small hover range. Tasks based on sequential or frequent remote content access are better accommodated by more flexible techniques, like the vacuum presented here. Finally, while our work focused on distance reaching techniques for wall displays, the vacuum could also be used on smaller devices like TabletPCs, where users may prefer working in a comfortable zone and not reaching to the display extremities.

Push-and-pop [Collomb et al., 2005a] is a later extension of drag-and-pop. Their main difference is that when an icon is being dragged, not only icons from inside the arc surrounding the user movement, but all icons relevant to the dragged icon are copied close to the user (Figure 4.28). Thus push-and-pop eliminates the issue of predicability of

the final location of proxies present in drag-and-pop, as well as problems arising from not choosing the appropriate sector of influence. Nevertheless, as proxies are still compressed in a grid layout, this technique cannot accommodate a large number of proxies or proxies of different sizes and types (other than icons). Finally, it is in effect a single target selection technique and does not support multiple operations or complex functionality.

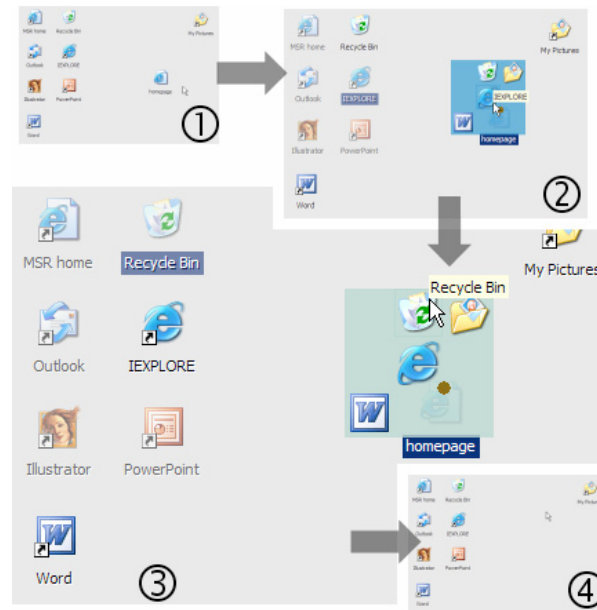


Figure 4.28: Push-and-pop. (1) As the user drags a document, (2) the system surrounds the pointer with copies of all relevant application icons in a condensed layout. (3) The user drops the document on the Recycle Bin icon and (4) the copies disappear. (Image from Collomb et al. [2005a])

Our design of the vacuum is based on the assumption that the hover zone over touch sensitive displays is a smooth range available to designers. Nevertheless, in many occasions hover is detected as a discrete property: the pen is either touching, inside the hover zone, or exiting the hover zone. In these surfaces the design of the vacuum needs to be adjusted. The vacuum can still be active in the hover zone, but more feedback is needed when the user is about to dismiss the tool, since the smooth transparency adjustment based on distance from the display is no longer available. We propose that when the user

exits the hover zone the pop-ing animation starts, but the tool remains active for another 600ms [Hinckley et al., 2006]. Thus if the user has accidentally exited the hover zone, she sees the animation starting and can quickly enter it again. Depending on the display, other options may be available. For example if a display is pressure sensitive, then the complex vacuum behavior (ex. selecting multiple targets) can be performed using different pressure states. Finally, complex behavior can be supported using alternative single stroke mechanisms, such as crossing [Apitz and Guimbretière, 2004]).

4.6.4 Potential refinements

Our work has indicated how small design variations can impact an interaction technique. For example, the use of hover enabled us to design the vacuum to work well with multiple targets. Further design alternatives could also be explored. For example, an interesting variation to the vacuum would be to use a hard boundary right before dismissal to clearly warn the user of what is about to happen, rather than our current design of smoothly fading the widget as the pen moves away.

Through observing our users and based on their comments, we found that they occasionally accidentally crossed the bull's eye center and reset the vacuum angle. In attempting to improve our design, we could remove the bull's eye center. Nevertheless, the fluidity and flexibility of the vacuum would suffer. If for instance users wanted to make significant angle changes, they would have to either perform complicated adjustments to the arc of the vacuum through crossing, or re-invoke the tool.

When using the vacuum, a single target is selected by dragging the pen until the target is reached and lifting it off the display, while multiple targets are selected by sequentially touching and lifting the pen inside the hover zone. Given our decision to cross the edges of the arc to manipulate them, general use of crossing for selection has been considered. For example users could select a single item by crossing it and lifting the pen, while for multiple targets they could cross all targets and then lift the pen. This design can be

problematic in somewhat cluttered layouts, where the user has to bypass or cycle through multiple items in order to cross desired targets. Moreover it takes away the consistency of the technique, as for similar selections (for example items that are close), the user might have to follow very different dragging paths. Another approach would be to make a scribble or pigtail mark on desired objects while dragging. Although this solution preserves the fluid and consistent nature of the technique, it requires users to learn a new interaction paradigm, thus removing its transparent nature. Besides, in either crossing design, it is unclear if users will only want to perform selections of items to be copied close to their location. Depending on the interaction, users might want to manipulate menus, pallets, or adjust sliders directly on their proxies, actions involving a significantly different interaction paradigm in crossing interfaces [Apitz and Guimbreti re, 2004].

While the user drags on the vacuum widget (for example during the first invocation or when resetting the widget angle), interaction becomes size independent in terms of the user’s stroke and can be partially performed outside the widget confounds. In other words the user can perform angle adjustments through crossing, without dragging inside the vacuum’s visual representation (Figure 4.17). It would thus be possible to make the vacuum widget essentially invisible and remove some of the visual clutter in the area in front of the user due to the visual representation of the widget. The issue that might arise though is whether ease of parameter adjustment would suffer. Although this is an interesting design variation, a more promising investigation would be to create an entirely eyes free interaction by removing even the visual representations of the proxies, with a transition from novice to expert such as that of a marking menu [Kurtenbach and Buxton, 1993]. Obviously, such a variation would require careful design of transitions between novice and expert user, as well as detailed investigation on whether the transition from novice to expert user is possible and examine the performance benefits, if any.

While the vacuum’s arc of influence reduces the number of proxies that are brought close to the user, in very dense layouts, due to the fanning of the arc, the number of

copied proxies can still be large. To reduce this number, we could envision the vacuum as a "spiral" instead of a circle. Based on this idea, the vacuum's arc can be divided into concentric slices, and depending on how far from the invocation point the user drags, a different slice would be copied close to the user. The user could then lift the pen inside the hover zone to make appropriate selections. Alternatively the user could "dial" around the tool to progressively bring different slices close. Although the spiral vacuum is an interesting conception, it complicates the fluid vacuum interaction and burdens users with detailed parameter adjustments, that seem to lend better to tools of a more permanent nature, for prolonged interaction (as discussed in Chapter 3).

Although the discussed refinements open up further possibilities for new designs, they violate some of our design principles or require major redesign attempts. Given that the vacuum was well perceived by users and performed well in the given tasks, we decided to stop its exploration. We have by no means solved all aspects of remote reaching issues on wall displays, but we feel we have provided useful design guidelines for future research and have explained the design process of a fluid and useful remote reaching mediator.

4.7 Chapter summary and closing remarks

In this section we have presented the design process for a remote reaching technique, the vacuum. The process began by examining drag-and-pop/pick [Baudisch et al., 2003a], an existing remote reaching technique, and identifying some of its characteristics. We then went onto implementing several exploration prototypes attempting to follow our design principles of application independence, effort minimization, predictability and consistency. Through user feedback sessions, we dismissed designs that violated our design goals. Furthermore, we gained useful insights and identified issues in our most promising prototype and drag-and-pop/pick. And finally, we further refined our design principles based on user feedback. New design principles for remote reaching techniques

include the need: for transparent techniques that integrate seamlessly into the normal interaction tasks performed by users; for transient use that does not burden the user with large invocation and setup costs; as well as the need for flexibility in terms of parameter adjustment and tool functionality. Based on our new design principles we redesigned one of our prototypes, the vacuum, which we evaluated against drag-and-pick and direct unaided selection. Our studies showed the vacuum to perform similarly to direct picking in single target selection tasks and slower than drag-and-pick when there are few distracters along the path. However, the vacuum scales nicely to multiple target selection, and performs better in this scenario than existing techniques.

Our work has indicated how a combination of design explorations and user feedback can help pinpoint important aspects of a targeted problem, such as distance reaching. This process has allowed us to refine our original design goals in order to create a general purpose, intuitive and well received technique for wall displays.

Chapter 5

Presenting Invisible Changes

In current computing displays, dynamic visual changes regularly happen without the user being aware of them. For example, a download manager might complete its task in the background, emails might be received in an occluded window, or changes might occur in a partially hidden live video stream. Unless explicit notification or history mechanisms are provided, chances are that the user will miss these changes, or only become aware of them by chance or by explicitly trying to reconstruct them at a later time. Indeed, studies have shown that people are rarely able to spot visual changes when they occur during disruptions as short as eye saccades [Rensink, 2002]. Changes occurring during longer periods of invisibility should be even harder to reconstruct, because users might have partly forgotten the previous state or missed significant intermediate changes. In particular, visual transitions often play an important role in the understanding of visual changes [Woods and Watts, 1997, Chang et al., 1998, Mertz et al., 2000].

While some applications do provide notifications, these typically work well for single discrete events and not for multiple or continuous events over a period of time. Further, current window managers do not provide any integrated system-wide support for visualizing such dynamic changes occurring in the background, in the users periphery, or over some past time period. These problem gets worst as display size increases, for

example in wall displays, and more visual space remains outside the user’s visual field at any given time. Moreover, the highly dynamic nature of data in many applications for wall displays, aggravates this situation.

In this section we explore the idea of recording and presenting visual changes that might have been missed by the user due to the changes occurring in the background, periphery, or at a time where the user was otherwise preoccupied. As part of this exploration, we first survey existing approaches and discuss their limitations. We then develop a general image-based storage, visualization, and implicit interaction paradigm to provide better support for handling visual changes, called mnemonic rendering. The term mnemonic is intended to capture the essence of storing, cueing and access to memory

The design of mnemonic rendering attempts to provide an application agnostic mechanism for storing and revealing changes that have happened in invisible locations. Thus users need not memorize previous states of the large visual canvas of the wall display, or constantly move around the display to update their mental model of the display content. We thus satisfy our application independence and minimal user effort design goals. Through our design exploration we discuss how predictability and consistency is also achieved. Although mnemonic rendering is applicable to both small and large displays [Bezerianos et al., 2006], this chapter focuses on wall display interaction.

5.1 Existing approaches

Current window environments have been used for more than two decades and most of the time users seem to have no trouble integrating dynamic visual information. The main reason is that windowed applications don’t expect to be continuously visible and thus rely on implicit or explicit strategies for easing visual information integration. Commonly used strategies are persistence, structuring, reduction, history and notification. These are semantic strategies, in that they depend on the meaning of the visual content:

Persistence The visual content of a window at a given time is often sufficient to infer what happened. For example, new emails don't simply flash in an email client; they accumulate inside a container and can be accessed via a scrollbar if not visible any more. Persistence ensures that most changes are not lost.

Structuring Window contents are not just messy visual scenes; they are structured in meaningful ways so that they can be quickly interpreted into discrete, higher-level concepts. Structuring visual information allows users to infer changes using their semantic memory whenever visual short-term memory is not exploitable. For example, previously received emails can be remembered based on subject or sender.

Reduction Often the number of possible changes in a window is limited, making them predictable. Changes can also have privileged locations or salient visual features, making them faster to detect. For example, new emails usually appear in a predictable location and in bold font.

History Time is often part of visualizing information. Combined with persistence, time visualization provides histories that indicate the order in which changes occurred or their respective ages, when relevant. For example, emails are stacked in order of arrival and are time-stamped. Histories can be displayed statically or based on re-visitation.

Notification Some applications additionally maintain awareness by notifying users of meaningful changes as they occur. For example, users can be easily alerted each time a new email is received.

These semantic strategies are widely adopted in computer applications and seem to provide some support for the user to easily integrate dynamic visual information. However, they have some limitations:

Notifications (for an overview see McCrickard and Chewar [2006]) are only useful if they are time-critical or relevant to the current goal. Evidence shows that most users are not willing to be interrupted while absorbed in their primary task (for email notifications, see for example [Venolia et al., 2001]). Especially when large amounts of dynamic data are presented over a large area, as in a wall display monitoring task, users might not desire to be constantly interrupted by notifications, but may choose to proactively poll for changed information in different locations on the display.

Semantic structuring alone is not enough to support efficient integration of visual information. Using memory and visual search for spotting changes limits the amount of visual information that can be monitored. For example, the user will hardly remember an entire text document or a folder's content. Likewise, this strategy does not scale well to inherently continuous information: even the value of a single continuous variable displayed as a progress bar or a color can't be accurately memorized. Memorization and visual search become even more challenging in wall display environments where the amount of information that can be displayed at a given time is larger. Similarly reduction and history mechanisms are strongly dependent on the nature of displayed information and are always designed and implemented separately in each application, as discussed in Section 2.5.3. Recently Baudisch et al. [2006b] inspired by techniques from chronophotography and comic book transitions, used animation and abstraction to indicate changes in user interface elements. This approach assumes high-level system knowledge of the data presented and the types of dynamic changes likely to occur.

Because there is no general approach for supporting visual information integration, we can expect an uneven level of support among applications and the existence of flaws that make the detection of some changes difficult or impossible. Some system wide history support mechanisms that could potentially be used to identify hidden changes [Freeman and Fertig, 1995, Rekimoto, 1999, Renaud, 2000], assume the user revisits states already visited and thus rely heavily on navigation and search, rather than fluid review.

5.2 An image based approach

Although individual applications could be improved by paying even more attention to visualization, history or notification, it is worth considering simpler approaches that might work globally in an application agnostic manner.

Consider for example what happens when a number of new emails are downloaded while a) the mail client is minimized and subsequently reactivated for use, versus b) launching the mail client afresh. In the first case, the user is instantaneously shown a screen with new and old emails, and has to rely on visual searching for application provided cues such as highlights and dates to discern the new emails from the old. In the second case, newly downloaded emails appear one by one in the inbox and some folders are progressively bolded. No visual search and no semantic memory are involved because all changes simply pop out: motion perception is more powerful than most other strategies [Rensink, 2002] and requires nothing more than the actual visual changes to be presented over time.

Following the above example, we argue that a reasonable strategy to pursue would be an image based approach that stores invisible changes at the pixel level for later redisplay, or restitution, to provide the user with a simple visual understanding of changes that they may have missed. Such an approach assumes a pixel-level model of visual changes, which we introduce in the next section.

5.2.1 Surfaces and Pixel Visibility

Suppose that visual information made available to the user is spread among surfaces. A surface can be either physical or virtual, and contains dynamically rendered visual information. Whether they are physical or virtual, it is convenient to think about surfaces as unstructured matrices of pixels, which can be rendered off-screen as bitmap images. Although rendered and made available to the user, a pixel may or may not be actually

visible. There are two reasons why a pixel may be invisible to the user (Figure 5.1):

- *It is not displayed.* In layered graphical models, surfaces are allowed to overlap so that a pixel may be hidden by a surface of higher priority in terms of z-order. For example, all or part of a window's content can be obscured by another window (Figure 5.1a). In peephole graphical models such as virtual desktops, potentially very large surfaces are cropped so that only part of them is visible (Figure 5.1b), known as the keyhole effect [Woods and Watts, 1997]. Windows can also be minimized or temporarily hidden while the underlying application keeps running and the window still exists in the user's mental model and/or in the computer's memory (Figure 5.1c).
- *It is not seen by the user.* Even when physically displayed, a pixel can be obscured by a physical object, such as user's hand on a touch screen (Figure 5.1d) or a physical input device. But most of the time, a pixel is not seen simply because it is outside the user's field of view. In desktop computing, users are not continuously looking at the screen: a pixel can remain unseen during the time of an eye-blink, a brief talk with a colleague, or absence during the night (Figure 5.1e). When large and/or multiple displays are being used, a pixel might not be seen because the focus of the user's attention is on another part of the display (Figure 5.1f).

The first type of invisibility (the pixel is not shown) mainly concerns small displays and desktop computing, whereas it is less of an issue on large, wall and/or multiple displays, that allow more visual information to be shown concurrently. Conversely, the second type of invisibility (the pixel is not seen) occurs on small displays but is a more serious issue on wall displays where visual information resides outside the user's field of view, especially during close interaction, as is the case in our work.

Of course, pixel visibility is rarely a binary property. For example, in layered models supporting translucency such as see-through tools [Bier et al., 1993], a pixel can be

more or less shown. The concept of seen is even more subtle. First, human vision is physiologically non uniform and abilities such as acuity, color discrimination and movement sensitivity are unevenly distributed within the visual field [Baily, 1989], which is more or less covered by wall displays. Second, vision is a global perceptual phenomenon hardly decomposable into individual pixels and involving cognitive mechanisms such as attention [Rensink, 2002, Itti et al., 2005].

Though human vision is a complex phenomenon, we can make significant advances by adopting a minimalist model of visual perception that is purely optical (if something is visible to the eye, then it is perceived) and binary (we ignore partially visible information). Given that user's locus of attention is difficult to infer precisely even using specialized hardware [Hunt and Andkingstone, 2003], such an approximate model may be used as a practical substitute that captures a subset of possible use cases.

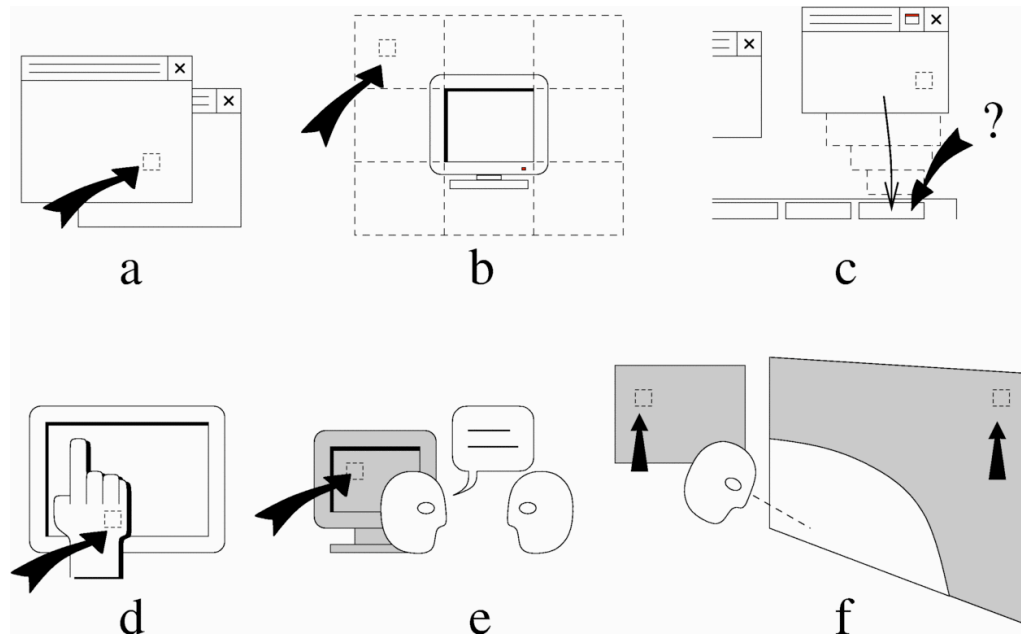


Figure 5.1: A pixel might be invisible to the user because: (a) of occlusion by another window, (b) it is on a virtual canvas outside the current display area, (c) its window is minimized; or it is displayed but not seen by the user due to: (d) physical occlusion, (e) interruption, (f) being outside the focus of attention.

5.3 Mnemonic Rendering

As the user navigates visual information, the visibility of each pixel is subject to change over time. On surfaces conveying dynamic visual information, the color of each pixel is also subject to change. Discontinuities may arise when changes occur on pixels while they are invisible.

Mnemonic rendering involves buffering pixel changes then restituting these changes on the screen. A visible pixel is rendered normally, whereas an invisible pixel stores a time-stamped history of its color changes, so that no transitional information is lost (Figure 5.2). When the pixel becomes visible again it restitutes its buffered history. After the restitution the pixel is displayed normally.

The goal of buffering and restitution is to aid users in maintaining an accurate mental representation of the state of visual displays, in the presence of invisible changes. This is achieved by storing and presenting the changes themselves in a manner that does not require memorization and comparison with previous states.

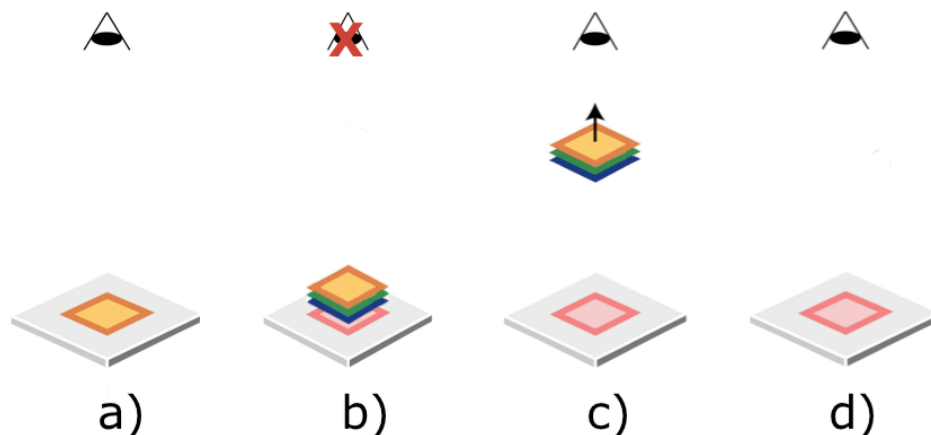


Figure 5.2: Mnemonic rendering. a) A pixel is visible and rendered normally. b) If the pixel becomes invisible, its visual changes get buffered. c) When the pixel becomes visible again, the stored visual changes get restituted or reviewed to the user, d) until they reach the current visual state of the pixel.

5.4 Mnemonic rendering design space

Although the idea behind mnemonic rendering is simple to understand, its design space is very rich. In this section we will discuss the rationale behind different design decisions used in the creation of a wall display prototype (Figure 5.3), the Mnemonic Wall. For details on a desktop prototype see [Bezerianos et al., 2006].



Figure 5.3: Mnemonic wall. A prototype of mnemonic rendering on a wall display.

5.4.1 Detecting visibility on wall displays

The two main components of mnemonic rendering are the process of buffering or storing invisible changes, and restituting or reviewing the stored information. In order to initiate both the buffering and restitution of visual information, there is a need to determine when pixels are visible.

Detecting whether a pixel is shown or not (see Figure 5.1a,b,c) is more relevant in desktop computer settings and can be easily done using software techniques. Detecting whether a pixel is seen or not (see Figure 5.1d,e,f), is more relevant in wall displays where

overlapping of content and layering is less common, and areas of the display often reside outside the user’s visual field. Determining if a pixel is seen or not on wall displays could be done using a variety of approaches, including:

- *Software-based inference*: on interactive systems, user absence can be coarsely inferred from periods of inactivity.
- *Presence detection*: absence can be inferred with better time accuracy using simple vision algorithms or sensors.
- *Head tracking*: tracking location and/or orientation of a user’s head allows inferring the user’s field of view, thus gaining spatial accuracy (for a comprehensive face tracking bibliography list see Price [2007]).
- *Eye-tracking*: eye tracking gives even more accurate spatial information about the user’s locus of attention [Duchowski, 2002].
- *Occlusion detection*: in some cases, physical occlusions can be inferred using software approaches (e.g., hand location on a touch screen). In other cases, occlusions can be accurately detected using additional hardware [Jaynes et al., 2001].

Non-intrusive hardware solutions exist for all of these approaches [Turk, 2004] and can be spread among displays. For example, displays could be aware of eye contact [Nakanishi et al., 2002, Vertegaal et al., 2005].

Although eye-tracking technology may seem appealing, coarser approaches for visibility detection might actually yield better results. They limit the Midas Touch problems [Jacob, 1993] and might provide the user with finer control over the restitutions when conservative inference strategies are used. Using head direction tracking for example (by head or face tracking [Park et al., 2002]), a conservative inference strategy, would not assume visibility inside the entire field of vision. Rather, it would add an area of invisibility around the user’s central vision. The way these invisible pixels are rendered

reflect tradeoffs between peripheral awareness and distraction reduction. For example, such pixels can display their last visible color, resulting in a non-distractive static rendering. Or actual pixel colors can be shown instead, resulting in a dynamic rendering serving peripheral detection. In our prototype, we decided on using head tracking for determining the visibility of pixels on the wall display. To ensure that no information is lost, we assumed that pixels in the area of invisibility around the user’s central vision are in fact invisible, and were buffered together with invisible pixels.

5.4.2 Restitution techniques

The second major component of mnemonic rendering is restitution. Restituting a pixel buffer consists of rendering it in a synthetic way during a limited period of time, possibly dismissing part of the stored information. Given that an entire color sequence for a pixel is stored, the two more obvious ways to display it is either expanded in space (displaying the entire history concurrently) or in time (presenting an animation of the history). We thus first describe two simple types of restitution: a static one called persistence and a dynamic one called flashback (Figure 5.4).

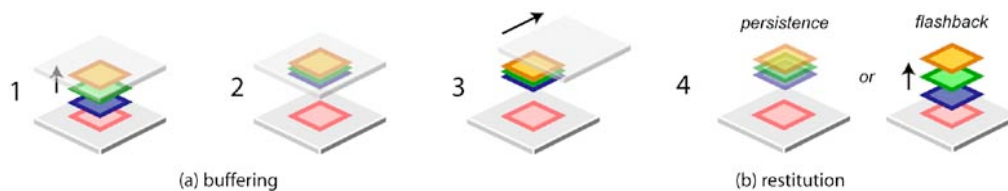


Figure 5.4: All changes occurring since a pixel is hidden (1) are buffered (2). Removing the occluding element (3), results in restitution of the entire buffered history, blended (persistence) or played back (flashback) (4).

Persistence When a pixel becomes visible it temporarily displays its buffered history as a single color obtained from blending all colors previously buffered. On a macroscopic scale, persistence typically results in motion trail effects, that resemble motion sequences

in comics [McCloud, 1994] and non-photorealistic rendering [Masuch, 1999].

Suppose that an entire surface is hidden, then shown after a change happened on this surface. When the visual change has a simple motion quality (e.g., a translation or a rotation) persistence produces a temporary semi-transparent trail showing the motion path (the moving gauge in Figure 5.5.a left). When the change is more complex (e.g., scrolling text or an animated icon), persistence blurs the areas where changes happened (the weather icons in Figure 5.5.a right).

Flashback A pixel that becomes visible plays back its history in a sped-up timeline. On a macroscopic scale, pixel flashback displays the previously visible state, then replays the stored changes in fast-forward motion until the present state is reached (Figure 5.5.b).

When adjacent pixels are made visible or invisible at different times, mnemonic rendering results in transient time-warping effects and visual breaks (Figure 5.6). The time-warping effect is produced by an animation that is progressively occluded or progressively shown (e.g., a window moves over another one). Visual breaks can arise, for example, if a part of a window is shown after a change happened on the entire window.

Animation has been used in user interfaces [Baecker and Small, 1990, Baecker et al., 1991, Stasko, 1993] for explaining visible changes, including smooth transitions based on cartoon or movie techniques [Chatty, 1992, Chatty and Beaudouin-Lafon, 1992, Chang and Ungar, 1993, Thomas and Calder, 2001, Baudisch et al., 2003b]. Instead of interpolating motion for better explaining discontinuous changes, our flashback approach restores visual continuity, occasionally disrupted by occlusion or attention shift.

Initial user response session

An early wall display prototype of mnemonic rendering was presented to a group of 10 users in an informal discussion session. Users really appreciated the idea behind mnemonic rendering and provided us with useful comments.

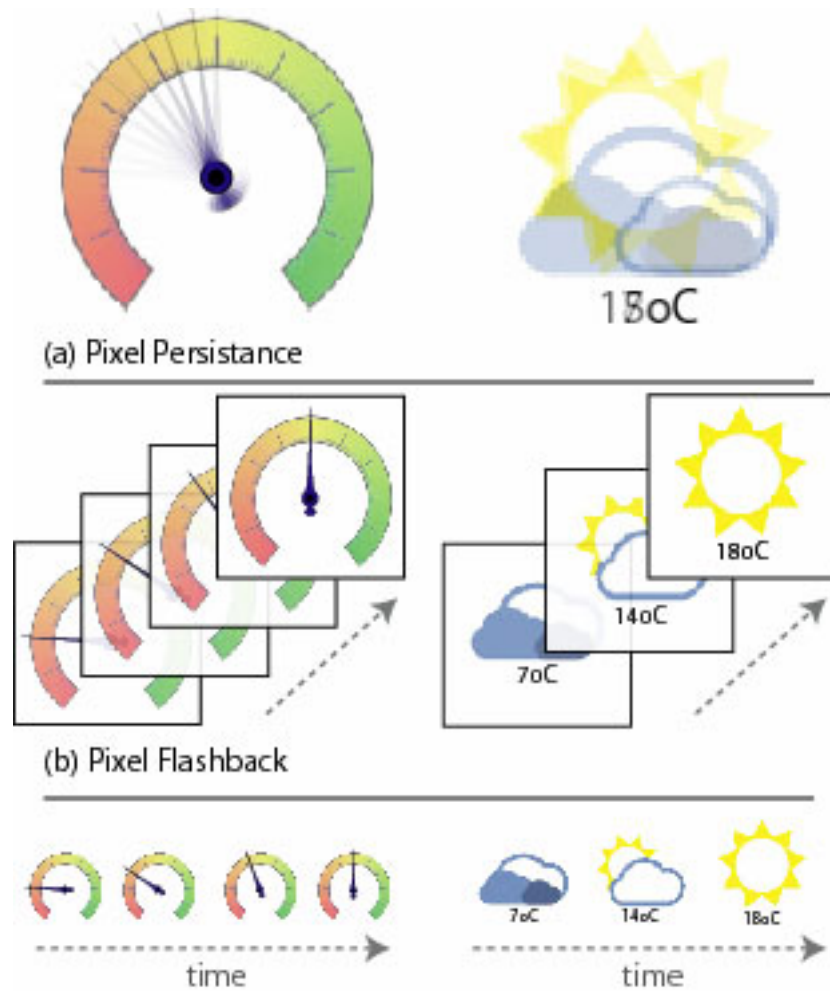


Figure 5.5: Two examples of using our restitution techniques. (a) Persistence, (b) Flashback. Notice how in persistence motion changes (gauge dial on the left) leave behind a motion trail, but in non-continuous changes a less informative blurring occurs (weather icons on the right).

Users understood easily both restitution techniques and commented on how they each provide the same information in two different ways. This led us to experiment with a third type of restitution, that we call *combination*, which is essentially a blend of persistence and flashback. In the combination restitution, invisible areas present a static component of their buffered history, and when they become visible again the animated component of their history is played back.

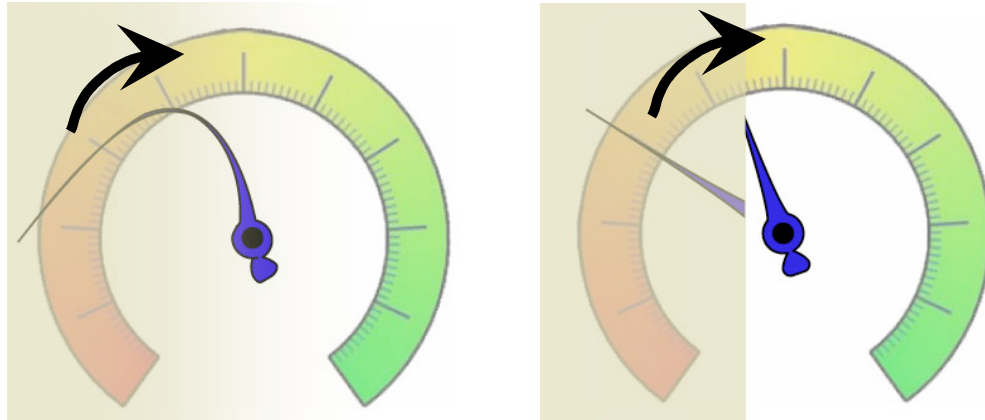


Figure 5.6: Assume a moving dial that is partially hidden. (Left) Time-warping with the flashback technique: The gauge is originally partly hidden (area of red pixels). As it is slowly uncovered (from right to left), pixels are gradually revealed and start their restitution at different times (red pixel saturation indicates buffered history left for restitution—redder pixels have more history to reconstitute). So pixels closer to the center of the gauge begin their restitution before pixels further away. (Right) Visual break with the flashback technique: The gauge is originally partly hidden (area of red pixels). It suddenly becomes entirely visible again. The previously hidden pixels are "out of sync" with the pixels that were visible. In both cases red shading conveys pixel age.

Other types of restitutions were also discussed, such as highlighting changed areas (similarly to [Matthews et al., 2006]) or using some form of clarification [Thomas and Calder, 2001], or summarization, explanation and abstraction [Baudisch et al., 2006b]. Highlighting changed areas, although can be easily incorporated in the mnemonic rendering design, does not really provide a change transition from previously viewed to current visual states. Explanation and abstraction, although may serve the same purpose as mnemonic rendering, are based on the assumption that the data manipulated is of a specific type (for example application icons, sliders or radio buttons, etc) and that the potential changes they may go through are well defined. On the other hand, mnemonic

rendering makes no assumption as to the nature of the changed data.

An issue raised by the group was the rendering of the buffered history. Users indicated that it was not always clear which rendering was part of the history and which was part of the current state of the display. This lead us to further explore different visual effects for displaying the restituted history.

Finally, our group noticed that some of the changes that might take place in invisible areas may be complex and hard to understand from a brief animation. To that effect we decided to add the option to revisit restituted history in mnemonic rendering.

Based on the comments from our user group we further explored the design space of mnemonic rendering. Persistence and flashback can be implemented in several ways and each small variation can result in different visual effects. Second, there are several ways to make the technique more sophisticated and improve its scalability. In the next sections, we go into more details on such design variations.

5.4.3 Visual Effects

There are several ways of implementing the persistence technique. The blended color can be obtained by averaging all colors present in the history, which amounts to alpha-blending each pixel with the previous ones using alpha values that follow a geometric progression of scale factor $-\frac{1}{2}$. Other averaging weights can be used, as well as other color composition schemes. For example, having weights depending on timestamps can produce fading-out or fading-in trails (Figure 5.7). Simple pixel superimposition can be used instead of alpha-blending when pixels already include an alpha channel. Objects moving on a transparent background would then leave sharp trails. Other compositing techniques can be used for preserving more information, such as multi-blending [Baudisch and Gutwin, 2004].

Color manipulation in all restitution techniques may also convey status information

on a pixel (invisible, restituting or normally shown). For example, applying de-saturation or sepia effects during restitution gives a feel of oldness (see Figure 5.7), while preserving most of the visual information (similarly to the use of color transitions to indicate time by Woodring and Shen [2003]). Such an effect can fade out throughout the restitution, enhancing the film metaphor while allowing smooth transition to the current visual state.

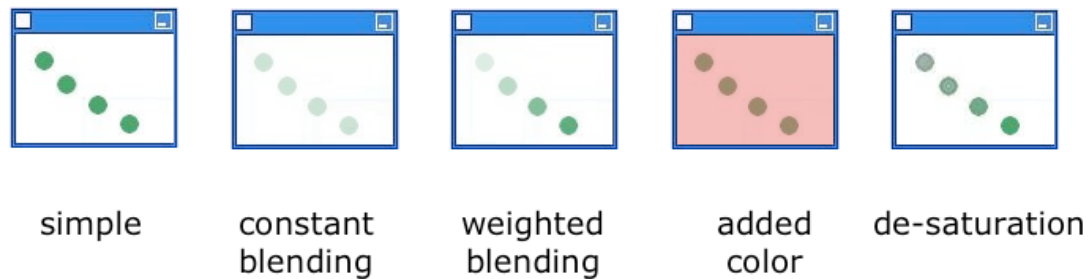


Figure 5.7: Different visual effects for rendering buffered history. Restitution of a moving ball. *Simple* rendering of all buffered history at full opacity. *Constant blending* provides a constant transparent rendering of previous history states, while *weighted blending* is based on history timestamps and results in fading-in trails. Adding *color* may also indicate the areas that are currently displaying buffered information, while *de-saturation* communicates the feeling of oldness in buffered history.

If time between glances is short or old data is unimportant, a transient variant of persistence can be obtained by adding temporal response to color changes, similar to phosphor persistence of old monitors. Thus older stored information will slowly fade out as time passes, even if it does not become visible. On oscilloscopes for example, such effect is sought after for viewing complex data.

On a wall sized display where a large amount of information is visible at any given time, extra clutter should be avoided to minimize user effort in discerning what information is part of a restitution, and what is actual current data. To that effect, we decided to render previous history states are always as semi-transparent, with de-saturation, in order to clearly distinguish them from current rendered data.

5.4.4 Timing

Restitution length can be variable (e.g., it can depend on the invisibility duration), but must be short enough to give the user a rapid glimpse of the changes [Stasko, 1993]. This is especially true if surfaces are interactive, because the user may want to quickly resume interaction with the application. Typical lengths for GUI animations (i.e., an opening menu) are below one second. So, restitutions must similarly be achievable in a small bounded time.

With the persistence restitution, the averaged pixel color rapidly fades out to reveal the actual pixel color. With the flashback restitution playback can be compressed linearly to achieve bounded restitution time. Frames can be dropped or blended with adjacent frames (a technique known as motion blur). The simplest non-linear time compression technique involves playing color changes successively without taking their timestamp into account. But this technique can desynchronize adjacent pixels, unless grouping is used (see the section on grouping below).

Another timing issue is handling pixel changes during restitution. If restitutions are fast compared to the speed of the actual changes, these pixel changes can be ignored. The techniques can also be easily extended to support concurrent buffering and playing. For example, updating the playback sequence and speed as new pixels are being buffered will ensure continuity with animations currently occurring.

5.4.5 Ordering

The order in which stored frames are restored or dismissed greatly affects the overall visual effect of the restitution.

In the persistence restitution for example, when pixels become visible they can all fade away concurrently. So if a motion trail is visible on the display it will slowly fade out at the same pace. On the other hand, a timestamped fading can be performed. Frames may

then fade out based on their seniority. Thus a motion trail will start fading out close to the start position of the trail and progressively reach the final position. This timestamped fading gives an "animation" feel to persistence, which is very different from the flashback restitution. Here all history states are visible, whereas in the flashback restitution only the state being played back is rendered. And there is no notion of highlighting history states one by one, as is the case in the progressive flashback animation.

In the flashback restitution the playback ordering is even more important. Imagine that one part of the wall display became invisible to the user. Depending on the ordering of restitution we can: a) leave the state of the invisible area exactly as it was the last time the user encountered it. When the area becomes visible again we display the buffered history forward (starting from the oldest stored history state) until we reach the current visual state of the area. b) We may render normally the area as it is being changed. When the area becomes visible again, we play the stored history backwards, starting from the current visual state until we reach the oldest stored history state. Finally c), we may render normally the invisible area as it is being changed, and when it becomes visible again go to the oldest stored visual change and move forward until the current visual change is reached again.

There are several drawbacks and advantages in either ordering of the flashback restitution. When the last visual state is persistently rendered, as the invisible area becomes visible again, restitution forms a continuous and fluid animation path that leads to the current visual state. But, when restitution begins, there is no information as to what the current visual state is until the playback finishes. In the other two options, the current visual state of the area is immediately apparent to the user. When restitution goes backwards in time from the current visual state, it eventually reaches the oldest stored history. At that point a visual break takes place as the state jumps from the last restituted state to the current visual state again. Another issue that may arise in this case is that if changes continue to happen during restitution, when the first restitution ends, the

area will jump to a completely new state (current visual state) and go backwards again to play the newly stored restitution. When restitution goes forward, we have an initial visual jump to the oldest stored state and start moving forwards until the current visual state is reached.

In a wall display, where a large number of changes might take place at any given time over extended distances, it is essential that the behavior of restitution is predictable. Imagine for example an icon that has moved while invisible in a new location, and the moment the user looks at it, it jumps to its previous location and starts moving. These types of sudden visual jumps may cause disorientation and confuse users. Thus any designs with visual breaks violate our predictability design goal. Based on this observation, for our wall display prototype in the flashback restitution, we decided to render the last seen visual state of an area, and play forwards its buffered history when the area becomes visible again.

5.4.6 Grouping

The time-warping effect mentioned earlier (Figure 5.6) provides a spatiotemporal visualization of the changes together with invisibility periods. However, such an effect might puzzle the user and visual breaks can hinder her understanding of more global changes.

There are several approaches for ensuring some level of synchronization. First, changes in pixel visibility can be deferred so that they occur at the same time. Delaying can rely on clock ticks or on user actions. For example, pixels won't start recording or be restituted before the user has finished dragging an occluding window. This type of solution is not well suited for wall display environments, where invisibility does not often occur due to explicit user interaction (for example moving windows around), but rather to more implicit types of actions (like moving one's head).

Another approach for ensuring synchronization is grouping pixels so that their visibilities are all updated at the same time. Several grouping strategies can be considered:

- *Surface-wide grouping*: If whole surfaces are seen as groups, then their content will remain visually consistent.
- *Dirty region-based grouping*: For performance concerns, most graphical toolkits repaint only rectangular regions that have changed. Although semantic, this information is very often available at a low-level on the system. Dirty regions often match fixed areas on the surface, but can also be parceled out and follow objects as they move. In that case, global motion groups can be obtained by iteratively accumulating dirty regions that intersect on two successive animation frames.
- *Image-based grouping*: More accurate motion groups can be directly derived from the images. One minimalist approach is drawing pixels that changed and grouping them based on proximity. More sophisticated motion segmentation techniques have been proposed in various fields, including vision and image compression [Salembier and Marques, 1999].

Once a grouping strategy has been adopted, several approaches can be considered for triggering the groups, i.e., starting the restitutions by marking all pixels visible:

- *Triggering on full visibility*: a group is triggered when it is fully visible. One issue with this approach is that it may be hard or impossible to make a group fully visible, for example if the group runs over the user's field of view, as might be the case in wall displays.
- *Triggering on partial visibility*: a group is triggered as soon as it is partially visible. Triggering is easy, but may lead to missed changes, for example if the group is partly occluded or runs over the user's field of view.
- *Incremental triggering*: triggering can be done on the first motion group, then on the second one and so on. For example, if an object has moved around the user

on a circular display, the user will have to look at the starting point and follow it until the end.

In wall displays potential invisible areas might be very large and users may have a hard time predicting which parts of the display are likely to have changed. This will often result in unexpected restitutions when the user looks at a changed area. Although this cannot be avoided, when coupled with visual breaks it may confuse users and make the restitution hard to understand. Furthermore, even with similar changes, their resulting restitution visual breaks may be radically different depending for example on the direction of movement of the user's head. To eliminate the confusion and to ensure consistency in our technique, we decided on using image based grouping in our wall display prototype.

Triggering in a wall display may be impossible to perform on full visibility, since motion groups might not be fully visible at any given time if they span the user's visual field. Furthermore, incremental triggering on large surfaces may be time consuming. Imagine for example an icon that has moved back and forth along the length of the wall display and has left a somewhat complicated motion trail. It may then be hard for the user to focus on the starting point of the motion trail and follow it through. For these reasons, in our prototype we trigger group restitution based on partial visibility.

5.4.7 Interactivity

Interaction with mnemonic rendering is most of the time passive in that it exclusively involves changes in pixel visibilities, implicitly controlled by the user (by moving her head). However, adding active interaction can be useful. For example, when histories are lengthy or complex, the user might prefer to have some control on the restitution, rather than be provided with longer playback times. We briefly describe an example of a lightweight mechanism that allows flexible navigation in the history of changes, which assumes the presence of a pointing device. This technique can be applied to any type of restitution, for replaying a flashback or restoring persistence.

During the restitution and for an extra second afterwards, a semi-transparent disc is visible. Before the disc fades out, the user can indicate interest in the restitution by depressing the pointer, in which case the disc changes into a circular representation of the restitution timeline (Figure 5.8). Restitution is controlled with circular eyes-free gestures, allowing both coarse and fine control. This is useful when history lengths differ by several orders of magnitude [Smith and schraefel, 2004, Moscovich and Hughes, 2004].

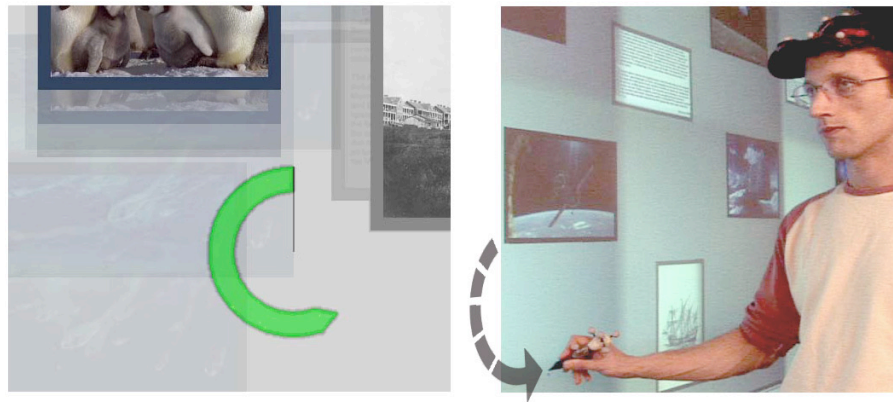


Figure 5.8: Control dial operated remotely by the user to control restitution playback in a wall display. The visual representation of the dial is rendered in the location the user is focusing on (left), while interaction takes place close to the user (right).

Feedback and input locations depend on the constraints of the particular interactive system. On a desktop computer, the input area and the disc feedback can be confounded and displayed at the mouse location when the restitution started, thus being both easy to trigger and to ignore. On a wall display, the input area should be close to the user whereas the disc feedback might be displayed elsewhere within the field of view. Thus for our wall display prototype, the visual representation of the dial is located at the position where the user is currently viewing the restitution, while input for the dial takes place close to the user. Thus user effort is minimized, since small arm movements are needed to interact with the dial, and the effect of the interaction is visible at the location the user is currently focused on.

5.5 The Mnemonic Wall

The Mnemonic Wall is a document organizer running on a wall display (Figure 5.9). Pictures and text boxes are spread out over the display and can be moved around using a pen, or are animated by the system to simulate a distant user. The Mnemonic Wall also includes a variant of the layout-matching game discussed in the user feedback section 5.6.

The entire wall display is treated as a single surface. Head-tracking using a Vicon (www.vicon.com) motion tracking system is used to determine pixel visibilities. The visible area is visualized by a red translucent quadrilateral, whose surface is chosen to encompass foveal and parafoveal vision (Figure 5.9). Since head movement and eye amplitudes during gaze shifts follow a tight linear relationship [Wang and Jin, 2001], we exaggerate horizontal and vertical head movement values to allow natural head movements.

Three restitutions are supported: persistence, flashback, and a combination of both. Pixels from the surface have alpha channels (some are transparent) and the combined technique shows persistence and flashback simultaneously. Also, a motion-based pixel grouping has been adopted and groups are triggered based on their partial visibility. Restitutions can be controlled with a dial.

5.6 User feedback

Mnemonic rendering involves a number of potentially disturbing effects such as overlapping graphics, delayed animations and uncommon use of context information. In order to make sure the techniques were viable, i.e. that at least for one particular type of task they would prove beneficial, we conducted an informal evaluation of the techniques using a layout-matching game. Our goal was to understand how users would utilize the three restitution techniques, to determine how effective or hindering they would find them, and to elicit comments on their design.



Figure 5.9: The Mnemonic Wall with pixel persistence: visual changes outside the user’s central vision leave motion trails that will fade out as the user glances at them. The user is wearing a hat with passive markers mounted. On the top of the image infrared Vicon motion capturing cameras can be seen.

Task 6 participants (1 female) played a series of games using regular rendering and the three mnemonic rendering techniques (persistence, flashback, persistence/flashback combination). Each session lasted about half an hour.

The layout-matching game showed four large grids at the four corners of the display, containing numbers. Periodically, a number was randomly moved to a free location by the system. These four grids were related to another four small number grids close to the user’s location in the middle of the display, whose numbers could be moved by the user (Figure 5.10). The task was to reorder the small interactive grids so that they match the large dynamic ones before a timer ran out. Given the display’s size, while working on one grid, the two opposite grids could not be seen without major body movement. No control dial was used.

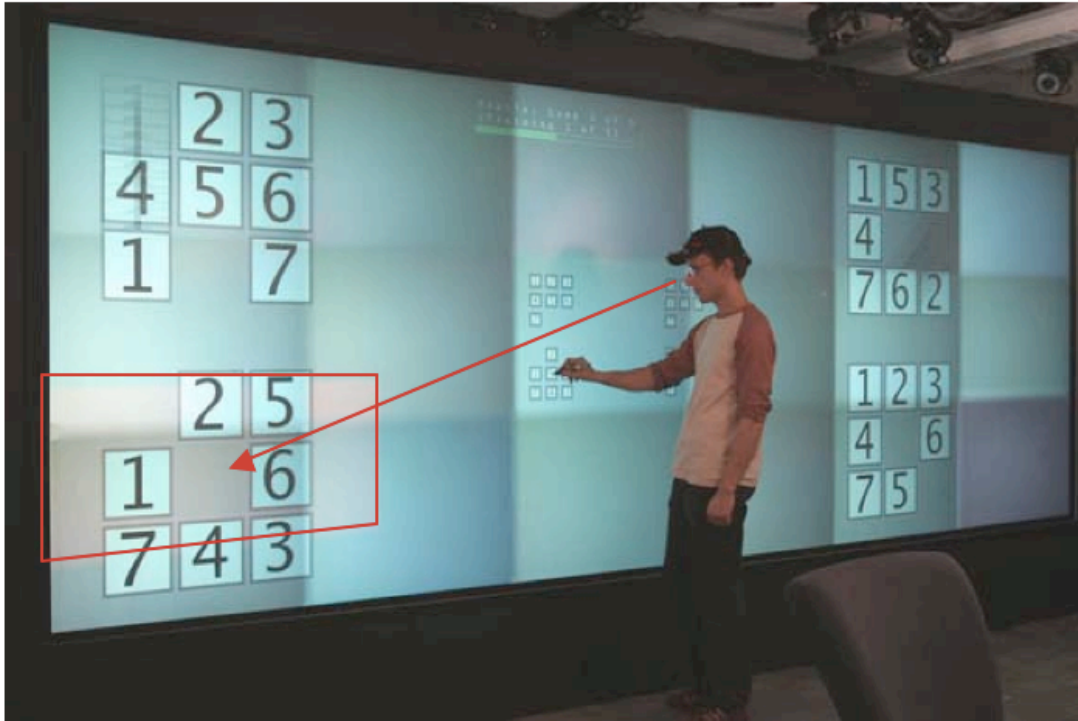


Figure 5.10: Layout matching game on the Mnemonic Wall, using pixel persistence. The red lines and arrow are there to emphasize our approximation of the user’s visual focus based on head tracking.

The reasons behind the informal nature of this evaluation vary: first mnemonic rendering does not provide an explicit notification mechanism, thus any performance measures will largely depend on the design of a notification component, making unclear the effectiveness of the restitution itself. To examine the explanation value of mnemonic rendering we would need a task that cannot be performed when changes are missed. This type of task is largely artificial, as when changes are indeed critical some support is provided by existing applications, while in most computer environments tasks can be performed, albeit more difficultly, when changes are missed. Mnemonic rendering is targeted mainly at the latter type of task, but may also improve the former through the detection of visibility of information. Thus more formal evaluations of mnemonic rendering would require it to be incorporated in a fully designed system. And even in such an evaluation,

results would greatly depend on the nature of the monitored information. It is thus unclear how to evaluate mnemonic rendering removed from a given system and monitored information. Incorporating mnemonic rendering to a system and evaluating its use under different monitored information is a clear direction in future research. At this initial design exploration we focused on acquiring general purpose insights on mnemonic rendering by using a toy task.

We wanted the task to differ from common computing activities in two respects. Firstly, a faster pace was required to evaluate the techniques in a reasonable time. Secondly, we wanted an engaging, well-defined task, and thus avoided concurrent tasks with unclear priorities. The layout-matching game involves a single primary task with a well-defined goal: match all the layouts as fast as possible. Nevertheless, the task shares salient characteristics with personal computing activities: it requires glancing for polling and monitoring information, and while seeing changes as they happen might be useful, the task can still be completed if changes are missed. Completing this task when all changes have been missed is easier than real-life search for changes, since it involves visual search alone (side-by-side comparison of two grids) instead of visual search plus semantic memory. To ensure participant familiarity with this side-by-side comparison strategy, non-mnemonic rendering was presented first. Hence, participants could ignore mnemonic rendering information if they found it unclear or useless.

5.6.1 Results

Relevance of mnemonic rendering. Participants easily understood the task. As expected, all participants used a strategy of matching individual grids sequentially. We counted the number of changes users had to treat at each step. Although this was a function of user's speed, values from none to three were the most common. All users made use of mnemonic rendering for reproducing changes. After playing the game, they also ranked all three restitution techniques higher than regular rendering that required

side-by-side comparison. They seem unaffected by Midas Touch, so using head tracking for passively controlling visual rendering seems realistic.

Usefulness and limits of Persistence With the persistence restitution, numbers moving outside parafoveal vision left a semi-transparent echo or trail behind them. Participants found the technique useful when one or two hidden changes happened, but found it challenging for more than two overlapping changes. However, they used the technique for quickly determining whether any changes happened on a grid, allowing them to spend less time on unchanged grids. Apart from proactively scanning for these areas, users commented that they made use of persistence passively, through peripheral vision, as an alert mechanism for changing areas on the screen.

Clearly persistence failed as an explanation mechanism for complex changes, but provided useful synthetic information on the changes that could be quickly interpreted.

Usefulness and limits of Flashback. With the flashback restitution, the numbers stood still if they were occluded or unseen but moved in fast forward motion as soon as the user glanced at them, if changes happened. All participants described the flashback technique as being very effective. They noted that after looking at the playback of changes they could replicate the sequence easily. Sometimes users could successfully mimic the motions even when the animation was partly missed or apparently chaotic, with up to four successive motions in one second. More than four motions could hardly be reproduced.

However, users commented that the lack of feedback in the presence of changes before the actual animations were played back prevented them from using peripheral vision. Some users required an indication as to whether the motion they were viewing was that of an actual change or a flashback.

Clearly the flashback visualization we used, although effective in explaining changes, lacks feedback and does not support peripheral awareness.

Persistence and flashback combined With the persistence-flashback restitution, trails were shown then faded out during the flashback. Unsurprisingly, the majority of participants preferred the combined technique. As they pointed out, the blurred trails left by persistence both clearly provided rapid information on changes and supported peripheral awareness, whereas the flashback component was useful in explaining in detail the actual changes. One participant felt distracted by the persistence in the combination condition.

Design Considerations Several design-related issues were raised. They mainly concerned the proper choice of visual effects and timing:

Visual effects. When performing the task with the persistence technique, several users commented positively on the use of de-saturation as a metaphor to indicate older states. Nevertheless, they requested a fainter representation of older states, so as not to clutter the screen. Three users commented that the intermediate history need not be too detailed, as long as some transitional information is present.

Timing. One design aspect that particularly concerned us was the timing of the restitution. We found the chosen 1 second to be appropriate for displaying one to three changes without hindering the task. Some participants found it occasionally too slow, while everyone found it too fast for more than four changes. One participant said that he would have liked the ability to replay the last restitution. We feel we have addressed this issue by introducing the interaction dial, a functionality we did not provide in the game task.

5.7 Chapter summary and closing remarks

In our increasingly dynamic computer environments visual changes often occur without being seen. After identifying the situations under which changes are missed, we introduced the idea of mnemonic rendering to present these missed changes in a lightweight

and fluid manner. Mnemonic rendering ensures that users need not memorize the visual state of a dynamic surface, by storing all changes in invisible locations. Thus it reduced user effort while interacting with a wall display. Variations of the design space were discussed and three visualization techniques developed. A wall display prototype was created, and design decisions on the implementation were such as to maximize predictability and consistency.

Mnemonic rendering is application agnostic and may benefit a variety of applications and tasks, a theme we have followed throughout this thesis. Nevertheless, seamless integration of generic and dedicated techniques for supporting change awareness requires further investigation. In addition to the basic system-wide support provided by mnemonic rendering, individual applications could further exploit information on pixel visibility to better handle specific types of changes.

Since mnemonic rendering is image-based, it can be currently implemented at the window manager level. A new trend in window management is compositing, a technique that allows windows to be rendered off-screen and manipulated as images [Chapuis and Roussel, 2005]. While its current application lies mainly in cosmetic effects, mnemonic rendering suggests a new compelling use of this graphics power. Moreover, video summarization and key-frame extraction techniques [DeMenthon et al., 1998, Cooper and Foote, 2002, Assa et al., 2005] could be incorporated in the restitution component of mnemonic rendering to both enhance explanations and achieve data compression.

The assumption in our technique is that the tasks performed by the users are not time-critical and thus notification mechanisms are not provided. Although our techniques are aimed at explaining hidden changes, they were not designed to explicitly provide notification of the occurrence of the changes. Clearly the two issues are interrelated and also require further investigation, since some of the monitoring tasks that may be performed in large displays could be time critical. A combination of mnemonic rendering with notification mechanism could significantly enhance its applicability.

Chapter 6

Conclusions and future directions

Wall displays present unique interaction opportunities because they can display vast amounts of data, and can both be viewed and interacted upon at close proximity. As they become a commercial reality, interaction challenges arising from these unique aspects need to be examined. This thesis focused on three aspects that may hinder wall display interaction at close proximity. First, challenges in organizing content and switching context across an extended interaction surface. Second, issues related to reaching across large distances to perform simple selection tasks. And third, problems arising while attempting to keeping track of potential visual changes across a surface that both extends across a large area and can simultaneously display large amounts of data.

Attempting to explore these interaction challenges we identified a set of desirable properties of interaction techniques for wall displays. We discussed the need for general purpose solutions and designs, given that the field is still in its infancy and future uses of the technology might greatly differ from current practices. Moreover, we advocated the need to minimize user effort while interacting within large distances and with visual content spread over extended areas, as well as the need for predicable and consistent designs. These properties were considered in three design explorations of new interaction techniques for wall displays, that focus on the identified interaction challenges.

In the first design exploration we introduced the canvas portals framework that encompasses existing context switching techniques applicable to wall displays, and acts as a catalyst for the development of new ones. In the most abstract sense, canvas portals are alternative views of display areas where interacting with the portal’s interior is equivalent to interacting with the depicted display area. Manipulating the various canvas portal parameters (for example the mapping between the canvas portals and the depicted area of the underlying display) lets us support a variety of novel interactions, of a somewhat persistent nature, that can serve as context switching and comparison mechanisms, as well as layout management aids.

To illustrate the generality of the canvas portals concept, we instantiated a representative sample of existing large display interaction techniques within its design space, and developed three new ones that address some of the challenges in new ways.

One of the newly introduced techniques, the ScaleView portal, was evaluated through user feedback sessions, and a usability study involving layout and comparison tasks that incorporated frequent context switching. Our evaluations allowed us to refine the design of ScaleView portals. We improved the mechanisms of focusing the portal to a remote location, as well as the visualizations that reveal the technique’s functionality, such as the potential to move content across portals.

Apart from these valuable design improvements, the usability session provided us with general insights as to the use of wall displays. We found that for the challenging layout and comparison tasks of our evaluation, users preferred the ScaleView portal (a fairly complex technique) to simpler techniques (such as a magnification lens) that did not adequately support them in their tasks. A group of users always performed tasks using ScaleView portals alone, because, as they mentioned, they found compelling the ability to have everything concentrated close to their position and minimize physical effort. Several participants spent a fair percentage of time using the large canvas as a peripheral reference to their tasks, by periodically glancing at remote locations, highlighting one of

the benefits of using a wall sized display: users may visually refer to the large, spread out content on the wall display, even if they prefer to interact with it close to their location.

The second exploration focused on remote reaching techniques that are likely to be invoked frequently, but will not be of a persistent nature, as the canvas portal techniques are. First we designed and evaluated, through a pilot session, new interaction designs to enable quick access to items on areas of a wall display that are difficult for users to reach without significant physical movement. Based on observations and findings of our evaluation session, apart from the original design goals outlined in Section 2.6, three new desirable design principles were identified: the need for transparent, transient and flexible designs.

Following the augmented set of design guidelines, one of the newly proposed techniques, the vacuum, was further refined. The new design allows for transparent invocation that does not distract from regular interaction, as it resembles the action of selecting a remote item without a mediator; moreover, the technique is fluid and transient as it is invoked and dismissed with simple pen-down, pen-up events; and finally it exhibits flexibility by allowing on the fly parameter adjustments and by supporting more complicated interactions than simple target selection. Results from a single target selection study in the presence of distracter items, showed the newly designed vacuum to perform similarly to direct picking and drag-and-pick [Baudisch et al., 2003a] in terms of time and errors, except when very few distracter targets were located along the path. However, the redesigned vacuum scales nicely to multiple target selection, and is faster and less error prone than drag-and-pick.

Managing large amounts of dynamic visual information on wall displays, common in monitoring type tasks, involves understanding changes happening out of the user's sight. Our last design exploration showed how current software does not adequately support users in this task, even in desktop settings, and motivated the need for a more general

approach. We proposed an image-based storage, visualization, and implicit interaction paradigm called mnemonic rendering, that provides better support for handling visual changes. Mnemonic rendering consists of buffering dynamic visual information on the pixel level while the pixel is not visible to the user. Once implemented on a system, mnemonic rendering techniques can benefit all applications. We discuss different methods of detecting visibility and explore in detail the rich design space of the technique.

Based on user feedback on a wall display prototype that uses head tracking to detect pixel visibility, benefits as well as limitations of the technique were identified. While performing a layout matching game using mnemonic rendering, users seemed unaffected by Midas Touch effects, so our decision to use head tracking for passively controlling visual rendering seems realistic. Persistence, a static representation of the buffered history, was used passively, through peripheral vision, as an alert mechanism for changing areas on the screen. Nevertheless, users felt it did not adequately explain visual changes of a complex nature. According to participants, flashback, an animated rendering of the buffered history, was effective in explaining hidden changes, but lacked feedback on the location of these changes and did not support peripheral awareness. Thus a combination of the two renderings was used, which as users pointed out, both clearly provided rapid information on the location of the through the persistence component, and explained in detail the actual changes through the flashback component.

6.1 Combining the three designs

The three exploration designs are presented in the thesis as independent techniques, nevertheless in real use they coexist and interact. In this section we will describe how the three designs can be combined together in a single interaction environment and discuss potential issues or benefits derived from using them in conjunction.

6.1.1 Canvas portals and vacuum

In our current implementation canvas portals and vacuum instances coexist and may be invoked concurrently. For example a canvas portal with a zoomed-out attribute maybe focusing on a location where a vacuum is being invoked. Thus a "miniature" vacuum can be seen inside the portal. As interaction within the portal is equivalent to interaction with the canvas in general, the user can manipulate the "miniature" vacuum from within the portal if they wish. Similarly, a user may "vacuum" a remote canvas portal, bring it at a proximal location, by using the vacuum.

Given this behavior second order effects may occur. By second order effects we refer to events initiated through a tool that affect the tool itself. For example imagine a portal located at the left corner of the display focusing at the right end of the display. Invoking a vacuum through this portal initiates the vacuum at the left end of the display, at the area of focus of the portal. If the arc of influence of the vacuum is towards the left area of the display, the vacuum might pull, among other items, the portal itself and move it to the left area of the display, jumping away from the user. In our implementation these effects are dealt with through prioritizing our designs. All our techniques are implemented as filters that work on top of regular system behavior. Canvas portals are given higher priority than the vacuum. Thus a new input event is first checked against existing canvas portals. If the event takes place within a portal, the specific portal (and portals belonging to the same group) becomes immediately "invisible" to other filters, and thus actions of other techniques (like the vacuum) do not affect it. If events are not initiated or affecting a portal, then it is treated as a regular interaction element and so it can be vacuumed.

Allowing the coexistence of portals and vacuums provides flexibility and allows users to combine them in interesting ways. For example users can briefly identify targets of interest with the vacuum and might use portals to focus on specific locations they interact often with. Portals can help users explore in detail the display space and then use the vacuum for quick access actions. If a user finds the size of some vacuumed items

too small, she can have a magnification portal close to her location in order to invoke the vacuum through it (acting as a vacuum magnification lens). A user that relocates physically to a new location on the display may pull her portals to her using the vacuum.

Apart from existing connections of these two designs others can be implicitly designed to leverage user familiarity with the techniques. For example the magnification of a portal could act as a mechanism for resizing the scale of vacuumed items across invocations. Or the quick and fluid functionality of the vacuum could be used as a focusing mechanism for creating a new portal, as was suggested by a participant who was shown both designs after a study session.

6.1.2 Canvas portals, vacuum and mnemonic rendering

Combining mnemonic rendering with the other two designs is a somewhat more complicated procedure, as it involves the notion of visibility. An area on the display can be assumed to be visible either directly or through a mediator (for example a canvas portal or the vacuum). We will be discussing the effect of portals and mnemonic rendering, but the points raised are also applicable in the use of mnemonic rendering and the vacuum, if the area of influence of the vacuum is assumed to define a "view" of the display.

In our current implementation we ignore mediated visibility. Thus even if an area is viewed by a user through a portal, it is deemed invisible and its changes are buffered. This design decision was made because the existence of the portal does not necessarily provide an adequate view of the remote focal area. For example if the portal is a zoomed-out view of the display, the fact that the entire display is visible inside the portal does not guarantee that the resolution of potential changes is such that users will be able to perceive them in the scaled down view.

By disregarding mediated visibility we are being conservative in buffering and displaying changes. For example imagine a portal with a dynamic remote focal area. If the user is ignoring both the portal and focal area for a time, changes are buffered through

mnemonic rendering in both locations. When she returns to the portal she will see the changes being restituted inside it, and when she focuses on the remote focal area the changes will be restituted there as well. If a user is looking through a portal to a remote focal area where changes are taking place, the changes are buffered and will be shown to her again when she later looks at the remote area. Finally if the user is paying attention to a dynamic area of the screen that happens to be depicted in a portal, the portal pixels actually buffer these changes. When the user returns to the portal, she will briefly see the changes taking place there as well.

An interesting side-effect of this behavior is that user actions get buffered themselves. For example moving an item inside a portal causes the item in the remote location to move and thus this change is being buffered. In this instance, when the user looks again at the remote location she is briefly reminded of all the actions she performed from within the portal. Similarly, if a user interacts with an area of the display depicted in a portal, when she returns to the portal she is briefly shown her changes again. Apart from smoothly synchronizing the last observed state of the portal to the current state of the display at the remote focal area, this restitution may also reinforce the connection between the portal and remote focal area, as the actions in the remote focal area are later seen to affect the portal. Unfortunately, when users move items between portals, in the remote focal areas items actually disappear from their previous locations. Thus any buffered history at those locations results in sudden appearances or disappearances of items. We believe this issue can be alleviated by incorporating a small "fading" or "shrinking" animation to any item crossing the borders of portals. Such an animation could inform users in restitutions (but also any other observers of the display), that the item has been "teleported" to a new location through a portal. This is an aspect we intend to examine further.

Through our conservative buffering and restitution we hope that users will not miss any changes, at the expense of occasionally restituting changes the users might actually

be aware of. We hope that the short restitution time (1 sec) is small enough as not to disturb users, but clearly more formal investigation is need.

6.2 Contributions

Taken individually, each exploration contributes significantly to its domain in the specific aspects it investigates.

The first exploration identified how different types of layout and context switching tasks could be performed using interactive alternative views. The idea of passing content across views in order to bridge distances also appeared elsewhere in literature [Khan et al., 2004]. The development of the canvas portal framework was a contribution in itself, as it allowed us to identify potential areas of improvement in existing alternative view techniques when applied to wall displays, and provided a systematic way for creating three novel interaction designs. One of these designs, the ScaleView portal was evaluated through an informal feedback session, pilot studies and a usability session. The last usability session in particular is novel in several respects: first of all the tasks performed resemble tasks that are likely to be performed on wall displays; secondly, it indicated that complex behavior in a wall display technique is often preferred to simple functionality that requires extended motor effort; moreover, general trends of differences in how different users use the display visually were identified, by tracking the visual focus of the user; and finally the interaction patterns were graphically presented for the different tasks, indicating how the display was viewed differently in each one of them by different users.

The second design exercise most importantly contributed the vacuum, a fluid and lightweight remote reaching technique that allows for complicated functionality when needed. In finalizing this design, several intermediate benefits were achieved. First of all, a set of alternative interaction designs was proposed and their relative strengths and weaknesses were discussed. Although only one of these designs was refined further, there

is a potential for extending the others to eliminate their weaknesses and take advantage of their strengths. In the process of redesigning the vacuum, we derived several new design guidelines for remote reaching techniques. These, while followed in redesigning the vacuum, may also benefit and inform any future remote reaching design for wall displays. Finally, our work presented one of the first systematic evaluations and comparisons of remote reaching techniques on wall displays in the presence of distracter targets.

The last design exercise explored the idea of recording and presenting visual changes that might have been missed by the user due to the changes occurring in the background, periphery, or at a time where the user was otherwise preoccupied. The first contribution of this work is the characterization of the conditions under which visual information is not seen by users. The remaining contributions are related to the specifics of designing mnemonic rendering and its design space. The technique is applicable both in wall displays, but also in desktop settings [Bezerianos et al., 2006]. Since it is image-based, it may benefit all types of dynamic applications and data. The exploration of its design space has lead to several designs for storing and presenting the stored data. Among others, three restitution techniques are discussed together with their design variations, as well as ways of grouping changed content and triggering restitution. Most importantly, mnemonic rendering opens up the potential for new research in storing and visualizing hidden changes in everyday computing environments.

The main contributions of this thesis are the derived, general purpose designs. Nevertheless, insights into the design process may also prove helpful to researchers attempting to provide solutions to the same or similar problems on wall displays.

Our designs are always motivated by tasks we feel are relevant to wall display interaction, which may be used as a reference point for future interaction research and evaluations. By considering the identified tasks we were able to create a list of possible interaction challenges and discuss desirable properties for new interaction designs. These properties were followed in our designs, but may also prove useful in future research on

wall displays, as they are based on observations regarding the nature of the medium, namely its still uncharted application and use, and its vast real-estate that requires effort minimization and careful design.

Finally, our design process is initiated by reflecting on interaction challenges unique to the new medium, and refined through user feedback sessions. This practice in the case of ScaleView portals provided some validation for our design principles, mainly effort minimization, as well as indications that the real-estate of wall displays is used as a reference by users, while interaction preferably takes place close to their location. In the case of the vacuum, it enabled us to further refine our design principles: flexibility, but not on the expense of transient use and fluidity, is desirable in remote reaching techniques. This process motivated us to think of alternatives for enhancing the functionality of the vacuum, which led to the use of hover. Finally, in the case of mnemonic rendering, it motivated a thorough exploration of its design space and identified potential shortcomings of the technique. Nevertheless, this design process needs to be further augmented by long term studies of wall display use in real work environments, once wall displays become a commercial commodity.

6.3 Future directions

6.3.1 Layout management

Even though the canvas portal framework supports all types of alternative view techniques, our exploration of layout management and context switching designs for wall displays has focused on solutions based on alternative views simultaneously present on the display. Although this assumption is realistic, other designs need to be explored. For example the use of virtual *wall*tops and switching between them might be desirable in some use scenarios. Or the use of canvas portals that are decoupled from the display and may exist on PDAs or other interaction surfaces should be considered.

Another assumption of our designs is that content on wall displays will rarely suffer from issues like overlap and occlusion due to the extended display real-estate. Although this is true for now, future uses of wall displays might involve multiple overlapping elements and techniques for dealing with overlap need to be developed. Existing techniques for desktops [Beaudouin-Lafon and Lassen, 2000, Beaudouin-Lafon, 2001, Dragicevic, 2004, Baudisch and Gutwin, 2004] need to be re-evaluated for wall display interaction, given issues that might arise with large distances (for example dragging over other windows to flip between them might be cumbersome).

For the designs proposed in this chapter, detailed evaluation under everyday use is needed to fully assess their benefits. Further design improvements can also be considered. For example the focal areas of the described techniques have so far been rectangles. We might let users create other shapes as well. We expect that more free-form shapes could help users to better remember item groupings and portal and remote focal area connections. Similarly, specific focal area shapes could have some schematic meaning, for example skinny rectangles could be used for capturing areas of the display that contain tools or menu bars.

We’ve observed that horizontal Division Bands are most often used for facilitating the management of working and storage spaces, whereas vertical bands are most often used for quick glancing actions or fast shortcuts and their lifespan on the display is more limited. Although we’ve sorted the ordering of multiple Division Bands based on this observation, further investigation of usage patterns of the different types of bands could lead to a sorting and visualization mechanism better tailored to user behavior. Furthermore, the direct creation mechanism of Division Bands and the bands’ ephemeral nature indicate that they can be used to take state snapshots of the display and act as a form of history of the virtual canvas, thus leveraging canvas portals’ time attribute. However, we’ll need to solve the visualization challenges of depicting and navigating groups of such history bands related to their spatial and temporal nature, as well as

issues with system wide buffering, similar to mnemonic rendering.

Because weve focused on single-user techniques, our alternative views share the same instances of on-screen objects. We use wall sized displays, however, so it is desirable that the techniques be extendable to multi-user interaction. Although multiple users could use some of our tools simultaneously without interfering much with one another (such as the ScaleView and Window portals), adding functionality for context-sensitive objects would require enhancing others, such as the Division Bands, to support multiple users. This area clearly requires significant future research.

6.3.2 Remote reaching

Our exploration of remote reaching techniques that copy content close to the user, has indicated a tradeoff between the size of proxies and their relative layout. Small proxies can preserve the original layout of the display, but maybe hard to interact with. Full scale proxies are easily viewed, but laying them out in a predictable and efficient way close to the user is challenging. Although violating the application agnostic approach advocated in this thesis, some work [Baudisch et al., 2003a, Collomb et al., 2005a] proposes the use of semantic filtering for reducing the number of candidate icon proxies, by determining which application icons are appropriate. This filtering addresses the problem specifically for application icons. Nevertheless, more general filtering mechanisms can be thought of. For example proxies could be grouped into clusters and presented as such to the user, with most frequently selected proxies being more prominent. Thus frequently selected items may be accessed at full resolution, while less used ones might need extra interaction steps in order to be expanded at full scale. Clearly such filtering mechanisms need to be evaluated in detail.

All our prototypes for remote reaching techniques have been designed such that both the interaction and the visual effect of the interaction takes place close to the user. Clearly such designs need to be evaluated against other types of remote reaching techniques, like

virtual picking [Rekimoto, 1998, 1997], throwing interactions [Hascoët, 2003, Collomb and Hascoët, 2004, Scott et al., 2005, Reetz et al., 2006] and input amplification [Robertson et al., 2005, Forlines et al., 2006b]. Such a study has been performed for horizontal display reaching on tabletops [Nacenta et al., 2005], where interaction and visibility issues are very different to those of horizontal large displays, yet accuracy was not taken into account.

Recent work in psychology and neurophysiology [Làdavas, 2002, Witt et al., 2005] has indicated that the use of physical tools extends the notion of "near" or "peripersonal" space, the way humans understand distance and plan their actions with the environment. Wall displays are of the first digital surfaces that extend beyond the confines of peripersonal space and it would be interesting to investigate if the use of virtual mediators (tools) similarly extends human peripersonal space.

Finally, the majority of remote reaching designs presented in previous and current work make the implicit assumption that they are operated by a single user. Although many such scenarios can be thought of, eventually multi-user remote reaching has to be considered. Existing designs that introduce graphical elements across the display (such as rubber bands [Baudisch et al., 2003a, Collomb et al., 2005a], arcs of influence [Bezerianos and Balakrishnan, 2005a], etc) need to be refined to minimize clutter across the interaction surface. Others that involve animated transitions of objects across the display [Hascoët, 2003, Collomb and Hascoët, 2004, Scott et al., 2005, Reetz et al., 2006] need to be completely redesigned.

6.3.3 Presentation of invisible changes

Mnemonic rendering has the advantage of being an application and data agnostic technique and may thus benefit any type of application or dynamic visual data. On the other hand, because it is such a general approach, it can never lead to the sophistication of change identification and explanation present in application or data specific designs

[Endsley et al., 2003, Banbury and Tremblay, 2004, Tam and Greenberg, 2006, Baudisch et al., 2006b]. Another issue that mnemonic rendering does not address is the display of causality in hidden visual changes. Some changes in one part of the display might be directly or indirectly responsible for changes in another location. If these types of changes are not graphically or temporarily connected, their restitution might take place separately and users may miss the underlying connections. Such causal effects may only be explained when application and data specific designs are used. Thus integration of generic and dedicated techniques for supporting change awareness requires further investigation. Nevertheless, the pixel visibility notion introduced in this thesis can be used as-is to better handle existing designs aimed at revealing specific types of changes.

The mnemonic rendering interaction model assumes that the tasks performed by users are not time-critical and encourages proactive navigation in hidden changes. Thus no notification mechanism of the occurrence of changes is explicitly provided. Clearly tasks in wall displays may also be time critical or buffered changes in some invisible locations might have reached a point where the user needs to be notified. Thus a combination of mnemonic rendering with notification mechanisms could significantly enhance its applicability.

Nevertheless, current notification designs [McCrickard and Chewar, 2005] need to be re-examined when applied on large interactive surfaces, where content may reside outside the user's visual field. Moreover, not all applications dealing with dynamic data provide notifications. In the same way that mnemonic rendering provides an application agnostic buffering approach, approaches of notifying users of buffered changes, irrespective of applications, may be considered. For example users could be provided with a "notification minimap" of the display, that highlights (using say color) the areas where visual changes have taken place, indicating the most densely changed areas (by darker color). Or at the edges of the area that is deemed visible to the user, markers can indicate that changes have occurred in that direction. Other techniques could aim at attracting the user's

attention through peripheral vision. As motion detection remains fairly accurate across the human vision range, we can imagine areas with many buffered changes wobbling at a high sequence or becoming visually distorted, whereas remote areas with few changes remaining fairly static. Of course all these designs need detailed examination, as they introduce further user distractions.

Mnemonic rendering introduces potentially disturbing visual effects. Initial exploration indicated that users are not hindered in their interaction while using mnemonic rendering restitutions. Nevertheless, long term studies of everyday use need to be performed to ascertain their benefits and potential usability issues.

Finally, mnemonic rendering was designed under the assumption of a single user scenario. If it is to be used while multiple users concurrently interact with the wall display, major design alterations need to be performed in order to minimize user distraction. An approach for solving the multi-user issue may include social protocols. For example different users may be responsible for different areas of the display and only content not seen by any user will be buffered. Obviously such solutions need to be carefully designed and restitution triggering mechanisms based on multiple users need to be explored.

6.3.4 Beyond these designs

This thesis presented different designs for addressing layout management, remote reaching and hidden change presentation issues on wall displays. Clearly different designs, even ones motivated by our prototypes and early designs, can be developed. We do not claim to have completely solved the related challenges, but we feel we have provided both motivational designs, and interesting design processes that can inform future research in the field.

The issues discussed in this thesis and the developed designs address challenges for single user interaction on wall displays. Yet wall displays are well suited for multi-user concurrent interaction, due to their large real-estate that can accommodate multiple

people, collaborating or performing independent tasks, for example in public settings [Izadi et al., 2003, Vogel and Balakrishnan, 2004]. Thus existing research on single display groupware (SDG), usually performed in desktop environments, needs to be adapted for wall displays. For example interference between users might no longer pose an issue [Tsandilas and R.Balakrishnan, 2005] as users can work concurrently on the large display real-estate, at the potential cost of losing awareness of collaborators' actions due to the space between them.

Bibliography

Gregory D. Abowd, Christopher G. Atkeson, Jason Brotherton, Tommy Enqvist, Paul Gulley, and Johan LeMon. Investigating the capture, integration and access problem of ubiquitous computing in an educational setting. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 440–447, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-30987-4. doi: <http://doi.acm.org.myaccess.library.utoronto.ca/10.1145/274644.274704>.

J. Alimohideen, L. Renambot, J. Leigh, A. Johnson, R. Grossman, and M. Sabala. Pavis pervasive adaptive visualization and interaction service. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems. Workshop on Information Visualization and Interaction Techniques for Collaboration Across Multiple Display*, 2006.

G. Apitz and F. Guimbretière. Crossy: a crossing-based drawing application. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 3–12, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-957-8. doi: <http://doi.acm.org/10.1145/1029632.1029635>.

Apple. Apple time machine in mac os x leopard- <http://www.apple.com/macosx/leopard/timemachine.html> - visited january 2007, 2007.

J. Assa, Y. Caspi, and D. Cohen-Or. Action synopsis: Pose selection and illustration. pages 667–676. ACM Press, 2005.

- J. Baar, T. Willwacher, S. Rao, and R. Raskar. Seamless multi-projector display on curved screens. In *Eurographics Workshop on Virtual Environments (EGVE)*, pages 281–286, 2003.
- R. Baecker and I. Small. Animation at the interface. In B. Laurel, editor, *The Art of Human-Computer Interface Design*, pages 251–267. Addison-Wesley, Reading, MA, 1990.
- R. Baecker, I. Small, and R. Mander. Bringing icons to life. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1–6, New York, NY, USA, 1991. ACM Press. ISBN 0-89791-383-3. doi: <http://doi.acm.org/10.1145/108844.108845>.
- R.W. Baily. *Human performance engineering: Using human factors/ergonomics to achieve computer system usability (2nd ed.)*. Englewood Cliffs, New Jersey: Prentice Hall, 1989.
- J.Z. Bakdash, J.S. Augustyn, and D.R. Proffitt. Large displays enhance spatial knowledge of a virtual environment. In *ACM Siggraph Symposium on Applied Perception in Graphics and Visualization*, pages 59–62, 2006.
- R. Balakrishnan, G. Fitzmaurice, G. Kurtenbach, and W. Buxton. Digital tape drawing. In *UIST '99: Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 161–169. ACM Press, 1999. ISBN 1-58113-075-9. doi: <http://doi.acm.org/10.1145/320719.322598>.
- R. Ball and C. North. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1196–1199, New York, NY, USA, 2005a. ACM Press. ISBN 1-59593-002-7. doi: <http://doi.acm.org.myaccess.library.utoronto.ca/10.1145/1056808.1056875>.

- R. Ball and C. North. An analysis of user behavior on high-resolution tiled displays. In *Tenth IFIP International Conference on Human-Computer Interaction (INTERACT 2005)*, pages 350–364, 2005b.
- R. Ball, C. North, and D.A. Bowman. Move to improve: promoting physical navigation to increase user performance with large displays. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 191–200, New York, NY, USA, 2007. ACM Press. ISBN 978-1-59593-593-9. doi: <http://doi.acm.org/myaccess.library.utoronto.ca/10.1145/1240624.1240656>.
- B. Banbury and S. Tremblay, editors. *A cognitive approach to situation awareness: theory and application*. Ashgate, 2004.
- J.E. Bardram, T.R. Hansen, and M. Soegaard. Awaremedia: a shared interactive display supporting social, temporal, and spatial awareness in surgery. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 109–118, New York, NY, USA, 2006a. ACM Press. ISBN 1-59593-249-6. doi: <http://doi.acm.org/10.1145/1180875.1180892>.
- J.E. Bardram, T.R. Hansen, and M. Soegaard. Large interactive displays in hospitals - motivation, examples, and challenges. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems. Workshop on Information Visualization and Interaction Techniques for Collaboration Across Multiple Display*, 2006b.
- P. Baudisch and C. Gutwin. Multiblending: displaying overlapping windows simultaneously without the drawbacks of alpha blending. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 367–374, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-702-8. doi: <http://doi.acm.org/10.1145/985692.985739>.
- P. Baudisch, N. Good, and P. Stewart. Focus plus context screens: combining display

- technology with visualization techniques. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 31–40. ACM Press, 2001. ISBN 1-58113-438-X. doi: <http://doi.acm.org/10.1145/502348.502354>.
- P. Baudisch, N. Good, V. Bellotti, and P. Schraedley. Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 259–266. ACM Press, 2002. ISBN 1-58113-453-3. doi: <http://doi.acm.org/10.1145/503376.503423>.
- P. Baudisch, E. Cutrell, D. Robbins M., Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *Proceedings of Interact 2003*, pages 57–64, August 2003a.
- P. Baudisch, E. Cutrell, and G. Robertson. High-density cursor: A visualization technique that helps users keep track of fast-moving mouse cursors. In *Proceedings of Interact 2003*, pages 236–243, August 2003b.
- P. Baudisch, E. Cutrell, K. Hinckley, and R. Gruen. Mouse ether: accelerating the acquisition of targets across multi-monitor displays. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1379–1382, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-703-6. doi: <http://doi.acm.org/10.1145/985921.986069>.
- P. Baudisch, M. Sinclair, and A. Wilson. Soap: a pointing device that works in mid-air. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 43–46, New York, NY, USA, 2006a. ACM Press. ISBN 1-59593-313-1. doi: <http://doi.acm.org/10.1145/1166253.1166261>.
- P. Baudisch, D. Tan, M. Collomb, D. Robbins, K. Hinckley, M. Agrawala, S. Zhao,

- and G. Ramos. Phosphor: explaining transitions in the user interface using afterglow effects. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 169–178, New York, NY, USA, 2006b. ACM Press. ISBN 1-59593-313-1. doi: <http://doi.acm.org/10.1145/1166253.1166280>.
- M. Beaudouin-Lafon. Novel interaction techniques for overlapping windows. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 153–154, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-438-X. doi: <http://doi.acm.org/10.1145/502348.502371>.
- M. Beaudouin-Lafon and H.M. Lassen. The architecture and implementation of cpn2000, a post-wimp graphical application. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 181–190, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-212-3. doi: <http://doi.acm.org/10.1145/354401.354761>.
- B.B. Bederson and J.D. Hollan. Pad++: a zooming graphical interface for exploring alternate interface physics. In *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 17–26, New York, NY, USA, 1994. ACM Press. ISBN 0-89791-657-3. doi: <http://doi.acm.org/10.1145/192426.192435>.
- C. Beynon and T. Reilly. Spinal shrinkage during a seated break and standing break during simulated nursing tasks. *Applied Ergonomics*, 32(6):617–622(6), 2001. doi: [10.1016/S0003-6870\(01\)00038-2](http://doi.org/10.1016/S0003-6870(01)00038-2).
- A. Bezerianos and R. Balakrishnan. The vacuum: facilitating the manipulation of distant objects. In *CHI '05: Proceeding of the SIGCHI conference on Human factors in computing systems*, pages 361–370, New York, NY, USA, 2005a. ACM Press. ISBN 1-58113-998-5. doi: <http://doi.acm.org/10.1145/1054972.1055023>.

- A. Bezerianos and R. Balakrishnan. Interaction and visualization techniques for very large scale high resolution displays. DGP-TR-2004- 002, University of Toronto, 2004.
- A. Bezerianos and R. Balakrishnan. Canvas portals: View and space management on large displays. *Special issue of IEEE Computer Graphics and Applications*, 25(4): 34–43, July/August 2005b.
- A. Bezerianos, P. Dragicevic, and R. Balakrishnan. Mnemonic rendering: an image-based approach for exposing hidden changes in dynamic displays. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 159–168, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-313-1. doi: <http://doi.acm.org/10.1145/1166253.1166279>.
- J.T. Biehl, M. Czerwinski, G. Smith, and G.G. Robertson. Fastdash: a visual dashboard for fostering awareness in software teams. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1313–1322, New York, NY, USA, 2007. ACM Press. ISBN 978-1-59593-593-9. doi: <http://doi.acm.org.myaccess.library.utoronto.ca/10.1145/1240624.1240823>.
- E.A. Bier, M.C. Stone, K. Pier, W. Buxton, and T.D. DeRose. Toolglass and magic lenses: The see-through interface. *Computer Graphics*, 27(Annual Conference Series): 73–80, 1993.
- R.A. Bolt. put-that-there: Voice and gesture at the graphics interface. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270, New York, NY, USA, 1980. ACM Press. ISBN 0-89791-021-4. doi: <http://doi.acm.org/10.1145/800250.807503>.
- D.A. Bowman and L.F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *SI3D '97: Proceedings of the*

- 1997 symposium on Interactive 3D graphics*, pages 35–38, New York, NY, USA, 1997. ACM Press. ISBN 0-89791-884-3. doi: <http://doi.acm.org/10.1145/253284.253301>.
- H. Brignall and Y. Rogers. Enticing people to interact with large public displays in public space. In *Proceedings of INTERACT'03*, pages 17–24, 2003.
- H. Brignull, S. Izadi, G. Fitzpatrick, Y. Rogers, and T. Rodden. The introduction of a shared interactive surface into a communal space. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 49–58, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-810-5. doi: <http://doi.acm.org/10.1145/1031607.1031616>.
- W. Buxton. Chunking and phrasing and the design of human-computer dialogues. In *Proceedings of the IFIP World Computer Congress*, pages 475–480, 1986.
- W. Buxton. Three-state model of graphical input. In *Interact Conference*, pages 449–456, 1990.
- W. Buxton. Living in augmented reality: Ubiquitous media and reactive environments. In K. Finn, A. Sellen, and S. Wilber, editors, *Video Mediated Communication*, pages 363–384. Hillsdale, N.J.: Erlbaum, 1997.
- William Buxton, George Fitzmaurice, Ravin Balakrishnan, and Gordon Kurtenbach. Large displays in automotive design. *IEEE Computer Graphics and Applications*, 20(4):68–75, 2000. ISSN 0272-1716. doi: <http://doi.ieeecomputersociety.org/10.1109/38.851753>.
- J.J. Cadiz, G. Venolia, G. Jancke, and A. Gupta. Designing and deploying an information awareness interface. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 314–323, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-560-2. doi: <http://doi.acm.org/10.1145/587078.587122>.

- X. Cao and R. Balakrishnan. Interacting with dynamically defined information spaces using a handheld projector and a pen. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 225–234, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-313-1. doi: <http://doi.acm.org/10.1145/1166253.1166289>.
- X. Cao and R. Balakrishnan. Visionwand: Interaction techniques for large displays using a passive wand tracked in 3d. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*. ACM Press, 2003.
- S. Carpendale and C. Montagnese. A framework for unifying presentation space. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 61–70, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-438-X. doi: <http://doi.acm.org/10.1145/502348.502358>.
- B. Chang, J.D. Mackinlay, P.T. Zellweger, and T. Igarashi. A negotiation architecture for fluid documents. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 123–132, New York, NY, USA, 1998. ACM Press. ISBN 1-58113-034-1. doi: <http://doi.acm.org/10.1145/288392.288585>.
- B.W. Chang and D. Ungar. Animation: from cartoons to the user interface. In *UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 45–55, New York, NY, USA, 1993. ACM Press. ISBN 0-89791-628-X. doi: <http://doi.acm.org/10.1145/168642.168647>.
- O. Chapuis and N. Roussel. Metisse is not a 3d desktop! In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 13–22, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-271-2. doi: <http://doi.acm.org/10.1145/1095034.1095038>.
- S. Chatty. Defining the dynamic behaviour of animated interfaces. In *Proceedings of the*

- IFIP TC2/WG2.7 Working Conference on Engineering for Human-Computer Interaction*, pages 95–111, Amsterdam, The Netherlands, The Netherlands, 1992. North-Holland Publishing Co. ISBN 0-444-89904-9.
- S. Chatty and M. Beaudouin-Lafon. Integrating animation with interfaces. In *CHI '92: Posters and short talks of the 1992 SIGCHI conference on Human factors in computing systems*, pages 70–70, New York, NY, USA, 1992. ACM Press. doi: <http://doi.acm.org/10.1145/1125021.1125087>.
- R.E. Christ. Review and analysis of color coding research for visual displays. *Human Factors*, 17(6):542–570, 1975.
- R.E. Christ. Research for evaluating visual display codes: an emphasis coding. In R. Easterby and H. Zwaga, editors, *Information Design: The design and evaluation signs and printed material*, pages 209–228. New York, NY: John Wiley and Sons Ltd, 1984.
- E. Churchill, L. Nelson, L. Denoue, and A. Girgensohn. The plasma poster network: Posting multimedia content in public places. In *Proceedings of INTERACT'03*, 2003. URL citeseer.ist.psu.edu/churchill03plasma.html.
- A. Cockburn, S. Greenberg, S. Jones, B. McKenzie, and M. Moyle. Improving web page revisitation: Analysis, design and evaluation. *IT and Society*, 3(1):159–183, 2003.
- M. Collomb and M. Hascoët. Speed and accuracy in throwing models for large displays. In *HCI2004*, page Volume 2. British HCI Group, 2004.
- M. Collomb, M. Hascoët, P. Baudisch, and B. Lee. Improving drag-and-drop on wall-size displays. In *GI '05: Proceedings of the 2005 conference on Graphics interface*, pages 25–32, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005a. Canadian Human-Computer Communications Society. ISBN 1-56881-265-5.

- M. Collomb, M. Hascoët, P. Baudisch, and B. Lee. Improving drag-and-drop on wall-size displays. In *Graphics Interface 2005*, 2005b.
- M. Cooper and J. Foote. Summarizing video using non-negative similarity matrix factorization. In *2002 IEEE Workshop on Multimedia Signal Processing*, pages 25–28, 2002.
- J. Cutting and M. Vishton. Perceiving layout and distances: The integration, relative potency, and contextual use of different information about depth. *Perception of space and motion: Handbook of perception and cognition (2nd ed.)*, 5:69–117, 1995.
- M. Czerwinski, G. Smith, T. Regan, B. Meyers, G. Robertson, and G. Starkweather. Toward characterizing the productivity benefits of very large displays. In M. Rauterberg et al, editor, *Human-Computer Interaction–INTERACT ’03*, pages 9–16. IOS Press, 2003.
- D. DeMenthon, V. Kobla, and D. Doermann. Video summarization by curve simplification. In *MULTIMEDIA ’98: Proceedings of the sixth ACM international conference on Multimedia*, pages 211–218, New York, NY, USA, 1998. ACM Press. ISBN 0-201-30990-4. doi: <http://doi.acm.org/10.1145/290747.290773>.
- P. Dietz and D. Leigh. Diamondtouch: a multi-user touch technology. In *UIST ’01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-438-X. doi: <http://doi.acm.org/10.1145/502348.502389>.
- P. Dragicevic. Combining crossing-based and paper-based interaction paradigms for dragging and dropping between overlapping windows. In *UIST ’04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 193–196, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-957-8. doi: <http://doi.acm.org/10.1145/1029632.1029667>.

- A.T. Duchowski. A breadth-first survey of eye tracking applications. *Behavior Research Methods, Instruments, & Computers (BRMIC)*, 34(4):455–470, 2002.
- W.K. Edwards, T. Igarashi, A. LaMarca, and E.D. Mynatt. A temporal model for multi-level undo and redo. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 31–40, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-212-3. doi: <http://doi.acm.org/10.1145/354401.354409>.
- Electrosonic. http://www.electrosonic.com/command_profiles.asp - visited 2006, 2006.
- S. Elrod, R. Bruce, R. Gold, D. Goldberg, F. Halasz, W. Janssen, D. Lee, K. McCall, D. Pedersen, K. Pier, J. Tang, and B. Welch. Liveboard: A large interactive display supporting group meetings. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 599–607. ACM Press, 1992.
- M.R. Endsley, B. Bolt, and D.G. Jones. *Designing for Situation Awareness: An Approach to User-centered Design*. Taylor and Francis, 2003. ISBN 0748409661.
- A. Fass, J. Forlizzi, and R. Pausch. Messydesk and messyboard: two designs inspired by the goal of improving human memory. In *DIS '02: Proceedings of the conference on Designing interactive systems*, pages 303–311, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-515-7. doi: <http://doi.acm.org.myaccess.library.utoronto.ca/10.1145/778712.778754>.
- G. Fitzmaurice, A. Khan, G. Kurtenbach, and G. Binks. Cinematic meeting facilities using large displays. *IEEE Computer Graphics and Applications*, 25(4):17–21, 2005. ISSN 0272-1716. doi: <http://doi.ieeecomputersociety.org/10.1109/MCG.2005.76>.
- G.W Fitzmaurice, A. Khan, W. Buxton, G. Kurtenbach, and R. Balakrishnan. Sentient data access via a diverse society of devices. *ACM Queue*, 1(8):53–62, 2003.

- C. Forlines, R. Balakrishnan, P. Beardsley, J. van Baar, and R. Raskar. Zoom-and-pick: facilitating visual zooming and precision pointing with interactive handheld projectors. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 73–82, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-271-2. doi: <http://doi.acm.org/10.1145/1095034.1095046>.
- C. Forlines, A. Esenther, C. Shen, D. Wigdor, and K. Ryall. Multi-user, multi-display interaction with a single-user, single-display geospatial application. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 273–276, New York, NY, USA, 2006a. ACM Press. ISBN 1-59593-313-1. doi: <http://doi.acm.org/10.1145/1166253.1166296>.
- C. Forlines, D. Vogel, and R. Balakrishnan. Hybridpointing: fluid switching between absolute and relative pointing with a direct input device. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 211–220, New York, NY, USA, 2006b. ACM Press. ISBN 1-59593-313-1. doi: <http://doi.acm.org/10.1145/1166253.1166286>.
- E. Freeman and S. Fertig. Lifestreams: Organizing your electronic life. In *AAAI Fall Symposium: AI Applications in Knowledge Navigation and Retrieval*, November 1995.
- T. Funkhouser and K. Li. Onto the wall: Large displays. *Special issue of IEEE Computer Graphics and Applications*, 20(4):–, 2000.
- G.W. Furnas. Generalized fisheye view. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23. ACM Press, 1986.
- J. Geißler. Shuffle, throw or take it! working efficiently with an interactive wall. In *CHI '98 extended abstracts on Human factors in computer systems*, 1998.
- S. Greenberg and M. Rounding. The notification collage: posting information to public and personal displays. In *CHI '01: Proceedings of the SIGCHI conference on Human*

- factors in computing systems*, pages 514–521, 2001. URL citeseer.ist.psu.edu/greenberg00notification.html.
- T. Grossman and R. Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor’s activation area. In *CHI '05: Proceeding of the SIGCHI conference on Human factors in computing systems*, pages 281–290, New York, NY, USA, 2005. ACM Press. ISBN 1-58113-998-5. doi: <http://doi.acm.org/10.1145/1054972.1055012>.
- T. Grossman and R. Balakrishnan. The design and evaluation of selection techniques for 3d volumetric displays. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 3–12, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-313-1. doi: <http://doi.acm.org/10.1145/1166253.1166257>.
- T. Grossman, D. Wigdor, and R. Balakrishnan. Multi-finger gestural interaction with 3d volumetric displays. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 61–70, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-957-8. doi: <http://doi.acm.org/10.1145/1029632.1029644>.
- T. Grossman, K. Hinckley, P. Baudisch, M. Agrawala, and R. Balakrishnan. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 861–870, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-372-7. doi: <http://doi.acm.org/10.1145/1124772.1124898>.
- J. Grudin. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 458–465. ACM Press, 2001. ISBN 1-58113-327-8. doi: <http://doi.acm.org/10.1145/365024.365312>.
- Y. Guiard, R. Blanch, and M. Beaudouin-Lafon. Object pointing: a complement to

- bitmap pointing in guis. In *GI '04: Proceedings of the 2004 conference on Graphics interface*, pages 9–16, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society. ISBN 1-56881-227-2.
- F. Guimbretière and T. Winograd. Flowmenu: Combining command text and parameter entry. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 213–216. ACM Press, 2000.
- F. Guimbretière, M. Stone, and T. Winograd. Fluid interaction with high-resolution wall-size displays. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 21–30, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-438-X. doi: <http://doi.acm.org/10.1145/502348.502353>.
- E. Hall. *The Hidden Dimension*. Anchor Books, 1966. ISBN 0-385-08476-5.
- J. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-271-2. doi: <http://doi.acm.org/10.1145/1095034.1095054>.
- M. Hascoët. Throwing models for large displays. In *HCI'2003*, pages 73–77. British HCI Group, 2003.
- K. Hawkey, M. Kellar, D. Reilly, T. Whalen, and K.M. Inkpen. The proximity factor: impact of distance on co-located collaboration. In *GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 31–40, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-223-2. doi: <http://doi.acm.org/10.1145/1099203.1099209>.
- M.A. Hibbs, N.C. Dirksen, K. Li, and O.G. Troyanskaya. Visualization methods for statistical analysis of microarray clusters. *BMC Bioinformatics*, 6(115), 2005.

- K. Hinckley, R. Pausch, J.C. Goble, and N.F. Kassell. A survey of design issues in spatial input. In *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 213–222, New York, NY, USA, 1994. ACM Press. ISBN 0-89791-657-3. doi: <http://doi.acm.org/10.1145/192426.192501>.
- K. Hinckley, G. Ramos, F. Guimbretière, P. Baudisch, and M. Smith. Stitching: pen gestures that span multiple displays. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 23–31, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-867-9. doi: <http://doi.acm.org/10.1145/989863.989866>.
- K. Hinckley, F. Guimbretière, M. Agrawala, G. Apitz, and N. Chen. Phrasing techniques for multi-stroke selection gestures. In *GI '06: Proceedings of Graphics Interface 2006*, pages 147–154, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society. ISBN 1-56881-308-2.
- N.P. Holmes and C. Spence. The body schema and the multisensory representation(s) of peripersonal space. *Cognitive Processing*, (5):94–105, 2004.
- E.M. Huang and E.D. Mynatt. Semi-public displays for small, co-located groups. In *CHI '03: Proceedings of the conference on Human factors in computing systems*, pages 49–56. ACM Press, 2003. ISBN 1-58113-630-7. doi: <http://doi.acm.org/10.1145/642611.642622>.
- A.R. Hunt and A. Andkingstone. Covert and overt voluntary attention: linked or independent? *Cognitive Brain Research*, 18:102–105, 2003.
- P. Irani, C. Gutwin, and X.D. Yang. Improving selection of off-screen targets with hopping. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 299–308, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-372-7. doi: <http://doi.acm.org/10.1145/1124772.1124818>.
- L. Itti, G. Rees, and J. Tsotsos, editors. *Neurobiology of attention*. Elsevier, 2005.

- S. Izadi, H. Brignull, T. Rodden, Y. Rogers, and M. Underwood. Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 159–168, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-636-6. doi: <http://doi.acm.org/10.1145/964696.964714>.
- R.J.K. Jacob. Eye-movement-based human-computer interaction techniques: Toward non-command interfaces. In R. Hartson and D. Hix, editors, *Advances in Human-Computer Interaction*, volume 4, chapter 6, pages 151–190. Ablex Publishing Corporation, Norwood, New Jersey, 1993.
- C. Jaynes, S. Webb, R.M. Steele, M. Brown, and W.B. Seales. Dynamic shadow removal from front projection displays. *IEEE Visualization*, pages 175–182, 2001.
- P.A. Jedrysik. Interactive large-screen displays for command and control, 2006. URL <http://www.afrlhorizons.com/Briefs/Sept01/IF0012.html-visited2006>.
- H. Jiang, E. Ofek, N. Moraveji, and Y. Shi. Direct pointer: direct manipulation for large-display interaction using handheld cameras. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1107–1110, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-372-7. doi: <http://doi.acm.org/10.1145/1124772.1124937>.
- B. Johanson, A. Fox, and T. Winograd. The interactive workspace project: Experiences with ubiquitous computing rooms. In *IEEE Pervasive Computing*, pages 67–74, 2002a.
- B. Johanson, G. Hutchins, T. Winograd, and M. Stone. Pointright: experience with flexible input redirection in interactive workspaces. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 227–234, New York, NY, USA, 2002b. ACM Press. ISBN 1-58113-488-6. doi: <http://doi.acm.org/10.1145/571985.572019>.

- A. Johnson, J. Leigh, P. Morin, and P. Van Keken. Geowall: Stereoscopic visualization for geoscience research and education. *IEEE Computer Graphics and Applications*, 26(6):10–14, 2006.
- J. Johnson, T.L. Roberts, W. Verplank, D.C. Smith, C.H. Irby, M. Beard, and K. Mackey. The xerox star: A retrospective. *Computer*, 22(9):11–26, 28–29, 1989. ISSN 0018-9162. doi: <http://dx.doi.org/10.1109/2.35211>.
- A. Kendon, editor. *Gesture: visible action as utterance*. Cambridge University Press, 2004.
- A. Khan, G. Fitzmaurice, D. Almeida, N. Burtnyk, and G. Kurtenbach. A remote control interface for large displays. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 127–136, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-957-8. doi: <http://doi.acm.org/10.1145/1029632.1029655>.
- A. Khan, J. Matejka, G. Fitzmaurice, and G. Kurtenbach. Spotlight: directing users' attention on large displays. In *CHI '05: Proceeding of the SIGCHI conference on Human factors in computing systems*, pages 791–798, New York, NY, USA, 2005. ACM Press. ISBN 1-58113-998-5. doi: <http://doi.acm.org/10.1145/1054972.1055082>.
- C. Kirstein and H. Muller. Interaction with a projection screen using a camera-tracked laser pointer. In *Multimedia Modeling; MMM'98*, pages 191–192, 1998.
- D. Kurlander and S. Feiner. A history-based macro by example system. In *UIST '92: Proceedings of the 5th annual ACM symposium on User interface software and technology*, pages 99–106, New York, NY, USA, 1992. ACM Press. ISBN 0-89791-549-6. doi: <http://doi.acm.org/10.1145/142621.142633>.
- G. Kurtenbach and W. Buxton. The limits of expert performance using hierarchic marking menus. In *Proceedings of InterCHI'93*, pages 482–487, 1993.

- E. Làdavas. Functional and dynamic properties of visual peripersonal space. *Trends in Cognitive Science*, 6(1):17–22, 2002. doi: 10.1016/S1364-6613(00)01814-3.
- J. Lin, H. Duh, H. Abi-Rached, D. Parker, and T. Furness. Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. In *VR '02: Proceedings of the IEEE Virtual Reality Conference 2002*, page 164, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1492-8.
- A.M. Lund. The influence of video image size and resolution on viewing-distance preferences. *Society of Motion Picture and Television Engineers Journal*, 102:406–415, 1997.
- B. MacIntyre, E.D. Mynatt, S. Volda, K.M. Hansen, J. Tullio, and G.M. Corso. Support for multitasking and background awareness using interactive peripheral displays. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 41–50, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-438-X. doi: <http://doi.acm.org/10.1145/502348.502355>.
- I. S. MacKenzie and S. Jusoh. An evaluation of two input devices for remote pointing. In *in Proc. of the 8th IFIP Working Conf. on Engineering for HCI - EHCI 2001*, pages 235–249. Springer-Verlag, 2001.
- P.P. Maglio and C.S. Campbell. Tradeoffs in displaying peripheral information. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 241–248, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-216-6. doi: <http://doi.acm.org/10.1145/332040.332438>.
- S. Malik and J. Laszlo. Visual touchpad: a two-handed gestural input device. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 289–296, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-995-0. doi: <http://doi.acm.org/10.1145/1027933.1027980>.

- S. Malik, A. Ranjan, and R. Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 43–52, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-271-2. doi: <http://doi.acm.org/10.1145/1095034.1095042>.
- M. Masuch. Speedlines: depicting motion in motionless pictures. In *SIGGRAPH '99: ACM SIGGRAPH 99 Conference abstracts and applications*, page 277, New York, NY, USA, 1999. ACM Press. ISBN 1-58113-103-8. doi: <http://doi.acm.org/10.1145/311625.312369>.
- T. Matthews, M. Czerwinski, G. Robertson, and D. Tan. Clipping lists and change borders: improving multitasking efficiency with peripheral information design. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 989–998, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-372-7. doi: <http://doi.acm.org/10.1145/1124772.1124920>.
- S.V. Matveyev and M. Göbel. The optical tweezers: multiple-point interaction technique. In *VRST '03: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 184–187, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-569-6. doi: <http://doi.acm.org/10.1145/1008653.1008685>.
- S. McCloud. *Understanding Comics*. Perennial Currents, 1994.
- D. Scott McCrickard and C.M. Chewar. Designing attention-centric notification systems: Five hci challenges. research report, Department of Computer Science and Center for HCI, Blacksburg VA 24061-0106, 2005. URL <http://eprints.cs.vt.edu/archive/00000720/01/TR-05-13.pdf>.
- D.S. McCrickard and C.M. Chewar. Designing attention-centric notification systems: Five hci challenges. In J.C. Forsythe, M.L. Bernard, and T.E. Goldsmith, editors, *Cog-*

- nitive Systems: Human Cognitive Models in Systems Design*, pages 67–89. Lawrence Earlbaum, 2006.
- M. McGuffin, N. Burtnyk, and G. Kurtenbach. FaST Sliders: Integrating Marking Menus and the Adjustment of Continuous Values. In *Proceedings of Graphics Interface (GI) 2002*, pages 35–41, May 2002.
- M.J. McGuffin, L. Tancu, and R. Balakrishnan. Using deformations for browsing volumetric data. In *Proceedings of IEEE Visualization (VIS) 2003*, pages 401–408, October 2003.
- C. Mertz, S. Chatty, and J. Vinot. Pushing the limits of atc user interface design beyond s&m interaction: the digistrips experience. In *Proceedings of the ATM'00 seminar*, 2000. URL citeseer.ist.psu.edu/mertz00pushing.html.
- T. P. Moran, P. Chiu, and W. van Melle. Pen-based interaction techniques for organizing material on an electronic whiteboard. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 45–54. ACM Press, 1997.
- M. Ringel Morris, A. Huang, A. Paepcke, and T. Winograd. Cooperative gestures: multi-user gestural interactions for co-located groupware. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1201–1210, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-372-7. doi: <http://doi.acm.org/10.1145/1124772.1124952>.
- T. Moscovich and J. Hughes. Navigating documents with the virtual scroll ring. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 57–60, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-957-8. doi: <http://doi.acm.org/10.1145/1029632.1029642>.

- B. A. Myers, R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller, and A. C. Long. Interacting at a distance: measuring the performance of laser pointers and other devices. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing system*, pages 33–40. ACM Press, 2002.
- E. Mynatt, T. Igarashi, W. Edwards, and A. LaMarca. Flatland: New dimensions in office whiteboards. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 346–353. ACM Press, 1999.
- M. Nacenta, S. Sallam, B. Champoux, S. Subramanian, and C. Gutwin. Perspective cursor: perspective-based interaction for multi-display environments. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 289–298, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-372-7. doi: <http://doi.acm.org/10.1145/1124772.1124817>.
- M.A. Nacenta, D. Aliakseyeu, S. Subramanian, and C. Gutwin. A comparison of techniques for multi-display reaching. In *CHI '05: Proceeding of the SIGCHI conference on Human factors in computing systems*, pages 371–380, New York, NY, USA, 2005. ACM Press. ISBN 1-58113-998-5. doi: <http://doi.acm.org/10.1145/1054972.1055024>.
- Y. Nakanishi, T. Fujii, K. Kiatjima, Y. Sato, and H. Koike. Vision-based face tracking system for large displays. In *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*, pages 152–159, London, UK, 2002. Springer-Verlag. ISBN 3-540-44267-7.
- W. Newman. A system for interactive graphical programming. In *AFIPS Spring Joint Computer Conference*, pages 47–54, 1968.
- D.R. Olsen, Jr. and T. Nielsen. Laser pointer interaction. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–22. ACM Press, 2001. ISBN 1-58113-327-8. doi: <http://doi.acm.org/10.1145/365024.365030>.

- OSHA. Occupational safety & health administration: "ergonomics for the prevention of musculoskeletal disorders guidelines for retail grocery stores" - <http://www.osha.gov/ergonomics/guidelines/retailgrocery/retailgrocery.html> - visited january 2007, 2007.
- K.R. Park, J.J. Lee, and J. Kim. Facial and eye gaze detection. In *Lecture Notes in Computer Science*, volume 2525. Springer Berlin / Heidelberg, 2002.
- J.K. Parker, R. Mandryk, and K.M. Inkpen. Tractorbeam: Seamless integration of local and remote pointing for tabletop displays. In *GI'05: Graphics Interface*, pages 33–40, New York, NY, USA, 2005. ACM Press. doi: <http://doi.acm.org/10.1145/1099203.1099209>.
- H.E. Pashler. *The Psychology of Attention*. A Bradford book Publisher, 1998.
- E. R. Pedersen, K. McCall, T. P. Moran, and F. G. Halasz. Tivoli: an electronic whiteboard for informal workgroup meetings. In *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 391–398. ACM Press, 1993.
- J.S. Pierce and R. Pausch. Comparing voodoo dolls and homer: exploring the importance of feedback in virtual environments. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 105–112, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-453-3. doi: <http://doi.acm.org/10.1145/503376.503396>.
- J.S. Pierce, A.S. Forsberg, M.J. Conway, S. Hong, R.C. Zeleznik, and M.R. Mine. Image plane interaction techniques in 3d immersive environments. In *SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 39–ff., New York, NY, USA, 1997. ACM Press. ISBN 0-89791-884-3. doi: <http://doi.acm.org/10.1145/253284.253303>.
- J.S. Pierce, B.C. Stearns, and R. Pausch. Voodoo dolls: seamless interaction at multiple scales in virtual environments. In *SI3D '99: Proceedings of the 1999 symposium on*

- Interactive 3D graphics*, pages 141–145, New York, NY, USA, 1999. ACM Press. ISBN 1-58113-082-1. doi: <http://doi.acm.org/10.1145/300523.300540>.
- I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80, New York, NY, USA, 1996. ACM Press. ISBN 0-89791-798-7. doi: <http://doi.acm.org/10.1145/237091.237102>.
- K. Price. Section 21.3 - <http://www.visionbib.com/bibliography/contentspeople.html#Analysis%20of%20Images%20of%20People,%20Faces,%20Fingerprints> - visited february 2007, 2007.
- D.R. Proffitt. Distance perception. *Current Directions in Psychological Research*, 15: 131–135, 2006.
- R. Raskar, J. Baar, and J. Chai. A low-cost projector mosaic with fast registration. In *Asian Conference on Computer Vision (ACCV)*, 2002.
- A. Reetz, C. Gutwin, T. Stach, M. Nacenta, and S. Subramanian. Superflick: a natural and efficient technique for long-distance object placement on digital tables. In *GI '06: Proceedings of the 2006 conference on Graphics interface*, pages 163–170, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society. ISBN 1-56881-308-2.
- J. Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 31–39. ACM Press, 1997. ISBN 0-89791-881-9. doi: <http://doi.acm.org/10.1145/263407.263505>.
- J. Rekimoto. Time-machine computing: a time-centric approach for the information environment. In *UIST '99: Proceedings of the 12th annual ACM symposium on User*

- interface software and technology*, pages 45–54, New York, NY, USA, 1999. ACM Press. ISBN 1-58113-075-9. doi: <http://doi.acm.org/10.1145/320719.322582>.
- J. Rekimoto. A multiple device approach for supporting whiteboard-based interactions. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 344–351. ACM Press/Addison-Wesley Publishing Co., 1998.
- J. Rekimoto. Smartskin: An infrastructure for freehand manipulation on interactive surfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing system*, pages 113–120. ACM Press, April 2002.
- J. Rekimoto and M. Saitoh. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 378–385, New York, NY, USA, 1999. ACM Press. ISBN 0-201-48559-1. doi: <http://doi.acm.org/10.1145/302979.303113>.
- K. Renaud. Expediting rapid recovery from interruptions by providing a visualization of application activity. In *Proceedings of OZCHI 2000*, pages 348–355, 2000.
- R.A. Rensink. Change detection. *Annual Review of Psychology*, 53:245–277, 2002.
- G. Rizzolatti, L. Fadiga, L. Fogassi, and V. Gallese. The Space Around Us. *Science*, 277(5323):190–191, 1997. doi: 10.1126/science.277.5323.190. URL <http://www.sciencemag.org>.
- G. Robertson, E. Horvitz, M. Czerwinski, P. Baudisch, D.R. Hutchings, B. Meyers, D. Robbins, and G. Smith. Scalable fabric: flexible task management. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 85–89, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-867-9. doi: <http://doi.acm.org/10.1145/989863.989874>.

- G. Robertson, M. Czerwinski, P. Baudisch, B. Meyers, D. Robbins, G. Smith, and D. Tan. The large-display user experience. *IEEE Computer Graphics and Applications*, 25(4): 44–51, 2005. ISSN 0272-1716. doi: <http://doi.ieeecomputersociety.org/10.1109/MCG.2005.88>.
- D.M. Russell. Large interactive public displays: Use patterns, support patterns, community patterns. In *CSCW '02: Workshop on Public, Community and Situated Displays. ACM conference on Computer supported cooperative work*, 2002.
- D.M. Russell and R. Gossweiler. On the design of personal & communal large information scale appliances. In *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, pages 354–361, London, UK, 2001. Springer-Verlag. ISBN 3-540-42614-0.
- D.M. Russell, C. Drews, and A. Sue. Social aspects of using large public interactive displays for collaboration. In *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*, pages 229–236, London, UK, 2002. Springer-Verlag. ISBN 3-540-44267-7.
- K. Ryall, C. Forlines, C. Shen, and M. Ringel Morris. Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 284–293, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-810-5. doi: <http://doi.acm.org/10.1145/1031607.1031654>.
- K. Ryall, C. Forlines, C. Shen, M. Ringel Morris, and K. Everitt. Experiences with and observations of direct-touch tabletops. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 89–96, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2494-X. doi: <http://dx.doi.org/10.1109/TABLETOP.2006.12>.

- P. Salembier and F. Marques. Region-based representations of image and video: Segmentation tools for multimedia services. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8):1147–1169, 1999.
- S.D. Scott, M. Sheelagh, T. Carpendale, and K.M. Inkpen. Territoriality in collaborative tabletop workspaces. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 294–303, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-810-5. doi: <http://doi.acm.org/10.1145/1031607.1031655>.
- S.D. Scott, S. Carpendale, and S. Habelski. Storage bins: Mobile storage for collaborative tabletop displays. *Special issue of IEEE Computer Graphics and Applications*, 25(4): 58–65, July/August 2005.
- A. Sears and B. Shneiderman. High precision touch screens: Design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies*, 43(4):593–613, 1991.
- D. Semeraro, B. Wilhelmson, D. Bramer, J. Leigh, D. Porter, and J. Ahrens. Collaboration, analysis, and visualization of the future. In *Meteorological Data 20th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, 2004.
- Chia Shen, Neal B. Lesh, Frederic Vernier, Clifton Forlines, and Jeana Frost. Sharing and building digital group histories. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 324–333, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-560-2. doi: <http://doi.acm.org/10.1145/587078.587124>.
- B. Shneiderman. Touch screens now offer compelling uses. *IEEE Software*, 8(2):93–94, 107, 1991.
- A. Skopik and C. Gutwin. Improving revisitation in fisheye views with visit wear. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*,

- pages 771–780, New York, NY, USA, 2005. ACM Press. ISBN 1-58113-998-5. doi: <http://doi.acm.org.myaccess.library.utoronto.ca/10.1145/1054972.1055079>.
- SmartTechnologies. Smart technologies inc. digital vision touch technology <http://www.smarttech.com/dvit/> - visited december 2006, 2006.
- G. Smith and mc. schraefel. The radial scroll tool: scrolling support for stylus- or touch-based document navigation. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 53–56, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-957-8. doi: <http://doi.acm.org/10.1145/1029632.1029641>.
- R. Sommer. *Personal Space: The Behavioral Basis of Design*. Prentice-Hall, 1969.
- J.T. Stasko. Animation in user interfaces: principles and techniques. pages 81–101, 1993.
- R. Stoakley, M.J. Conway, and R. Pausch. Virtual reality on a wim: interactive worlds in miniature. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-84705-1. doi: <http://doi.acm.org/10.1145/223904.223938>.
- N. Streitz, J. Geißler, T. Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz, and R. Steinmetz. i-land: an interactive landscape for creativity and innovation. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 120–127. ACM Press, 1999.
- N.A. Streitz, J. Geißler, J.M. Haake, and J. Hol. Dolphin: integrated meeting support across local and remote desktop environments and liveboards. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 345–358, New York, NY, USA, 1994. ACM Press. ISBN 0-89791-689-1. doi: <http://doi.acm.org/10.1145/192844.193044>.

- K. Swaminathan and S. Sato. Interaction design for large displays. *interactions*, 4(1): 15–24, 1997. ISSN 1072-5520. doi: <http://doi.acm.org/10.1145/242388.242395>.
- J. Tam and S. Greenberg. A framework for asynchronous change awareness in collaborative documents and workspaces. *Int. J. Hum.-Comput. Stud.*, 64(7):583–598, 2006. ISSN 1071-5819. doi: <http://dx.doi.org/10.1016/j.ijhcs.2006.02.004>.
- D. Tan, D. Gergle, P. Scupelli, and R. Pausch. Physically large displays improve path integration in 3d virtual navigation tasks. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 439–446, New York, NY, USA, 2004a. ACM Press. ISBN 1-58113-702-8. doi: <http://doi.acm.org/10.1145/985692.985748>.
- D. Tan, B. Meyers, and M. Czerwinski. Wincuts: manipulating arbitrary window regions for more effective use of screen space. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1525–1528, New York, NY, USA, 2004b. ACM Press. ISBN 1-58113-703-6. doi: <http://doi.acm.org/10.1145/985921.986106>.
- D.S. Tan, J.K. Stefanucci, D. R. Proffitt, and R. Pausch. The infocockpit: providing location and place to aid human memory. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–4, New York, NY, USA, 2001. ACM Press. doi: <http://doi.acm.org/10.1145/971478.971526>.
- D.S. Tan, D. Gergle, P. Scupelli, and R. Pausch. With similar visual angles, larger displays improve spatial performance. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 217–224, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-630-7. doi: <http://doi.acm.org/10.1145/642611.642650>.
- M. Tani, M. Horita, K. Yamaashi, K. Tanikoshi, and M. Futakawa. Courtyard: integrating shared overview on a large screen and per-user detail on individual screens. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing*

- systems*, pages 44–50, New York, NY, USA, 1994. ACM Press. ISBN 0-89791-650-6. doi: <http://doi.acm.org/10.1145/191666.191690>.
- B. Thomas and P. Calder. Applying cartoon animation techniques to graphical user interfaces. *ACM Trans. Comput.-Hum. Interact.*, 8(3):198–222, 2001. ISSN 1073-0516. doi: <http://doi.acm.org/10.1145/502907.502909>.
- F. Tissot, K. Messing, and S. Stock. Standing, sitting and associated working conditions in the quebec population in 1998. *Ergonomics*, 48(3):249–269(21), 2005. doi: 10.1080/00140130512331326799.
- TOTal. http://www.city.chiba.jp/fire/english/system_e.html - visited 2006, 2006.
- J. Trimble, R. Wales, and R. Gossweiler. Nasas merboard: An interactive collaborative workplace platform. In K. OHara et al., editor, *Public and Situated Displays: Social and Interactional Aspects of Shared Display Technologies*, pages 18–44. Kluwer Academic Publisher, 2003.
- T. Tsandilas and R. Balakrishnan. An evaluation of techniques for reducing spatial interference in single display groupware. In *ECSCW’05: Proceedings of the ninth conference on European Conference on Computer Supported Cooperative Work*, pages 225–245, New York, NY, USA, 2005. Springer-Verlag New York, Inc. ISBN 978-1402040221.
- M. Turk. Computer vision in the interface. *Commun. ACM*, 47(1):60–67, 2004. ISSN 0001-0782. doi: <http://doi.acm.org/10.1145/962081.962107>.
- G.D. Venolia, L. Dabbish, J.J. Cadiz, and A. Gupta. Supporting email workflow. Technical Report Tech Report MSRTR-2001-88, Microsoft Research, 2001.
- F. Vernier, N.B. Lesh, and C. Shen. Visualization techniques for circular tabletop interfaces. In *ACM Advanced Visual Interfaces (AVI 2002)*, May 2002.

- R. Vertegaal, A. Mamuji, C. Sohn, and D.Cheng. Media eyepliances: using eye tracking for remote control focus selection of appliances. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1861–1864, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-002-7. doi: <http://doi.acm.org/10.1145/1056808.1057041>.
- D. Vogel and R. Balakrishnan. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 137–146, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-957-8. doi: <http://doi.acm.org/10.1145/1029632.1029656>.
- D. Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 33–42, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-271-2. doi: <http://doi.acm.org/10.1145/1095034.1095041>.
- X. Wang and J. Jin. A quantitative analysis for decomposing visual signal of the gaze displacement. In *Pan-Sydney area workshop on visual informtaion processing conference on visual information processing*, pages 153–159, 2001.
- Y. Wang and C.L. MacKenzie. The role of contextual haptic and visual constraints on object manipulation in virtual environments. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 532–539, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-216-6. doi: <http://doi.acm.org/10.1145/332040.332494>.
- C. Ware and M. Lewis. The dragmag image magnifier. In *CHI '95: Conference companion on Human factors in computing systems*, pages 407–408, New York, NY, USA, 1995. ACM Press. ISBN 0-89791-755-3. doi: <http://doi.acm.org/10.1145/223355.223749>.
- B. Wei, C. Silva, E. Koutsofios, S. Krishnan, and S. North. Visualization research with

- large displays. *IEEE Computer Graphics and Applications*, 20(4):50–54, /2000. URL citeseer.ist.psu.edu/wei00visualization.html.
- A. Wexelblat and P. Maes. Footprints: history-rich tools for information foraging. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 270–277, New York, NY, USA, 1999. ACM Press. ISBN 0-201-48559-1. doi: <http://doi.acm.org/10.1145/302979.303060>.
- D. Wigdor, D. Leigh, C. Forlines, S. Shipman, J. Barnwell, R. Balakrishnan, and C. Shen. Under the table interaction. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 259–268, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-313-1. doi: <http://doi.acm.org/10.1145/1166253.1166294>.
- A. Wilson and H. Pham. Pointing in intelligent environments with the world cursor. In *Proceedings of INTERACT 2003*, 2003.
- T. Winograd and F. Guimbretière. Visual instruments for an interactive mural. In *CHI '99 extended abstracts on Human factors in computer systems*, pages 234–235. ACM Press, 1999.
- J.K. Witt, D.R. Proffitt, and W. Epstein. Tool use affects perceived distance but only when you intend to use it. *Journal of Experimental Psychology: Human Perception and Performance*, 31:880–888, 2005.
- J. Woodring and H.W. Shen. Chronovolumes: a direct rendering technique for visualizing time-varying data. In *VG '03: Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics*, pages 27–34, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-745-1. doi: <http://doi.acm.org/10.1145/827051.827054>.
- D. D. Woods and J. C. Watts. How not to have to navigate through too many displays.

- In *Handbook of Human-Computer Interaction*. The Netherlands: Elsevier, Amsterdam, 2nd ed edition, 1997.
- M. Wu and R. Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 193–202, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-636-6. doi: <http://doi.acm.org/10.1145/964696.964718>.
- M. Wu, C. Shen, K. Ryll, C. Forlines, and R. Balakrishnan. Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 185–192, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2494-X. doi: <http://dx.doi.org/10.1109/TABLETOP.2006.19>.
- S. Zhai. User performance in relation to 3d input device design. *SIGGRAPH Comput. Graph.*, 32(4):50–54, 1998. ISSN 0097-8930. doi: <http://doi.acm.org/10.1145/307710.307728>.