

Waugh, Kieran David (2026) *Designing efficient interaction techniques for mid-air gesture interfaces.* PhD thesis.

https://theses.gla.ac.uk/86296/

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This work cannot be reproduced or quoted extensively from without first obtaining permission from the author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Enlighten: Theses
https://theses.gla.ac.uk/
research-enlighten@glasgow.ac.uk

Designing Efficient Interaction Techniques for Mid-Air Gesture Interfaces

Kieran David Waugh

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow



May 2026

Abstract

Mid-air gesture interaction allows users to control touchless displays without physical contact. Most touchless displays adopt the cursor-based pointing paradigm used with conventional input modalities, which demands a level of precision that is challenging for an unsupported hand in mid-air. Precise cursor targeting is challenging with an unsupported hand due to instability and fatigue. Furthermore, cursor activation suffers from a lack of clear feedback for initiating and confirming state changes. This thesis investigates novel interaction techniques that improve the efficiency of cursor-based touchless interaction without departing from the familiar cursor-and-click paradigm, addressing: the gesture used to activate the cursor (e.g., for target selection), the targeting mechanism used to acquire targets with the cursor, and the control-display gain that determines the rate of hand to cursor movement.

Experiments 1 and 2 compared four activation gestures (push, cursor dwell, and two variations of finger-to-thumb pinch) across single and multi-widget input tasks. These experiments found that pinch acquired targets faster than push, was less vulnerable to target slips, and was the most preferred technique, while dwell was poorly suited to continuous control.

Experiments 2 and 3 investigated targeting with dynamic area cursors, using the *proxemic cursor*, a Bubble Cursor-style technique which automatically targets the widget closest to the cursor without requiring that the cursor precisely overlap the target. The proxemic cursor reduced widget acquisition and transition times without sacrificing accuracy, and users adapted their targeting strategy to the layout rather than reverting to standard point-cursor selection.

Experiments 4 and 5 characterised the joint effect of area cursor targeting and control-display gain, across different configurations of display size and target density. These experiments found that that increased gain significantly reduced hand travel without reducing accuracy when paired with the proxemic cursor. However, dense layouts left less room for the area cursor to absorb the input noise that increased gain amplifies. The findings show that an area cursor and increased gain form a mutually beneficial combination.

These findings show that by reducing the precision and physical demands of activation, targeting, and traversal on mid-air input, users can interact quicker and with less physical effort, making touchless input a more viable option for everyday computing tasks on planar displays.

Contents

Abstract **i**

Abbreviations **xii**

Acknowledgements **xiii**

Declaration **xv**

1 Introduction **1**

1.1 Motivation 1

1.2 Research Problem & Scope 2

1.2.1 Gestures are Unfamiliar and Lack Signifiers 3

1.2.2 Cursor Activation can be Ambiguous and Induce Inefficiency 4

1.2.3 Cursor Control can be Inefficient and Imprecise 5

1.3 Thesis Statement 6

1.4 Thesis Research Questions 6

1.5 Contributions 6

1.6 Thesis Structure 7

2 Literature Review **9**

2.1 Introduction 9

2.2 Mid-Air Gesture Interactions in HCI 9

2.2.1 Early Foundations of Gesture Input 9

2.2.2 A Brief Consumer History 10

2.2.3 Defining “Mid-Air”, “Touchless”, and “Spatial” Interaction 11

2.2.4 Applications of Touchless Displays 13

2.2.5 Cursor-Based Pointing and WIMP 16

2.3 Mid-Air Cursor-Based Pointing 17

2.3.1 Theoretical Foundations of Cursor-and-Click 17

2.3.2 Fitts’ Law and Target Acquisition 18

2.3.3 Continuous Control and The Steering Law 19

2.4 Human Factors of Touchless Mid-Air Interaction 20
2.4.1 Absence of Physical Affordance and Support 20
2.4.2 Fatigue and Consumed Endurance 21
2.4.3 Precision, Instability and Target Slips 23
2.4.4 Standardisation, Gesture Vocabularies, and Learnability 24
2.4.5 Summary 25
2.5 Mid-Air Activation Techniques 26
2.5.1 Push: A Depth-Based Activation Gesture 26
2.5.2 Dwell: Temporal Activation via Steadied Cursor 27
2.5.3 Pinch: A Finger-to-Thumb Mode Switch Gesture 28
2.5.4 Alternative and Multimodal Activation Approaches 29
2.5.5 Summary 29
2.6 Cursor Movement Mechanics 30
2.6.1 Area Cursors & Dynamic Area Cursors 31
2.6.2 Control Display Gain 34
2.6.3 Summary 36
2.7 Overall Summary and Research Gap 37

3 Experiment 1: Activation Gestures 39
3.1 Introduction 39
3.1.1 Push Gestures 40
3.1.2 Mid-Air Continuous Interactions 41
3.2 Chapter Structure 41
3.3 Interaction and Gesture Design 41
3.3.1 Gesture Design 42
3.3.2 Visual Feedback 45
3.4 Experiment 1 45
3.4.1 Research Aims 45
3.4.2 Experiment Design 46
3.4.3 Task Design 47
3.4.4 Apparatus 47
3.4.5 Measures 48
3.4.6 Procedure 50
3.5 Results 50
3.5.1 Task Time 50
3.5.2 Time to Acquire 51
3.5.3 Time to Target 51
3.5.4 Error Distance 52
3.5.5 Success Rate 53

3.5.6 Survey Results . . . 54
3.5.7 Preference Ranking . . . 54
3.6 Discussion . . . 56
3.6.1 Acquiring Slider Control . . . 56
3.6.2 Using Area vs. Precise Targeting . . . 57
3.6.3 Enhanced Cursor Targeting and Translation . . . 59
3.7 Chapter Conclusions . . . 59
3.7.1 Chapter Contributions . . . 60
3.8 Research Question 1 . . . 61

4 Experiments 2 & 3: The Proxemic Cursor 62
4.1 Introduction . . . 62
4.1.1 Area Cursor Interaction . . . 63
4.2 Chapter Structure . . . 64
4.3 Chapter Experimental Design . . . 65
4.4 Interaction Design . . . 65
4.4.1 The Proxemic Cursor . . . 66
4.4.2 Gesture Design . . . 67
4.4.3 Visual Feedback . . . 69
4.5 Measures . . . 69
4.6 Implementation and Apparatus . . . 71
4.7 Experiment 2: Multiple Slider Input . . . 72
4.7.1 Research Aims . . . 72
4.7.2 Experiment Design . . . 72
4.7.3 Task Design . . . 73
4.7.4 Procedure . . . 74
4.7.5 Results . . . 74
4.8 Experiment 3: Discrete and Continuous Interactions . . . 86
4.8.1 Research Aims . . . 86
4.8.2 Experiment Design . . . 86
4.8.3 Task Design . . . 87
4.8.4 Procedure . . . 88
4.8.5 Results . . . 88
4.9 Overall Discussion . . . 101
4.9.1 Area Cursor Usability . . . 101
4.9.2 Targeting Behaviours . . . 102
4.9.3 Activation Gestures . . . 104
4.9.4 Layout and Widget Spacing . . . 105
4.9.5 Design Implications . . . 106

4.10 Chapter Conclusions . . . 106
4.10.1 Chapter Contributions . . . 107
4.11 Research Questions . . . 108

5 Experiments 4 & 5: Control-Display Gain 110
5.1 Introduction . . . 110
5.1.1 Control-Display Gain . . . 111
5.1.2 Target Density and Area Cursor Behaviour . . . 112
5.2 Chapter Structure . . . 113
5.3 Chapter Experimental Design . . . 113
5.4 Interaction Design . . . 113
5.4.1 Area Cursor . . . 113
5.4.2 Control Display Gain Mapping . . . 114
5.4.3 Activation Gesture . . . 114
5.4.4 Simulating Display Size . . . 115
5.4.5 Visual Feedback . . . 115
5.5 Implementation and Apparatus . . . 116
5.6 Experiment 4: Display Size x Gain . . . 116
5.6.1 Experimental Design . . . 117
5.6.2 Task Design . . . 117
5.6.3 Procedure . . . 120
5.6.4 Results . . . 120
5.7 Experiment 5: Display Size x Gain x Density . . . 136
5.7.1 Experimental Design . . . 136
5.7.2 Task Design . . . 137
5.7.3 Procedure . . . 138
5.7.4 Results . . . 138
5.8 Overall Discussion . . . 152
5.8.1 Dynamic Area Cursors Support Increased Gain Levels . . . 152
5.8.2 Designing Mid-Air Cursor Interactions and User Interfaces . . . 157
5.9 Chapter Conclusions . . . 160
5.9.1 Chapter Contributions . . . 161
5.10 Research Questions . . . 162

6 Thesis Discussion 164
6.1 Introduction . . . 164
6.2 Three Techniques, Three Residual Problems . . . 164
6.2.1 Findings Across The Experiments . . . 165
6.3 Where This Work Sits: Other Modalities and Paradigms . . . 166

6.3.1 The Commercial Introduction of Pinch Interactions 166
6.3.2 Multimodal Interactions 167
6.3.3 Towards a More General Theory of Interaction 168
6.4 Looking Into The Future: Reflecting On What Stands 168

7 Conclusions & Recommendations 170
7.1 Introduction 170
7.2 Research Questions 171
7.3 Thesis Contributions 173
7.4 Designing Efficient Interaction Techniques for Mid-Air Gesture Interfaces . . . 175
7.5 Thesis Conclusion 180

Appendices 182

A Experiment 1 183

B Experiment 2 184

C Experiment 4 185

D Experiment 5 186

List of Tables

3.1 Summary of Likert scale responses. Note responses were on a scale from 1 ("Strongly Disagree") to 7 ("Strongly Agree"). 54
3.2 ANOVA results, and post-hoc test results for Experiment 1. Significant effects are highlighted* 55

4.1 ANOVA results, and post-hoc test results for Experiment 2. Significant comparisons are highlighted* 83
4.2 ANOVA results, and post-hoc test results for Experiment 3. Significant comparisons are highlighted* 99

5.1 Percentage decrease in mean hand movement distance across Control-Display Gain (CDG) multipliers and screen sizes. 124
5.2 ANOVA results, and post-hoc test results for task 1. Significant comparisons are highlighted* 127
5.3 Percentage change in mean hand distance across CDG and screen sizes. Negative values indicate an increase in hand distance. 133
5.4 ANOVA results, and post-hoc test results for task 2. Significant comparisons are highlighted* 134
5.5 Percentage decrease in mean hand movement distance across CDG multipliers and screen sizes. 144
5.6 ANOVA results, and post-hoc test results for Experiment 5. Significant comparisons are highlighted* 149
5.7 Summary of design recommendations for mid-air cursor interfaces. 160

List of Figures

1.1 Screenshots from the films *Minority Report* (2002, image ©Twentieth Century Fox Film Corporation & DreamWorks Productions, LLC) and *Iron Man 2* (2010, image ©MVL Rights, LLC) depicting mid-air interaction as a futuristic input method for engaging with large holographic display configurations. 1

2.1 Milestone commodity mid-air interaction devices 12
2.2 Example of COVID-19 wariness over shared touch displays. Image taken by the author in the National Museum of Scotland, May 2021 14
2.3 Example of a distal touchless interaction. 15
2.4 Illustration of three common activation gestures. 26
2.5 Comparison of area cursor and dynamic area cursor targeting. 32
2.6 Illustration of how control-display gain affects cursor movement. 34

3.1 Visualisation of the interaction environment, showing the relationship between hand movement in mid-air and cursor movement on-screen 42
3.2 Illustration of the four selection techniques. The cursor position was mapped to the centre of the palm. Note that while these images show an extended index finger, this was not required. 43
3.3 Illustrations of the visual feedback states 46
3.4 Screenshots showing each task type as shown to the participants. The semi-transparent red circle is the slider handle. For the digit and letter tasks, slider ticks were rendered to indicate target width. For the scale task, participants had to use the slider to adjust the size of the semi-transparent red shape so that it matched the black shape. 48
3.5 Hardware used for the experiment: 27” monitor, Leap Motion Controller and wireless keyboard 48
3.6 This timeline shows how task timing measures are calculated from interaction events, where T denotes time. 49
3.7 Mean task time per condition. Error bars show 95% CIs. 51
3.8 Mean time to acquire the slider handle per condition. Error bars show 95% CIs. 52
3.9 Mean time to reach the target per condition. Error bars show 95% CIs. 52

3.10 Mean error distance per condition. Error bars show 95% CIs. 53
3.11 Mean success rate (%) per condition. Error bars show 95% CIs. 53
3.12 Preference ranking distribution for each interaction technique. 55
3.13 Scatterplot showing distribution of cursor x-axis coordinate when the user first acquired control of the slider handle via the Area Pinch gesture. The x-axis shows the slider start, mid, and end points. The large dot shows mean start position. 58

4.1 Visualisation of the interaction environment, showing the relationship between hand movement in mid-air and cursor movement on-screen. 66
4.2 The nearest widget is defined as the minimum distance to the widget handle. In this case, the leftmost slider would be targeted. 67
4.3 Illustration of the two selection techniques with and without area targeting. The cursor position was mapped to the centre of the palm. Note that while these images show an extended index finger, this was not required as part of the tracking implementation. 68
4.4 Example of the in-focus visual state. Left slider is in-focus 69
4.5 This timeline shows how task timing measures are calculated from interaction events where T denotes time. 71
4.6 Hardware used for both experiment 2 and 3: 27-inch monitor, Leap Motion Controller and wireless button 71
4.7 Screenshots of the three slider layouts. Each slider has 26 regions, labelled from A to Z. 74
4.8 Mean task time per condition. Error bars show 95% CIs. 76
4.9 Mean time to acquire the first slider per condition. Error bars show 95% CIs. . 77
4.10 Mean time to transition between sliders per condition. Error bars show 95% CIs. 78
4.11 Mean time to arrive at the target per condition. Error bars show 95% CIs. . . . 79
4.12 Mean error distance for both sliders per condition. Error bars show 95% CIs. . 80
4.13 Mean success rate for both sliders per condition. Error bars show 95% CIs. . . 81
4.14 Mean activation distance both sliders per condition. Error bars show 95% CIs. . 81
4.15 Area cursor positions when the participant took control of the slider handles with brighter colours indicating greater density of cursor positions. 82
4.16 Mean cursor displacement caused by the Heisenberg effect. Error bars show 95% CIs. 83
4.17 Preference ranking distribution for each interaction technique. 83
4.18 Experiment 3 task screenshot. There are 26 slider notches (A–Z) and three buttons (1–3). Shows a vertical and horizontal separation of 64px and 256px, respectively. 87
4.19 Mean task time per condition. Error bars show 95% CIs. 90

4.20 Mean time to acquire the first widget per condition. Error bars show 95% CIs. . 90
4.21 Mean time transition between the first and second widget, per condition. Error bars show 95% CIs. 91
4.22 Mean time to hit the slider target, per condition. Error bars show 95% CIs. . . . 92
4.23 Mean slider error distance, per condition. Error bars show 95% CIs. 93
4.24 Mean distance participants took control over the slider, per condition. Error bars show 95% CIs. 95
4.25 Mean distance participants activated the buttons, per condition. Error bars show 95% CIs. 96
4.26 Cursor positions at the time of widget activation, with brighter colours marking higher density areas. Slider and button activations are marked by circles and triangles, respectively. 97
4.27 Heisenberg displacement per condition. Error bars show 95% CIs. 98
4.28 Mean task success rate per condition. Error bars show 95% CIs. 99

5.1 Visualisation of the interaction environment, showing the relationship between hand movement in mid-air and cursor movement on-screen for a CDG multiplier g. 115
5.2 Example of the simulated display sizes. Black bars are disabled areas of the 55" display 115
5.3 Hardware set-up used for both Experiment 4 and 5: a 55-inch display at head height, with an Ultraleap Stereo IR 170 and wireless button positioned on a table in front of the user. 117
5.4 Examples of both tasks used in Experiment 4: (a) a slider task that involved matching the size of a target square by moving the slider handle; and (b) a selection task comprising 15 sequential target selections. 118
5.5 Mean task time per condition. Error bars show 95% CIs. 121
5.6 Mean time to reach the target per condition. Error bars show 95% CIs. 122
5.7 Mean error distance per condition. Error bars show 95% CIs. 123
5.8 Mean time to reach the target per condition. Error bars show 95% CIs. 124
5.9 Mean slider handle activation distance per condition. Error bars show 95% CIs. 125
5.10 Mean success rate per condition. Error bars show 95% CIs. 126
5.11 Borg CR10 Perceived fatigue scale per condition. Error bars show 95% CIs. . . 127
5.12 Mean total Task Time for all selections per condition. Error bars show 95% CIs. 129
5.13 Mean selection time per condition. Error bars show 95% CIs. 130
5.14 Mean error rate per task. Error bars show 95% CIs. 131
5.15 Mean Selection Distance using the area cursor per trial. Error bars show 95% CIs. 132
5.16 Mean hand movement distance per trial. Error bars show 95% CIs. 133
5.17 Borg CR10 Perceived fatigue scale per condition. Error bars show 95% CIs. . . 134
5.18 Experiment 5 task layouts ranging from density values 0.075 (left) to 0.6 (right) 138

5.19 Mean Selection Time per trial. Error bars show 95% CIs. 140
5.20 Mean Error rate per trial. Error bars show 95% CIs. 141
5.21 Mean Sensitivity to distractors widgets per trial. Error bars show 95% CIs. . . . 142
5.22 Mean selection distance of targets using the area cursor per trial. Error bars show 95% CIs. 143
5.23 Mean distance the hand moved per trial. Error bars show 95% CIs. 145
5.24 Borg CR10 Perceived fatigue scale per condition. Error bars show 95% CIs. . . 146
5.25 Perceived control of the cursor per condition. Error bars show 95% CIs. 147
5.26 Perceived selection difficulty per condition. Error bars show 95% CIs. 148
5.27 Example touchless user interface layouts inspired by real-world examples [142, 157]: text-entry interface with high-density small targets (left); and menu interface with low density, and both small and large targets (right). 158

7.1 An illustration of the main design recommendations supported by this thesis. Mid-air cursor interfaces should use a control-display gain proportional to the display size and target density (DR3), a dynamic area cursor that allows proximal target selection (DR2), and a pinch activation gesture to confirm selection or change user interface state (DR1). Note the magnitude of the area cursor is only shown here for illustration 179

A.1 Agreement survey used in Experiment 1 183
A.2 Technique rank question used in Experiment 1 183

B.1 Technique rank question used in Experiment 2 184

C.1 Borg CR10 scale question used in Experiment 4 185

D.1 Perceived control and difficulty survey used in Experiment 5 186
D.2 Borg CR10 scale question used in Experiment 4 187

Abbreviations

HCI Human-Computer Interaction

CDG Control-Display Gain

XR Extended Reality

UI User Interface

WIMP Windows, Icons, Menus, Pointer

CE Consumed Endurance

Acknowledgements

I have been incredibly lucky to have amazing people in my life who, each in their own way, helped me throughout this PhD. My thanks here will never be enough.

Firstly, I would like to thank my supervisors, Euan Freeman and Mark McGill, for their guidance, encouragement and expertise throughout the entire PhD. I have been extremely fortunate to have them as supervisors and as friends. They took a chance on me, and I am a better researcher for it.

Secondly, I would like to thank my examiners, Eve Hoggan and Mathieu Chollet, and my convener, Ilyena Hirskyj-Douglas, for the great discussions and questions (hard ones included) during the viva. Together you created a comfortable atmosphere and made the viva an experience I really enjoyed.

Special thanks go to Julie Williamson, Stephen Brewster and Aidan Kehoe for their ongoing support and friendship. Each of you inspires me, and your constant encouragement has got me to where I am today. I hope we keep working together for a long time to come, because I have enjoyed every minute.

I have had the privilege of being part of more than one lab group during my time at Glasgow. First, the Multimodal Interaction Group, who welcomed me even though I was never actually a member! The weekly meetings, advice, brutal paper feedback, and steady supply of biscuits and cake were reason enough to keep coming. Second, the Fusion team, who showed me how to bring fun back to work while being some of the hardest workers I know. Their copious supply of beer and wine was, again, reason enough to keep coming! Thank you all for being there for me.

To my PhD family: first, my officemates (past and present), Graham, Jake, Diego, Laura, Iain and Joseph for all their chat and guidance throughout. Second to our next-door neighbours, Tom, Shaun, Paddy, Melvin and Ammar for their friendship and for showing me the lighter side of PhD life alongside an incredible work ethic. You all created the best work environment, and for that I'm always grateful. The Thursday drinks rant sessions will forever remain an integral part of my PhD experience (a special shoutout to the Grosvenor).

I would not be here without my friends, and no words of thanks can properly express how much I appreciate them. To Amy and Kate, for their love and support throughout the PhD

and for long before. From our undergraduate days of sending packets to recreate an image of Shrek, to surviving MIPS, I will never get over how far we have come. Whatever was in the air encouraging us all to do PhDs ought to be investigated! To Kat, for her advice, friendship and company: thank you for always being a phone call away, however much I miss having you nearby. I will always relish our (far too many) trips to the QMU Chinese takeaway. To Hugo and Kelsey, for their kindness and humour. To Jules, for his friendship and reliability, and for keeping me grounded over the years. Without you all, this thesis would not exist.

To my biggest supporters, my family (Mum, Dad, Emily, Aidan ... and Bonnie), who have always backed me in everything I do, no questions asked. Each of you has shaped me into who I am today and encouraged me to keep going. Especially my dad, who never stopped believing in me and always gave me the push I needed not to give up. You finally have your answer to "*are you actually done with university yet?*".

Finally, to Javi[1]. His only advice, given from the middle of his own PhD, was not to do a PhD. This thesis stands as evidence that I do not listen, and has loved and supported me. He will never quite understand how much he has helped me through these years. I am forever the luckiest to have him by my side — TQM.



And to our mascot RoVR, for simply existing.

[1] A good friend.

Declaration

The research presented in this thesis is entirely the author's own work. Research in this thesis has been published at the following venues; this thesis uses only the parts of these papers that are directly attributable to the author:

Experiment 1 in Chapter 3 was published and presented as a full paper at **NordiCHI 2022**: Kieran Waugh, Mark McGill, and Euan Freeman. 2022. *Push or Pinch? Exploring Slider Control Gestures for Touchless User Interfaces.* In Nordic Human-Computer Interaction Conference (NordiCHI '22). Association for Computing Machinery, New York, NY, USA, Article 37, 1–10. https://doi.org/10.1145/3546155.3546702

Experiments 2 and 3 in Chapter 4 have been published and presented as a full paper at **ACM SUI 2023**: Kieran Waugh, Mark McGill, and Euan Freeman. 2023. *Proxemic Cursor Interactions for Touchless Widget Control.* In Proceedings of the 2023 ACM Symposium on Spatial User Interaction (SUI '23). Association for Computing Machinery, New York, NY, USA, Article 21, 1–12. https://doi.org/10.1145/3607822.3614525

Experiments 4 and 5 in Chapter 5 have been published and presented as a full paper at **ACM CHI 2025**: Kieran Waugh, Mark McGill, and Euan Freeman. 2025. *Everything to Gain: Combining Area Cursors with increased Control-Display Gain for Fast and Accurate Touchless Input.* In Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25). Association for Computing Machinery, New York, NY, USA, Article 726, 1–16. https://doi.org/10.1145/3706598.3714021

Chapter 1

Introduction

1.1 Motivation

Science fiction has envisioned a future where users control digital content on large ubiquitous displays through mid-air hand movements, referred to here as **gestures** (e.g., Figure 1.1). While some aspects of these design fictions remain distant (e.g., ad-hoc holographic displays), new advances in spatial computing technology have taken steps towards bringing mid-air gesture interactions into everyday reality with existing personal computing devices and public displays. These everyday computing contexts will benefit from mid-air interaction as an alternative input modality, complementing other input modalities and devices by offering a style of interaction that may be more convenient, engaging, or efficient across a range of application areas.

Mid-air interaction has long attracted interest within the field of Human-Computer Interaction (HCI), with research first emerging in the 1980s and growing significantly since. In years since, consumer systems like the Nintendo Wii and Xbox Kinect brought mid-air interaction experiences into the home, and more recently, smartphones and Extended Reality (XR) devices have enabled mid-air gesture interaction on the go, across a much wider range of usage contexts.

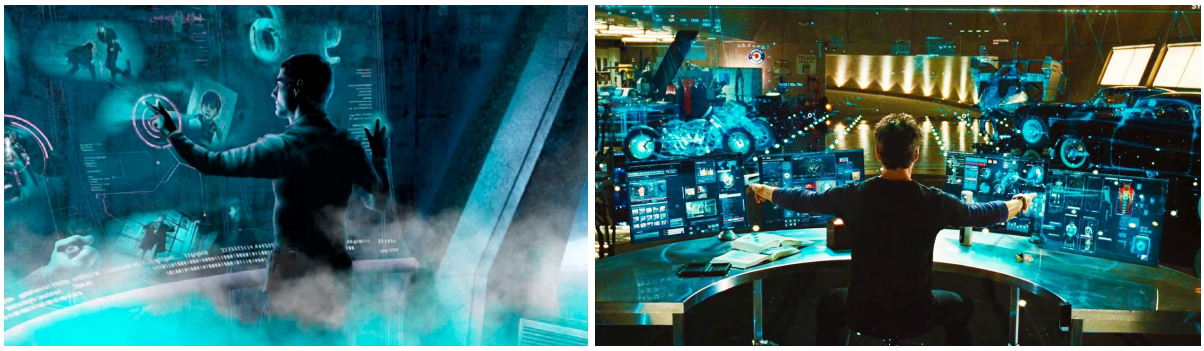


Figure 1.1: Screenshots from the films *Minority Report* (2002, image ©Twentieth Century Fox Film Corporation & DreamWorks Productions, LLC) and *Iron Man 2* (2010, image ©MVL Rights, LLC) depicting mid-air interaction as a futuristic input method for engaging with large holographic display configurations.

Beyond personal devices, a range of touchless sensing approaches have brought mid-air gesture to public displays, offering new ways of interacting without direct touch with a shared surface.

Mid-air interaction is a compelling input modality, though its benefits are yet to be fully realised. Contemporary sensing solutions and recognition approaches are highly capable and can be easily integrated into a range of computing products, though gesture has yet to see widespread adoption for computing tasks in everyday interaction scenarios. This thesis takes the position that fundamental usability issues mean mid-air gesture remains a challenging interaction modality, for novice and experienced users alike. It therefore aims to investigate interaction solutions that enhance usability and efficiency, taking steps towards realising visions that clearly attract interest in both science fiction and in modern day consumer technology.

1.2 Research Problem & Scope

Touchless input can enable users to interact with distal displays via mid-air gestures, giving an alternative to direct physical contact with the screen or without the use of additional input devices (e.g., handheld controllers, trackpads). Mid-air gestures afford a more expressive and dextrous form of input through increased degrees of sensing freedom [152], can enhance the hygiene of using public or shared displays by reducing the need for physical contact [42, 125, 138], and can allow interaction with content and displays that are beyond reach [58, 126, 168] (e.g., behind shop windows or enclosed in museum or art exhibits). Touchless input is now being explored across a range of everyday application areas in public display settings, including use in retail [158, 161], restaurants [160], tourism [151, 159, 174], and advertising [102]. Touchless technology has also been explored commercially in personal computing devices, e.g., with smartphones, laptops, desktop computing peripherals, and mixed reality headsets.

These numerous application areas often differ in terms of display size, interaction objectives, and required levels of precision, though most share the same fundamental interactions used to select and engage with content, through the use of 'pointing' at user interface elements. Touchless displays often use **cursor-based pointing interactions**, where users manoeuvre an on-screen cursor by moving their hand in mid-air. User interface widgets, such as buttons and sliders, can be targeted by the cursor then activated by a state change action, e.g., a gesture that indicates intent to activate. Enhancing the usability of these fundamental interactions remains a key challenge to more widespread adoption of the interaction modality.

These cursor-based interfaces are similar to existing pointing interfaces with conventional inputs (touchscreens, trackpads, mouse, etc). One factor that influences this is *legacy bias*, the tendency to utilise familiar design paradigms that are not necessarily ideally suited to the affordances of an input modality. Related to this is the desire to *retrofit new interaction modalities to existing interfaces*. Many touchless input peripherals (e.g., Leap Motion, Intel RealSense) were designed to work alongside existing computing systems, so that touchless input could be used

without redesigning the user interface; this drive for rapid adoption was especially compelling when touchless peripherals started to gain traction during the COVID-19 pandemic. Finally, use of existing cursor-based pointing interaction supports a goal of providing *parity across different input methods* so that users have comparable choice of input modalities; e.g., choosing between handheld controllers or hand-tracking to interact with a mixed reality headset, or choosing the most appropriate interaction modality for the user's desired level of engagement [141].

Cursor-based interaction is thus the predominant interaction paradigm for touchless displays, which introduces many interaction challenges – some of which this thesis will address – because the precision demands of this interaction style are not necessarily suited to free-hand mid-air input. This thesis investigates interaction adaptations that aim to improve the efficiency of cursor-based interaction with distal planar displays. It will not consider other forms of interaction (e.g., direct manipulation of spatial content, or discrete gesture-based shortcuts, etc) as these are outside the scope of this research. The following sections introduce key usability issues that affect touchless cursor control, some of which this thesis will address.

1.2.1 Gestures are Unfamiliar and Lack Signifiers

Although touchless input has gradually become more widely supported by mainstream computing devices, it remains unfamiliar to most users and the interfaces often lack in-situ affordances to help users discover how to interact successfully. In conventional interfaces, the physical appearance and tactile properties of an input device provide interaction *affordances* and *signifiers* [131, 132], and a *frame of reference* to guide the body [81, 186]. For mid-air interfaces, however, the sensing space in which users interact is not visible and typically lacks a well-defined boundary, offering fewer signifiers that indicate where the user can and should interact [56, 57, 102], e.g., the surface of a touchscreen shows the user where inputs can be registered, the form of a mouse button encourages 'clicking', though mid-air gestures lack similar cues.

Without prior experience and signifiers, users must make assumptions about how to interact effectively: e.g., about where to perform gestures [56, 57] and about which gestures should be performed to achieve a certain outcome [56]. When performing gestures, users must also rely on proprioception [120] to help guide their movements, at the risk of decreasing accuracy and increasing imprecision, especially for fine-motor tasks. Moreover, the lack of guidance could make it difficult for systems to distinguish between an intentional gesture and a natural, incidental movement [6, 85], often leading users to over-correct their movements [168].

Such usability issues are further complicated by the lack of standardisation in mid-air interface design [169] which mean users cannot necessarily rely on prior experience to help them learn to use new systems, new applications and new operations. Users may be familiar with how cursors behave in other contexts, through transferring knowledge from conventional interaction paradigms to a touchless context is not straightforward – e.g., the necessary action to 'activate'

or 'select' with the cursor does not reveal itself and so must be learned or communicated.

This thesis will not investigate methods for revealing interaction possibilities to users (e.g., through feedforward [166]). However, its focus on improving fundamental cursor interactions is a first step towards learning how to design and communicate effective cursor-based interactions for touchless displays. Many of these issues are established prior to interaction when the user *addresses* the system [16, 56] and other research contributions have explored solutions to help guide users towards successfully initiating interaction with a touchless display [42, 56, 57, 102]. This thesis contribution focuses more on what happens *after* the user has engaged the system and is control of a mid-air cursor.

1.2.2 Cursor Activation can be Ambiguous and Induce Inefficiency

A significant challenge in hand-tracked interactions is the absence of a binary physical state change equivalent to a mouse click or a screen tap [27, 171], which can impair the activation of user interface widgets (e.g., when activating a button or selecting a list item). With conventional input devices, the mechanical click of a mouse button, or the contact of a finger on a touchscreen, for example, provide both a tactile confirmation as feedback that an input action has occurred, and provides a clear temporal marker of intent to inform the system that the action happened on purpose. Mid-air interfaces lack such a clearly defined means of changing input state, which can create uncertainty for both user and system about when an intentional action has occurred.

A range of activation gestures have been explored in research and commercial systems. Many platforms utilise a 'push' gesture that supports the metaphor of pressing a realistic button in mid-air, a potentially engaging input metaphor, although one that introduces interaction issues. Since gestures are typically performed with the hand unsupported in-air, there is a risk of target slips [168] when the act of performing the push gesture causes the hand to move [181], inadvertently shifting the cursor off the intended widget before the activation action registers. Users may experience additional uncertainty about what exactly constitutes a push (e.g., its start and end points) since this mid-air metaphor lacks the familiar physical cues that support button pressing: e.g., feeling the surface of a button, the resistance against movement, and the point where it has fully 'clicked'. Another widespread alternative is the use of cursor dwell, where users instead activate targets by stabilising the cursor on top of the widget for a short period. This potentially enables the hand to remain still (to mitigate slips) though decreases interaction efficiency since throughput is limited by the dwell period. Moreover, users receive limited physical feedback as their hand moves the cursor on to, or off from, a target, making state change less noticeable. Hand tracking sensors can support a range of activation gestures and this thesis will consider finger-to-thumb pinch as a promising alternative for touchless displays.

This thesis investigates the use of finger-to-thumb pinch gestures for activation on touchless displays, comparing this with alternatives found in commercial systems (e.g., push gestures and cursor dwell). Its results will show that users can utilise pinching as an effective state-change

mechanism for selection when using the point-and-click interaction paradigm typical in most touchless interfaces.

1.2.3 Cursor Control can be Inefficient and Imprecise

Another critical barrier for mid-air interaction is the reduced efficiency and precision compared to conventional input methods. Previous works have extensively highlighted relatively longer interaction times for spatial computing tasks, marking them as unsuitable for extended interactions [81]. Longer interaction time largely stems from precision issues, where users must make smaller, more frequent adjustments to combat the inherent instability of maintaining a raised hand. This requires corrective and stabilising adjustments needed when, e.g., selecting a target, thereby reducing interaction efficiency and prolonging task time.

Issues of inefficiency are particularly pronounced in environments where the interaction space is limited or when interface elements are out of reach. When displays are positioned farther away, users are forced to operate at the edges of the tracking area, often requiring UI scaling adjustments. Consequently, these interactions are impaired by the precision-fatigue tradeoff [80, 168]. As the duration of mid-air selection increases to ensure accuracy, so does physical fatigue arising from increased muscle tension (i.e., the "gorilla arm" effect of raising the arm for extended periods) and lack of a physical surface to support the hand or arm [80, 185]. This fatigue, in turn, impairs fine motor skills, making precise selection increasingly difficult and less comfortable. Fatigue can create a negative feedback loop where the user becomes slower and less precise the longer they interact [87]. To mitigate these precision issues, interfaces could be redesigned (e.g., scaled up) to compensate for the difficulty of selecting standard-sized UI elements. However, this reduces parity between input devices in multimodal systems (i.e., most contemporary touchless systems that get used alongside other input devices) and would be a detriment to the UI; expanding buttons and widgets often reduces information density, resulting in a less efficient user interface.

This thesis explores interaction design solutions that aim to mitigate inefficiency without requiring whole user interface redesign. It considers, specifically, the use of dynamic area cursors as an interaction method that can allow for more efficient and less precise targeting, with a goal of reducing interaction time and increasing user comfort. Dynamic area cursors allow implicit targeting of UI elements without needing to overlap them. Furthermore, this thesis investigates use of increased gain ratio between hand and cursor movement, as an additional method of increasing efficiency via reduced physical movement demand. In conjunction, these solutions could offer more efficient and precise touchless interactions.

The experiments in this thesis do not include a mouse or touchscreen baseline. The performance gap between mid-air and surface input is already well established by direct comparisons [17, 52, 88], and this thesis does not claim that its techniques close that gap. Touchless input is typically used where a mouse or touchscreen is unavailable, impractical or undesirable

(e.g., for hygiene, or with displays beyond reach), so the relevant comparison is against current mid-air practice rather than against other devices. Each experiment therefore uses a mid-air baseline: the Push gesture, a standard point cursor, and a 1:1 control-display mapping. The findings should be read as improvements relative to those baselines, not as evidence of parity with conventional input.

1.3 Thesis Statement

This thesis demonstrates that using finger-to-thumb **pinch gestures** for state change activation, in conjunction with **increased control-display gain** and **dynamic area cursor pointing**, can improve input efficiency in cursor-based touchless interfaces. This combination of interaction techniques reduces the need for pointing precision and supports relaxed targeting behaviours, whilst also mitigating concerns relating to depth ambiguity often seen in spatial interfaces. The benefits of these techniques scale across a range of interface configurations, e.g., from sparse to dense target layouts, from small to large displays, and for both activation and manipulation operations. As a result, users can interact more quickly and with less physical effort, making touchless input a more viable option for everyday computing tasks on planar displays.

1.4 Thesis Research Questions

This thesis aims to answer the following research questions:

RQ1 - How does the choice of target activation technique affect the usability of a mid-air user interface?

RQ2 - How can we reduce the need for traversal and selection accuracy in a mid-air user interface across different display sizes and widget densities?

RQ3 - How can we design mid-air interactions such that the required effort is reduced while interaction efficiency is increased?

1.5 Contributions

This thesis makes novel contributions in three main areas. First, it presents two experiments (Experiment 1, Experiment 2) that directly examine how target activation technique affects usability in mid-air cursor-based interfaces. While a push-to-activate gesture is a common industry-standard activation method, users experience uncertainty about state change and may have difficulty maintaining a consistently activated state, which can lead to target slips and a diminished sense of control. Modern hand-tracking sensors can detect richer and more fine-grained poses

like finger-to-thumb pinching, allowing alternative activation gestures which are less prone to these issues. These first two experiments compare the conventional 'push' gesture with alternatives, including cursor dwell and 'pinch', finding that pinch gestures are a more effective alternative because they have clearly defined states (i.e., pinching, not pinching) and enable the hand (and therefore the cursor) to remain steady during activation, helping mitigate target slips and avoiding ambiguous hand movement to/from the display.

Second, this thesis contributes a novel investigation into the use of dynamic area cursors to improve targeting efficiency and reduce the physical effort required for mid-air interaction. Current mid-air cursor-based systems require precise aiming, a process that can be slow, fatiguing, and prone to errors. This thesis addresses this limitation by proposing and evaluating the *proxemic cursor*, a dynamic area cursor variant that automatically targets the widget closest to the user-controlled cursor without requiring the cursor to be directly overlapping the target. Two experiments (Experiment 2, Experiment 3) compared the performance of the area cursor against a standard point cursor across tasks involving both continuous slider manipulation and discrete button selections, demonstrating that area cursors can significantly enhance usability by allowing users to acquire control of interface elements more quickly via more relaxed movements.

Lastly, this thesis contributes an investigation into the optimisation of cursor control within mid-air interfaces, particularly for large displays and interfaces with varying target densities. While area cursors reduce the need for precise aiming, significant cursor traversal across a spatial environment can still be physically demanding, and simply increasing the control-display gain to reduce arm movement typically results in a loss of precision. Two experiments (Experiment 4, Experiment 5) investigate this trade-off and characterise the complex relationship between an area cursor, control-display gain, display size, and target density, revealing a mutually beneficial combination that supports effective target selection. This allows for more efficient and accurate interaction with smaller and more controlled hand movements, even on large displays with dense target layouts.

1.6 Thesis Structure

Chapter 2 discusses related research. It introduces the current state of touchless interface technology, reflects on key usability challenges that currently limit widespread adoption of touchless input as a mainstream input modality, and introduces novel interaction techniques that could be used to improve touchless input performance.

Chapter 3 presents **Experiment 1**, which investigates the effect of activation method (push, dwell, pinch, area cursor pinch) on usability across three slider manipulation tasks. Its findings partially answer **RQ1**, showing that pinch - especially with area cursor characteristics - offers an effective method for selecting and manipulating widgets on a touchless display.

Chapter 4 presents **Experiment 2** and **Experiment 3**, which both investigate the effect of area

cursor targeting on slider manipulation and button activation. Their findings partially answer **RQ1** and **RQ2**, showing that area cursor targeting aids target acquisition and supports efficient input without requiring extraneous pointing movements to reach targets.

Chapter 5 presents **Experiment 4** and **Experiment 5**, which both investigate the complex relationship between control-display gain and area cursor behaviour, across a range of interface configurations comprising different sized screens and target densities. Their findings answer **RQ2** and **RQ3**, showing increased gain and area cursors to be mutually beneficial by enabling users to more rapidly reach distant targets whilst mitigating instability.

Chapter 6 discusses the thesis as a whole: how the interaction techniques proposed in this thesis relate to one another, how the work compares to other input methods such as gaze-and-pinch in XR headsets, and which of its findings are likely to hold as hardware improves.

Chapter 7 concludes the thesis by reflecting on its experimental findings and distilling these into design recommendations about how to design efficient mid-air cursor interfaces.

Chapter 2

Literature Review

2.1 Introduction

Despite growing commercial deployment of touchless display systems that use mid-air cursor control for input, there are fundamental interaction challenges that remain largely unresolved and are likely to impact the long-term adoption and use of this interaction modality. Touchless displays inherit the conventions of conventional pointer interaction, yet mid-air hand input lacks the physical support, interaction affordances, tactile feedback, and binary state-change mechanisms that those conventions depend upon. This literature review situates those challenges within the broader HCI research landscape, beginning with a broad overview of mid-air touchless interaction (Section 2.2), then discussing touchless cursor interfaces (Section 2.3) and the human factors challenges that affect their usability (Section 2.4), e.g., the instability and ambiguity introduced by unsupported hand movement. This chapter then reviews a range of cursor activation methods (Section 2.5) and interaction techniques (Section 2.6) that in-part address some of the usability challenges of mid-air interaction, such as , dynamic area cursors and control-display gain. This thesis builds upon these foundational research topics as it aims to address the efficiency and precision deficits identified in the introduction and discussed further in this chapter.

2.2 Mid-Air Gesture Interactions in HCI

2.2.1 Early Foundations of Gesture Input

Mid-air gestures have attracted considerable interest in HCI research for decades, grounded in both psychological theory and physical ergonomics [92,137]. Gesture interfaces have the potential to leverage interaction patterns humans already use in everyday life when completing manual tasks and when communicating with other people. As McNeill [118] notes, hand gestures are not merely accessories to speech; they are fundamentally intertwined with cognitive processes

and language production. As such, gestures have been repeatedly considered for spatial computing contexts, where users can manipulate digital objects through actions that closely mirror real-world physical interactions (e.g., pinching, grasping, pushing virtual objects as though they were real) [8]. This potential to exploit familiar action patterns has inspired the vision that users will be able to interact "*like a natural*" [177] when engaging with user interfaces via their hands.

Whilst mid-air gestural input is often associated with modern consumer hardware (especially through its wider adoption in VR headsets), its conceptual foundations as a novel interaction modality were established decades earlier. Bolt's seminal *Put-That-There* system [20] demonstrated in the 1980s that mid-air pointing could successfully serve as an input channel alongside speech, enabling users to issue commands to a computer using gesture to augment speech in similar ways that they might offer instructions to a human. By relying on a magnetic sensor to track wrist position (in 3D space), the system allowed users to provide spatial context to verbal commands; e.g., when speaking the instruction "*Put that over there*", the user's pointing gesture resolved the inherent ambiguity of what "*that*" was and where "*there*" was located. Bolt's work demonstrated that pointing and gesture could serve as a compelling input channel for computing systems, laying the foundation for future mid-air interaction systems.

A few years later, the *DataGlove* [187], developed by Thomas Zimmerman, tracked both finger movement, and hand position and orientation, using a wearable glove. The device was an early implementation of fine motor control over virtual objects within a computing system. Zimmerman showed that with appropriate sensing approaches, computing systems could recognise complex gestural postures, which would enable users to perform dexterous manipulations of virtual objects with skill. In the early 1990s, the *Charade* system [14] made similar use of wearable tracking to enable gesture control over a presentation system, allowing speakers to use gestures to communicate to audience and computer simultaneously. Along with Bolt's research on multimodal pointing, these early works demonstrated a compelling potential for computer interfaces that use hand movements and postures to establish user intention.

2.2.2 A Brief Consumer History

Despite the promise shown by these early works on gesture interaction, it has taken several decades for mainstream consumer products to adopt mid-air input. The introduction of Nintendo's Wii videogame console (2006) brought gesture-based computer control to a wide general audience for the first time. The system controller (Wiimote), whilst requiring users to maintain a physical grip on a controller, utilised inertial motion and infrared sensors to map mid-air arm movements to on-screen actions, leading to an interaction vocabulary inspired by physical action metaphors. Users could, for example, swing their arm to roll a bowling ball or swing a tennis racquet, or point directly at the screen to select an object. While not a free-hand interaction (since users had to hold a peripheral), the Wii familiarised the general public with mid-air input on distal displays [23] and was a key mainstream commercial success of mid-air interaction.

Touchless interactions without a tracked peripheral later gained large-scale traction with the introduction of the Microsoft Kinect (2010) for the Xbox videogame console. By utilising depth-sensing cameras to track a 3D skeletal representation of the human body, the sensor allowed for direct gestural computing, tracked at a coarse level through larger skeletal joints (e.g., wrist, elbow, shoulder). The Kinect had a mixed reception as a gaming peripheral but quickly found success as a key tool within maker communities and the emerging *natural user interfaces* research domain [115] because the commodity hardware and strong software support made it possible to rapidly prototype novel gesture interfaces and interaction techniques. The increased interest in gesture interaction, driven by the Kinect sensor's wide and low-cost availability, demonstrated the potential of using body movements and gestures for everyday computing [91, 155].

While the Kinect worked well for full-body gesture input (due to coarse-grained skeleton tracking), it lacked the optical fidelity and sensing resolution required for reliable fine-grained, small-scale hand tracking (i.e., individual fingers). The Leap Motion controller (2013) addressed this limitation by utilising high-precision infrared sensors to capture sub-millimetre movements of the hands and fingers within a constrained tracking space. This shifted the interaction focus from large arm movements and whole body gestures to smaller and more precise hand gestures, allowing for dextrous and nuanced control without the need for wearable sensors (e.g., gloves) or physical peripherals (e.g., handheld controllers). The Leap Motion provoked a broad range of research exploring dexterous, free-hand gesture interactions that could exploit the numerous degrees of freedom offered by this kind of sensing [152].

In the years since, hand-tracking sensors have been explored in a variety of commercial devices with mixed success, including personal computing (e.g., HP Envy laptops with integrated Leap Motion sensing), smartphones (e.g., Samsung Air Gesture, Google Pixel Motion Sense), and in-vehicle interfaces (e.g., Volkswagen Gesture Controls). More recently, hand tracking has found success as a mainstream interaction modality for immersive computing, utilised by myriad virtual and augmented reality (XR) headsets [9], e.g., the Meta Quest headsets as the best-selling example. These devices have driven the widespread consumer adoption of mid-air peripheral-free gestures, establishing gesture interaction as a first-class input modality. Consequently, direct gesture interaction, such as pinching to select or sweeping to scroll, has become the default and preferred [5] input method for spatial computing environments, marking the realisation of mid-air gesture control in the mainstream consumer market.

2.2.3 Defining "Mid-Air", "Touchless", and "Spatial" Interaction

Across the timeline presented so far in this chapter, from Bolt's seminal research in the 1980s through to the modern commercial use of hand-tracking in off-the-shelf XR headsets, there has been a broad and often inconsistent vocabulary used to describe this style of interaction. Terms like *mid-air*, *touchless*, and *spatial* have been used interchangeably to describe this interaction paradigm, often used synonymously. This thesis does not consider these terms to be equivalent,



(a) Wiimote (b) Kinect (c) LeapMotion (d) Meta Quest 3

Figure 2.1: Milestone commodity mid-air interaction devices, spanning the last two decades of consumer experience of mid-air input.

and the differences between them matter for establishing the scope of this research contribution. Each term emphasises a different property of the same broad family of techniques.

Spatial interaction is the broadest of the three. Greenwold [68] characterises spatial computing as "*human interaction with a machine in which the machine retains and manipulates referents to real objects and spaces*", a use of three-dimensional space as a coordinate system shared between user and system rather than the specific input channel used to operate within it. Under this framing, spatial interaction encompasses tangible interfaces, body-tracked games, virtual and augmented-reality, and gaze and gesture-controlled environments.

Mid-air interaction narrows the framing to intentional body or hand movements performed in 3D space, absent any physical supporting surfaces. Surveys of gesture-based HCI [6, 169] and 3D User Interface (UI) [24] consistently use the term in this sense. The defining property is not the absence of contact with the display, but the absence of mechanical support for the hand/arm: the input signal is subject to the postural and perceptual constraints of a hand or arm raised against gravity (with or without a peripheral), regardless of whether the user is interacting with any 2D or 3D display (e.g. touchless kiosk, XR headset, volumetric display, etc). Many of the human-factors issues discussed later in this chapter are consequences of this condition, not of the display itself.

Touchless interaction is best understood as a type of mid-air interaction for a particular interface. Garzotto and Valoriani [62], define touchless interaction as "system control via body movements and gestures" without "physical contact with technology", explicitly contrasting it with handheld controllers and on-body wearables. In touchless-display literature [7, 34, 35, 45, 63], the term is more or less synonymous with mid-air interaction of a planar display: the user's hand is the input, the output remains a conventional two-dimensional screen, and the interaction is typically intended for use in shared or public space.

Based on the above, spatial interaction refers to any interaction in a 3D space; mid-air interaction refers to hand, arm, and body movements in 3D space without physical support; and touchless interaction refers to mid-air interaction specifically with 2D planar displays (hereafter referred to as "touchless displays"). This thesis explores **mid-air** interactions within **touchless display** contexts and uses both terms throughout the following chapters. Whenever it is

important to differentiate the two, the distinction is made explicitly.

2.2.4 Applications of Touchless Displays

While XR headsets are the current mainstream application for mid-air input, there has been significant interest in using mid-air input for interaction with touchless *planar displays* across many application domains, e.g., information kiosks, retail terminals, museum exhibits, transit ticket machines, and digital signage. In this usage context, users stand in front of a flat upright display – typically in public, occasionally shared or communal – and provides input without physical contact with the screen or another input device. Interaction typically takes place at close distance using hand and finger tracking to give the user the ability to communicate their intention to interact. This particular interaction context, i.e., mid-air gesture interaction with a touchless display, is the primary motivation for this thesis.

Public displays are often explored in the literature as a primary usage context for touchless input, given that conventional input methods (e.g., mouse, voice and touch) are often awkward or unavailable in these settings. However, deploying a mid-air interface "in the wild" on a public display presents many challenges. Foundational work by Vogel and Balakrishnan [168] identified key issues including precision under unsupported reach, fatigue, and the social cost of gestural input. These problems are compounded for inexperienced users encountering touchless displays without instruction. Ackad et al. [4] documented a stark gap between lab and observed usability, while Mäkelä et al. [124] noted that touchless input lacks the in-situ affordances needed to transition passers-by into engaged public display users, especially in the absence of a "honey-pot effect" [126] where they can learn spontaneously by observing others. The lack of obvious affordance contributes to uncertainty about how to 'address' mid-air gesture systems [16, 56] and some people may not even know that a display can be controlled through hand or body movements in the first place, even with explicit "call to action" signage which tries to encourage users to gesture [102]. Work is ongoing to address these issues (and this thesis contribution aims to mitigate some concerns about interaction inefficiency), towards a long-term vision of everyday public interfaces that can be controlled through gesture.

Hygiene

Touchless input has been seen as a potential solution to cleanliness, health and hygiene concerns around shared surfaces and computing devices. Prior to the COVID-19 pandemic, O'Hara et al. [134] explored the potential of touchless interaction in clinical settings, where sterility is a key concern that results in numerous interaction obstacles [42]. The potential benefits of gesture in clinical settings motivated the investigation of clinical HCI work, including the development of a touchless gesture vocabulary tailored to operating-room workflows and demonstrated feasibility for the manipulation of 2D and 3D medical images [61, 82]. Lopes et al. [107] evaluate



Figure 2.2: Example of COVID-19 wariness over shared touch displays. Image taken by the author in the National Museum of Scotland, May 2021

three-dimensional hand cursors against conventional mouse input for the exploration of medical datasets, with evaluation by senior surgeons and radiologists; the touchless interface supported faster and more fluent navigation, although tasks requiring fine precision remain better with mouse. Cronin et al. [42] also investigated the use of touchless gestures for navigating medical imaging systems, finding that medical professionals were receptive to touchless interaction because it gives them the ability to engage with images, notes and data directly and when needed.

The COVID-19 pandemic rapidly increased public awareness about the risks of shared touch surfaces (including screens, control panels and input devices) and inspired accelerated interest in the use of touchless technology as a safe solution for shared display use. As users grew wary of public surfaces (Figure 2.2), touchless alternatives became perceived as highly acceptable [42, 125, 138]. Touchless software solutions, including *Intel RealSense Touchless* and *UltraLeap TouchFree*, were rapidly developed to meet this demand, bringing the capacity to retrofit mid-air input to existing systems. However, despite the unfortunate circumstances that inspired visions of a touchless future, mid-air interaction did not achieve the widespread adoption that many envisaged. This thesis posits that fundamental usability issues, which still persist, limited adoption and use. Whilst many reports often present touchless input as a mature and market-ready technology [84, 110, 158], the academic consensus remains cautious [98, 164], with a need for novel interaction solutions to improve interaction efficiency and simplicity to mitigate concerns around lack of input precision and fatigue over longer term use.

Distal Input

Touchless interactions can also facilitate *distal display interaction*, where the displays are physically out of reach of the user and direct contact is either inconvenient or impossible. These include wall-sized displays [43, 86, 176], information displays mounted high in transit and retail environments, and content positioned behind protective glass, such as shop window installations [126] or enclosed museum and gallery exhibits [65]. Here, the primary benefit of touchless input is that it enables at-a-distance tracking of user action and intent, when other input modalities are not available. Direct touch is inherently not possible as the user cannot reach the display, and shared peripherals or controllers would be impractical and must be passed between users, and represents an additional artefact that the passer-by must first notice, pick up, and orient correctly before they can interact at all [168]. Mid-air input overcomes these concerns because the user interacts with from a position separate from the display, without any external peripherals.

Displays in public settings are placed for visibility, often elevated, recessed, or oriented for specific viewing angles that are not always conducive to direct touch interaction. Mid-air input, sensed by optical sensors co-located with the display, allows the display to remain wherever its content is best presented while still being interactive. Previous research on distal pointing has established the technical viability, e.g., Vogel and Balakrishnan [168] demonstrated that distant pointing using ray-cast with a cursor could substitute for direct touch on a wall display, Nancel et al. [128–130] have characterised the design space of mid-air pointing on wall-sized displays in considerable detail, and Markussen et al. [116] report on selection techniques designed specifically for the requirements of distal-display use.

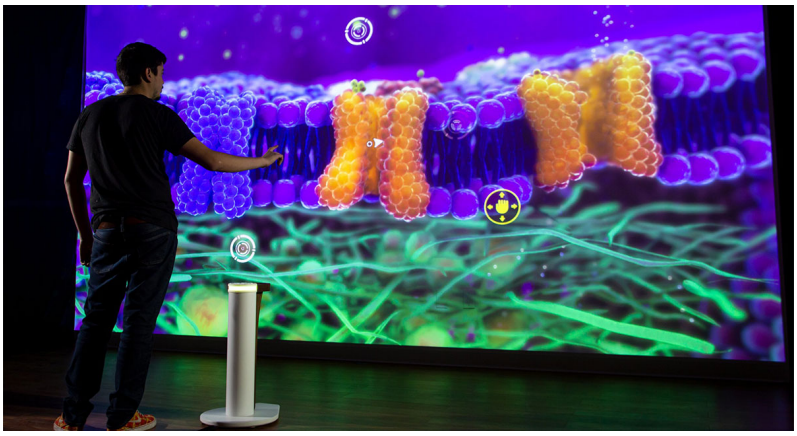


Figure 2.3: Example of a distal touchless interaction on a large wall-sized display. Image taken from Ideum[1], showing their touchless wall display.

[1] `https://ideum.com/news/touchless-pedestal-videowall-projection` - Last accessed May 2026

2.2.5 Cursor-Based Pointing and WIMP

Touchless displays predominantly rely on conventional cursor-based interactions with Windows, Icons, Menus, Pointer (WIMP) interfaces [4, 102, 158, 164], where users control an on-screen cursor or pointer which can be used to 'point at' and 'activate' user interface elements, analogous to using a pointing device to *point-and-click*. This reliance is largely driven by what Morris et al [121] term *legacy bias*, the inherent inclination of users, designers, and gesture-elicitation participants to replicate familiar input. Evidence of this is apparent in mid-air and touchless gesture-elicitation studies [69, 162, 169], which consistently reveal that users gravitate toward pointing, tapping, and dragging metaphors taken from mice and touchscreens. Consequently, designers face a significant constraint when attempting to diverge from these established norms. As Norman and Nielsen [1,133] warn, systems that abandon expected cursor semantics in favour of novel gestural vocabularies frequently suffer from poor discoverability, leading to user frustration and disengagement. Ultimately, this adherence to legacy interaction standards is not a failure of imagination, but rather a pragmatic choice. If a touchless display represents only a fraction of a user's daily technological interactions, the cognitive effort required to master a novel gestural language for a single system rarely justifies the expressive return.

Another pressure influencing this attraction towards cursor-based touchless interaction is the desire to retrofit touchless input to existing computer systems, introduced earlier. Commercial touchless software solutions, such as Ultraleap's TouchFree platform [110] and Intel RealSense Touchless [84], were designed to be added to existing computing systems without redesigning the user interface, and so deployed mid-air input as an alternative way of controlling the on-screen cursor supported at the operating system level.

Many of the public display case studies in which touchless interfaces have been deployed at scale [158, 160, 161] typically involve adapting existing kiosk interfaces to accept mid-air input, rather than designing the interface from scratch around the gesture modality. The cursor-and-click paradigm is, in this context, the path of least resistance: it requires little-to-no change to the UI, no change to the widget toolkit, and allows users to leverage their prior experience using WIMP interfaces. The pandemic-era interest in touchless deployment reinforced this, as the primary goal was to make existing interfaces operable without physical contact, rather than to design (and evaluate) wholly new mid-air interfaces.

An additional challenge, hinted at previously, is the demand for cross-modal parity with other input devices. Many touchless interface deployments, particularly in personal computing and mixed reality environments, are intended to operate alongside other input methods rather than replace them entirely. For example, Xbox Kinect often used cursor pointing for UI navigation, so that users could use gesture or controllers with the same interface layouts. Similarly, an XR headset user may alternate between controller-based and hand-tracked input within a single session, and a public kiosk user may interact with the same user interface via touch, a controller or keypad, or mid-air gesture. Designing for parity means that the underlying interface must be

operable through any of the supported inputs, which in practice means it must support the cursor-and-click semantics afforded by conventional input modalities. The result is that even when designers are not constrained by legacy bias or the need to retrofit, they often retain cursor-based pointing because doing otherwise would fragment the user experience across input methods.

This thesis will focus on the cursor-and-click interaction paradigm as this has emerged as the predominant form of touchless display interaction, and current trends, especially in immersive computing, point towards the continuation of this style of mid-air interaction. Recent years have seen great interest but ultimately muted uptake of touchless displays. This thesis believes that there are fundamental interaction issues that limit the usability and desirability of touchless displays, although interaction solutions that simplify input and increase efficiency could help drive future adoption. The follow sections of this literature review take a more detailed look at mid-air cursor interfaces (Section 2.3) and reflects on some of the human factors challenges that impact the usability of these (Section 2.4), before turning attention towards promising directions for interaction solutions.

2.3 Mid-Air Cursor-Based Pointing

2.3.1 Theoretical Foundations of Cursor-and-Click

Touchless systems predominantly inherit the established cursor-and-click interaction style, as discussed previously. The concept of the on-screen cursor originates from Engelbart, English, and Berman's [50] foundational work, which demonstrated the mouse's superiority over alternative input devices for on display selection. Card et al. [28] subsequently validated this via the first comparative Fitts' Law analysis in HCI, establishing that the mouse operated near the theoretical limits of human motor performance. In the right circumstances, cursor pointing can be an efficient way to navigate information and issue commands to a computing device.

The longevity of this interaction paradigm, which continues to be adapted and explored with novel input modalities (including touchless interfaces) stems from distinct human-motor and cognitive advantages. From a motor control perspective, indirect cursor control (i.e., an input device which is separate from the display and often oriented on a different plane) decouples the physical and visual spaces, enabling system designers to optimise movement and mappings [30, 66] whilst interacting in an ergonomic way (e.g., moving a mouse on a horizontal desk surface for comfort and support, rather than moving a mouse vertically in alignment with the display). For mid-air gestures, this decoupling is valuable; it can reduce fatigue, translating small hand movements into screen coordinates. Decades of interactive computing have shown humans to be highly adept at performing the necessary spatial transformations to support mental models of effective indirect pointing, a skill that also transfers from conventional cursor-control devices (e.g., mice and trackpads) to mid-air interaction.

From a cognition perspective, the cursor-and-click paradigm also benefits from the fact that the cursor visualises user intent prior to commitment. By separating cursor positioning from UI widget activation, computing systems are able to provide a continuous visual feedback loop. As established by Shneiderman [146] and Hutchins et al. [83], co-locating the cursor, the target object, and the available actions within a single visual field is what creates the perceived "directness" of graphical interfaces. In the absence of the tactile feedback that physical surfaces provide, this continuous visual information is essential for guiding users through unsupported, mid-air interactions [55]. Further, as discussed previously, touchless displays predominantly retain WIMP interaction conventions to facilitate rapid deployment and ensure parity with other input modalities. A mid-air cursor ensures this compatibility. The rest of this section breaks down some of the fundamental principles of cursor control and performance.

2.3.2 Fitts' Law and Target Acquisition

Perhaps the best known empirical interaction model in the domains of Human Factors and HCI is Fitts' Law [54], which models the time required for a rapid aimed movement given the size of a target and its position relative to an input cursor or end-effector. In his original 1954 experiments, Fitts observed participants performing reciprocal tapping movements between two strips of varying widths separated by varying distances. He found that the average movement duration was approximately linearly related to a logarithmic "Index of Difficulty," calculated from the target's distance and width. Meaning, movements to more difficult targets (i.e., those that are smaller or further away) require proportionally more time to target. This principle was applied in the context of interactive computing by Card et al. [28], who demonstrated that mouse positioning obeyed Fitts' Law, operating near the theoretical limit of human motor performance. The original 1D formulation of the reciprocal pointing task was later extended to more common 2D interactions used on computer displays [3, 114] varying target width and height, the type of interfaces for cursor point and click. Subsequent decades of methodological refinement [150] have resulted in a standardised evaluation protocol (ISO 9241-411) allowing pointing devices to be compared on a common quantitative scale. It accounts for the speed-accuracy trade-off, allowing techniques that sacrifice accuracy for speed to be objectively compared. Ultimately, it reinforces that making a target effectively closer or larger inherently speeds up interaction.

Aspects of Fitts' Law have been applied to mid-air control, revealing a narrative of how traditional pointing translates (and degrades) within touchless and 3D environments. Vogel and Balakrishnan's [168] early freehand pointing study established a baseline for unsupported mid-air interaction highlighting poorer performance in freehand selection tasks. Building on this, Kopper et al. [96] fundamentally shifted the model for distal pointing, demonstrating that movement time scales quadratically with angular, rather than linear, target amplitude and size. Later studies used the ISO 9241-9 multi-directional tapping task to compare hand-tracked mid-air pointing directly against conventional devices. Bessa Seixas et al. [17] found that the Leap

Motion performed poorly against both a mouse and a touchpad; Jones et al. [88] found lower throughput, more fatigue and lower preference than a mouse, even with a standardised arm posture; and Fallah and MacKenzie [52] reported a throughput of 1.97 bps and an error rate of 12.8% for mid-air pointing with a tap gesture, against 2.26 bps and 3.4% for a touchpad. These studies consistently confirm a fundamental performance gap between surface and mid-air input, driven by inherent systemic challenges, such as tracking jitter [154] and instability arising from the lack of support for the hand [168]. With the emerging popularity of hand-tracking interactions for XR, the literature further unpacks this by exploring depth, posture, and multimodal gaze-hand alignment [38, 70, 111, 172] within virtual environments and with virtual planar displays.

For this thesis, these studies validate that mid-air cursor control remains constrained by the same underlying speed–accuracy trade-offs as surface interactions, with the added difficulties of cursor instability, arising from technical limitations (e.g., tracking jitter) and ergonomic concerns (e.g., lack of support and hand stability, fatigue). This thesis does not explicitly look at interaction efficiency through the lens of Fitts' Law, although it does use systematic empirical methods and similar task protocols to characterise input performance.

2.3.3 Continuous Control and The Steering Law

While Fitts' Law models discrete cursor-and-click target acquisition tasks (which generalise to UI operations like button activation), it does not account for interactions that require continuous control, i.e., interaction tasks like dragging a slider, moving a scrollbar, or navigating a cursor through hierarchical drop-down menus. Such continuous interaction tasks require the user to maintain control of the cursor within a strictly constrained path over a sustained duration, which is not compatible with the *rapid aimed movement* nature modelled by Fitts' Law.

To model such examples of continuous movement, Accot and Zhai [2] introduced the Steering Law, which takes its name from the continual precision steering of the cursor along a bounded trajectory. Derived mathematically as an infinite series of Fitts' Law tasks, the Steering Law treats continuous path navigation as a goal-crossing trajectory. It asserts that the movement time required to steer a cursor through a tunnel of length A and width W is directly proportional to the task's index of difficulty. The model has since been refined to accommodate paths whose properties vary along their length [104, 105], UI menus [183], and direct-touch dragging behaviours that conform to a steering rather than a Fitts profile [40]. Across these, the Steering Law remains the principal theoretical anchor for continuous widget control and draws attention to the more complex nature of continuous interaction tasks often found in WIMP interfaces.

Continuous widget control in mid-air environments requires users to do three things simultaneously: keep the cursor within the widget boundaries, maintain an active *held* state, and sustain both for the duration of the task. While a mouse or touchscreen provides physical mechanisms to support this (e.g., mechanical button as a clear mode-switch, or surface friction to support and guide movement), touchless systems must rely entirely on hand-tracking. This lack of phys-

ical grounding introduces significant vulnerabilities to jitter, proprioceptive drift, and postural changes [35, 175].

In summary, touchless displays often used cursor-and-click interactions where users control an on-screen cursor, which they can use to 'point at' and 'activate' UI elements. Theoretical foundations of cursor control suggest that input performance and efficiency will be impacted by similar characteristics as conventional input modalities and, to an extent, interaction can be supported through adhering to empirically-informed similar design principles (e.g., to increase target width where possible to reduce the index of difficulty). However, mid-air input faces additional usability challenges arising from lack of affordance and support, resulting in a performance gap between mid-air input and surface-based input. The following section explores these issues in more detail.

2.4 Human Factors of Touchless Mid-Air Interaction

The cursor-and-click interaction paradigm was designed around input devices whose physical and perceptual properties differ from those of an unsupported hand. This section explores the human-factors problems that distinguish mid-air cursor use from its surface-supported counterparts to highlight some of the usability issues that currently impact touchless input performance.

2.4.1 Absence of Physical Affordance and Support

The most fundamental difference between mid-air and surface-supported input is that mid-air interaction provides almost none of the physical structure that surface-supported devices contribute to the user's understanding of what is happening during an interaction, or what *should* happen in order to successfully achieve a desired interaction outcome. For example, the physical form of a mouse signifies [131, 132] how the user should touch it and its position on a desk supplies a stable surface that helps resolve the user's hand into a two-dimensional input. Whilst not spatially aligned with a vertical orientation screen, users can quickly establish a mental model of the relationship between mouse movements, constrained by the physical surface, and corresponding cursor behaviour on screen. Similarly, a touchscreen supplies a physical surface whose contact event is unambiguous and self-localising, and a stylus on a tablet supplies the haptic cue of the pen tip meeting glass: users know how to plan their actions because they can see directly where they need to tap on screen, and they get implicit feedback that they have succeeded because they perceive the collision between finger, or stylus, and screen.

In contrast, mid-air input lacks these haptic signifiers that support interaction. The hand operates in free space without a stabilising surface, with no tactile or physical feedback to show the boundaries of the interaction region, and with no haptic confirmation to inform the user about activation or registration of their actions [26, 35, 111]. Users must instead rely on visual

feedback to determine the current system state and the ways that this changes in accordance with their actions (if they are even recognised), supplemented by proprioception (awareness of the position and movement of the body) for indication of where their own hand is [81, 120].

Lack of physical support has been recognised since the earliest evaluations of free-hand pointing interaction. Vogel and Balakrishnan [168] reported that the absence of kinaesthetic feedback in freehand pointing was a primary contributor to the issues participants experienced with the modality, proposing subtle auditory and visual cues as partial compensations for loss of haptic information. Chattopadhyay and Bolchini [35] found that the absence of haptic feedback during large-display touchless interaction made visual feedback the only source of state information, and that the design of this visual feedback materially affected user performance in drag-and-drop tasks. This reliance of visual and lack of haptic feedback has been linked to increased cognitive load [26]

Further, this reliance on visual feedback highlights a second challenge concerning the sensing apparatus itself. Hand tracking operates within a finite, three-dimensional sensing volume and its boundaries often remain invisible to the user, while accuracy degrades when getting closer to it [9, 57, 73, 88]. While manufacturer guidance on device placement [108] and explicit signage [102] can help mitigate this issue, it does not resolve the core problem: the user cannot see or feel where the sensing volume is located, meaning the sensor's response degrades before failing outright (since a computing system cannot recognise gestures it cannot 'see'). Freeman et al. explored a range of feedback techniques [57] to help guide the hand towards the tracking "sweet spot", although the best-case provision of ultrasound haptic feedback is a costly solution. Ultimately, as the user is asked to operate within an invisible volume using an unsupported hand and a feedback channel that cannot signal the boundary, the resulting gestural input is predictably more variable than its surface-supported counterpart or one with a clearly visible interaction boundary (e.g., a touchscreen).

Beyond the spatial uncertainty of invisible boundaries, the absence of physical affordances gives rise to the Midas-touch problem [85], where false-positive inputs occur when users unwittingly engage the system and result in a registered input event. In surface-supported interaction with a physical input peripheral or device, a mechanical 'click' often provides an unambiguous mode-switch event that clearly separates incidental movement from intentional input activation. Mid-air input lacks a similar perceptible physical threshold as the hand must continuously occupy the active sensing space simply to be tracked, meaning any incidental motion made becomes a candidate for misinterpretation. Here, a system can interpret these movements as deliberate actions, possibly activating widgets without clear intention.

2.4.2 Fatigue and Consumed Endurance

The lack of physical support in mid-air interactions results in interactions that can be physically difficult to sustain over longer interaction sessions. The common term for the accumulated

shoulder fatigue that results from prolonged mid-air interaction is the *gorilla-arm* effect, and is a consistently reported usability complaint in the touchless and mid-air literature [88, 134].

Fatigue and discomfort in mid-air interaction typically occur because this interaction style requires prolonged arm raising (e.g., to maintain the continuous control of a cursor), which increases the amount of time muscles spend under tension (e.g., shoulder articulation to keep the arm raised in front of the body). Hincapié-Ramos et al. [80] modelled this time-under-tension using a metric called Consumed Endurance (CE), a biomechanically derived metric that estimates the cumulative shoulder torque produced by an unsupported arm raised against gravity. Their analysis shows both that fatigue accumulates as a function of the duration and amplitude of arm motion, and that fatigue-efficient interface design is a quantifiable goal rather than a stylistic one. Additionally, fatigue is not just an issue that promotes discomfort, but is also something that actively degrades motor performance during the interaction itself: Jang et al. [87] report a feedback loop in which accumulated arm fatigue progressively degrades fine motor control, which in turn raises error rates, requiring prolonged interaction time to compensate and correct errors, which in turn accumulates further fatigue, and so on.

The biomechanical origin of this fatigue is well understood. Hincapié-Ramos et al. [80] derive CE from the static torque that the deltoid (shoulder muscle) and supporting musculature must produce to hold the upper limb against gravity, and Jang et al. [87] extend the analysis to include the kinetic component of arm motion, showing that perceived exertion is a joint function of the load and the work done in moving the limb between positions. From this, holding an unsupported hand in an elevated position is ergonomically taxing, as the gravitational pull on the shoulder must be resisted continuously regardless of whether any motion is happening. Importantly, the extent of the fatigue is dependant on the joint doing the work. Lin et al. [103] compared three joint-based pointing methods (whole-arm, elbow-based, and wrist-based) on a Fitts' Law style pointing task, finding that wrist-based pointing produced substantially less perceived fatigue. This highlights an important consideration for mid-air interaction: gestures which can be performed with the wrist or fingers, rather than gestures which require shoulder-driven motion, carry a measurable ergonomic advantage, since these more subtle movements originate from smaller muscle groups.

These factors combine to make some mid-air manipulations more fatiguing than others. Continuous interactions (Section 2.3.3) could be particularly costly from a fatigue perspective due to their requirement of a sustained activation state during a precise ongoing motion, combining the static and kinetic contributions to fatigue across the duration of the interaction [77, 88]. Selections requiring high precision (i.e. high Index of Difficulty in Fitts' Law terms: Section 2.3.2) will lead to slower and more precisely controlled motions that increase the load on the shoulder and accelerate the accumulation of fatigue.

Given the 3D nature of mid-air input and the lack of a physical surface to constrain movement against, there is also an ergonomic cost of trying to constrain motion to certain planes of the input

region – most notably with regards to the depth of the hand to/from a vertical display. Extending the unsupported arm forward (e.g., along the depth axis) lengthens the arm at the shoulder, raising the gravitational torque the shoulder must resist, so postures that require sustained or repeated forward reach accumulate load faster than postures performed closer to the body [80]. Chen and Wang [36] report a behavioural correlate for distal mid-air pointing on large displays, observing that users compensate for the cost of reaching forward by adopting hand-stretching, shortening the effective depth distance.

2.4.3 Precision, Instability and Target Slips

The unsupported hand is also a less precise input device than a surface-supported counterpart. Even when well-positioned (with respect to tracking), the hand is a noisier input mechanism because even a steadied hand experiences inherent instability, especially as fatigue grows. Naturally occurring tremor in the hand and fingers has been shown to be large enough to be detected during interaction [178]. This can degrade Fitts' Law throughput [13], with further issues occurring when tracking near the edge of the sensor's effective range [9, 73].

Beyond mechanical and physiological noise, there is also a feedback deficit: while proprioception is the principal source of spatial sensory information received by users during mid-air interaction, it remains necessary but insufficient on its own for precise cursor control [120]. Hinckley et al. [81] demonstrated this limitation, showing that passive haptics provide additional proprioceptive grounding that improves 3D interaction performance. Not having a physical ground or haptic replacement is consistently the largest single contributor to negative input performance, carrying a severe precision cost [186]. Ultimately, this lack of physical support creates a cycle: as arm fatigue accumulates, fine motor control worsens [87]. This makes tasks artificially harder and slower to complete, which in turn causes even more fatigue and precision loss. The lack of physical support during mid-air interaction is an inherent characteristic of this input modality and is difficult to remove entirely, although novel interaction solutions can be used to mitigate input task difficulty and reduce potential for fatigue – an approach taken here.

A specific instance of a precision problem in mid-air interaction is The *Heisenberg effect of spatial interaction* [22]. Formally characterised by Wolf et al. [181], where the physical act of a discrete activation (e.g., pressing a button) unintentionally displaces the input end effector, often causing target slips [25]. Distinct from sensor or hand jitter, the effect is solely user induced at the moment a physical activation ation is performed. In a Fitts' Law style experiment, Wolf et al. approximated that 30% of overall selection errors were caused by the effect and software compensation could only recover a quarter of those errors [181]. This phenomenon is a biomechanical consequence of the activation gesture itself [173]: any gesture that requires hand motion could move the position of the hand (which directly or indirectly determines the cursor position on-screen) at the moment of selection, causing the cursor to 'slip' off the target.

The Heisenberg effect highlights an underlying property of mid-air pointing that not all axes

within the 3D environment are equally precise. The depth axis (z) is intrinsically less stable than along the lateral plane (x, y). Lacking a physical surface to provide kinaesthetic feedback or stabilising friction, users must rely on visual and proprioceptive estimates of arm extension [120]. Chan et al. [31] found that targeting errors along the depth axis are an order of magnitude larger than in-plane errors, while Lubos et al. [111] similarly observed that selection errors increase along the depth axis. Furthermore, three-dimensional extensions of Fitts' law confirm that targets along the depth axis disproportionately degrade both movement time and accuracy [38, 70]. Crucially, this instability degrades not only target acquisition but also keeping depth. Fischer et al. [53] demonstrated that even when users intend to execute purely planar movements, the unsupported hand drifts, triggering frequent unintended state changes across continuous dragging and sketching tasks. Ultimately, as depth motion is biomechanically coupled to the shoulder joint and is reliant on poorer visual perception, the depth axis remains the least precise dimension of human motor control.

Often the practical manifestation of these issues is the *target slip* [25, 181]: a cursor that successfully arrives at the intended target but then drifts off it before the selection event completes, causing the system to register either the wrong selection or no selection at all.

2.4.4 Standardisation, Gesture Vocabularies, and Learnability

The human-factors problems considered so far in this section are inherent properties of the touchless modality itself, largely occurring due to the physically challenging nature of unsupported mid-air hand movement. There are additional usability problems arising from interface and interaction design conventions, which are not grounded in physiology or biomechanics but are the result of design inconsistency and a lack of user familiarity and intuition.

Even where the cursor-and-click model has been inherited, the gesture action vocabulary that controls the cursor is not standardised across touchless systems, and the absence of standard gestures imposes a usability cost on first-time and inexperienced users. Vogiatzidakis and Koutsabasis [169] review of 47 gesture-elicitation studies for mid-air interaction find that the same task is often associated with different gestures across studies, that the elicitation methods themselves are inconsistently applied, and that no convergent vocabulary has emerged. Groenewald et al. [69] reach similar conclusions across 65 papers where their systematic review finds that the gestures associated with selection, navigation, and manipulation are heterogeneous enough that no developer can rely on user familiarity with them. Similarly, Aigner et al. [6] find wide variability in the gesture types that participants prefer for any given task. Commercial systems have similarly used a variety of gestures for common user interface operations, e.g., distinct 'click' or activation gestures, which are discussed in more detail in Section 2.5.

The cost of this is placed on the user, especially during first time or infrequent use, which would reasonably be assumed to be common in touchless public display scenarios (though also affecting everyday use). Where a touchscreen affords a direct touch 'tap' and a mouse affords

a direct 'click' of a mouse button, a touchless display lacks signifiers [132] that would suggest how to interact with it. User are left to infer both the location of the sensing region and the gesture that will commit a selection. Limerick et al. [102] investigate this in the context of touchless public displays, reporting that users failed to recognise that gestural input is available even when explicit signage is provided, and would disengage attention from the display before discovering what the system can do. This discoverability failure is the consequence of designing gestural systems around vocabularies that assume user familiarity they cannot rely on [1, 133] and depends on whether the user can map onto a familiar action they would perform with a corresponding physical object [67]. This lack of a gestural map also increases the likelihood of performing an accidental gesture (i.e., the Midas Touch issue of false-positive recognition). The cursor-and-click metaphor, although apparent for *pointing*, does not on its own reveal *how to activate*: the click action that would denote a selection on a mouse or touchscreen has no clear gestural counterpart, and the user must therefore either guess at the activation method or fail to interact at all.

2.4.5 Summary

The human-factors problems set out in this section of the review describe a modality in which physical and perceptual characteristics differ substantively from those of the surface-supported pointing devices that cursor-and-click interfaces have long been designed for. These usability issues are also not independent of one another: fatigue degrades precision, depth-axis instability compounds the Heisenberg effect, the absence of haptic feedback raises the cost of an unstandardised vocabulary, and over-correction in response to ambiguity and the Midas Touch problem raises both the precision and the fatigue costs of mid-air interaction. This thesis will explore interaction solutions that could help address these concerns. The following section of the literature review introduces some foundational concepts on which these interaction solutions will be based.

There is also a lack of standardisation and discoverability which affects mid-air interaction, with users lacking clear indicators about how to interact successfully. Gesture vocabularies are often inconsistent across mid-air systems and devices, and elicitation studies find a range of gestures trying to achieve the same goal, suggesting that typical interaction operations do not necessarily have an obvious and 'intuitive' gesture. This thesis does not aim to investigate or recommend the 'best' gestures for touchless display input as this is beyond the scope of this work, and it would be naive to ignore the technical constraints that also influence gesture selection. However, the range of interaction techniques investigated here will take important steps towards identifying promising interaction solutions and approaches for designing and evaluating efficient touchless gesture interfaces.

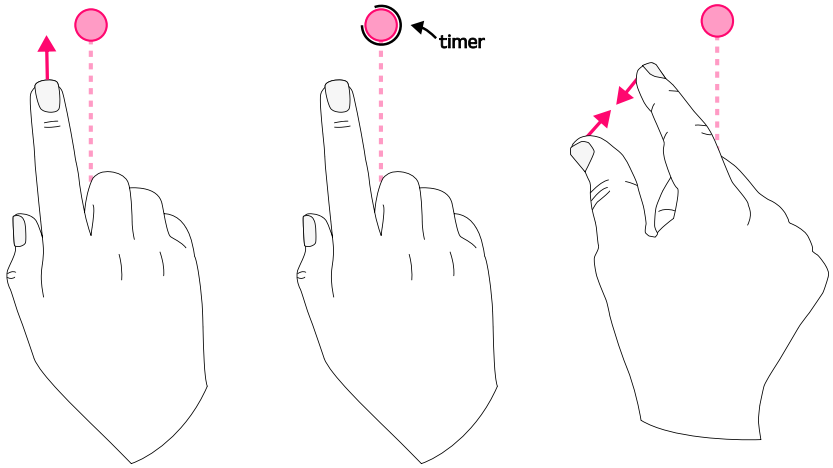


Figure 2.4: Illustration of the Push (left), Dwell (middle) and Pinch (right) activation gestures.

2.5 Mid-Air Activation Techniques

Cursor-and-click interaction requires two distinct operations: continuous *positioning*, which moves the cursor across the UI, and discrete *activation*, a mode-switch event which causes a change in selection state at the current cursor location. These are often enacted through distinct physical actions, e.g., moving a mouse to reposition the pointer *then* clicking a button to activate the UI element targeted by the cursor. With a touchscreen these actions jointly occur, with the touchscreen tap simultaneously providing positional data and the occurrence of the activation event. With these physical input devices, input is unambiguous in its timing, self-localising in its position, and accompanied by tactile confirmation that the user has engaged the device [27, 171].

In a mid-air cursor-and-click interface, positioning and activation are distinct events, analogous to mouse or trackpad use. However, the implicit tactile properties that support peripheral clicking do not hold in mid-air. The unsupported hand has no physical surface to strike, no perceptible tactile threshold to cross, and no haptic confirmation to register, and the sensing pipeline cannot, on its own, tell the user's incidental motion apart from their deliberate intent to select [85, 181]. Users reposition the cursor through hand movement (a mapping that will be explored in detail later in this thesis) and perform a distinct gesture action for activation.

This section reviews common activation techniques for mid-air interface hand-based activation in the literature, namely Push, Dwell, and Pinch (Figure 2.4), and considers the multimodal alternatives that have been proposed alongside them. This review introduces activation gestures that will be explored in the experimental chapters of this thesis.

2.5.1 Push: A Depth-Based Activation Gesture

The most direct skeuomorphic translation of mouse and touch activation into mid-air is the *push* gesture (Figure 2.4–Left): the user moves their hand forward, along the depth axis toward the display, and crosses a virtual plane that the system treats as a click event [64, 168, 184]. This

metaphor is appealing as it preserves the cause-and-effect semantics of pressing a button while requiring no additional gesture vocabulary, and it has been adopted as the default in widely deployed commercial touchless display platforms such as Ultraleap's TouchFree [110] and the Intel RealSense touchless module [84]. Vogel and Balakrishnan's foundational evaluation of freehand pointing on large displays included an *AirTap* variant in which selection was triggered by a forward finger motion [168], and Yoo et al.'s *To Dwell or Not to Dwell* study found push to be the most preferred selection technique on public information displays even when dwell-based alternatives were more accurate [184]. Gentile et al. [64] explicitly frame push as the "*artificial push-button gesture*" that designers introduce into touchless interfaces precisely to preserve the familiar cursor-and-click model.

Push, however, is prone to many of the issues described in Section 2.4. The forward push into the z-axis plane is invisible, so the user has no real-world visual or haptic signal of its location and must rely entirely on the visual feedback of cursor size or shape change to know when the gesture has registered [35, 55]. The forward motion itself displaces the hand and causes the Heisenberg effect of spatial interaction [22, 181]. Additionally, the repeated forward extension to enter activation also lengthens the shoulder muscle, accelerating the accumulation of consumed endurance [80, 87], and can push the hand further towards the limit of a sensor's reliable tracking area, where accuracy is known to degrade [9, 57, 73]. These issues compound further for continuous control widgets (like sliders or scrollbars). To use them, a user must push forward to keep the widget activated while simultaneously steering sideways along a path [2,93]. Even though pushing forward and moving sideways are separate directions in space, the user cannot easily separate them in practice [174].

Experiments in this thesis will consider Push as a baseline activation gesture, due to the prevalence of this interaction metaphor in both the literature and in commercial touchless tracking solutions, especially at the point in time when these experiments were conducted.

2.5.2 Dwell: Temporal Activation via Steadied Cursor

Dwell (Figure 2.4–Middle) bypasses the depth-axis problem by decoupling activation from space and physical actions, using temporal information to infer intent instead. The user is required to hold the cursor steady over a target for a fixed period, after which the system treats the elapsed time as evidence of intent and enters an activated state, removing the third dimension entirely [127, 184]. Müller-Tomfelde's exploratory studies of natural pointing identify a short, almost involuntary dwell period in human pointing behaviour and propose using it as a basis for selection feedback [127], and Yoo et al.'s public-display study reports that point-and-dwell was perceived as more accurate than push despite being less preferred [184]. Commercially, the original Microsoft Kinect interfaces used dwell for button activation, and gaze-based interfaces routinely apply dwell thresholds [78]. Dwell has the inherent advantage of reducing the likelihood of Heisenberg-style target slips that Push is prone to, since the gesture that commits the

selection is the *absence* of motion rather than the addition of it. However, the demands for stability over time, and the propensity of jitter resulting from human and sensor, mean this cannot be automatically assumed.

Dwell imposes a hard floor on selection throughput equal to the chosen timing threshold, so the time for each selection costs the dwell period in addition to the ballistic and corrective movement time predicted by Fitts' Law (Section 2.3.2) [127, 184]. The threshold must also be tuned to mitigate the risk of Midas Touch selections [85]: too short and incidental hand pauses commit unintended selections, too long and users perceive the interface as slow and disengage [95, 184]. Dwell additionally requires the user to hold a sustained, unsupported posture, which compounds rather than relieves the consumed-endurance cost of mid-air input [80, 87]. Finally, unlike repeated mouse clicks, successive dwell periods do not accumulate. If a user loses track of a target during a continuous-control task, they must re-acquire it and wait another dwell period each time. This repetitive resetting increases user frustration in slider-style interactions [95] and the need to correct and wait compounds fatigue costs [80]). Ultimately, while using a dwell time might reduce the physical burden of maintaining depth-axis activation, it simply trades that physical effort for increased time, cognitive load, and accumulated fatigue.

2.5.3 Pinch: A Finger-to-Thumb Mode Switch Gesture

Another type of activation gesture moves the burden onto the pose of the hand itself. The typical example of this is *pinch* (Figure 2.4–Right), in which the user brings the index finger and thumb together to enact a mode switch, then separates them to release it [12, 90, 153, 179]. Two properties of finger-to-thumb pinch make it well suited for mid-air input. First, the gesture creates a clear binary state, the fingers are either in contact or they are not, producing a tactile and proprioceptive threshold the system can also use to move into the activation state [153]. Second, as finger articulation is decoupled from hand translation, the cursor, when mapped to the palm or wrist, will not move during the activation gesture itself; this in principle removes the Heisenberg-style slip that the push gesture introduces [12, 181]. Additionally, biomechanical evidence further suggests that thumb-finger movements create a smoother and more controlled precise interaction compared to extended finger movements [143, 149].

Pinch has become the dominant activation gesture in XR interactions. Surale et al.'s analysis of eleven bare-hand mid-air mode-switching techniques in virtual reality identifies pinch as a baseline that consistently performed well across all tasks [153]. Similarly, Jude et al. reports a strong preference for pinch over open-hand alternatives for in-air gestural manipulation tasks [90] and when comparing pinch-based selections (in XR) they are competitive with controller-button baselines [123]. At the commercial level, pinch is the default activation gesture of recent XR headsets such as the Meta Quest 3 and Apple Vision Pro [5], often combined with gaze to support distal selection [140, 147]. Pinch is not without cost: it requires precise finger tracking, which is more sensor-demanding than skeletal pose alone and therefore more sensi-

tive to occlusion. Further, due to the increased demand in tracking, false negatives are more common, increasing fatigue [99, 170].

2.5.4 Alternative and Multimodal Activation Approaches

Beyond these activation methods, the wider design space contains several alternative activation techniques that are highlighted here for completeness. Instead of manoeuvring the cursor directly over a target before performing an action to activate it, *crossing* adapts the fine-point targeting model into a goal-crossing model, such that activation is triggered by moving the cursor *across* a widget [2, 95, 112] reducing the need to decelerate and make corrective adjustments to land the cursor over the target. *Semaphoric* gestures such as wave, swipe, or thumbs-up have been proposed as discrete activators for system-control tasks [92, 169] decoupling activation gestures from movement in a similar manner to finger-to-thumb pinch. However, these still inherit the same discoverability problem discussed in Section 2.4.4.

Multimodal activation explicitly separates the pointing and selection across distinct input modalities. An early example of this is the Put-That-There system [20] , which combines mid-air pointing gestures with spoken commands, as described in Section 2.2.1. More recently, targeting via eye gaze or head-pointing and then using the hand for selection, either via finger-to-thumb pinch or other discrete gestures, removes both the precision cost of mid-air pointing and the depth ambiguity of push [33, 140, 147]; such *gaze-ad-pinch* interactions have become popular with XR devices.

Another popular multimodal method is to integrate haptic feedback with gesture input, in an attempt to "add touch to touchless interfaces" [55]. As the absence of a physical click is implicated in both selection error and reduced sense of agency [41], focused ultrasound haptics has been shown to create a perceptible cue at the moment of activation without the user contacting any surface [41, 55, 167].

While these alternatives can work well under the right circumstances, they come with the cost and complexity of additional hardware integration and an increased difficulty in touchless display deployment. This thesis therefore concentrates on mid-air gesture-only techniques, while noting that multimodal augmentation remains a viable option for systems where the additional channels are available and practical.

2.5.5 Summary

The three primary activation techniques reviewed in this section take distinct approaches to mitigate the human factors concerns discussed earlier. *Push* adopts the familiar metaphor of a physical button press, though requires problematic forwards hand motion [31, 111] and has increased likelihood of target slips [181]. *Dwell* attempts to mitigate target slips through steady cursor position, making a trade-off for lower throughput and potentially increased fatigue [80,

184]. *Pinch* decouples activation from cursor position by using finger-to-thumb pinch gesturing as a mode switch, and removes the need for forward motion. While *Pinch* has emerged as the default activation gesture in many contemporary XR environments, it is less commonly used than *Push* and *Dwell* on planar touchless displays. Moreover, mid-air cursor interaction research predominantly focuses on discrete target selection, with limited understanding of how facets of cursor interaction design affect more complex continuous operations.

The first research question of this thesis (**RQ1**, stated below) considers the effect of activation method on usability across discrete selection and continuous manipulation tasks in a mid-air cursor interface. This thesis believes *Pinch* to be a promising alternative to more established activation approaches (*Push* and *Dwell*), with potential to mitigate the Heisenberg effect [181] by permitting cursor stability at the point of activation, and its distinct states could be especially beneficial for continuous interactions (e.g., sliding), with users aided by the touch cues experienced during the finger-to-thumb pinch posture.

> **RQ1** - How does the choice of target activation technique affect the usability of a mid-air user interface?

2.6 Cursor Movement Mechanics

Mid-air cursor interfaces generally adopt a mapping between hand position and cursor position, such that moving the hand in mid-air results in a corresponding on-screen cursor movement: e.g., moving the hand to the left results in the cursor moving some corresponding amount to the left. The mapping between physical and virtual space is nuanced and will be discussed in more detail later, but fundamentally, hand movements will directly produce a cursor response so long as the hand is within the sensor space and the system is actively responding to the user.

As discussed earlier in this chapter (Section 2.4), the usability of this kind of cursor movement is affected by a variety of factors which, ultimately, increase the difficulty of precisely moving the cursor towards UI targets (e.g., buttons, list items) and keeping it there whilst performing the activation method (as described in Section 2.5). Targeting movements are made more challenging via lack of stability, poor precision, lack of control across multiple axes, and accumulated fatigue from sustained interaction movements.

From a user interface design perspective, interaction models like Fitts' Law hint towards potential layout solutions, like increasing the *target width*, though adapting designs to compensate for touchless input is less ideal when trying to achieve parity with alternative input modalities or when retrofitting touchless to existing systems that were designed for other input devices. Alternatively, finding novel ways to reduce the effective *distance to target* could offer efficiency improvements without altering UI designs. In conventional pointing modalities (i.e., non-touchless), the literature has highlighted potential mechanisms for manipulating cursor behaviour or movement mechanics, with a view to ultimately reducing distance to target. Two of

which are investigated in this thesis are discussed below.

The first approach modifies the behaviour (and occasionally the appearance) of the cursor such that the cursor sizes increases. Such *area cursors* expand the effective selection region beyond a single point (or pixel) such that targets (or parts of targets) may be enclosed within the cursor area. The increased target size reduces the effective distance to the target (which Fitts' Law suggests could improve efficiency) and reduces the necessary precision required to manoeuvre the cursor onto a target, especially with relatively smaller target sizes. Section 2.6.1 discusses the development of area cursor techniques and their application in pointing interfaces, across conventional and touchless interaction contexts.

The second approach modifies the mapping between physical movement (i.e., movement of a handheld peripheral, or the hand itself in a touchless system) and the corresponding cursor movement that occurs on-screen. This physical to virtual movement ratio is the *control-display gain*, which can be used to amplify or attenuate the effect that input movement has on cursor displacement. An increased control-display gain, whereby the same physical movement results in *more* cursor movement, could potentially reduce the distance to target and require less physical movement (with potential reduction in fatigue), although this could also come with a cost of increased difficulty stabilising the cursor. Section 2.6.2 discusses prior work on control-display gain manipulation across a range of input devices.

Both approaches will be investigated in later thesis experiments, as potential interaction solutions that ultimately aim to improve usability and efficiency when interacting with a mid-air gesture controlled cursor.

2.6.1 Area Cursors & Dynamic Area Cursors

Pointing interfaces conventionally use 'point cursors', where selection occurs at the smallest possible point beneath the cursor, in terms of display or sensor resolution. This allows precise per-pixel selection, although increases the difficulty of target selection when targets are small or when the input device experiences a lack of stability and guidance. Area cursors, which allow activation or selection within a targeting region, instead of at a single point, can potentially improve selection performance by reducing the need to move the cursor directly over the target.

Area cursor and target-expansion techniques have a long history in pointing research. Their key principle is that the effective selectable region of a target can be made larger than its visual appearance, either by enlarging the target near the cursor or by enlarging the cursor itself so that targets may fall within its wider selection area, even if the centre of the cursor (or the tip of the pointer) is not directly over the target. This increases the effective target width (in Fitts' Law terms, Section 2.3.2), reducing the index of difficulty of the selection and so reducing the necessary movement time to select that target.

Early expansion techniques operated statically by enlarging the target rendered beneath the cursor (e.g, the MacOS dock). For instance, Gutwin's speed-coupled flattening [75] reduced

fisheye distortion during focus targeting, while subsequent research explicitly expanded targets near the cursor's predicted destination [15,117]. While this approach is effective in linear sparse layouts, its performance degrades in dense environments. When multiple targets compete for the same enlarged region, the system is forced to pre-emptively commit to a single expansion before the user's explicit intent is clear.

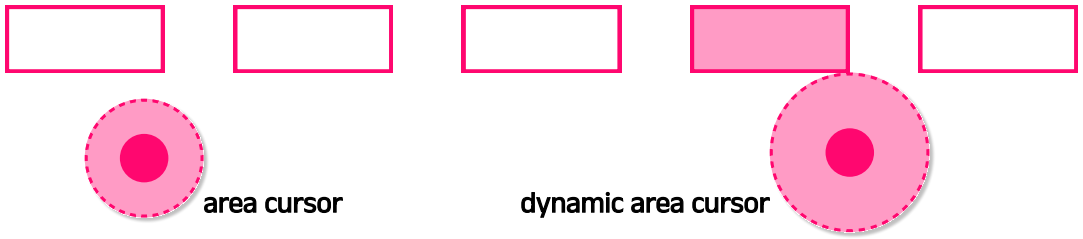


Figure 2.5: Illustration of an area cursor (left), which targets items within its region, and a dynamic area cursor (right), which dynamically scales to reach the nearest target.

Perhaps the best known instance of an area cursor is the *Bubble Cursor* [71], which dynamically expands the cursor size, rather than the target size: the cursor's activation radius is dynamically resized so that it always encloses exactly the nearest target, guaranteeing that one and only one target is selectable at any moment. This thesis makes the distinction between an *area cursor* (of static size) and *dynamic area cursor* (which scales to implicitly select the nearest target), as shown in Figure 2.5. The *Bubble Cursor* is a dynamic area cursor, which – like a regular area cursor – has the benefit of enlarged target width; however, its dynamic scaling removes the ambiguity of a fixed size area cursor (which might enclose multiple targets). *Bubble Cursor* consistently outperforms a standard cursor across a range of target sizes and densities [79] and has inspired further research into dynamic 'bubble-like' targeting. For instance, the *Bubble Lens* [122] later addressed an issue with the *Bubble Cursor*: when targets are small and densely packed, the cursor radius shrinks to the point where it behaves like a standard point cursor, and the *Lens* responds by automatically magnifying nearby targets. A closely related alternative makes the activation region dynamic with respect to a different variable. Rather than scaling the cursor to the layout of surrounding targets, Chapuis et al.'s *DynaSpot* [32] couples the width of the cursor's activation area to the cursor's movement speed: the activation area grows as the cursor accelerates and collapses back to a point cursor at low speed or when motionless, with a short lag before the area is withdrawn so that the enlarged region remains available at the end of a rapid ballistic movement. When several targets fall within the active region, *DynaSpot* resolves the ambiguity in the same manner as the *Bubble Cursor*, by designating the target nearest the cursor centre; when none do, the cursor addresses empty space, so region selection and other point-cursor interactions remain available without an explicit mode switch. Critically, *DynaSpot* deliberately minimises its visual feedback. It does not render a bubble that inflates and contracts to enclose whichever target is currently nearest, and its designers argue that the large, abrupt visual changes required by the *Bubble Cursor* are themselves a source of visual distraction; *Dy-*

naSpot instead renders only a small translucent spot that changes gradually and smoothly, and highlights the designated target.

Target density imposes the primary limitation on the theoretical advantages of a dynamic area cursor. As Hertzum and Hornbæk [79] report, the performance advantage of the *Bubble Cursor* over a standard point cursor diminishes as target density increases. When targets are tightly packed, there is insufficient empty space for the cursor's activation area to expand, causing the performance of the two techniques to ultimately converge. Mott and Wobbrock [122] observed a similar limitation for the *Bubble Lens*. Density constrains speed-coupled activation in a different way: because *DynaSpot*'s spot width is capped independently of the surrounding layout, Chapuis et al. [32] report that its movement time rises with target density in a way that the *Bubble Cursor*'s did not, since the latter is free to expand into whatever empty space the layout affords. Either way, the usable benefit of an enlarged activation region is bounded by how crowded the interface is. Blanch and Ortega [19] later formalised this dynamic by incorporating a distractor-density factor into the standard Fitts' paradigm. By proposing an extension to Fitts' Law that explicitly accommodates sensitivity to surrounding distractor widgets, they demonstrated that target density must be treated as a factor in the evaluation of pointing techniques. Consequently, the efficacy of an area cursor cannot be reliably established through a single layout configuration. Any interaction technique that relies on leveraging surrounding empty space must be systematically characterised across a range of target densities to determine the precise thresholds where it remains effective and where its advantage diminishes. This thesis thus considers target density (and uses controlled experiment protocols by Blanch and Ortega [19]) in its experimental evaluation of touchless area cursor interaction.

When adapted to mid-air and ray-cast pointing, area-based targeting techniques generally inherit both the Fitts Law benefits and the underlying issues with target density. For instance, Dover et al. [48] applied the *Bubble Cursor* and *Bubble Lens* to mid-air gestural pointing, confirming that both techniques significantly improve small-target acquisition over a standard cursor, however, their effectiveness remains highly dependent on UI layout. Similar adaptations have been explored in XR, where Baloup et al.'s *RayCursor* [11] translated area cursor targeting to ray-cast selection, utilising a snapping mechanism to mitigate issues of precision. To address more complex 3D layouts, Vanacken, Grossman, and Coninx [163] introduced a *3D Bubble Cursor* and a *Depth-Ray* technique designed for dense, occluded environments, later expanded [46] with the *MultiFinger Bubble*, created specifically for highly dense UI where the bubble could not reliably isolate a target. These works suggest that while the theoretical benefits of area cursors and target expansion could transfer to mid-air interaction paradigms, their critical sensitivity to target density must also be taken into consideration.

2.6.2 Control Display Gain

Another method that can potentially be used to reduce the interaction cost of mid-air pointing is to adjust the mapping between input and cursor displacement. This mapping is best known as the *control-display gain* (CDG): the ratio between the extent of input displacement (e.g., mid-air hand movement or on-desk mouse movement), and the corresponding extent of cursor movement on-screen.

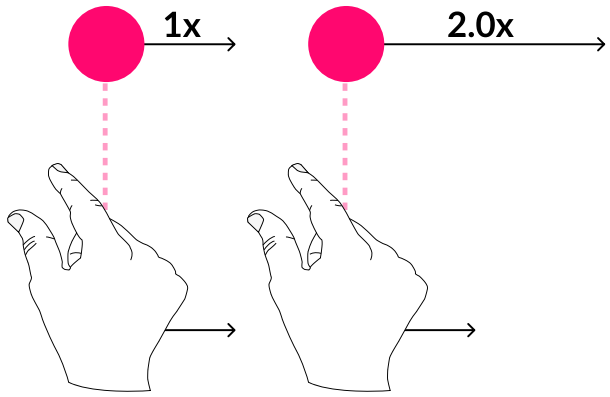


Figure 2.6: Illustration of how control-display gain affects cursor movement.

Figure 2.6 demonstrates the effect of CDG on cursor movement. A CDG of 1.0 means the cursor moves the same distance as the hand, e.g., a 10 cm hand movement produces a corresponding 10 cm cursor movement. A CDG of 2.0 doubles this, so a 10 cm hand movement produces a corresponding 20 cm cursor movement. Increasing the CDG amplifies movement, but reducing the ratio can attenuate movement as well: e.g., a CDG of 0.5 halves the amount of hand displacement, so 10 cm of hand motion produces only 5 cm cursor movement on screen. The higher the gain multiplier, the less the user has to move their hand to reach a target; the lower the gain, the user must increase hand motion for the same on-screen distance.

Control-display gain can have a notable impact on usability. Increased gain ratios can reduce movement time towards targets because the cursor moves at a higher relative speed, and could potentially reduce fatigue by enabling less physical movement to control the cursor; however, the increased gain also has potential to amplify instability and lack of control, which could have a detrimental effect on movement time and fatigue by necessitating more corrective and compensatory adjustments. Physical reach is especially a concern for mid-air interaction. When interacting in mid-air where the hand is unsupported, large screens might demand cursor traversals approaching the limit of comfortable arm reach, and so this ratio becomes a significant design decision for how physically demanding the interaction is.

Gibbs [66], in their early investigation of CDG, demonstrated in a series of joystick experiments that the ‘optimal’ gain for a cursor controlled system depends jointly on the controlling limb, the system’s time lag, and the displacement to be produced. This relationship manifests

as the speed-accuracy trade-off [30]. Applying a higher gain reduces the physical distance required to traverse an on-screen distance, but it also amplifies any inherent noise within the input signal. Conversely, while a lower gain effectively reduces this noise, it incurs a biomechanical cost by requiring larger arm movements. On sufficiently long traversals, this increased physical requirement results in repeated clutching to reposition the input device (or hand).

On conventional desktop input, the speed-accuracy trade-off has been well documented. Casiez et al. [30] provide an empirical account investigating CDG on a mouse input, they find that low gain impose a performance penalty due to increased clutching and hand velocity, and that high gain increases election time with small targets. Their subsequent work across operating systems [29] finds that an increased CDG over baseline improves pointing performance by up to 24%. Blanch et al. introduced *semantic pointing* [18] which varies gain according to the cursor's proximity to known targets. Similarly, *PRISM* [60] varies gain according to the user's velocity, supplying fine control at low speeds and increased speed at high ones. Furthermore, Thompson et al. [156] demonstrate that the effect of gain on movement time depends heavily on the target's size, its distance, and the angle of the user's movement. Across these, however, the input device sits on a desk: tremor is small, sensing is largely noise-free, and clutching is the primary cost of underestimating gain all of which are more challenging in mid-air.

For mid-air input, the same speed-accuracy trade-off exists with the benefit of increasing gain being more attractive than on the desktop. Reducing the physical amplitude of hand motion can be seen as a direct response to the fatigue issues characterised in Section 2.4.2: less arm motion per selection means less gravitational torque accumulated at the shoulder and a slower rise in cumulative fatigue [80,87]. Reducing the lateral extent of hand motion also reduces the risk of pushing the hand into the unreliable tracking area, reducing accuracy (Section 2.4.1) [9, 57, 73]. The mid-air CDG literature has accordingly explored how gain might be manipulated. Nancel et al. analysed mid-air pointing on large wall displays [129], demonstrating that standard pointing in mid-air is inadequate for the high index of difficulty that wall display targetting imposes, and that increased CDG can recover substantial precision loss. Kim et al.'s *Motion-Display Gain* [94] extends the adaptive-gain idea to distal mid-air pointing by tuning the mapping to the users' natural hand motion. Kota and Young [97] report that vertical control-display mapping produces throughput approximately 0.5 bits/s greater than other mappings on average in free-hand pointing, with lower gain producing a further 0.14 bits/s advantage when paired with vertical mapping. Lin et al. [103] confirm that the joint doing the work still matters as much as the CDG applied to it, finding that wrist-based pointing accumulates substantially less fatigue than elbow- or shoulder-based pointing under the same conditions.

However, the drawbacks of increased CDG in mid-air follow directly from the human-factors problems characterised in Section 2.4. Any noise in the input signal is amplified linearly by the gain multiplier: tremor in the hand/fingers [178], jitter introduced by sensor [13], and the natural drift of the unsupported arm [175] all become more visible on screen as gain increases.

The Heisenberg effect [181] amplifies in the same way: the cursor displacement produced by the act of selection is multiplied by the same factor as the user's intended motion, so a level of disturbance that would have been absorbed under a 1:1 mapping can be enough to cause a target slip under a higher gain. Riyal et al. [145] evaluate dynamic-CDG transfer functions for hand and finger based touchless control on a 2D Fitts task and report that high gain significantly raises error with touchless performance remaining below mouse performance. Ram et al. [144] report that increased CDG, horizontal mapping, and the use of armrests each significantly reduced steering success. Chen et al. [37] examine three CDG levels paired with wrist, elbow, and shoulder based selections using distal pointing tasks, finding that increased gain is tolerable only for the wrist-based method and is works poorly with the more proximal joints.

The 'ideal' CDG may need to scale with display size, as the larger travel area required by larger screens necessitate high multipliers that would be unusable on smaller displays [128,129]. Furthermore, as displays can be larger than the tracking area (especially when considering public display applications), CDG must be calibrated to ensure the user's hand remains within the sensor's reliable 'sweet spot' [57], a threshold dictated entirely by the display's size [9, 73]. However, compounding these limitations is target density, which constrains usable gain. As more dense layouts reduce the margin for unintentional cursor movement [19], a gain level that feels responsive in a sparse interface may become unusable in a dense one. Ultimately, because display size, tracking area, and target density are linked, accurately evaluating mid-air CDGshould be done together.

Experiments in this thesis will investigate the effect of different CDG ratios on touchless displays, paired with a dynamic area cursor (as described in Section 2.6.1). This novel combination makes a contribution to research on control-display gain through the use of area cursor targeting as a mechanism for attenuating the drawbacks of increased gain instability.

2.6.3 Summary

The two cursor-movement mechanics reviewed in this section take complementary approaches to reducing the underlying interaction cost of mid-air pointing. *Dynamic area cursors* can reduce the precision required for target acquisition by enlarging the cursor's effective selection region beyond a single point, so that the user does not need to place the cursor precisely on top of a target [71, 79, 122]. This reduces the effective distance to target and reduces the need for corrective adjustments to cursor position after targeting movements, since the user only needs to align the target with the cursor area. *Control-display gain* can be used to amplify the cursor's response to hand motion, so that the user does not need to traverse the full physical extent of a display to reach distant targets [30, 66, 129]. This could also be used to reduce the effective distance to target and could mitigate fatigue via smaller range of motion.

Each approach has been characterised in considerable detail in isolation, predominantly with a mouse on discrete-button selection tasks. Mid-air adaptations of area cursors have largely

retained these discrete selection tasks, and have not considered how increased CDG interacts with a cursor whose effective target width has already been enlarged by area-based behaviour. Further, there is a lack of understanding about how crucial factors like display size and target density affects these cursor targeting manipulations. This thesis addresses this knowledge gap.

The second and third research questions of this thesis (**RQ2** and **RQ3**, stated below) will consider the effects of cursor manipulation and control-display gain on usability in mid-air cursor interfaces, both individually and in combination, and across the display-size and target-density factors that are anticipated to impact the usability of touchless cursor input.

> **RQ2** - How can we reduce the need for traversal and selection accuracy in a mid-air user interface across different display sizes and widget densities?
>
> **RQ3** - How can we design mid-air interactions such that the required effort is reduced while interaction efficiency is increased?

2.7 Overall Summary and Research Gap

This chapter reviewed research on five topics: (1) a brief history of mid-air gesture interaction in HCI, from early foundational systems to its current adoption in touchless public displays and XR headsets (Section 2.2); (2) the theoretical foundations of cursor-and-click pointing and how the underlying models of target acquisition and continuous steering translate into mid-air (Section 2.3); (3) the human-factors problems that distinguish mid-air cursor input from its surface-supported counterparts, including the absence of physical affordance and support, accumulated fatigue, depth ambiguity, the Heisenberg Effect, and the lack of a standardised gesture vocabulary (Section 2.4); (4) the activation techniques most commonly used to within mid-air pointing for discrete selection, namely *Push*, *Dwell*, and *Pinch*, together with multimodal alternatives that have been proposed alongside them (Section 2.5); and (5) two cursor movement mechanics that have been proposed to reduce the cost of mid-air pointing: i.e., dynamic area cursors and control-display gain (Section 2.6).

Three research gaps emerge from this review. First, the dominant activation techniques on planar touchless displays (Push and Dwell) have known precision and fatigue costs, and the most promising alternative (Pinch) has been studied almost exclusively in XR environments, where target selection occurs via ray-cast pointing in a 3D environment, rather than cursor translation mechanics used with 2D touchless displays. Additionally, comparisons across them have been confined to discrete button selection tasks and so there is a limited exploration of their impact on continuous manipulation tasks. Second, dynamic area cursors can evidently help target acquisition, but their sensitivity to display size and widget density has not been investigated on touchless displays. Third, control-display gain has been studied on the desktop and on large

wall displays, but not in combination with area-based targeting, and not across the range of display sizes and widget densities that touchless displays span.

These gaps motivate the research questions of this thesis:

RQ1 - How does the choice of target activation technique affect the usability of a mid-air user interface?

RQ2 - How can we reduce the need for traversal and selection accuracy in a mid-air user interface across different display sizes and widget densities?

RQ3 - How can we design mid-air interactions such that the required effort is reduced while interaction efficiency is increased?

This thesis will now addresses these research questions in turn. Chapter 3 presents **Experiment 1**, which primarily addresses **RQ1** by comparing Push, Dwell, and Pinch as activation techniques across both discrete selection and continuous-control tasks in a mid-air cursor interface. Chapter 4 presents **Experiment 2** and **Experiment 3**, which primarily address **RQ2** by characterising dynamic area cursor use in mid-air across discrete and continuous input tasks. Chapter 5 presents **Experiment 4** and **Experiment 5**, which primarily address **RQ3** by evaluating the effects of control-display gain alongside area based targeting, with the aim of identifying mid-air cursor designs that reduce the required effort while increasing interaction efficiency.

Chapter 3

Experiment 1: Activation Gestures

3.1 Introduction

Touchless displays often use fine-grained cursor-pointing interactions used with conventional displays and input devices, where users 'point at' user interface elements (e.g., buttons, list items) using an on-screen cursor controlled by their hand in mid-air, and perform some gesture to 'select' or 'activate' the targeted item. As discussed earlier in this thesis (Section 2.4), the unsupported nature of mid-air interaction, along with additional degrees of freedom for sensing, can make precise point-and-select operations challenging [174].

Many touchless systems use a 'push' gesture to enter an activation state, akin to reaching forward to touch a physical touchscreen [64]. However, unlike a physical screen which signifies how and where to touch, the threshold for push activation in mid-air is invisible. Commercial touchless systems [84, 110] typically use a depth threshold crossed by a forward 'push' of the hand [64], though this threshold is invisible, and users have a limited and inconsistent understanding of where their hand sits in relation to this activation threshold, diminishing their ability to purposefully activate the push gesture when they intend to, and to avoid activation when they do not. The forward push motion is also prone to lateral cursor movement (the *Heisenberg effect* introduced in Section 2.4.3), which may cause the cursor to *slip* [25] off the target before activation occurs. Cursor instability compounds during continuous interactions such as slider manipulation, where the unsupported hand remains in an activated state over time while simultaneously controls position along a constrained trajectory [2, 77, 93].

Whilst the push gesture is widely used, these usability issues likely stem from a lack of suitability for activating targets on flat touchless displays (where virtual content and user interface elements are not positioned at different depths). The push gesture introduces unnecessary movement in the forward plane and is prone to incidental movement in the lateral plane, causing the cursor to fall off the target during activation. Alternative activation gestures could mitigate these issues by minimising cursor movement during the activation gesture, whilst also having a design that makes it more clear to the user what they must to do transition in/out of the activation state.

The first research question in this thesis therefore considers how this fundamental design choice affects usability:

> **RQ1** - How does the choice of target activation technique affect the usability of a mid-air user interface?

This chapter presents Experiment 1, which takes first steps towards answering this research question with a controlled comparison of four mid-air activation gestures (push, cursor dwell, and two finger-to-thumb pinch variations), for a mid-air slider control task that involves acquiring a target (i.e., slider handle) and then performing a controlled sliding manipulation along a track whilst maintaining activation of the handle. This experiment assesses usability and task performance to identify interaction techniques that mitigate the stability and depth-control issues associated the status quo push interaction metaphor.

3.1.1 Push Gestures

A common target-activation method in cursor-based touchless interfaces is the 'push' gesture, whereby users control the position of a virtual cursor with side-to-side and up-and-down hand movement (parallel with the screen), then confirm selection by moving their whole hand forward towards the screen, as if pushing a physical button. This interaction takes place in two movement planes: (i) the *xy* plane where the user directly controls the cursor position, and (ii) the *z* plane where the user moves their hand forwards to perform the push gesture.

Users receive direct visual feedback about their movement in the *xy* plane via the cursor position and movement on screen, allowing them to determine the effects of their hand movement on the cursor. Users are also able to establish how their hand position relates to the boundaries of the interaction sensing area, since this is explicitly revealed through cursor position. However, users typically receive limited or no feedback about their movements in the *z* plane, which can introduce significant usability issues. If users are unaware of their hand position in this plane then they risk false-positive activation through incidental movements towards the screen, e.g., due to hand drift when pointing or when initially approaching the display [85]. Conversely, users may fail to activate an intended target if their hand does not move sufficiently far forward in the *z* plane, failing to complete the push gesture. The core issue is not the absence of a physical boundary (a trait of all mid-air interactions [168]), but rather that the user interface poorly convey the invisible spatial constraints that bound the activation.

Beyond the difficulty of determining the correct activation depth, the physical execution of a push-activation gesture can also introduce additional usability issues. The relationship between lateral hand movements and on-screen cursor motion in the *xy* plane is easily apparent, exhibiting greater precision along the horizontal and vertical axes than along the depth axis [13, 111]. The forward hand motion required for the push selection gesture inherently interferes with this fine-grained lateral pointing accuracy, since moving the hand forward often results in

incidental movement in the *xy* plane as well; this phenomenon is nicknamed the *Heisenberg Effect* (Section 2.4.3), where incidental cursor translation causes target slips [25], when the cursor leaves the boundary of a target (thus de-selecting it) before the push gesture is complete.

Despite these limitations, the push interaction paradigm can be effective when used to activate large buttons, since there is reduced likelihood of the cursor slipping off large targets during the forward push gesture. Furthermore, activation is a discrete event where users only need to maintain cursor stability until the target is activated. However, issues with the push gesture could become more problematic when used during continuous interactions, e.g., when dragging sliders and scrollbars, since users must sustain the pressed state over an extended trajectory.

3.1.2 Mid-Air Continuous Interactions

Continuous interactions (e.g., dragging operations with slider bars) require maintaining an activated state while simultaneously controlling position, often along a constrained trajectory. This is particularly challenging because the hand lacks tactile cues to stabilise movements [77]. When using the push activation gesture to select the slider handle, the user must guide the cursor towards the handle, perform the push gesture to activate it, translate the handle to its desired target value while maintaining the pressed state (at the correct depth), then finally release control of the handle. This effectively constitutes a 3D *steering* task [2]: users must track a specific trajectory while simultaneously managing hand depth to maintain the selection [93], a constraint that significantly increases interaction difficulty.

3.2 Chapter Structure

Section 3.3 details the interaction and gesture design used in this chapter, including the four activation techniques compared. Section 3.4 presents Experiment 1, covering the research aims, experimental design, task design, apparatus, measures, and procedure. Section 3.5 reports the results across all dependent measures. Section 3.6 presents an overall discussion of the findings. Conclusions are given in Section 3.7, also revisiting the research question raised earlier in the introduction in Section 3.8.

3.3 Interaction and Gesture Design

This section describes the design and implementation of the interaction mechanics used throughout this chapter. The core interaction paradigm was to enable the selection and manipulation of UI widgets on a 2D display using mid-air hand gestures. Hand translation was mapped to a 2D on-screen cursor at a 1:1 ratio, such that x mm of hand movement produced x mm of corresponding cursor movement, providing continuous visual feedback of pointing location (see

Figure 3.1). Cursor position was taken from the centre of the palm rather than a specific fingertip, decoupling finger movement from the pointing mechanism, ensuring the cursor remained more stable during the finger-to-thumb pinch gesture.

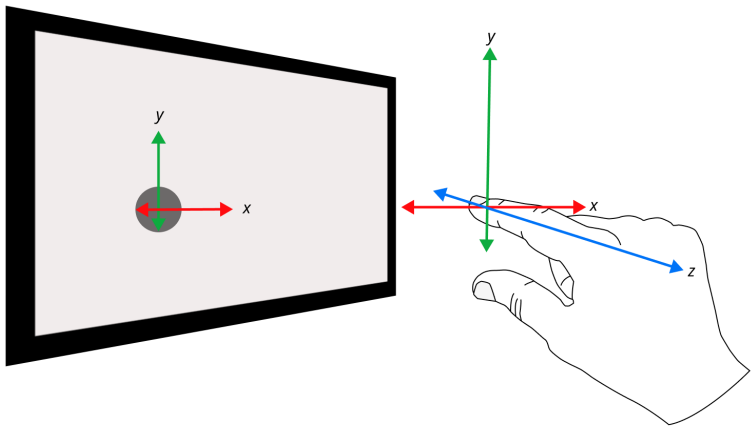


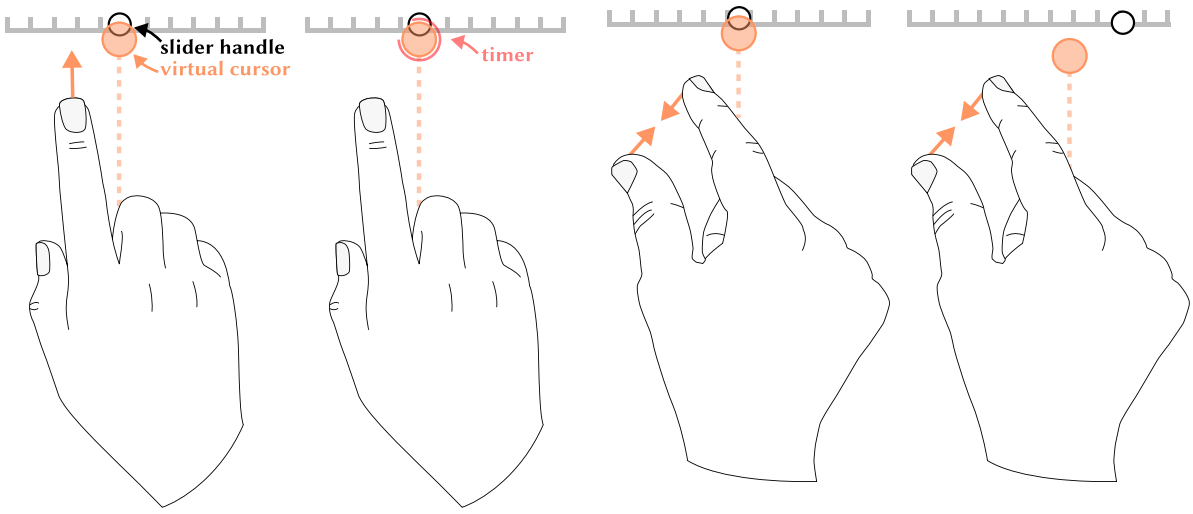
Figure 3.1: Visualisation of the interaction environment, showing the relationship between hand movement in mid-air and cursor movement on-screen

3.3.1 Gesture Design

Four mid-air selection techniques were chosen: *Push* (the baseline described above) and three alternatives, *Dwell*, *Pinch*, and *Area Pinch*. Each alternative was selected because it relaxes a different one of the constraints that make *Push* difficult, allowing the experiment to characterise which property of the activation gesture most affects usability. *Dwell* is an already established touchless activation method that removes hand displacement from the selection phase entirely: it requires no forward motion (increasing cursor stability) but introduces a fixed temporal threshold. *Pinch* retains an explicit, intentional gesture as the activation event, but moves it from the whole hand onto the fingers; this isolates the question of whether the push gesture's usability issues stem from the act of crossing a depth threshold itself, or from the lateral instability that whole-arm forward motion introduces. *Area Pinch* additionally relaxes the requirement that the cursor overlap the widget at the moment of activation, allowing the pinch to be performed from any comfortable hand position; this isolates the effect of spatial-precision constraints during target acquisition from the effect of the activation gesture itself. The following subsections detail how each technique was implemented.

Push

The *Push* technique (see Figure 3.2 (a)) required the user to move their hand forward along the z axis (towards the display) to transition from a pointing state to an active selection state,



(a) Push (b) Dwell (c) Pinch (d) Pinch Area

Figure 3.2: Illustration of the four selection techniques. The cursor position was mapped to the centre of the palm. Note that while these images show an extended index finger, this was not required.

matching the commercial-system behaviour described in Section 3.1.1. To activate a UI widget, the user first moved the cursor towards the widget (*xy* plane) and then moved their hand beyond a depth threshold (+15% from a moving-window average, to differentiate between depth arising from an intentional push versus hand position incidentally drifting over time) from their resting hand position, at which point the system registered the interaction as active. When in this state, movable UI widgets such as sliders could be manipulated. Upon bringing the hand back beyond the depth threshold (returning to the original z position), the active state was stopped, buttons were activated, and sliders were no longer manipulable.

Dwell

The *Dwell* technique (see Figure 3.2 (b)) required the user to hold the cursor over a UI widget for a short time to transition into an active selection state (temporal mode switch). *Dwell* was included here due to its prior use in mid-air interactive systems, and since it did not require hand displacement to enact selection (fewer degrees of freedom). *Dwell* was designed to replicate the selection mechanics of the Microsoft Xbox Kinect V1 [184]. To activate a UI widget, the user first moved the cursor towards the widget (*xy* plane), and once the cursor was on top of the widget, it had to be held for 800 ms (dwell time) to enter the active state. This time was chosen based on pilot testing between 600 ms [78] and 2 seconds [184]. When in an activated state, sliders could be manipulated, and buttons were activated. Once the cursor was removed from the UI widget, the active state was stopped (if it was already active), and the dwell time was reset.

Pinch

The *Pinch* technique (see Figure 3.2 (c)) uses an index finger-to-thumb pinch to transition from a pointing state to an active selection state (physical mode switch). *Pinch* was included here because of its emergence as the predominant activation method in modern Virtual Reality headsets [182], although less so in conventional touchless displays. To activate a UI widget, the user first moved the cursor towards the widget (*xy* plane) and then brought their index finger and thumb together ($\leq$ 35 mm apart to account for tracking noise), at which point the system registered the interaction as active. When in this state, movable UI widgets such as sliders could be manipulated. Upon bringing the index finger and thumb back apart (> 35mm), the active state was stopped, buttons were activated, and sliders were no longer manipulable. Unlike the *Push* gesture, which requires displacing the hand, the *Pinch* gesture could ostensibly be performed whilst maintaining a stable palm position to mitigate the cursor slipping off the target during a forward push motion. This gesture also has the potential advantage of providing implicit tactile feedback, with the finger-on-thumb contact providing a perceptible state change to support users belief about how the system functions.

Area Pinch

The *Area Pinch* technique (see Figure 3.2 (d)) extended the *Pinch* technique with dynamic area cursor behaviour (see Section 2.6.1, a concept also expanded upon in later chapters). Rather than requiring the cursor to directly overlap the widget to enter the *Pinch* activated state (see above), widget control was gained no matter where the cursor was located. *Area Pinch* was included to investigate whether removing the requirement for precise cursor-to-widget overlap before activation, by expanding the effective activation region to encompass the entire on-screen interaction area, could provide further performance improvements beyond those offered by the standard *Pinch* technique. By eliminating the fine-targeting step from the acquisition phase, the user could select and manipulate the slider from a more comfortable spatial position and minimise the necessary movement to acquire then reposition the slider. This allowed further analysis of how the *Pinch* activation technique served as a mode switch when isolated from spatial-precision demands. To activate a UI widget, the user moved the cursor (*xy* plane) without the need to overlap a widget, then brought their index finger and thumb together ($\leq$ 35 mm apart), at which point the system registered the interaction as active. When in this state, movable UI widgets such as sliders could be manipulated. Upon separating the index finger and thumb (> 35mm), the active state was stopped, buttons were activated, and sliders were no longer manipulable.

3.3.2 Visual Feedback

Across all selection techniques, users received continuous visual feedback via an on-screen circular cursor following their hand position. The cursor's appearance changed to communicate the current interaction state, helping users understand their proximity to activation and whether a selection had been registered (see Figure 3.3).

Idle

In its idle state, the cursor appeared as a grey circle at its default size (30 px), indicating that the hand was being tracked but no widget was currently targeted.

Hover

When the cursor overlapped with a widget such as a slider handle, the system entered a hover state. The widget was visually highlighted to indicate that it could be selected. When using the *Area Pinch* technique, this hover feedback remained permanently active, reinforcing that interaction was available from a broader range.

Activation Progress

The cursor also provided continuous feedback on the user's progress toward activation. For the *Push* technique, the cursor decreased in size proportionally as the user's hand moved forward along the depth axis. For the *Pinch* and *Area Pinch* techniques, the cursor similarly decreased in size as the distance between the user's finger and thumb closed, providing a continuous indication of how close the user was to completing the gesture. For the *Dwell* technique, a circular progress bar was rendered around the perimeter of the cursor during the dwell period, filling progressively to indicate the time remaining before activation was complete.

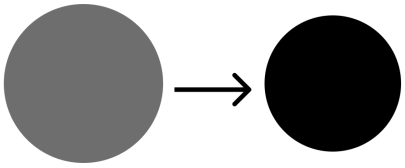
Active

Upon crossing the activation threshold and entering the active state, the cursor transitioned from grey to black and reduced in size to 25 px, providing a binary signal that selection had been registered. The cursor remained in this visual state when moving with the hand position. Once leaving the active state, the visual feedback returned to its idle state.

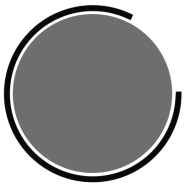
3.4 Experiment 1

3.4.1 Research Aims

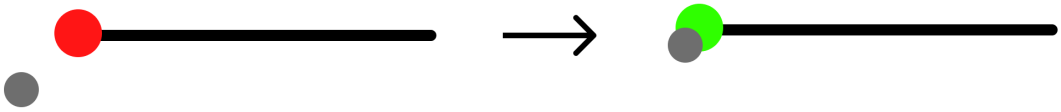
Experiment 1 investigates how different activation techniques affect the usability and control of mid-air slider bars, focussing on the problems highlighted in Section 2.4 associated with



(a) Cursor idle to active state change



(b) Example of dwell timer displayed around the idle cursor



(c) Example of the idle cursor and idle widget on a slider bar (left) and of the hover state (right)

Figure 3.3: Illustrations of the visual feedback states

depth-based activation during continuous mid-air pointing interactions. This experiment aimed to determine whether activation mechanisms with clearer state transitions, specifically the binary action of pinching or the temporal threshold of dwelling, could offer a more stable alternative for acquiring and maintaining control of slider widgets.

A secondary aim was to explore how relaxing the spatial precision required for target acquisition affects subsequent interaction performance. Fitts' Law predicts that a larger activation region should reduce acquisition time; this experiment investigates whether the manner in which a user acquires control of a slider handle affects the subsequent accuracy and stability of the steering phase. By including the *Area Pinch* technique, this experiment investigated whether the benefits of relaxed spatial constraints during target acquisition extend to the continuous control required for slider manipulation.

3.4.2 Experiment Design

A within-subjects experiment design was used, with selection technique and task type as independent variables, providing 12 conditions (four selection techniques $\times$ three task types). Condition order was balanced using a Latin Square, and participants completed 12 selection trials per condition, resulting in 144 selections per participant.

3.4.3 Task Design

Three task types were chosen for this experiment, all involving moving a horizontal slider handle of fixed length to a designated target position. The **Digit** task (Figure 3.4 (a)) required users to move the slider handle to select a value from 0–9, representing a relatively wide target width across ten discrete positions. The **Letter** task (Figure 3.4 (b)) required users to select a character from A–Z, producing narrower targets across twenty-six positions. Both tasks represent slider selections one might expect to find on a public display, such as selecting numerical quantities at a supermarket checkout or filtering the start letter of destinations on a ticket machine. The third **Scale** task (Figure 3.4 (c)) required users to adjust the slider to resize a square (red) until it matched a reference square (black) shown behind, demanding near pixel-perfect positioning and thus representing the most difficult condition.

All tasks started with the slider in the left-most position of the bar to replicate real-world usage, where the slider handles were not stepped and allowed free movement across the entire bar. For the digit and letter tasks, fixed sets of targets (randomised per participant) were used, with no selections made within the first three slider targets (i.e., 0–2 and A–C), ensuring some distance between the start and target positions. For the scale task, fixed sets of targets (randomised per participant) were located along the slider bar, with the target square proportional to the maximum and minimum values of the slider. Target values were at least 5 cm along the bar to ensure distance between the start and target positions. The screen showed a single horizontal slider bar (Figure 3.4) that occupied 55% of the screen width, for a real width of 35.75 cm.

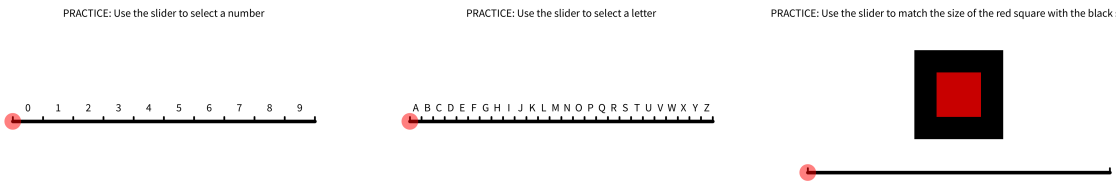
Widget Styling

Widgets used a deliberately plain, high-contrast style on a white background, modelled on the flat widgets of commercial touchless kiosk interfaces [157]: a black slider track (approximately 4 mm thick) with tick marks and labels, and a semi-transparent red circular handle. The handle was approximately 19 mm (75 px) in diameter, deliberately larger than the 30 px cursor so that the cursor remained visible when overlapping it and so that precise (overlap-based) targeting was not unduly difficult. No audio feedback was given; all feedback was visual. This styling and feedback design was carried over to Experiments 2–5 so that results could be compared across experiments.

3.4.4 Apparatus

The experimental software ran full-screen on a 27” display (2560 $\times$ 1440 px, 65 $\times$ 49.1 cm) with the visual feedback and GUI rendered using Processing[1] to enable customisation of cursor presentation and appearance.

[1] https://processing.org/ - Last accessed in January 2026



(a) Digit Task (b) Letter Task (c) Scale Task

Figure 3.4: Screenshots showing each task type as shown to the participants. The semi-transparent red circle is the slider handle. For the digit and letter tasks, slider ticks were rendered to indicate target width. For the scale task, participants had to use the slider to adjust the size of the semi-transparent red shape so that it matched the black shape.

A Leap Motion controller was used to track users' hands and fingers for input. The sensor's field of view is 150°×120°and provides reliable finger tracking at short distances. Gesture detection and hand tracking were facilitated using the Ultraleap Java SDK[2]. Following Ultraleap's recommendations [108], the sensor was placed 15cm below and 10cm in front of the display, centred underneath it, with a keyboard positioned in front. The sensor was angled so it pointed 15° towards the participant, allowing them to place their hand farther from the sensor and gesture from a more relaxed posture (to improve sensing quality and reduce fatigue [80]).



Figure 3.5: Hardware used for the experiment: 27" monitor, Leap Motion Controller and wireless keyboard

3.4.5 Measures

For each task, the overall **task time** was measured, the **time to acquire slider** handle, and the time taken for the slider cursor to first land on the target (**time to target**). This division helped understand how each selection technique and task type influenced different interaction phases

[2]https://www.leapmotiontechnology.com/developer-sub/documentation/java/index - Last Accessed January 2026

and how these may affect the initial *ballistic* phase of cursor movement [119] as they begin moving the slider towards the target. The period not covered by these measurements relates to user adjustments after overshooting or releasing the handle, as the focus was primarily on slider acquisition and initial movement. Figure 3.6 shows how these measurements relate to key interaction events. Time was started when the participant's hand reached the sensor and continued until the space bar was pressed on the keyboard.

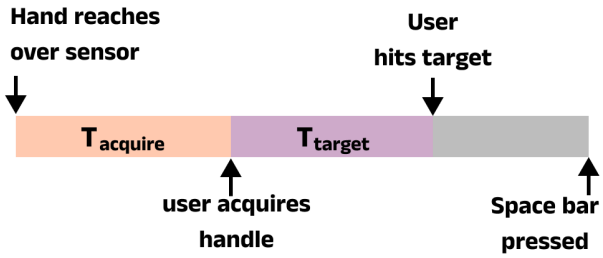


Figure 3.6: This timeline shows how task timing measures are calculated from interaction events, where T denotes time.

Additionally, the final cursor position when the slider handle was released was measured and used to calculate the horizontal **error distance** in pixels between the cursor and the intended target (for the digit and letter tasks, this was the centre of the target). Additionally, **task success**, based on whether the final cursor had acquired the intended target; for the scale task, which required a much greater degree of precision (i.e., pixel-perfect match), the task was considered a success if the cursor was within a 25 pixel (5 mm) threshold of the intended target position.

At the end of the session, after experiencing all four techniques, participants were asked to rank the selection techniques in order of preference (Appendix A), then to discuss why they ranked them in that order and reflect on their experience. Within the session, after each block of tasks for a given technique, participants also rated their agreement with the following statements on a seven-point Likert scale (Appendix A), capturing their immediate impressions of the technique they had just used:

1. I was able to easily move the slider handle to a specific point once the handle was activated
2. I was in control of the slider once the handle was activated
3. The slider was responsive to my movements once the handle was activated

All semi-structured interviews were audio recorded with consent and transcribed. Each transcript was read independently by two readers — the author and a member of the supervision team, who separately highlighted passages relevant to the research questions; the highlighted extracts were then compared and, where both readers had identified the same passage, carried forward and grouped into descriptive categories by consensus. No formal thematic analysis was undertaken and no inter-rater reliability statistic was calculated, so the quotes reported here are

illustrative participant commentary that contextualises the quantitative results rather than themes generated through a formal coding process.

3.4.6 Procedure

Before beginning the trials, participants provided informed consent and were introduced to each selection technique through practice tasks. Participants completed a minimum of 12 practice targets per technique and task type, though they were free to continue practising beyond this minimum until they felt comfortable. Each trial began with the participant's hands resting by their side; on-screen instructions indicated the required action (e.g., *"Use the slider to select the letter N"*), and the participant then reached forward into the Leap Motion tracking area. There were no time constraints despite being asked to interact *"as quickly and as accurately as possible"* , and participants were permitted to release the slider and make corrections at any point rather than being limited to a single continuous movement. A trial ended after pressing the keyboard spacebar, as described before. After each block of tasks for the selection technique, participants completed the agreement statements. Finally, after all trials, preference rankings were taken, and reasoning was discussed in the interview.

3.5 Results

15 participants were recruited (7 male, 8 female, mean age 27.7 years, SD 5.61 years, 14 right-handed and 1 left-handed) through university mailing lists and poster advertising. Participants were compensated with £10 for taking part in the study, which lasted 1 hour. Ethics approval for this study was provided by the College of Science and Engineering Ethics Committee.

A repeated-measures ANOVA was used for analysis, followed by post-hoc t-test comparisons of estimated marginal means. If data did not have a normal distribution (via the Shapiro-Wilk test), then the Aligned-Rank Transform [180] was applied first, followed by post-hoc comparisons using ART-C [49] where appropriate.

Below are summary statistics and visualisations for each dependent variable. Table 3.2 shows the results and analyses across all conditions, including summary statistics for each dependent variable, and results from the repeated-measures ANOVA and post-hoc comparisons.

3.5.1 Task Time

Overall, participants took an average of 9188 ms (SD = 3620 ms) per trial (Table 3.2 rows 1-3). Across Selection Techniques as shown in Figure 3.7, Area Pinch had the fastest mean completion time (M = 8195, SD = 3547) ms, while Dwell was the slowest (M = 10100, SD = 3280) ms. Push (M = 8719, SD = 3687) ms and Pinch (M = 9704, SD = 3751) ms. The Scale task took

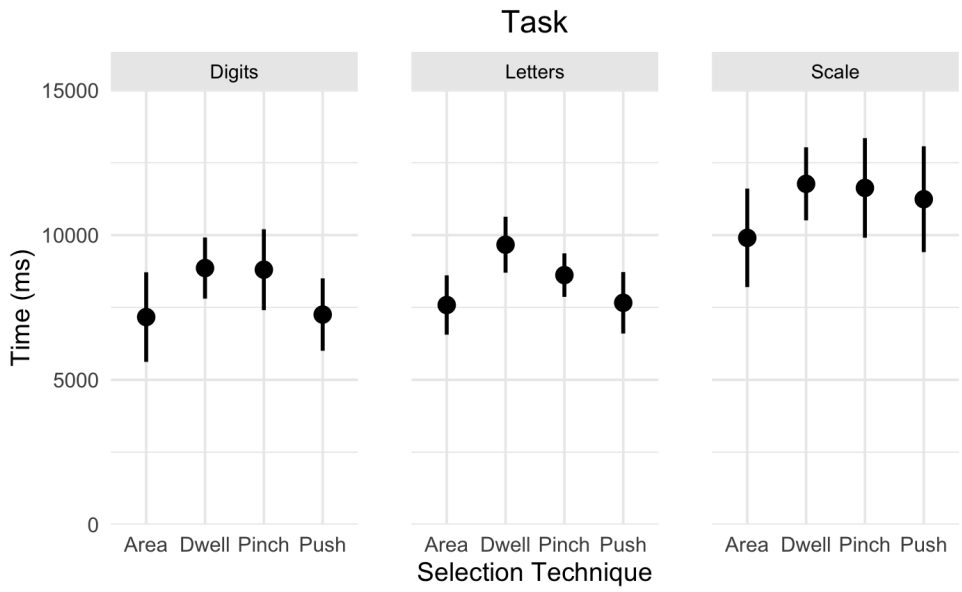


Figure 3.7: Mean task time per condition. Error bars show 95% CIs.

longer (M = 11157, SD = 4074) ms than both Digits (M = 8009, SD = 2873) and Letters (M = 8397, SD = 2986) ms.

3.5.2 Time to Acquire

Participants took an average of 1816 ms (SD = 793 ms) per trial to acquire the slider handle as shown in Figure 3.8 and Table 3.2 rows 4 - 6. Across Selection Techniques, Area Pinch had the fastest mean completion time (M = 1033, SD = 490) ms, while Push was the slowest (M = 2287, SD = 826) ms. Dwell (M = 2212, SD = 481) ms and Pinch (M = 1694, SD = 618) ms. The Digits task took noticeably longer (M = 1894, SD = 841) ms than both Scale (M = 1787, SD = 818) and Letters (M = 1766, SD = 723) ms.

3.5.3 Time to Target

Overall, participants took an average of 4659 ms (SD = 1488 ms) per trial to reach the target as shown in Figure 3.9 and Table 3.2 rows 7 - 9.. Across Selection Techniques, Area Pinch had the shortest time to reach the target (M = 3657, SD = 1570) ms, while Dwell was the slowest (M = 5565, SD = 1006) ms. Push (M = 4736, SD = 1463) ms and Pinch (M = 4633, SD = 1248) ms. The Scale task took longer (M = 5257, SD = 1538) ms than both Digits (M = 4528, SD = 1557) and Letters (M = 4192, SD = 1151) ms.

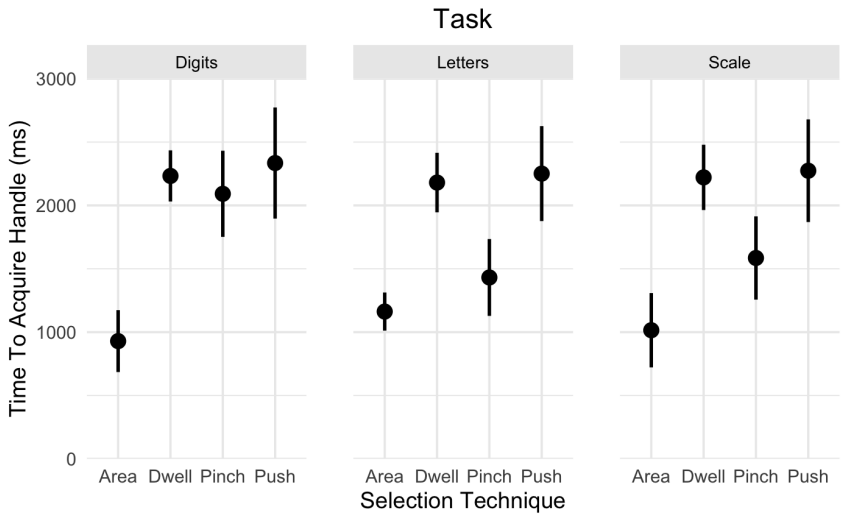


Figure 3.8: Mean time to acquire the slider handle per condition. Error bars show 95% CIs.

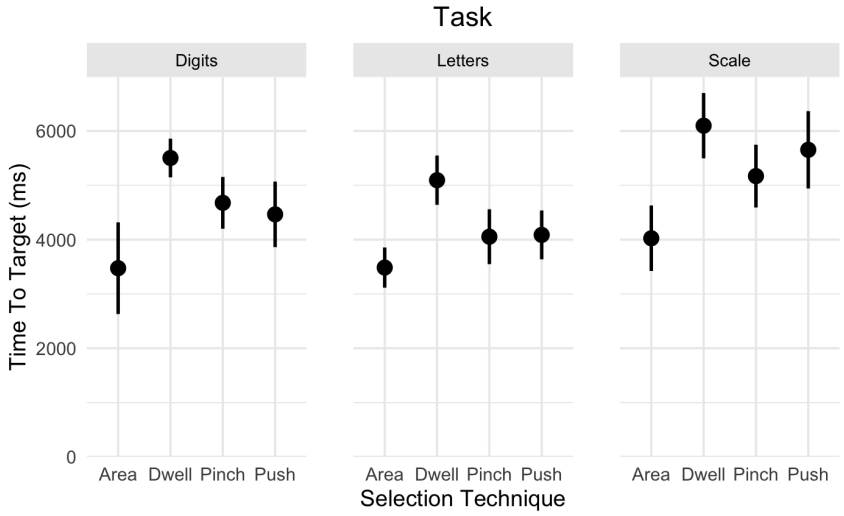


Figure 3.9: Mean time to reach the target per condition. Error bars show 95% CIs.

3.5.4 Error Distance

The mean error distance was 1.19 px (SD = 7.48 px) per trial as shown in Figure 3.10 and Table 3.2 rows 10 - 11.. Across Selection Techniques, Area Pinch was the only technique to undershoot the target on average (M = -1.19, SD = 10.51) px, while Dwell produced the largest overshoot (M = 2.71, SD = 7.23) px. Pinch (M = 2.07, SD = 5.69) px and Push (M = 1.09, SD = 5.06) px also overshot. Across tasks, error distances were broadly similar, though Digits showed the largest mean overshoot (M = 1.58, SD = 10.29) px compared to both Letters (M = 1.12, SD = 4.83) and Scale (M = 0.89, SD = 6.34) px.

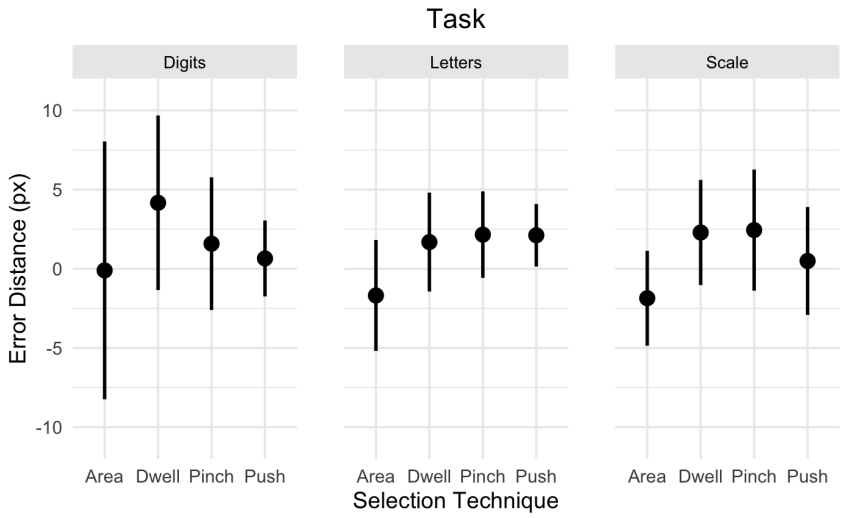


Figure 3.10: Mean error distance per condition. Error bars show 95% CIs.

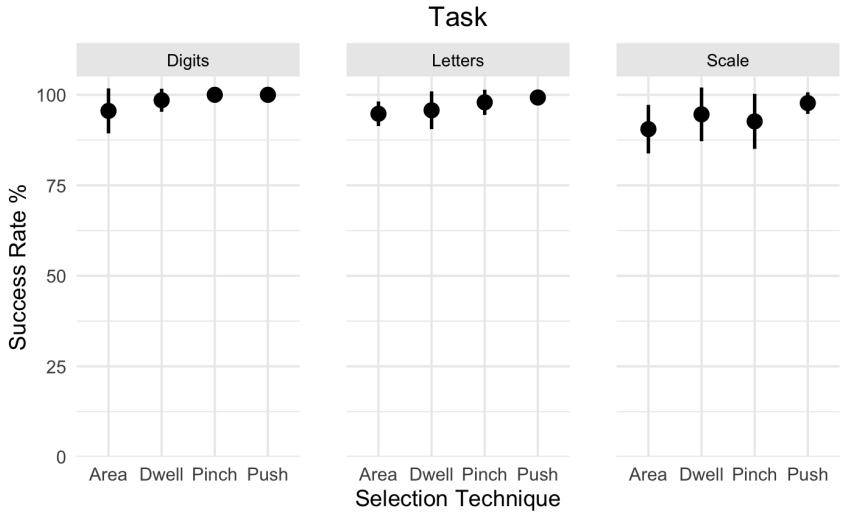


Figure 3.11: Mean success rate (%) per condition. Error bars show 95% CIs.

3.5.5 Success Rate

Participants achieved a mean success rate (finishing on the correct target) of 96.46% (SD = 8.99%) per trial as shown in Figure 3.11 and Table, 3.2 rows 14 - 15. Across Selection Techniques, Push had the highest mean success rate (M = 98.99, SD = 3.31) %, while Area Pinch was the lowest (M = 93.66, SD = 10.26) %. Pinch (M = 96.79, SD = 9.72) % and Dwell (M = 96.28, SD = 10.27) %. Across tasks, Digits achieved the highest success rate (M = 98.49, SD = 5.55) % compared to both Letters (M = 96.96, SD = 6.54) and Scale (M = 93.93, SD = 12.69) %.

Table 3.1: Summary of Likert scale responses. Note responses were on a scale from 1 ("Strongly Disagree") to 7 ("Strongly Agree").

	Push		Dwell		Pinch		Area Pinch	
Question	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
Q1	5.5	1.1	4.9	1.4	4.5	1.5	5.2	1.0
Q2	5.3	1.0	4.3	1.8	4.0	1.4	4.8	1.4
Q3	5.7	0.8	4.7	1.2	4.7	1.2	5.3	1.4

3.5.6 Survey Results

Table 3.1 shows an overview of the Likert scale results for agreement with the statements on slider control. For Q1 (*"I was able to easily move the slider handle to a specific point once the handle was activated"*), Friedman's test found a significant main effect of technique: $\chi^2 = 11.0, p = .012$. Post hoc Nemenyi tests found no significant differences (all $p \geq .07$). Participants generally agreed that all four techniques enabled them to easily control the slider, with all per-technique means above the neutral midpoint (Push: 5.5/7, Dwell: 4.9/7, Pinch: 4.5/7, Area Pinch: 5.2/7).

For Q2 (*"I was in control of the slider once the handle was activated"*), Friedman's test did not find a significant main effect of technique: $\chi^2 = 6.9, p = .074$. Participants generally agreed that Push (5.3/7) and Area Pinch (4.8/7) enabled them to stay in control of the slider, although responses for Dwell (4.3/7) and Pinch (4.0/7) were closer to the neutral midpoint of 4/7.

For Q3 (*"The slider was responsive to my movements once the handle was activated"*), Friedman's test found a significant main effect of technique: $\chi^2 = 9.1, p = .03$. Post hoc Nemenyi tests found no significant differences (all $p \geq .15$). Participants generally agreed that all four techniques were responsive to their gestures, with all per-technique means above the neutral midpoint (Push: 5.7/7, Dwell: 4.7/7, Pinch: 4.7/7, Area Pinch: 5.3/7).

3.5.7 Preference Ranking

Figure 3.12 shows the distribution of preference ranks for the techniques. Friedman's test found a significant difference in the rankings: $\chi^2(3) = 14.6, p = .002$. Post hoc Nemenyi tests found that Area Pinch was ranked significantly more highly than Dwell ($p = .01$) and Pinch ($p = .006$). No other significant differences were found (all $p \geq .19$).

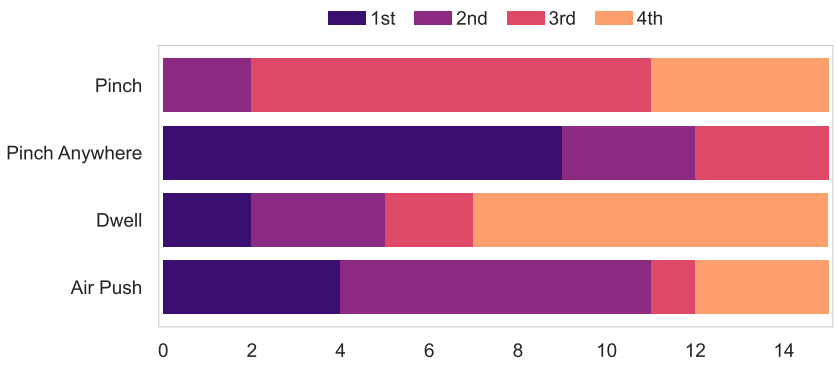


Figure 3.12: Preference ranking distribution for each interaction technique.

Table 3.2: ANOVA results, and post-hoc test results for Experiment 1. Significant effects are highlighted*

	Time	**Mean 9188 ms, SD 3619 ms, 95% CI [8651, 9725] ms**			
	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
1	*Technique*	F(3, 151) = 5.80	< .001*	.10	• Dwell > Area (p = .003) • Pinch > Area (p = .02) • Push > Dwell (p = .03)
2	*Task*	F(2, 151) = 24.28	< .001*	.24	• Scale > Digits (p <.001) • Scale > Letters (p <.001)
3	*Technique* × *Task*	F(6, 151) = 0.49	.82	.02	
	Time to Target	**Mean 4659 ms, SD 1488 ms, 95% CI [443, 4880] ms**			
4	*Technique*	F(3, 151) = 24.48	< .001*	.33	• Area < all (p <.001) • Pinch < Dwell (p <.001) • Push < Dwell (p <.001)
5	*Task*	F(2, 151) = 16.71	< .001*	.18	• Scale > all (p <.001
6	*Technique* × *Task*	F(6, 151) = 0.94	.47	.04	
	Time to Acquire	**Mean 1816 ms, SD 793 ms, 95% CI [1698, 1933] ms**			
7	*Technique*	F(3, 151) = 45.75	< .001*	.48	• Area < all (p <.001) • Dwell > Pinch (p <.001) • Push > Pinch (p <.001)
8	*Task*	F(2, 151) = 1.13	.33	.01	
9	*Technique* × *Task*	F(6, 151) = 1.97	.07	.07	

Continued on next page

Table 3.2: (continued)

	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
	Error Distance	**Mean 1.19 px, SD 7.48 px, 95% CI [0.08, 2.30] px**			
10	*Technique*	F(3, 151) = 3.55	.02*	.07	• Dwell > Area (p= .01)
11	*Task*	F(2, 151) = 0.18	.83	.00	
12	*Technique* × *Task*	F(6, 151) = 0.57	.76	.02	
	Success	**Mean 96 %, SD 9 %, 95% CI [95, 98] %**			
13	*Technique*	F(3, 151) = 7.24	< .001*	.13	• Push > Area ($p\leq$ .001) • Pinch (p= .007)
14	*Task*	F(2, 151) = 1.72	.18	.02	
15	*Technique* × *Task*	F(6, 151) = 1.52	.17	.06	

3.6 Discussion

3.6.1 Acquiring Slider Control

It took significantly less time for users to acquire control of the slider handle when using Area Pinch (1033 ms) and Pinch (1694 ms) than when using Push (2286 ms) and Dwell (2212 ms), as shown in Figure 3.8. The roughly 600 ms advantage of Pinch over Push is notable given that both techniques are explicit two-state gestures and require users to guide the cursor to the slider handle before activation. The difference is therefore unlikely to lie in the pointing phase or in the binary nature of the gesture itself, but rather in how the activation gesture interacts with the cursor: a forward push along the depth axis while maintaining lateral cursor accuracy introduces the Heisenberg effect discussed in Section 3.1.1, where incidental lateral translation during the push can cause target slips. Pinching, by contrast, is mechanically decoupled from the cursor's lateral position, the activation involves only finger articulation rather than displacement of the whole hand, so the cursor remains comparatively stable during the state transition. This also brings a biomechanical advantage: a finger-thumb pinch involves considerably less lean body mass than a whole-arm forward push, so the gesture can be initiated and completed more quickly and with smaller corrective adjustments along the way.

Area Pinch was faster still, reducing acquisition time by approximately 660 ms compared to standard Pinch. Since Area Pinch removed the requirement for the cursor to overlap with the handle before activation, users could initiate the pinch gesture from any comfortable spatial position and immediately gain control of the slider. This is consistent with Fitts' Law: by

effectively expanding the activation region to the wider interaction space, Area Pinch reduced the fine targeting phase, allowing users to *"focus on the task instead of the slider"* [P2] and *"take control without making an effort"* [P12]. The acquisition-time advantage suggests that relaxing spatial-precision constraints during target acquisition has a meaningful impact on interaction speed, a finding that extends to the subsequent steering phase.

Since the time to acquire slider control encompasses only the targeting and activation phases, the pinch gesture's speed advantage has implications beyond slider manipulation. This phase of the interaction (navigating to the slider handle and triggering a state change) is functionally equivalent to button pressing. Pinching could therefore serve as a promising complementary or alternative activation method to Push for discrete selections as well, particularly where rapid, reliable mode switching is desirable.[3]

It was interesting that there was no significant difference in slider acquisition time between Push and Dwell, given that the 800 ms dwell period was not subtracted from the timing data. Here, it would be reasonable to expect Dwell to be at least 800 ms slower, yet participants were observed having issues activating the Push gesture over the slider widget, often requiring multiple attempts to cross the depth threshold at the correct lateral position (consistent with earlier findings on push activation difficulty [174]). This difficulty appears to have increased Push acquisition times to offset the dwell period, producing comparable performance between the two. However, the dwell period's impact became more severe over the task duration, with Dwell being the slowest technique overall (10100 ms vs 8719 ms for Push). This was driven by a compounding effect in which participants accidentally released the slider by moving the cursor off the widget and were forced to re-dwell for another 800 ms to regain control, accumulating over the task. Some participants found that remaining steady during the dwell period made it *"easier to select"* [P4, P8], but for most, repeated dwelling over the course of slider manipulation became a source of frustration, described as more *"tiring"* [P10] and *"more time consuming"* [P12, P14]. This likely explains why Dwell was ranked the least preferred technique by a majority of participants despite its comparable initial acquisition speed.

3.6.2 Using Area vs. Precise Targeting

The Area Pinch technique can be understood as an implicit form of *area cursor* where, rather than requiring a single-point cursor to precisely overlap with the target widget, the pinch gesture effectively expands the activation region to encompass the available space. This decoupling of activation from cursor position meant that, upon pinching, the system assigned control of the nearest widget regardless of where the cursor was located. In later chapters, this concept of dynamically expanding the effective selection region is explored in much greater depth; here,

[3] Since this experiment was carried out, pinch as activation has become considerably more widespread, having been adopted as the primary input gesture across extended-reality headsets (e.g., Apple Vision Pro, Meta Quest 3) and wrist-worn input devices (e.g., the Meta Neural Wristband).

however, Experiment 1 provides early evidence that even a rudimentary area-based acquisition strategy can yield substantial performance and preference benefits over point-based targeting for mid-air interactions.

Participants largely preferred Area Pinch because it allowed them to control the slider from wherever hand was most comfortable at the time, bypassing the fine-motor demands of precise cursor-to-widget alignment. This reduced the acquisition phase to a simple mode switch with a pinch gesture performed at the user's current hand location rather than a compound action of steering followed by activation. The result was that users acquired slider control on average 700 ms faster than with the standard Pinch technique, which was the next fastest. The preference data reinforced this, with nine of fifteen participants ranking Area Pinch first, many noting they could *"focus on the task instead of the slider"* [P2] and *"take control without making an effort"* [P12].

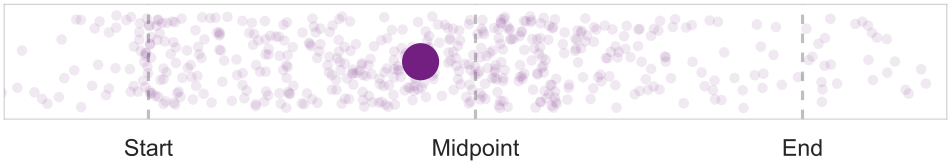


Figure 3.13: Scatterplot showing distribution of cursor x-axis coordinate when the user first acquired control of the slider handle via the Area Pinch gesture. The x-axis shows the slider start, mid, and end points. The large dot shows mean start position.

Figure 3.13 shows the distribution of cursor x-axis coordinates at the moment users first gained control of the slider handle via Area Pinch (mean start point was at 42% of the slider bar width). The starting positions were distributed across the full width of the slider bar, with some users even initiating control from positions beyond the slider's horizontal extent. This distribution illustrates the area cursor behaviour in practice, with participants not converging on the slider handle before activating, as they would with a standard cursor, but instead acquiring control from wherever their hand happened to be. Regardless of where the hand is positioned when the user takes control, they still need to make the same hand movements to translate the slider handle. Care must be taken to remain within the sensor's tracking range. For example, if a participant 'pinches' the slider handle with the cursor near the end of the slider bar, further sideways movement will move their hand even further from the centre of the tracking space, away from the "sweetspot" for tracking [57]. This could lead to tracking difficulties and, indeed, seemed to cause some issues with the Area Pinch technique. If a user took control of the slider with the cursor near the end, moving the full length of the slider would take their hand approximately 53.7cm to the right of the sensor centre (see Figure 3.5), approaching the limitations of the Leap Motion tracking range. As Figure 3.13 shows, some people even took control *beyond* the end of the slider, placing them in a range where sensing quality is not optimal.

This highlights an important design consideration: the UI's physical extent must be balanced

with the sensor's tracking volume. When users are free to acquire control from any position, the effective workspace they need during manipulation is the sum of their initial offset plus the widget's full traversal distance. Designers should therefore either constrain widget placement so that the maximum possible offset-plus-traversal distance remains within reliable tracking bounds or implement corrective mechanisms, such as gain adjustments (discussed in subsequent chapters) that compensate for peripheral hand positions after acquisition. Alternatively, widgets requiring large continuous movements (such as full-width sliders) may benefit from a hybrid approach, such as using area-based acquisition for the speed and comfort it affords, but snapping the cursor to the widget's current position upon activation, thereby eliminating the spatial offset before the steering phase begins.

3.6.3 Enhanced Cursor Targeting and Translation

Despite the user experience benefits and faster acquisition times for techniques that allow control of the slider handle from 'anywhere', precise cursor-based selection methods can still achieve the task. Push may have taken longer to gain slider control and first reach the target region of the slider, yet performed well in terms of task time, and had the highest overall task success rate (99% vs the mean of 96.5%). This is largely due to the movement dynamics participants employ when using Push. The need for precise targeting, hand stability and consistent hand depth led participants to make slower and more controlled hand movements, which meant they were less likely to lose control of the slider (e.g., "*I had to move very slowly to avoid falling off the circle and losing control*" [P3]). This is consistent with Kattinakere's model of mid-air steering [93], which shows that the need to maintain consistent depth leads to slower movement to maintain precision. In contrast, the relaxed constraints of Area Pinch may have led to less precision during the slider tasks, resulting in imprecise initial movements followed by additional corrective movements, as reflected in the overall task time.

3.7 Chapter Conclusions

This chapter presented Experiment 1, which compared four mid-air selection techniques (Push, Dwell, Pinch, and Area Pinch) for a series of continuous slider manipulation tasks. The push selection technique, despite its adoption in commercial systems, introduces usability challenges in the ambiguity of depth-based activation and its interference with lateral pointing accuracy. The results presented in this chapter demonstrate that the choice of activation technique has a meaningful impact on both the speed and subjective experience of acquiring and maintaining control of mid-air slider widgets.

Pinch-based techniques were significantly faster for acquisition than both Push and Dwell, largely because the closing finger-thumb gesture provides proprioceptive feedback that signals proximity to activation. This internal feedback is absent for Push, where activation occurs at

an invisible depth plane. Pinching also decouples activation from the pointing hand's position, avoiding the incidental cursor displacement that accompanies a forward push.

Pinch-based techniques were significantly faster for acquisition than both Push and Dwell. Here, the pinch gesture is decoupled from the cursor's lateral position, so executing an activation does not displace the cursor in the way that a forward push does, and the smaller, faster finger-thumb movement involves considerably less lean body mass than a whole-arm push. Additionally, using finger-thumb contact may also offer a useful proprioceptive cue at the moment of activation indicating to users when they have entered an activated state.

Area Pinch was the fastest and most preferred technique, reducing acquisition to a simple mode switch executed from wherever the hand was positioned. However, acquiring control from too far away could result in the hand moving towards the boundaries of reliable tracking during the slider manipulation part of the interaction, which could introduce new usability concerns. Dwell was the slowest and least preferred technique, in part because of the 800 ms pause required to establish intentional targeting. While its initial acquisition time was comparable to Push, accidental cursor slips during steering forced repeated re-dwell periods, compounding over the task. For continuous interactions requiring sustained activation, temporal mode switches are poorly suited. Push, though slower to acquire and less preferred, achieved the highest task success rate, suggesting that activation technique influences not only acquisition but also the movement dynamics of the steering phase.

These findings are within the context of a single-widget interface, giving formative insights into targeting and control behaviours without distractor targets; the next chapter describes experiments that involve dynamic area cursor interactions in the presence of multiple targets, giving insight into how these perform in more realistic user interfaces. This experiment (Experiment 1) also used a 1:1 control-display mapping on a 27-inch display; the balance between acquisition speed and steering precision under different gain mappings are also later examined in Experiments 4 and 5 (Chapter 5).

3.7.1 Chapter Contributions

This chapter makes the following contributions:

1. A comparative evaluation of four mid-air activation techniques (Push, Dwell, Pinch, and Area Pinch) for continuous slider control, demonstrating that pinch-based activation significantly reduces widget acquisition time and is preferred by users over depth-based and temporal alternatives.

2. Initial evidence of how activation technique influences movement dynamics beyond the acquisition phase, showing that the constraints imposed by different techniques shape the speed-accuracy trade-off during continuous slider steering.

3.8 Research Question 1

The outcomes of this experiment provide initial evidence towards answering the first research question of this thesis:

> **RQ1** - How does the choice of target activation technique affect the usability of a mid-air user interface?

The results here provide a first, partial answer to RQ1. The pinch gesture significantly reduces the time to acquire control of a slider widget compared to the commonly-used push activation method, with acquisition times approximately 600 ms faster. This advantage stems from that pinch gestures are mechanically decoupled from the cursor's lateral position, so it does not move the cursor during the state transition as a forward push can. This is reinforced by biomechanics, a finger-thumb pinch is a smaller, faster movement than a whole-arm push and therefore introduces less opportunity for incidental cursor displacement. Pinch gestures may also offer a proprioceptive/passive haptic indication of the state change, offering a clearer signal to users. Area Pinch extends the pinch advantage by eliminating the need for precise initial targeting, though at the cost of potentially placing users outwith reliable sensing regions during subsequent interaction. Push, while slower and less preferred, promotes more deliberate movements, resulting in marginally higher task accuracy. Therefore, the activation technique affects the usability of mid-air input, not only in terms of acquisition speed and subjective preference, but also in the movement strategies adopted during continuous widget manipulation.

The findings here are based on a single-widget interface, a fixed 1:1 control-display mapping, and a single display size. RQ1 is revisited in the chapters that follow: Chapter 4 investigates how a dynamic area cursor can manage the targeting phase across multiple widgets (building on the area-based acquisition principle demonstrated here); Chapter 5 examines how control-display gain can be tuned in conjunction with area cursors to balance speed and precision while diminishing the likelihood of leaving the optimal tracking area.

Chapter 4

Experiments 2 & 3: The Proxemic Cursor

4.1 Introduction

Reduced precision, in comparison to conventional pointing devices, is one of the main usability problems affecting mid-air cursor interaction. As discussed in Section 2.4.3, the unsupported hand is less stable than conventional input devices: natural hand tremor is can degrade Fitts' Law throughput [13, 178], the lack of physical support and grounding removes feedback that normally stabilises pointing [120, 186], and accumulating shoulder fatigue from a raised hand progressively degrades fine motor control [80, 87].

Chapter 3 established that the choice of activation technique has a measurable impact on the speed, accuracy and subjective experience of mid-air slider control. Among the four techniques evaluated, *Area Pinch* was the fastest and most preferred, largely because it decoupled widget acquisition from precise cursor positioning, where users could initiate a pinch gesture from a more comfortable hand location to activate their desired target. Such an approach could be a promising direction towards reducing the precision demanded by conventional cursor interfaces, especially in application contexts where the user interface layout cannot be wholly redesigned to accommodate a new input modality (as discussed earlier).

This style of targeting, in which activation can occur without directly overlapping the visual boundary of the target, can be seen as a rudimentary form of a dynamic area cursor (Section 2.6.1), where the effective selection region extends beyond the single point of the cursor to enclose at least one nearby target. However, Experiment 1 was deliberately constrained to a single on-screen widget (a UI slider) to isolate the effects of the activation technique on continuous interactions. In practice, mid-air interfaces contain many widgets arranged across the display, where acquiring and transitioning between them is a core part of the interaction flow. When more than one widget is present, area-based acquisition must navigate boundaries and resolve which widget the user intends to control. Whether the single-widget feasibility result from Chapter 3 generalises to multi-widget layouts, to discrete (button) selection, and to interfaces in which widget boundaries shift during interaction (e.g., as slider handles move) are open

questions. The work presented in this chapter investigates area cursor targeting, in continuation of the previous experiment, with a view towards answering the following questions:

> **RQ1** - How does the choice of target activation technique affect the usability of a mid-air user interface?
>
> **RQ2** - How can we reduce the need for traversal and selection accuracy in a mid-air user interface across different display sizes and widget densities?

This chapter presents two experiments to address these questions, alongside **the proxemic cursor**: a dynamic area cursor for touchless interfaces that implicitly targets the widget nearest to the cursor. **Experiment 2** examines whether the acquisition and transition benefits of area-based targeting hold for continuous (slider) manipulation when users must move between adjacent sliders across multiple layouts. **Experiment 3** extends the investigation to interfaces containing both continuous (slider) and discrete (button) widgets at varied inter-widget separations, testing whether area-based targeting also benefits discrete touchless input and how users manage the boundaries between adjacent widgets. Both experiments also continue the comparison of *Push* and *Pinch* activation gestures from Chapter 3, re-examining the activation-technique findings of Experiment 1 in multi-widget settings and in combination with area-based targeting.

4.1.1 Area Cursor Interaction

Area-based interaction techniques have been used across various input modalities to increase pointing and targeting performance. In desktop environments, techniques such as dynamically resizing targets near the cursor [75] or resizing the cursor itself to snap to the nearest target, as in the Bubble Cursor [71], reduce the index of difficulty predicted by Fitts' Law by effectively expanding the selectable region around each target. These concepts have been adapted for mid-air pointing, where augmented cursors that encompass the closest target have demonstrated improved gestural selection performance [48], and for ray-cast pointing in virtual reality [11]. Where interface elements cannot be resized, virtual proxy targets have been introduced to enable selection by targeting the area surrounding the nearest element [58]. Multimodal approaches have also been explored: Gaze-Touch [139] and Gaze+Gesture [33] combine eye tracking with touch or mid-air input for cursor refinement, while Gaze-Hand Alignment [113] minimises hand movement at the start of an input operation. Other approaches bypass cursor-based targeting entirely, such as summoning widgets to the cursor position [74] or using cursor-less temporal correlation gestures [39, 165] though these diverge substantially from the familiar cursor-and-click model and may increase learning costs. These techniques share a common principle: the effective interaction area is expanded beyond a single point, either by enlarging the target width, reducing the distance to the target, or both. Of these strategies, dynamic area cursors are particularly relevant here as they retain the familiar cursor-and-click model while relaxing the positional precision required for acquisition, as that made Area Pinch effective for single-widget control.

Dynamic Area Cursors

As introduced in Section 2.6.1, dynamic area cursors expand the effective activation area of widgets beyond their visual bounds, enabling selection without requiring precise overlap from the cursor. This concept has been explored previously in pointing research, shown to substantially improve target acquisition by reducing the effective distance to targets (in effect, reducing the index of difficulty in Fitts' Law terms).

One of the best known examples of a dynamic area cursor is Grossman and Balakrishnan's Bubble Cursor [71], where the cursor's activation area dynamically resizes to always enclose the nearest target, ensuring that exactly one target is selectable at any time. This mechanism effectively reduces the distance-to-target component of Fitts' Law while maintaining selection clarity, yielding significant performance improvements over standard point cursors across a range of target sizes and densities. Hertzum and Hornbæk [79] further investigated area cursors that dynamically change their activation area, comparing the Bubble Cursor with a cell-based alternative. Their findings indicated that while both techniques improved selection over a standard point cursor, the Bubble Cursor was particularly effective when targets were small and sparsely distributed; however, its advantage diminished as target density increased (as there was less free space for the target to expand in order to enclose targets).

These findings suggest that dynamic area cursors are a promising mechanism for improving mid-air targeting, where the instability of unsupported freehand pointing makes precise cursor-to-widget selection particularly difficult. However, previous evaluations of area cursors have focused on conventional pointing devices (e.g., mouse input) and discrete selection tasks (e.g., clicking on circular targets). While Chapter 3 (Section 3.3.1) demonstrated that area-based acquisition is viable in mid-air, it did so with a single widget and only for continuous (slider) manipulation. The extension of dynamic area cursors to multi-widget layouts, where widget proximity-based acquisition must resolve between competing widgets whose boundaries may shift during interaction, and to discrete (button) selection has not been examined. Whether users in such layouts adapt their targeting strategies to exploit area-based acquisition or revert to conventional, precise-targeting habits also remains unclear. Experiments 2 and 3 in this chapter directly investigate these gaps.

4.2 Chapter Structure

Section 4.3 describes the shared experimental design used across both experiments in this chapter. Section 4.4 details the interaction design, including the proxemic cursor and the activation gestures used. Section 4.5 outlines the measures used across both experiments. Section 4.6 describes the implementation and apparatus. Section 4.7 and Section 4.7 discuss Experiments 2 and 3, respectively. Section 4.9 presents an overall discussion of the findings across both experiments. Conclusions are given in Section 4.10, also revisiting the research questions discussed

earlier in the introduction in Section 4.11.

4.3 Chapter Experimental Design

Two within-subjects experiments (Experiments 2 and 3) were used to evaluate the effects of area cursor targeting on mid-air input. Therefore, continuing from Section 3.4, Experiment 2 focused exclusively on slider input tasks to establish how the proxemic cursor performs on continuous interactions across multiple widgets and layouts. Having demonstrated its efficacy for target acquisition (in a task with a single widget type), Experiment 3 then examined whether these benefits extended to interfaces containing multiple continuous (slider) and discrete (button) widgets at varying inter-widget separations, giving further insight into how these interactions perform in practice.

Across both experiments, the primary aim was to compare the impact of area cursor targeting with that of a traditional cursor during mid-air input tasks. Area cursor targeting could enhance task performance by reducing cursor movement time and physical effort during widget acquisition and transition, since the increased cursor area results in a reduced distance to target and means one target is always actively targeted. These experiments also sought to characterise *how* users took advantage of proxemic targeting, investigating effects on the proximity of the cursor to widgets at the moment of activation, how users managed the boundaries between adjacent widgets, and whether they adapted to proxemic targeting or maintained conventional, point-cursor targeting behaviours.

Both experiments continue the comparison of two activation gestures: Push and Pinch (see Figure 4.3). Although Chapter 3 demonstrated that Pinch gestures offered clear performance advantages for target acquisition with slider widgets, area-based targeting might provide more parity between the two gestures by mitigating the target slips associated with Push activation (i.e., the Heisenberg effect of activation; see Section 2.4.3).

4.4 Interaction Design

This section describes the design and implementation of the interaction techniques used in Experiments 2 and 3. The core interaction paradigm was to enable the selection and manipulation (state change) of UI widgets on a 2D display using mid-air hand gestures. Hand translation was mapped to a 2D on-screen cursor at a 1:1 ratio, such that x mm of hand movement produced x mm of corresponding cursor movement, providing continuous visual feedback of pointing location (see Figure 4.1). Cursor position was taken from the centre of the palm rather than a specific fingertip, decoupling finger movement from the pointing mechanism, ensuring the cursor remained stable during gestures that required users to bring their fingers together.

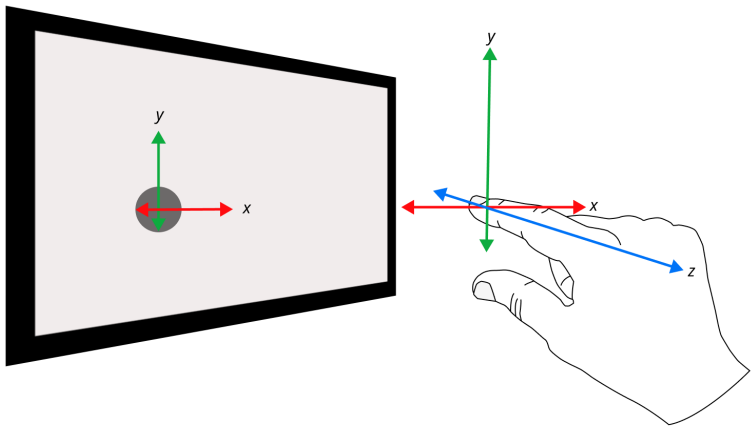


Figure 4.1: Visualisation of the interaction environment, showing the relationship between hand movement in mid-air and cursor movement on-screen.

4.4.1 The Proxemic Cursor

This chapter introduces the proxemic cursor, a variant of a dynamic area cursor that, when enabled, automatically targets the widget nearest to the cursor, allowing activation without requiring cursor overlap with the widget. The proxemic cursor builds directly on findings from Chapter 3: that relaxing the spatial precision required for widget acquisition (as demonstrated by Area Pinch) had substantial performance benefits. The proxemic cursor generalises this to interfaces containing multiple widgets by introducing a *proximity* mechanism that determines which widget the user intends to control. The proxemic cursor is influenced by the BubbleCursor [71], which always designates exactly one widget as the target.

The core mechanism of the proxemic cursor is continuous targeting based on the cursor's proximity to each widget. As the user moves their hand (and thus the on-screen cursor), the nearest widget to the cursor is identified and set as *in-focus*. This widget is visually highlighted, providing feedforward to the user about which element will respond to an selection gesture. When the user performs an activation gesture, the in-focus widget is activated. For discrete widgets such as buttons, activation triggers a press event upon gesture release. For continuous widgets such as sliders, activation initiates a mode switch: the user gains control of the slider handle and can adjust its position through corresponding hand movements, with control released when the gesture ends.

The proxemic cursor was designed such that there was minimal visual affordance on the display. Techniques such as the BubbleCursor visually dominate the interface by dynamically resizing the cursor to encompass the nearest target, effectively communicating the entire area mechanism through the cursor's appearance. While this provides feedforward, it also fundamentally alters the interface's appearance [32]. To isolate the effect of area-based targeting, the proxemic cursor deliberately avoids rendering explicit activation zones or resizing the cursor

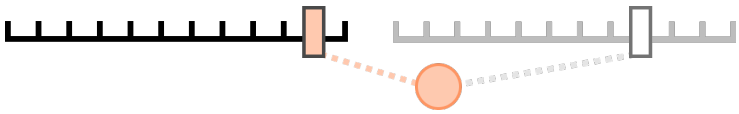


Figure 4.2: The nearest widget is defined as the minimum distance to the widget handle. In this case, the leftmost slider would be targeted.

to reflect proximity boundaries similar to Dynaspot [32]. The only visual change was a subtle highlight on the nearest widget (see Section 4.4.3), indicating which element was currently targeted. This minimal approach was intended to keep the comparison between the area cursor and standard cursor conditions focused on the area-based targeting mechanism rather than on additional visual affordance.

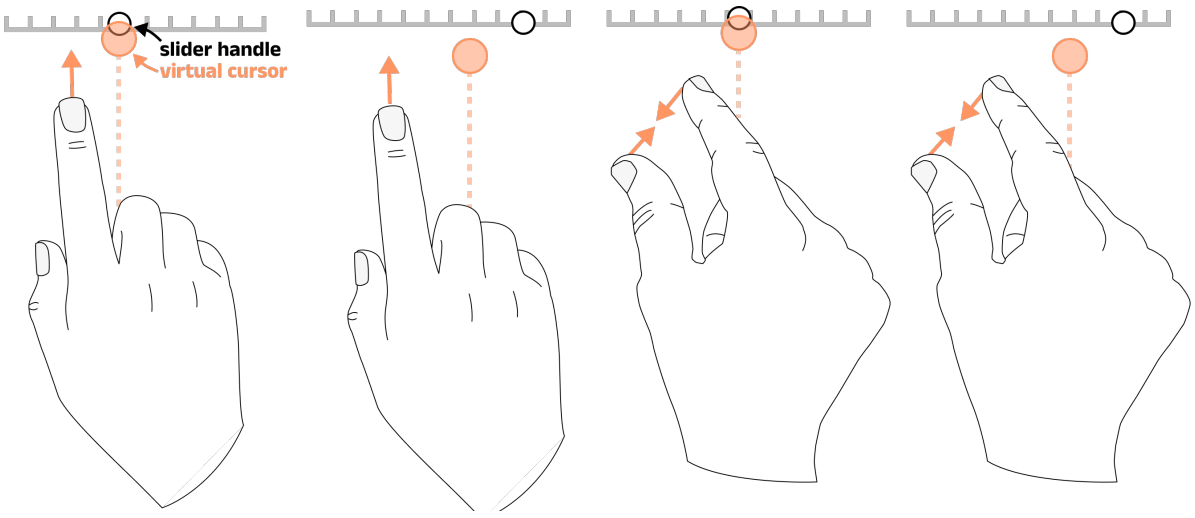
Euclidean distance is computed from the cursor position to the nearest point on each widget's bounding box, and the widget with the smallest such distance is selected as the target (see Figure 4.2). For more complex controls with multiple interactive parts, such as sliders (which consist of a static track and a movable handle), the handle bounding box was used to calculate proximity, rather than the slider track. Although using the entire slider bounding rectangle could be used as an alternative, pilot testing with a small group of interaction designers (n=5) suggested that the handle position would better align with users' perception of how neighbouring sliders should react to cursor movement, especially when handles are at the slider bar ends. Using the full bounding rectangle of each slider bar would keep the boundary between adjacent sliders fixed regardless of handle position, whereas handle-based distancing would allow it to shift dynamically as users manipulated the sliders.

4.4.2 Gesture Design

Two mid-air activation techniques were used to confirm selection: *Push* and *Pinch*, as used in Experiment 1. The Dwell technique from Experiment 1 was not used here due to its incompatibility with area-based targeting, in addition to its poor performance and user sentiment in that first experiment. Push and Pinch were retained for the same reasons as in Experiment 1 (Section 3.3.1): Push is widely supported in commercial touchless systems and Pinch offered a promising alternative due to its adoption within XR input.

Push

The Push gesture (Figure 4.3, (a), (b)) used a similar design to the version used in Experiment 1 (Section 3.3.1): i.e., the user moved their hand towards the display (along z-axis) to activate the targeted widget. The activation threshold was defined as a forward displacement of 100 mm from the hand's resting (moving average) depth position. This activation implementation replaced the percentage-based threshold used in Experiment 1 (where activation occurred at +15% from a continuously updated depth average), which some participants found less pre-



(a) Push (b) Area Push (c) Pinch (d) Area Pinch

Figure 4.3: Illustration of the two selection techniques with and without area targeting. The cursor position was mapped to the centre of the palm. Note that while these images show an extended index finger, this was not required as part of the tracking implementation.

dictable as the baseline shifted during interaction. A fixed displacement of 100 mm, determined through pilot testing, provided a more consistent and predictable activation boundary. When in the activated state, sliders could be manipulated by moving the hand along the slider's axis. Upon returning the hand beyond the 100 mm threshold (i.e., pulling back towards the resting position), the activated state was released. As in Experiment 1, the Push gesture required simultaneous management of lateral cursor position and hand depth.

Pinch

The Pinch gesture (Figure 4.3, (c), (d)) operated on the same principle as in Experiment 1 (Section 3.3.1): the user brought the index finger and thumb together to transition from a pointing state to an activated state, and separated them to release. However, the activation threshold was reduced from $\leq$ 35 mm (as used in Experiment 1) to $\leq$ 20 mm. This adjustment was made due to using an updated Leap Motion dev kit, which provided more precise hand tracking over a wider field of view than the sensor used in Experiment 1. The improved tracking fidelity made a tighter activation threshold both viable, and the system to more accurately infer intent to pinch and reduce the risk of false activations. When in the activated state, sliders could be manipulated by moving the hand along the slider's axis. Upon separating the index finger and thumb (> 20 mm), the activated state was released: for sliders, this ended manipulation; for buttons, this triggered the button press, analogous to a mouse-click-on-release paradigm.



Figure 4.4: Example of the in-focus visual state. Left slider is in-focus

4.4.3 Visual Feedback

Across both selection techniques, users received continuous visual feedback via an on-screen circular cursor following their hand position. The cursor and widget appearance changed to communicate the current interaction states. When the proxemic cursor was **not enabled**, the cursor visual feedback followed the states outlined in Experiment 1 (Section 3.3.2). When enabled, the visual feedback followed the states outlined below.

In-Focus

When the proxemic cursor was enabled, the nearest widget to the cursor was always visually highlighted (black outlined), providing continuous feedforward on which element would respond to an activation gesture (see Figure 4.4). Unlike the hover state, the in-focus state was permanent, where exactly one widget was always designated as the current target, and the highlight transitioned between widgets as the cursor moved through the interaction space. This meant there was no equivalent of the idle state (where no widget was targeted); the cursor always had an in-focus widget, regardless of its position on the screen. The in-focus highlight served as the primary visual cue for distinguishing between targeting by area or precision, allowing users to anticipate which widget would be activated before activating.

Active

Upon crossing the activation threshold, the cursor transitioned to its active state, behaving the same as Experiment 1 (Section 3.3.2). Widgets still turned green to indicate that the selection was registered, which was particularly important because activations could occur outside the widget's bounding box. Once the gesture was released, the cursor returned to its default appearance, and the in-focus highlight resumed on the nearest widget

4.5 Measures

Across both experiments, for each trial, four timing measures were recorded to capture distinct phases of the interaction workflow. **Task time** measured the total duration from the participant's

hand entering the tracking space to the task being confirmed as complete. **Time to acquire** measured the time from the hand entering the tracking space to the participant first gaining control of the first widget. **Time to target** measured the time from gaining control of a slider handle to the handle first reaching the designated target position. **Time to transition** measured the time between releasing control of the first widget and acquiring control of the second. Figure 4.5 illustrates how these measures relate to the key interaction events within each trial.

This decomposition was designed to isolate the effects of cursor and gesture types across different interaction phases. Time to acquire captured the targeting and activation phase, where differences between area and standard cursors were expected to be most pronounced. Time to transition captured the inter-widget movement phase, which was absent from Experiment 1's single-widget design and represented the primary new interaction introduced by multi-widget interfaces. Time to target captured the steering phase, during which users moved the slider handle toward the target after gaining control. The periods not covered by these measures correspond to corrective adjustments made after overshooting or accidentally releasing the handle, as the primary focus was on acquisition, transition, and initial movement dynamics.

Additionally, for sliders, the **Error distance**, the distance between the handle position and the exact intended target position was recorded (in pixels and converted to millimetres) to assess selection accuracy. **Cursor position** was tracked continuously from the start to the end of each widget interaction, deriving the **cursor distance** (the Euclidean distance between the cursor and the widget at the moment the user performed the activation gesture) for the area cursor conditions. This measure characterised the extent to which participants used the relaxed targeting of the proxemic cursor rather than reverting to conventional, precise-targeting behaviour. Additionally, the *Heisenberg displacement* [181] was calculated by taking the cursor displacement from activation onset and completion. This allowed the comparison of push and pinch gestures for unintended cursor movement due to the Heisenberg effect.

Upon completing all conditions, participants ranked the interaction techniques in order of preference (Appendix B). This ranking served as the starting point for a semi-structured interview in which participants discussed their reasoning and reflected on their experience with each technique. All semi-structured interviews were audio recorded with consent and transcribed. Each transcript was read independently by two readers — the author and a member of the supervision team, who separately highlighted passages relevant to the research questions; the highlighted extracts were then compared and, where both readers had identified the same passage, carried forward and grouped into descriptive categories by consensus. No formal thematic analysis was undertaken and no inter-rater reliability statistic was calculated, so the quotes reported here are illustrative participant commentary that contextualises the quantitative results rather than themes generated through a formal coding process.

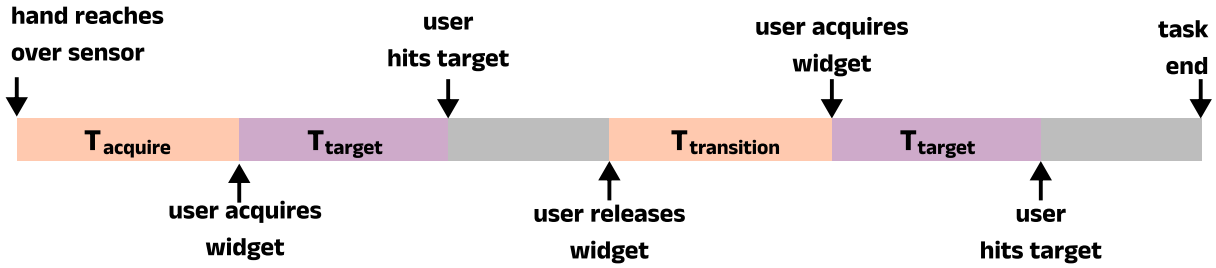


Figure 4.5: This timeline shows how task timing measures are calculated from interaction events where T denotes time.

4.6 Implementation and Apparatus

The experimental software ran full-screen on a 27" display (2560 × 1440 px, 65 × 49.1 cm) with the visual feedback and GUI rendered using Java Processing[1].

An Ultraleap Stereo IR 170 was used to track users' hands and fingers for input. The sensor's field of view is 170°x 170°and provides reliable finger tracking at short distances. Gesture detection and hand tracking were facilitated using the Ultraleap Java SDK[2]. Following Ultraleap's recommendations [108], the sensor was placed 15cm below and 10cm in front of the display, centred underneath it, with a wireless button (surface dial) positioned in front. As in Experiment 1, the sensor was angled so it pointed 15° towards the participant, allowing them to place their hand farther from the sensor and gesture from a more relaxed posture (to improve sensing quality and reduce fatigue [80]).

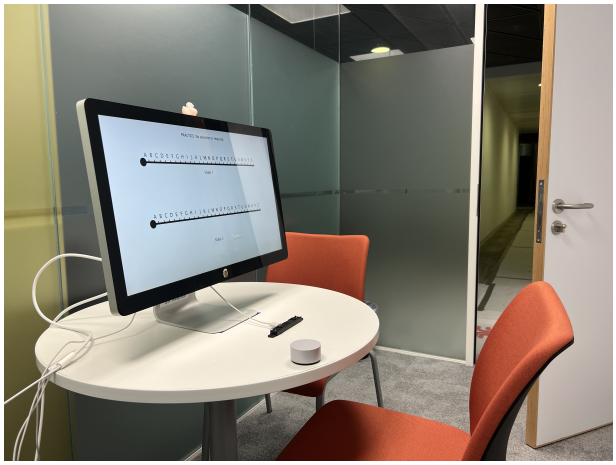


Figure 4.6: Hardware used for both experiment 2 and 3: 27-inch monitor, Leap Motion Controller and wireless button

[1]https://processing.org/ - Last accessed in January 2026

[2]https://www.leapmotiontechnology.com/developer-sub/documentation/java/index - Last Accessed January 2026

4.7 Experiment 2: Multiple Slider Input

4.7.1 Research Aims

Experiment 2 investigates how using area cursor targeting for input affects the usability of mid-air interactions on a touchless display with multiple sliders. The primary aim of this experiment was to evaluate how the performance and preference benefits of area-based targeting, as found in Experiment 1, are affected by multiple slider widgets. By requiring participants to interact with two adjacent sliders in sequence, this experiment introduced the need to acquire a widget, manipulate it, release it, transition to a second widget, and manipulate that in turn. Dynamic area cursors have predominantly been studied with button activation tasks where target boundaries are unaffected by interaction; in contrast, slider manipulation tasks (or similar scrolling/dragging tasks) affect proxemic targeting because the 'nearest' widget depends on the position of the dynamic parts of the widget, affected by previous input operations. This introduces inconsistency which may impact proxemic targeting behaviour.

This multi-widget task is representative of practical touchless interfaces where users routinely move between elements (e.g., adjusting multiple filter parameters on a public kiosk) and the complexities of this environment means deeper insight can be gained into how proxemic targeting is affected by multiple widgets. According to Fitts' law, using an area cursor is expected to reduce widget acquisition and transition times relative to a standard cursor, as users would not need to precisely guide the cursor to each slider handle before activation; however, this experiment investigates how users navigate between widgets and manage widget boundaries, especially when the relative boundaries are dynamic (e.g., as illustrated in Figure 4.2). Such inconsistency could negate the anticipated performance gains from area cursor use.

A secondary aim was to investigate how layout influenced targeting behaviour. The relative position and orientation of widgets on a display determine both the physical movements required to transition between them and the boundaries that separate their activation zones. By varying the slider layout and orientation, this experiment examined whether the benefits of area targeting were consistent across different widget configurations. Finally, this experiment revisited the comparison between Pinch and Push activation gestures in a multi-widget context.

4.7.2 Experiment Design

A within-subjects experimental design was used with three independent variables: Cursor Type (Area, Standard), Gesture Type (Pinch, Push), and 3 Layouts (see below), providing 12 conditions ($2 \times 2 \times 3$). Condition order was counterbalanced using a Latin square design. Each participant completed 12 trials per condition, with a balanced mix of starting sliders (i.e., half the trials began with Slider 1 and half with Slider 2), resulting in 144 trials per participant.

4.7.3 Task Design

Each trial required participants to make two sequential slider selections using a pair of adjacent sliders. Participants were asked to make alphabetical selections from A-Z, where each slider handle corresponded to a designated target. This task represents a realistic touchless interaction scenario, as alphabetical filtering is commonly found on public-facing interfaces such as ticket machines and retail kiosks. The use of two sequential selections was the minimum necessary to introduce a widget transition, allowing measurement of both initial widget acquisition and the transition between widgets.

All tasks began with both slider handles at 'A'. Target characters were selected beyond the first three positions (i.e., beyond A–C) to ensure meaningful traversal distances. The order in which sliders were used was alternated across trials (Slider 1 then Slider 2, or vice versa) to observe any asymmetries in participant behaviour when crossing widget boundaries in different directions. The position and orientation of the sliders were varied on-screen using three slider layouts (see below), providing variety in slider orientation (horizontal vs vertical) and transition direction (lateral vs vertical), and capturing a range of spatial configurations representative of practical interface designs.

Horizontal Side-by-Side (HSS)

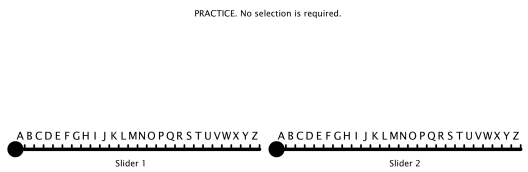
Two horizontal sliders were placed adjacent along the horizontal axis (Figure 4.7 (a)). This layout created a narrow vertical boundary between the two widgets and placed the highest demand on widget boundary management, as slider handles could be in close horizontal proximity depending on their current positions.

Horizontal Top-and-Bottom (HTB)

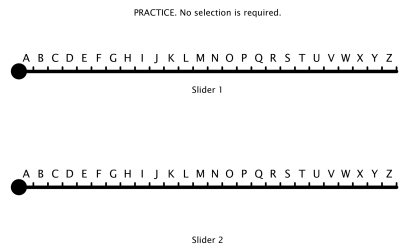
Two horizontal sliders stacked vertically (Figure 4.7 (b)). Transitioning between sliders required vertical cursor movement. This tested a wider horizontal slider with larger individual target widths compared to HSS.

Vertical Side-by-Side (VSS)

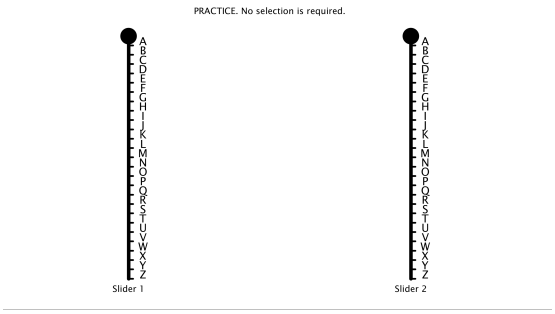
Two vertical sliders placed adjacent along the horizontal axis (Figure 4.7 (c)). This introduced vertical slider orientation, requiring vertical hand movement for manipulation, which is typically less comfortable during extended mid-air interaction due to the increased muscle tension from raising the arm and stabilising the hand at a greater height [80]. A vertical top-and-bottom layout was excluded as the landscape screen orientation would have rendered target widths impractically small.



(a) Horizontal side-by-side (HSS)



(b) Horizontal top-and-bottom (HTB)



(c) Vertical side-by-side (VSS)

Figure 4.7: Screenshots of the three slider layouts. Each slider has 26 regions, labelled from A to Z.

4.7.4 Procedure

Before beginning the trials, participants provided informed consent and were introduced to each gesture and the proxemic cursor through practice tasks. Participants completed a minimum of 12 practice targets per technique and task type, though they were free to continue practising beyond this minimum until they felt comfortable. Each trial began with the participant's non dominant hand resting on the wireless button; on-screen instructions indicated the required action (e.g., *"Select F on Slider 1 then Y or Slider 2"*), and the participant pressed the button to begin the trial, instructed to be make their selections *"as quick and as accurate as possible"*. There were no time constraints, and participants were permitted to release the slider and make corrections at any point rather than being limited to a single continuous movement. The trial stopped once the button was pressed again. After all trials, preference rankings were taken, and reasoning was discussed.

4.7.5 Results

21 participants were recruited (14 male, 6 female, 1 other; mean age 29 years, SD 8.19 years; 18 right-handed, 2 left-handed, 1 ambidextrous) through university mailing lists and poster advertising. Participants were compensated with £10 for taking part in the study, which lasted 1 hour. Ethics approval was provided by the College of Science and Engineering Ethics Committee.

Analysis was conducted using repeated-measures ANOVA, followed by post-hoc t-test com-

parisons of estimated marginal means. If data did not have a normal distribution (via the Shapiro-Wilk test), then the Aligned-Rank Transform [180] was applied first, followed by post-hoc comparisons using ART-C [49] where appropriate.

Below are summary statistics and visualisations for each dependent variable. Table 4.1 shows the results and analyses across all conditions, including summary statistics for each dependent variable, and results from the repeated-measures ANOVA and post-hoc comparisons.

Task Time

Overall, participants took an average of 13050 ms (SD = 2953 ms) per trial (Table 4.1 rows 1-7 and Figure 4.8). Across Gesture conditions, Pinch had the faster mean completion time (M = 12326, SD = 3092) ms compared to Push (M = 13762, SD = 2634) ms. Trials performed with the Area Cursor (M = 12549, SD = 2867) ms were completed faster than those with the Standard Cursor (M = 13543, SD = 2965) ms. Among Layout conditions, HSS was the fastest (M = 12497, SD = 2677) ms, followed by VSS (M = 13229, SD = 3011) ms and HTB (M = 13422, SD = 3108) ms.

For the Gesture $\times$ Area interaction, the difference between cursor types was more pronounced under Push, where the Area Cursor (M = 12921, SD = 2291) ms was substantially faster than the Standard Cursor (M = 14603, SD = 2702) ms, whereas under Pinch the two cursor types had similar times (Area Cursor: M = 12164, SD = 3336 ms; Standard Cursor: M = 12483, SD = 2853 ms). A similar pattern was observed in the Gesture $\times$ Layout interaction: Push was slowest in the HTB layout (M = 14397, SD = 2647) ms, while Pinch remained comparatively stable across layouts, with HSS the fastest (M = 11801, SD = 2645) ms. For the Area $\times$ Layout interaction, the Standard Cursor was notably slower than the Area Cursor in both the VSS (M = 14159 vs. 12299 ms) and HTB (M = 13945 vs. 12885 ms) layouts, whereas in the HSS layout the two cursor types performed similarly (M = 12525 vs. 12469 ms).

In the three-way Gesture $\times$ Area $\times$ Layout interaction, the slowest condition was Push with the Standard Cursor in the HTB layout (M = 15470, SD = 2634) ms, while the fastest was Pinch with the Standard Cursor in the HSS layout (M = 11489, SD = 2100) ms.

Time to Acquire Slider 1

Overall, participants took an average of 3366 ms (SD = 1281 ms) to acquire the first slider (See Table 4.1 rows 8-14 and Figure 4.9). Across Gesture conditions, Pinch was faster (M = 3177, SD = 1207) ms than Push (M = 3553, SD = 1329) ms. The Area Cursor (M = 3021, SD = 1221) ms showed notably shorter acquisition times than the Standard Cursor (M = 3706, SD = 1253) ms. Among Layout conditions, HSS was the fastest (M = 3171, SD = 1228) ms, followed by VSS (M = 3343, SD = 1195) ms and HTB (M = 3585, SD = 1394) ms.

For the Gesture $\times$ Area interaction, both gesture types benefited from the Area Cursor, though the advantage was more pronounced under Push: the Area Cursor (M = 3065, SD =

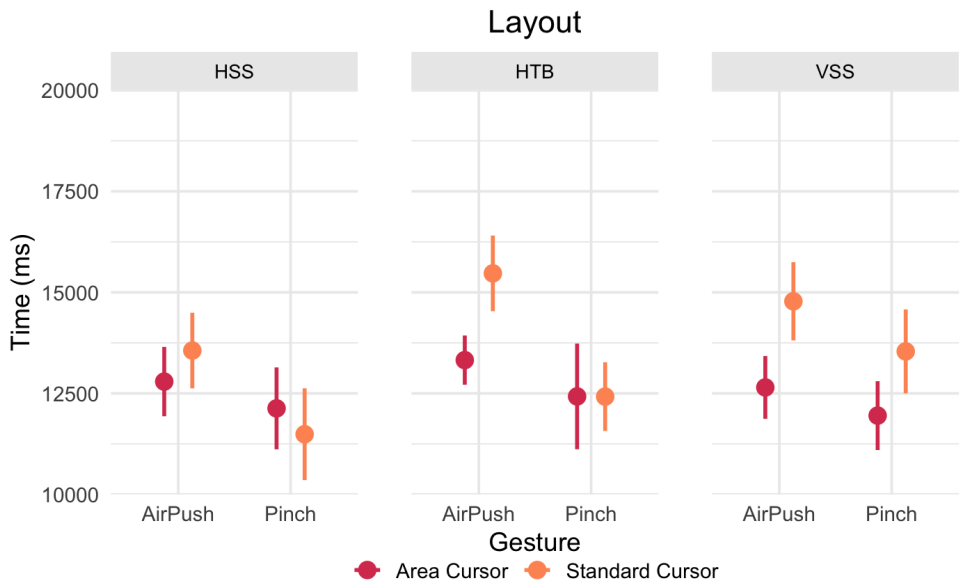


Figure 4.8: Mean task time per condition. Error bars show 95% CIs.

1081) ms was substantially faster than the Standard Cursor (M = 4040, SD = 1382) ms, a difference of nearly 1000 ms. Under Pinch, the gap was smaller (Area Cursor: M = 2975, SD = 1357 ms; Standard Cursor: M = 3372, SD = 1014 ms). In the Gesture × Layout interaction, Push was slowest in the HTB layout (M = 3802, SD = 1427) ms, while Pinch showed a similar pattern but with consistently lower means, being fastest in HSS (M = 2884, SD = 1135) ms. For the Area × Layout interaction, the Area Cursor was faster than the Standard Cursor across all layouts, with the largest difference in VSS (M = 2865 vs. 3821 ms) and the smallest in HTB (M = 3293 vs. 3870 ms).

In the three-way Gesture × Area × Layout interaction, the slowest condition was Push with the Standard Cursor in the HTB layout (M = 4266, SD = 1411) ms, while the fastest was Pinch with the Area Cursor in the HSS layout (M = 2762, SD = 1416) ms.

Time to Transition

Overall, participants took an average of 3894 ms (SD = 1679 ms) to transition between sliders (See Table 4.1 rows 15-21, and Figure 4.10. Across Gesture conditions, Pinch was faster (M = 3539, SD = 1577) ms than Push (M = 4244, SD = 1710) ms. The Area Cursor (M = 3381, SD = 1531) ms showed notably shorter transition times than the Standard Cursor (M = 4400, SD = 1671) ms. Among Layout conditions, HSS was the fastest (M = 3468, SD = 1437) ms, followed by VSS (M = 4053, SD = 1733) ms and HTB (M = 4160, SD = 1783) ms.

For the Gesture × Area interaction, both gesture types benefited from the Area Cursor, though the advantage was more pronounced under Push: the Area Cursor (M = 3579, SD = 1481) ms was substantially faster than the Standard Cursor (M = 4909, SD = 1674) ms, a difference of over 1300 ms. Under Pinch, the gap was smaller (Area Cursor: M = 3176, SD = 1567

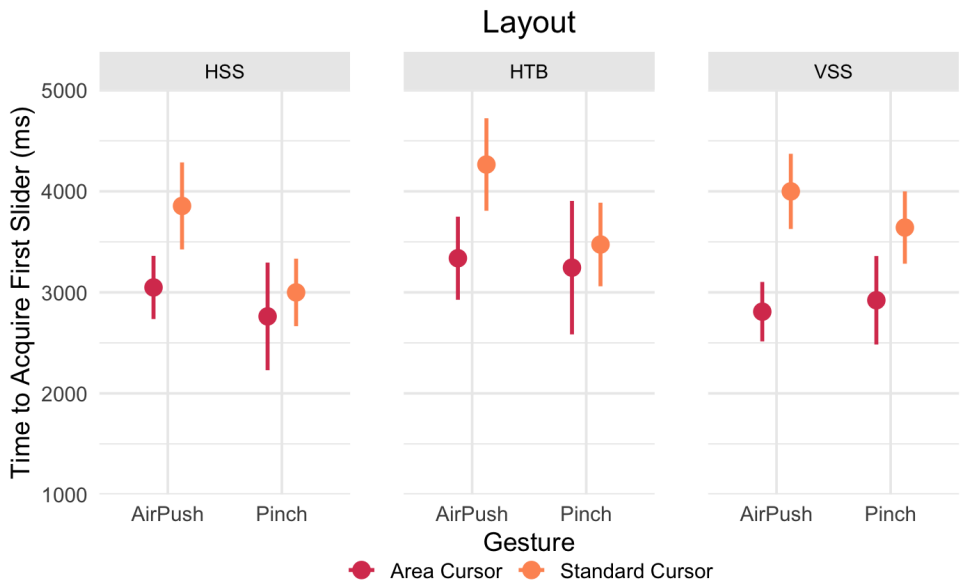


Figure 4.9: Mean time to acquire the first slider per condition. Error bars show 95% CIs.

ms; Standard Cursor: M = 3891, SD = 1517 ms). In the Gesture × Layout interaction, Push was slowest in the HTB layout (M = 4666, SD = 2004) ms, while Pinch showed a similar pattern but with consistently lower means, being fastest in HSS (M = 3135, SD = 1514) ms.

For the Area × Layout interaction, the Area Cursor was faster than the Standard Cursor across all layouts, with the largest difference in VSS (M = 3323 vs. 4784 ms) and the smallest in HSS (M = 3221 vs. 3709 ms).

In the three-way Gesture × Area × Layout interaction, the slowest condition was Push with the Standard Cursor in the HTB layout (M = 5444, SD = 1860) ms, while the fastest was Pinch with the Area Cursor in the HSS layout (M = 3025, SD = 1827) ms.

Time to Target

Overall, participants took an average of 1469 ms (SD = 615 ms) to reach the target value on each slider (See Table 4.1 rows 22-28, and Figure 4.11). Across Gesture conditions, Push (M = 1447, SD = 536) ms and Pinch (M = 1490, SD = 687) ms performed similarly. Likewise, the Area Cursor (M = 1429, SD = 630) ms and Standard Cursor (M = 1508, SD = 600) ms showed only a modest difference. Among Layout conditions, however, there was a more notable spread: HSS was the fastest (M = 1214, SD = 412) ms, followed by HTB (M = 1281, SD = 510) ms, while VSS was substantially slower (M = 1905, SD = 646) ms.

For the Gesture × Area interaction, the difference between cursor types was minimal under both Push (Area Cursor: M = 1390, SD = 534 ms; Standard Cursor: M = 1504, SD = 537 ms) and Pinch (Area Cursor: M = 1469, SD = 718 ms; Standard Cursor: M = 1511, SD = 661 ms). In the Gesture × Layout interaction, both gesture types followed the same pattern, with VSS the slowest (Push: M = 1935, SD = 583 ms; Pinch: M = 1876, SD = 709 ms) and HSS the fastest

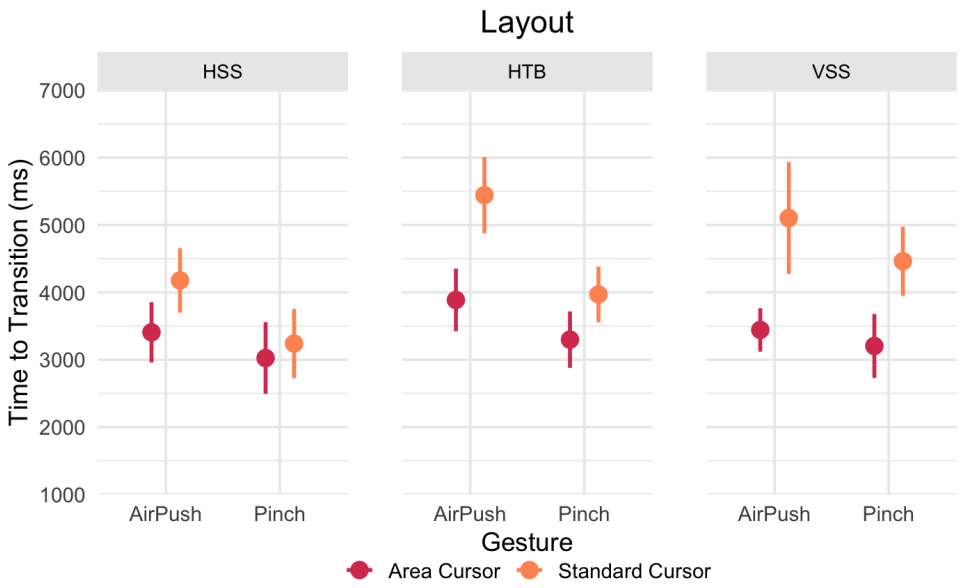


Figure 4.10: Mean time to transition between sliders per condition. Error bars show 95% CIs.

(Push: M = 1168, SD = 264 ms; Pinch: M = 1260, SD = 521 ms). For the Area × Layout interaction, the two cursor types performed similarly across all layouts, with both showing the largest times in VSS (Area Cursor: M = 1847 vs. Standard Cursor: M = 1964 ms) and the smallest in HSS (M = 1168 vs. 1258 ms).

In the three-way Gesture × Area × Layout interaction, the slowest condition was Push with the Standard Cursor in the VSS layout (M = 1993, SD = 559) ms, while the fastest was Push with the Area Cursor in the HSS layout (M = 1146, SD = 272) ms.

Error Distance

Overall, participants achieved a mean error distance of 0.11 mm (SD = 0.95 mm) per trial (See Table 4.1 rows 29-35, and Figure 4.12), indicating a slight tendency to overshoot the target. Across Gesture conditions, Push produced a small undershoot (M = −0.07, SD = 0.92) mm, while Pinch tended to overshoot (M = 0.29, SD = 0.95) mm. The Area Cursor (M = 0.03, SD = 0.97) mm had near-zero error compared to the Standard Cursor (M = 0.19, SD = 0.92) mm. For Layout conditions, VSS was the only layout to produce an undershoot (M = −0.13, SD = 1.10) mm, while HSS showed the largest overshoot (M = 0.31, SD = 0.75) mm and HTB fell between the two (M = 0.15, SD = 0.92) mm.

For the Gesture × Area interaction, Push with the Area Cursor was the only combination to undershoot the target (M = -0.17, SD = 0.81) mm, whereas Pinch with the Standard Cursor produced the largest overshoot (M = 0.36, SD = 0.80) mm. Under Pinch, both cursor types overshot the target (Area Cursor: M = 0.23, SD = 1.09 mm; Standard Cursor: M = 0.36, SD = 0.80 mm). In the Gesture × Layout interaction, the most notable result was Push in the VSS layout, which produced a clear undershoot (M = −0.49, SD = 1.17) mm, while Pinch

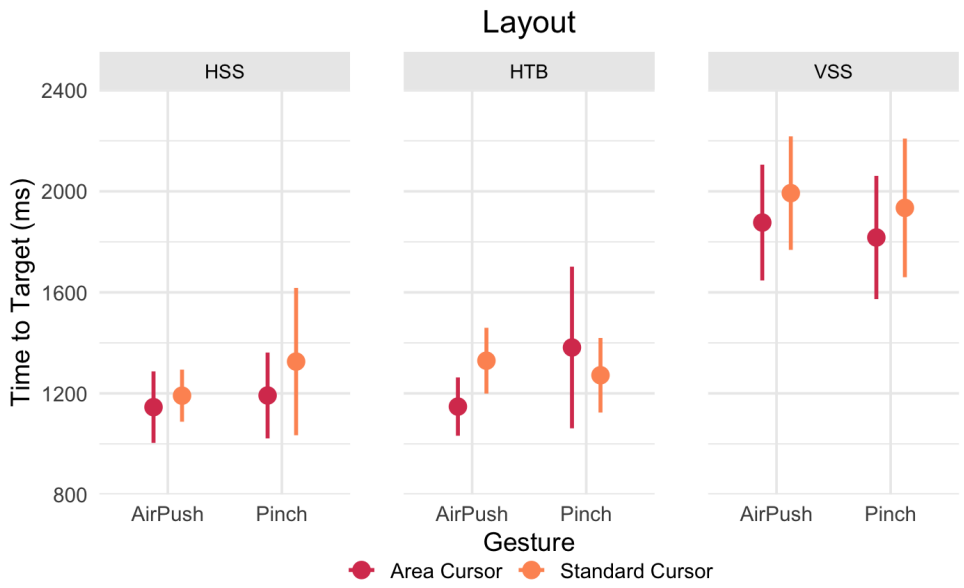


Figure 4.11: Mean time to arrive at the target per condition. Error bars show 95% CIs.

consistently overshot across all layouts, with HSS the highest (M = 0.39, SD = 0.88) mm. For the Area × Layout interaction, both cursor types performed similarly in HSS (Area Cursor: M = 0.30 vs. Standard Cursor: M = 0.32 mm), while in VSS the Area Cursor undershot the target (M = −0.27, SD = 1.04) mm and the Standard Cursor was close to zero (M = 0.02, SD = 1.16) mm.

In the three-way Gesture × Area × Layout interaction, the largest undershoot was observed for Push with the Area Cursor in the VSS layout (M = −0.66, SD = 0.92) mm, while the largest overshoot occurred under Pinch with the Standard Cursor in the VSS layout (M = 0.37, SD = 0.76) mm and Pinch with both cursor types in the HSS layout (both M = 0.39) mm.

Task Success

Participants achieved a high overall success rate of 97.91% (SD = 5.61%) across all conditions (See Table 4.1 rows 36-42, and Figure 4.13). Pinch had a higher success rate (M = 98.79, SD = 3.57)% than Push (M = 97.04, SD = 6.98)%, which also exhibited greater variability. The Area Cursor (M = 98.27, SD = 3.99)% outperformed the Standard Cursor (M = 97.55, SD = 6.84)%. Notably, the HTB layout achieved a perfect success rate of 100% with no variance, while HSS (M = 96.91, SD = 6.18)% and VSS (M = 96.83, SD = 7.09)% were slightly lower.

For the Gesture × Area interaction, Push with the Standard Cursor recorded the lowest success rate (M = 96.46, SD = 8.64)% alongside the highest variability, whereas Pinch with the Area Cursor was the most reliable combination (M = 98.95, SD = 2.85)%. In the Gesture × Layout interaction, both gesture types achieved perfect success in HTB. The lowest success rates were observed under Push in the VSS (M = 95.31, SD = 9.06)% and HSS (M = 95.81, SD = 7.26)% layouts, while Pinch remained consistently above 98% across all layouts. The

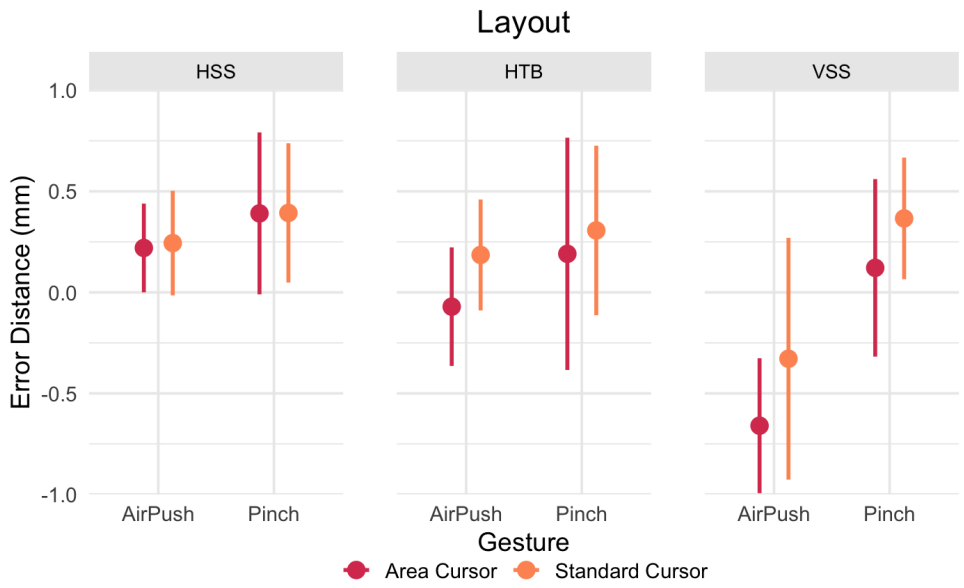


Figure 4.12: Mean error distance for both sliders per condition. Error bars show 95% CIs.

Area × Layout interaction revealed that both cursor types reached 100% in HTB; in HSS and VSS, the Area Cursor maintained slightly higher success rates than the Standard Cursor, with the Standard Cursor in HSS dropping to (M = 96.18, SD = 7.45)%.

For the three-way interaction, Gesture × Area × Layout, all conditions within the HTB layout achieved perfect success. Outside HTB, the lowest success rate was recorded for Push with the Standard Cursor in the VSS layout (M = 94.43, SD = 11.55)%, which also exhibited the greatest variability of any condition. Conversely, Pinch with the Area Cursor in the HSS layout (M = 98.72, SD = 3.23)% and Pinch with the Standard Cursor in the VSS layout (M = 98.53, SD = 4.18)% were among the highest-performing non-HTB conditions.

Activation Distance

When using the Area Cursor, participants activated sliders from an average distance of 26.90 mm (SD = 30.70 mm) from the widget centre (See Table 4.1 rows 43-45, and Figures 4.14), 4.15. The two gesture types produced similar activation distances, with AirPush (M = 27.75, SD = 32.68) mm marginally higher than Pinch (M = 26.04, SD = 28.66) mm. Layout had a more discernible effect: participants activated sliders from the closest distance in the HSS layout (M = 21.68, SD = 26.28) mm, at a moderate distance in HTB (M = 25.26, SD = 28.29) mm, and from the furthest distance in VSS (M = 33.69, SD = 35.76) mm.

For the Gesture × Layout interaction, both gesture types followed this overall pattern. In the VSS layout, AirPush exhibited the largest activation distance of any condition (M = 36.12, SD = 38.94) mm, while Pinch in the HSS layout recorded the smallest (M = 20.56, SD = 22.80) mm. In HTB, Pinch (M = 26.17, SD = 29.32) mm was slightly higher than AirPush (M = 24.37, SD = 27.57) mm, reversing the trend observed in the other two layouts. Across all conditions, the



Figure 4.13: Mean success rate for both sliders per condition. Error bars show 95% CIs.

relatively large standard deviations suggest considerable variability in the distance from which participants chose to engage with sliders.

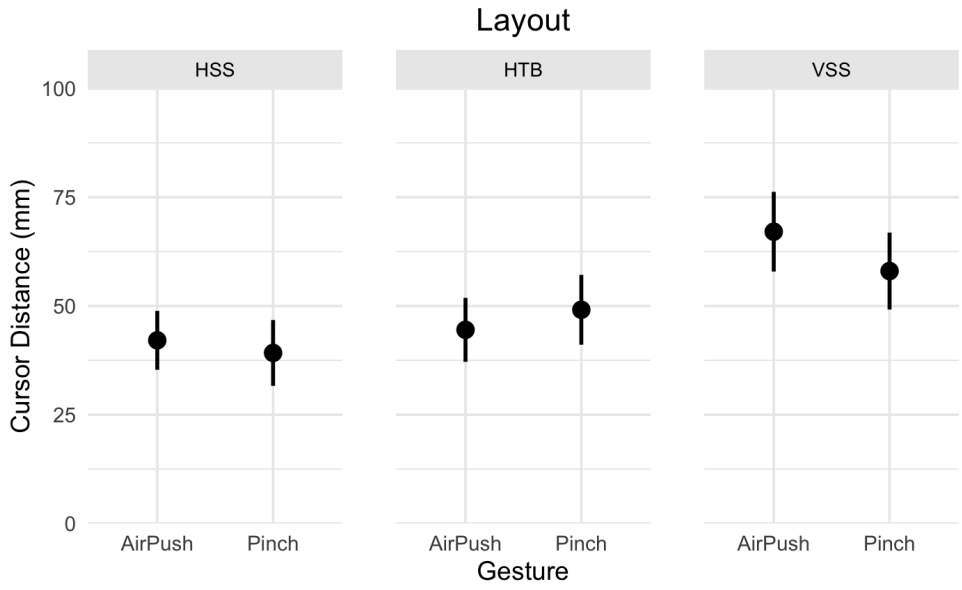


Figure 4.14: Mean activation distance both sliders per condition. Error bars show 95% CIs.

Heisenberg Displacement

Across all activations, participants displaced the cursor by an average of 15.65 mm (SD = 15.48 mm) between activation onset and completion (See Table 4.1 rows 46–48, and Figures 4.16). The two gesture types produced different displacements, with AirPush (M = 24.31, SD = 17.73) mm more than three times higher than Pinch (M = 6.98, SD = 3.89) mm. Layout had a far

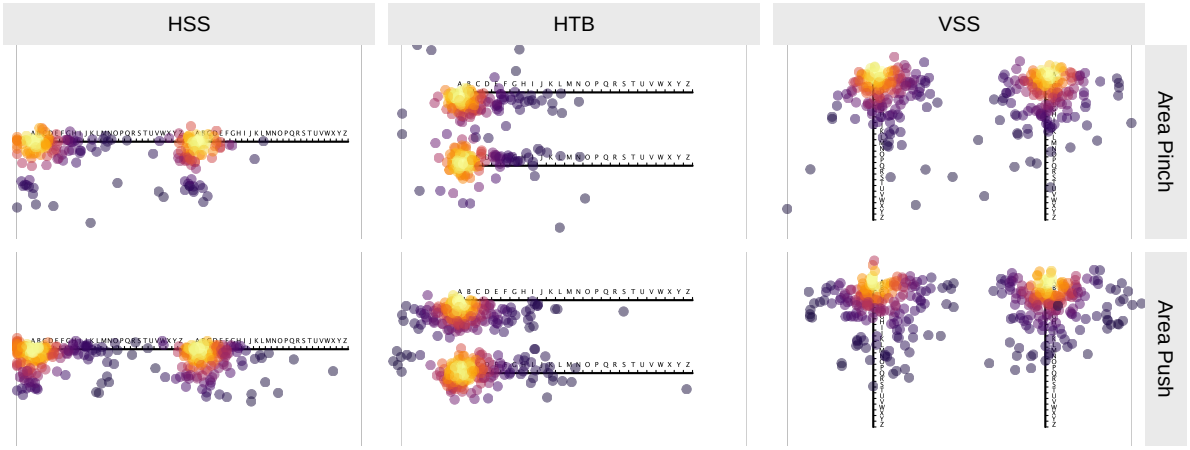


Figure 4.15: Area cursor positions when the participant took control of the slider handles with brighter colours indicating greater density of cursor positions.

more modest effect: displacement was smallest in the HSS layout (M = 14.33, SD = 14.63) mm, intermediate in HTB (M = 15.67, SD = 15.41) mm, and largest in VSS (M = 16.94, SD = 16.42) mm.

For the Gesture × Layout interaction, the two gesture types did not follow a common pattern. AirPush displacement increased across layouts, from HSS (M = 21.51, SD = 17.41) mm through HTB (M = 24.25, SD = 17.82) mm to VSS, which produced the largest displacement of any condition (M = 27.18, SD = 17.92) mm. Pinch, by contrast, remained stable and close to its overall mean in every layout, from HSS (M = 7.16, SD = 4.97) mm through HTB (M = 7.09, SD = 3.45) mm to VSS, which recorded the smallest displacement of any condition (M = 6.69, SD = 3.05) mm. The standard deviations mirror this: AirPush displacement varied considerably between activations (SDs of 17–18 mm, comparable to its own mean), whereas Pinch was tightly clustered (SDs of 3–5 mm), indicating that the Heisenberg effect under Pinch was both small and consistent, while under AirPush it was large and highly variable.

Preference Ranking

Figure 4.17 shows the distribution of preference ranks for the techniques. Area Pinch was the most preferred technique with 14 people ranking it as their most preferred. Friedman's test found a significant effect of interaction technique on rank: $\chi^2 = 37.97$, $p < .001$. Post hoc Nemenyi comparisons found that Push was significantly less preferred than all others (all $p \leq .021$), and Area Pinch was significantly more preferred than standard Pinch ($p = .007$); no other comparisons were significantly different ($p \geq .079$).

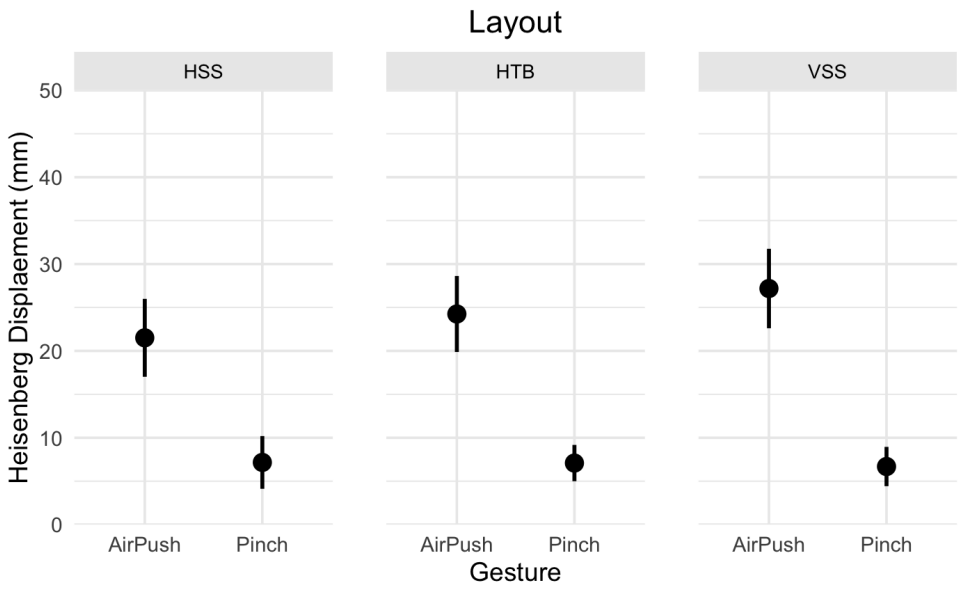


Figure 4.16: Mean cursor displacement caused by the Heisenberg effect. Error bars show 95% CIs.

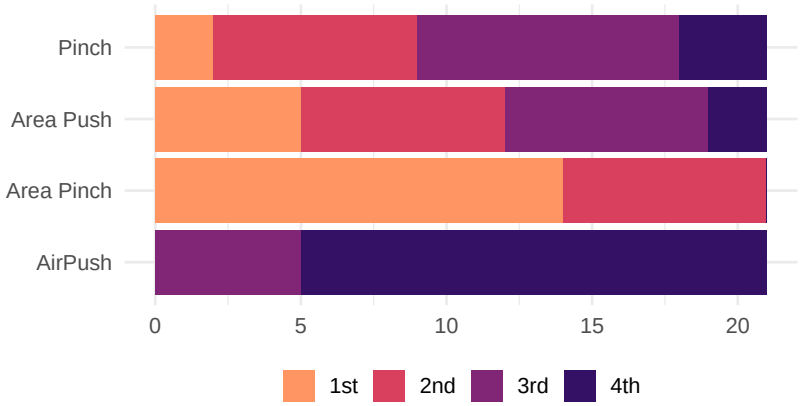


Figure 4.17: Preference ranking distribution for each interaction technique.

Table 4.1: ANOVA results, and post-hoc test results for Experiment 2. Significant comparisons are highlighted*

	Time	**Mean 13049 ms, SD 2953 ms, 95% CI [12682, 13418] ms**			
	ANOVA	*Result*	p	η_p^2	*Post-Hoc Comparisons*
1	*Gesture*	F(1, 218) = 39.09	< .001*	.15	• Push > Pinch ($p \leq .001$)
2	*Area*	F(1, 218) = 17.32	< .001*	.07	• Standard > Area ($p \leq .001$)
3	*Layout*	F(2, 218) = 4.54	.01*	.04	• HTB > HSS ($p = .01$)
4	*Gesture* × *Area*	F(1, 218) = 3.84	.05	.02	

Continued on next page

Table 4.1: (continued)

	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
5	*Gesture* × *Layout*	F(2, 218) = 2.31	.10	.02	
6	*Area* × *Layout*	F(2, 218) = 3.52	.03*	.03	• No sig comparisons
7	*Gesture* × *Area* × *Layout*	F(2, 218) = 0.46	.63	.00	
	Time to Acquire Slider 1	**Mean 3366 ms, SD 1281 ms, 95% CI [3207, 3526] ms**			
8	*Gesture*	F(1, 218) = 13.29	< .001*	.06	• Push > Pinch ($p \leq .001$)
9	*Area*	F(1, 218) = 54.63	< .001*	.20	• Standard > Area ($p \leq .001$)
10	*Layout*	F(2, 218) = 2.83	.06	.03	
11	*Gesture* × *Area*	F(1, 218) = 1.97	.16	.01	
12	*Gesture* × *Layout*	F(2, 218) = 1.87	.16	.02	
13	*Area* × *Layout*	F(2, 218) = 1.18	.31	.01	
14	*Gesture* × *Area* × *Layout*	F(2, 218) = 0.02	.98	.00	
	Time to Transition	**Mean 3894 ms, SD 1679 ms, 95% CI [3685, 4103] ms**			
15	*Gesture*	F(1, 218) = 44.11	< .001*	.17	• Push > Pinch ($p \leq .001$)
16	*Area*	F(1, 218) = 94.07	< .001*	.30	• Standard > Area ($p \leq .001$)
17	*Layout*	F(2, 218) = 13.18	< .001*	.11	• HTB > HSS($p \leq .001$) • VSS > HSS($p \leq .001$)
18	*Gesture* × *Area*	F(1, 218) = 4.46	.04*	.02	• No sig comparisons
19	*Gesture* × *Layout*	F(2, 218) = 1.24	.29	.01	
20	*Area* × *Layout*	F(2, 218) = 4.97	.01*	.04	• No sig comparisons
21	*Gesture* × *Area* × *Layout*	F(2, 218) = 0.60	.55	.01	
	Time to Target	**Mean 1467 ms, SD 615 ms, 95% CI [1392, 1545] ms**			
22	*Gesture*	F(1, 218) = 1.34	.25	.01	
23	*Area*	F(1, 218) = 5.68	.02*	.03	• Standard > Area ($p = .02$)
24	*Layout*	F(2, 218) = 71.43	< .001*	.40	• VSS > HSS ($p \leq .001$) • VSS > HTB ($p \leq .001$)
25	*Gesture* × *Area*	F(1, 218) = 0.05	.82	.00	
26	*Gesture* × *Layout*	F(2, 218) = 1.10	.34	.01	
27	*Area* × *Layout*	F(2, 218) = 0.05	.95	.00	
28	*Gesture* × *Area* × *Layout*	F(2, 218) = 0.56	.57	.01	

Continued on next page

Table 4.1: (continued)

	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
	Error Distance	**Mean 0.11 mm, SD 0.95 mm, 95% CI [-0.01, 0.23] mm**			
29	*Gesture*	F(1, 218) = 14.59	< .001*	.06	• Push > Pinch ($p \leq .001$)
30	*Area*	F(1, 218) = 1.95	.16	.01	
31	*Layout*	F(2, 218) = 8.30	< .001*	.07	• HSS > VSS ($p \leq .001$) • HTB > VSS ($p = .04$)
32	*Gesture* × *Area*	F(1, 218) = 0.03	.87	.00	
33	*Gesture* × *Layout*	F(2, 218) = 5.12	.01*	.04	• No sig comparisons
34	*Area* × *Layout*	F(2, 218) = 0.71	.49	.01	
35	*Gesture* × *Area* × *Layout*	F(2, 218) = 0.02	.98	.00	
	Task Success	**Mean 98 %, SD 6 %, 95% CI [97, 99] %**			
36	*Gesture*	F(1, 218) = 5.36	.02*	.02	Push < Pinch ($p = .02$)
37	*Area*	F(1, 218) = 0.31	.58	.00	
38	*Layout*	F(2, 218) = 22.35	< .001*	.17	• HTB > HSS ($p \leq .001$) • HTB > VSS ($p \leq .001$)
39	*Gesture* × *Area*	F(1, 218) = 4.40	.04*	.02	• Area Pinch > Standard Pinch ($p = .04$)
40	*Gesture* × *Layout*	F(2, 218) = 2.67	.07	.02	
41	*Area* × *Layout*	F(2, 218) = 1.89	.15	.02	
42	*Gesture* × *Area* × *Layout*	F(2, 218) = 3.64	.03*	.03	• HSS > HTB for Area Push ($p = .004$) • HTB > Standard push vs. Area Push ($p = .03$) • VSS > HTB for Area Push ($p \leq .001$) • VSS > HTB for standard Pinch ($p = .003$)
	Cursor Distance	**Mean 27 mm, SD 31 mm, 95% CI [23, 31] mm**			
43	*Gesture*	F(1, 225) = 0.08	.78	.00	
44	*Layout*	F(2, 224) = 3.95	.02*	.03	• VSS > HSS ($p = .02$)
45	*Gesture* × *Layout*	F(2, 224) = 0.13	.87	.00	

Continued on next page

Table 4.1: (continued)

	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
	Heisenberg Displacement	**Mean 16 mm, SD 15 mm, 95% CI [14, 18] mm**			
46	*gesture*	F(1, 226) = 205.17	< .001*	.48	• Push > Pinch ($p \leq .001$)
47	*layout*	F(2, 226) = 4.11	.02*	.04	• VSS > HSS ($p = .01$)
48	*gesture* × *layout*	F(2, 226) = 3.90	.02*	.03	• No significant comparisons

4.8 Experiment 3: Discrete and Continuous Interactions

4.8.1 Research Aims

The aim of Experiment 3 was to investigate how users use area cursors with different widget types (continuous input via sliders, discrete input via buttons), and in particular how they manage transitions between these fundamentally different interaction mechanics, shifting between continuous control and discrete activation within a larger, more complex interface. Experiment 2 suggested that area-based targeting significantly improved widget acquisition and transition times for slider interactions and was preferred over the standard cursor. Experiment 3 extended this to more densely clustered interface layouts featuring a broader range of interface tasks, combining both continuous and discrete input widgets. All conditions in this experiment used the area cursor.

A secondary aim was to investigate how inter-widget spacing affects area cursor targeting by systematically varying the horizontal and vertical distances between widgets to test whether the area cursor maintains its advantages as activation zone boundaries narrow. Finally, the comparison between Pinch and Push activation gestures was maintained to examine whether Push's performance deficit, consistently observed for continuous slider control, persisted for discrete button selections where a momentary forward motion more closely mirrors the action of pressing a physical surface.

4.8.2 Experiment Design

A within-subjects experimental design was used with three independent variables: Gesture Type (Pinch, Push), Vertical separation (64px, 128px, 256px), and Horizontal separation (64px, 128px, 256px), giving 18 conditions. Tasks were presented in blocks for each condition, in a counter-balanced order using a Latin square design. Each participant completed 12 trials per condition, for a total of 216 per participant.

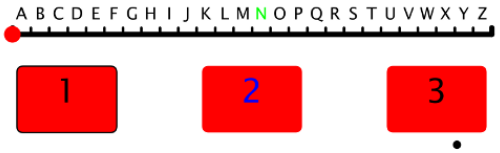


Figure 4.18: Experiment 3 task screenshot. There are 26 slider notches (A–Z) and three buttons (1–3). Shows a vertical and horizontal separation of 64px and 256px, respectively.

4.8.3 Task Design

Each trial required participants to make two sequential selections: one using a slider and one using a button. An alphabetical slider (A–Z) was positioned above a row of three numerically labelled buttons (1–3), as shown in Figure 4.18. This layout combined continuous and discrete widget types within a single interface, representing a common configuration in touchless systems where users might, for example, filter items by starting letter using a slider and then confirm a category using a button. The order of interaction was alternated across trials (slider then button, or button then slider) to observe any asymmetries in targeting behaviour when transitioning between different widget types in different directions.

All tasks began with the slider handle at ‘A’ for consistency. Slider targets were selected beyond the first three positions (i.e., beyond A–C) to ensure meaningful traversal distances. Button targets were distributed across all three buttons. The combination of a single movable slider and three discrete buttons introduced narrower effective target widths than Experiment 2, where each slider occupied a substantial portion of the screen, and required both horizontal targeting (to distinguish between the three adjacent buttons) and vertical targeting (to transition between the slider and button row). The three buttons were red rounded rectangles of approximately 76 $\times$ 51 mm (300 $\times$ 200 px), labelled 1–3, sized generously relative to the 64 px minimum separation recommended for touchless targets [72]; the slider used the same styling as in Experiments 1 and 2.

Horizontal and vertical widget separation were varied independently, each at three levels: 64, 128, and 256 pixels. Horizontal separation controlled the spacing between the three buttons, while vertical separation controlled the distance between the slider bar and the top edge of the button row. These values were chosen based on prior findings that touchless target separations below 64 pixels substantially increase selection difficulty [72], making 64 pixels the most challenging condition. The nine resulting spacing combinations (3 horizontal $\times$ 3 vertical) allowed

examination of how both the proximity of same-type widgets (buttons to each other) and that of different-type widgets (sliders to buttons) affected area cursor behaviour. Figure 4.18 shows an example layout with a vertical separation of 64 pixels and horizontal separation of 256 pixels.

Varying the spacing had direct implications for the area cursor zones. As the horizontal spacing between buttons decreased, the boundaries of their activation zones narrowed, increasing the precision required to target a specific button, even with area-based targeting. As vertical spacing between the slider and buttons decreased, the boundary between the slider's activation zone and the button row's zone moved closer to both widget types, potentially introducing ambiguity when the cursor was positioned between them. These spacing manipulations therefore tested whether the area cursor maintained its usability advantages under increasingly constrained spatial conditions, or whether tight spacing degraded the benefits of proximity-based targeting.

4.8.4 Procedure

Before beginning the trials, participants provided informed consent and were introduced to each gesture type through practice tasks. Participants completed a minimum of 12 practice targets per gesture, though they were free to continue beyond this minimum until they felt comfortable. Each trial began with the participant's non-dominant hand resting on the wireless button; on-screen instructions indicated the required action (e.g., *"Select F on the slider then press button 2"*), and the participant pressed the button to begin, instructed to be make their selections *"as quick and as accurate as possible"*. To assist with task comprehension, the first target widget was highlighted in green and the second in blue, ensuring participants could quickly identify the required sequence without re-reading the instructions mid-trial. There were no time constraints, and participants were permitted to release the slider and make corrections at any point rather than being limited to a single continuous movement. The trial ended once the wireless button was pressed again. Upon completing all conditions, participants verbally ranked the interaction techniques in order of preference, which served as the starting point for a semi-structured interview in which they discussed their reasoning and reflected on their experience.

4.8.5 Results

20 participants were recruited (14 male, 6 female; mean age 28 years, SD 7.4 years; 18 right-handed, 1 left-handed, 1 ambidextrous) through university mailing lists and poster advertising. Participants were compensated with £10 for taking part in the study, which lasted 1 hour. Ethics approval was provided by the College of Science and Engineering Ethics Committee.

Analysis was conducted using repeated-measures ANOVA, followed by post-hoc t-test comparisons of estimated marginal means. If data did not have a normal distribution (via the Shapiro-Wilk test), then the Aligned-Rank Transform [180] was applied first, followed by post-hoc comparisons using ART-C [49] where appropriate.

Below are summary statistics and visualisations for each dependent variable. Table 4.2 shows the results and analyses across all conditions, including summary statistics for each dependent variable, and results from the repeated-measures ANOVA and post-hoc comparisons.

Task Time

Overall, participants completed each trial in an average of 6057 ms (SD = 2061 ms) (See Table 4.2 rows 1-7 and Figure 4.19). Pinch was the faster gesture (M = 5585, SD = 1813) ms, while Push took noticeably longer (M = 6530, SD = 2187) ms. Varying the Horizontal spacing had little effect on task time: the three levels of 64 (M = 6029, SD = 2109) ms, 128 (M = 6009, SD = 2013) ms, and 256 (M = 6134, SD = 2075) ms were closely grouped. Similarly, Vertical spacing produced only minor differences, with 64 the slowest (M = 6236, SD = 2191) ms and 256 the fastest (M = 5958, SD = 2034) ms.

For the Gesture $\times$ Horizontal interaction, Push was consistently slower than Pinch at every spacing level, with the largest gap at 256 px (Push: M = 6672, SD = 2177 ms; Pinch: M = 5597, SD = 1832 ms). In the Gesture $\times$ Vertical interaction, a similar pattern held: Push remained slower across all vertical spacings, while Pinch was fastest at 128 px (M = 5356, SD = 1432) ms. The Horizontal $\times$ Vertical interaction revealed more variability; the slowest combination was 64 $\times$ 64 (M = 6653, SD = 2387) ms, while 128 $\times$ 128 was the fastest (M = 5681, SD = 1653) ms. The remaining combinations fell between these extremes, with no clear monotonic trend across either dimension.

In the three-way Gesture $\times$ Horizontal $\times$ Vertical interaction, the slowest condition was Push at 64 $\times$ 64 spacing (M = 7004, SD = 2492) ms, closely followed by Push at 256 $\times$ 128 (M = 7062, SD = 2483) ms. The fastest condition was Pinch at 64 $\times$ 256 (M = 5040, SD = 1406) ms, followed by Pinch at 128 $\times$ 128 (M = 5136, SD = 1418) ms.

Time to Acquire First Widget

Participants took an average of 1456 ms (SD = 668 ms) to acquire the first widget (See Table 4.2 rows 8-14 and Figure 4.20). Pinch was faster (M = 1308, SD = 604) ms than Push (M = 1604, SD = 698) ms. Across Horizontal spacing levels, all three conditions were closely grouped: 128 px was the fastest (M = 1412, SD = 532) ms, followed by 64 px (M = 1447, SD = 704) ms and 256 px (M = 1508, SD = 752) ms. Vertical spacing had virtually no effect on acquisition time, with all three levels producing near-identical means: 64 px (M = 1446, SD = 662) ms, 256 px (M = 1456, SD = 671) ms, and 128 px (M = 1465, SD = 678) ms.

For the Gesture $\times$ Horizontal interaction, Push was slower than Pinch at all spacing levels, with the largest difference at 64 px (Push: M = 1618, SD = 775 ms; Pinch: M = 1275, SD = 582 ms) and the smallest at 128 px (Push: M = 1551, SD = 573 ms; Pinch: M = 1272, SD = 449 ms). In the Gesture $\times$ Vertical interaction, the gap between gestures was remarkably consistent across all vertical spacing levels, ranging from approximately 250 to 330 ms, with Push uniformly

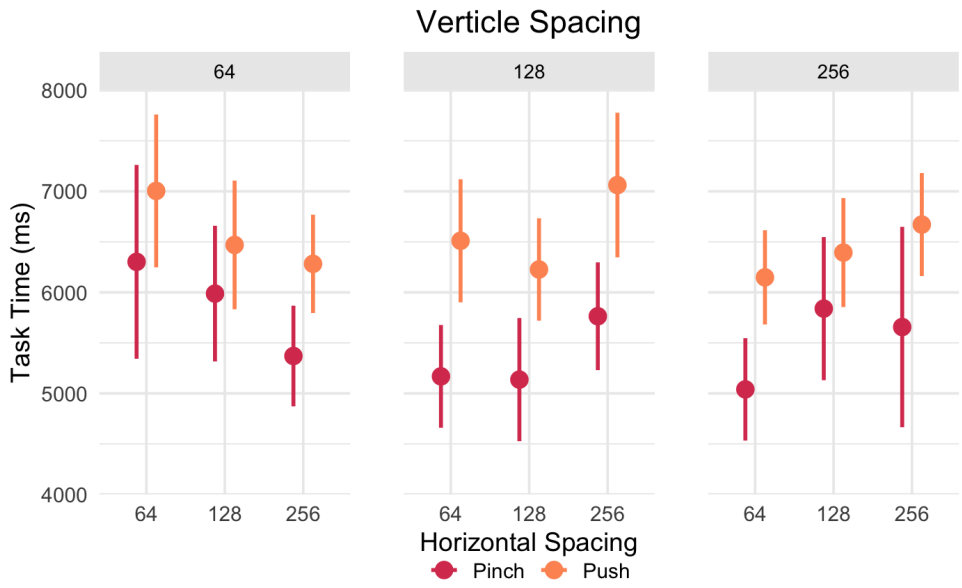


Figure 4.19: Mean task time per condition. Error bars show 95% CIs.

slower. The Horizontal × Vertical interaction revealed that the fastest combination was 128 × 128 (M = 1290, SD = 450) ms, tied with 256 × 64 (M = 1290, SD = 493) ms, while the slowest was 256 × 128 (M = 1669, SD = 833) ms.

In the three-way Gesture × Horizontal × Vertical interaction, the fastest condition was Pinch at 128 × 128 (M = 1119, SD = 330) ms, which also exhibited the lowest variability of any combination. The slowest condition was Push at 256 × 256 (M = 1784, SD = 812) ms, followed closely by Push at 256 × 128 (M = 1754, SD = 825) ms.

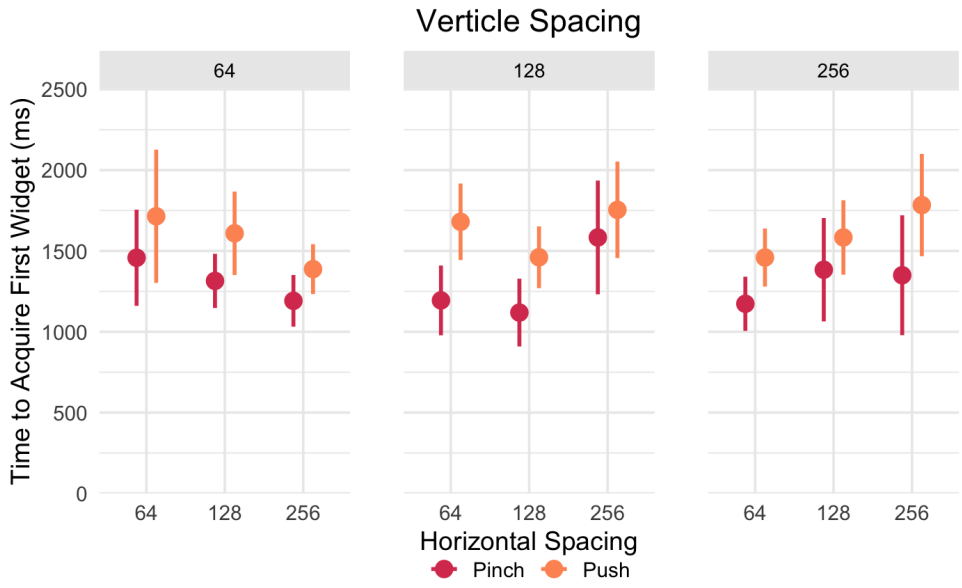


Figure 4.20: Mean time to acquire the first widget per condition. Error bars show 95% CIs.

Time to Transition

Participants took an average of 1247 ms (SD = 673 ms) to transition between widgets (See Table 4.2 rows 15-21 and Figure 4.21). Pinch facilitated faster transitions (M = 1132, SD = 682) ms compared to Push (M = 1363, SD = 646) ms. Horizontal spacing had a modest effect, with 128 px the fastest (M = 1205, SD = 632) ms and 256 px the slowest (M = 1317, SD = 773) ms, while 64 px fell between the two (M = 1219, SD = 603) ms. Vertical spacing followed a similar pattern, with 128 px yielding the shortest transition times (M = 1197, SD = 634) ms and 64 px the longest (M = 1296, SD = 711) ms.

For the Gesture × Horizontal interaction, Push was consistently slower than Pinch across all spacing levels, with the gap widest at 256 px (Push: M = 1483, SD = 794 ms; Pinch: M = 1151, SD = 721 ms). In the Gesture × Vertical interaction, the largest difference between gestures emerged at 128 px vertical spacing, where Pinch was notably fast (M = 1004, SD = 432) ms relative to Push (M = 1389, SD = 740) ms. The Horizontal × Vertical interaction revealed that the fastest combination was 128 × 128 (M = 1073, SD = 485) ms, while the slowest was 64 × 64 (M = 1384, SD = 782) ms. At the widest vertical spacing of 256 px, transition time increased with horizontal distance (64 px: M = 1125 ms; 128 px: M = 1254 ms; 256 px: M = 1368 ms).

In the three-way Gesture × Horizontal × Vertical interaction, the fastest condition was Pinch at 128 × 128 (M = 903, SD = 472) ms, followed closely by Pinch at 64 × 256 (M = 950, SD = 295) ms. The slowest condition was Push at 256 × 128 (M = 1610, SD = 1025) ms, which also exhibited the highest variability among all combinations.

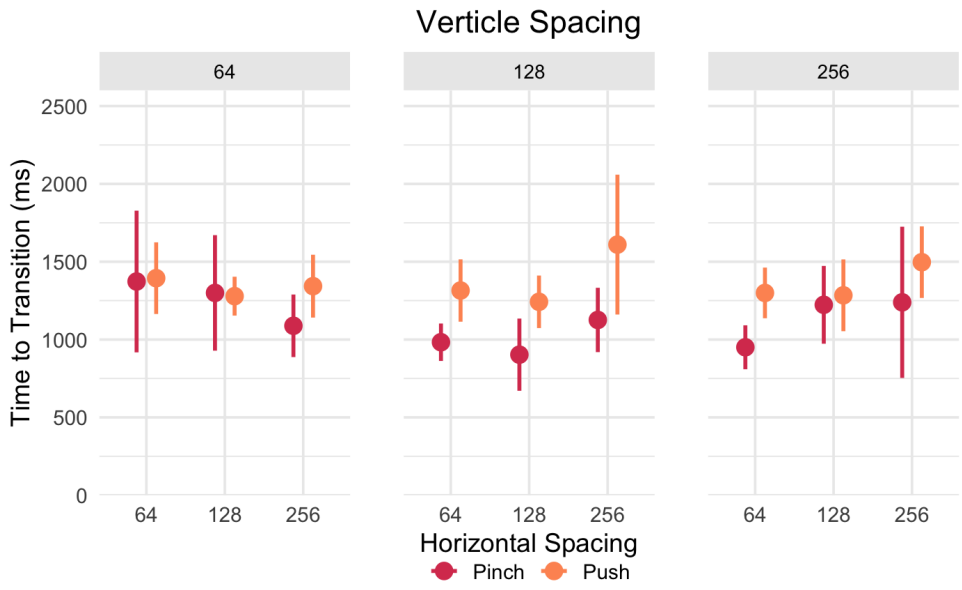


Figure 4.21: Mean time transition between the first and second widget, per condition. Error bars show 95% CIs.

Time to Slider Target

Participants reached the target value on each slider in an average of 983 ms (SD = 421 ms) (See Table 4.2 rows 22-28 and Figure 4.22). Pinch enabled faster targeting (M = 924, SD = 437) ms compared to Push (M = 1042, SD = 396) ms. Horizontal spacing had minimal influence on targeting time, with all three levels tightly clustered: 128 px (M = 978, SD = 432) ms, 64 px (M = 981, SD = 439) ms, and 256 px (M = 990, SD = 394) ms. Vertical spacing showed a slightly wider spread, where 128 px was the fastest (M = 947, SD = 360) ms and 64 px the slowest (M = 1036, SD = 514) ms, with 256 px in between (M = 968, SD = 369) ms. For the Gesture × Horizontal interaction, Pinch was faster than Push at every horizontal spacing level, with the difference remaining relatively stable across conditions (ranging from approximately 80 to 160 ms). In the Gesture × Vertical interaction, the largest gap between gestures appeared at 128 px vertical spacing, where Pinch (M = 857, SD = 291) ms was notably faster than Push (M = 1036, SD = 400) ms. The Horizontal × Vertical interaction revealed that the fastest combination was 128 × 128 (M = 898, SD = 298) ms, while the slowest was 64 × 64 (M = 1105, SD = 612) ms. Targeting times generally decreased with wider vertical spacing when horizontal spacing was narrow (64 px), but showed the opposite trend at 256 px horizontal spacing.

In the three-way Gesture × Horizontal × Vertical interaction, the fastest condition was Pinch at 128 × 128 (M = 803, SD = 202) ms, which also exhibited the lowest variability of any combination. The slowest condition was Pinch at 64 × 64 (M = 1119, SD = 806) ms, though Push at the same spacing was comparable (M = 1090, SD = 345) ms with substantially lower variability, suggesting that the high mean under Pinch may have been driven by a subset of difficult trials.

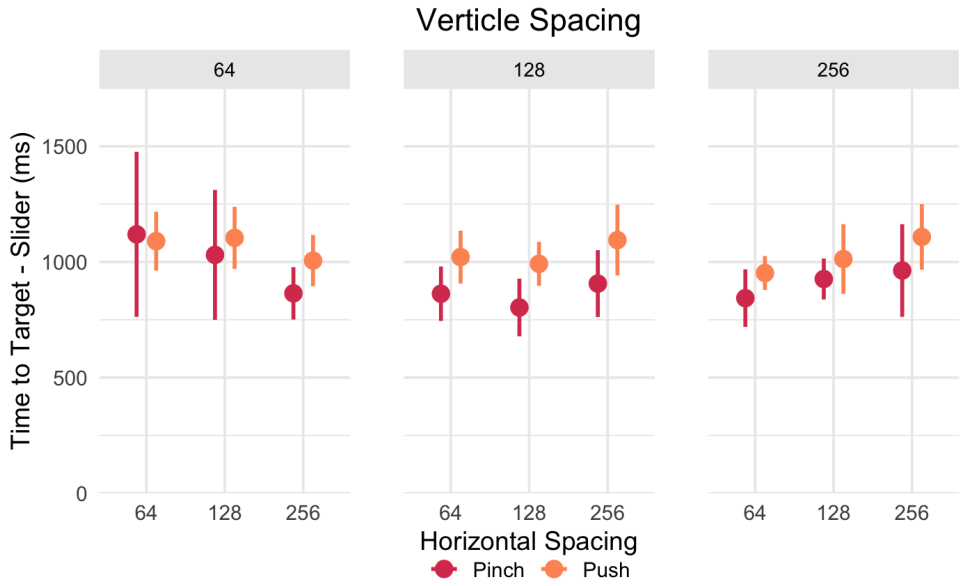


Figure 4.22: Mean time to hit the slider target, per condition. Error bars show 95% CIs.

Slider Error Distance

Participants achieved a mean error distance of 0.14 mm (SD = 1.01 mm) across all conditions (See Table 4.2 rows 29-35 and Figure 4.23), indicating a slight overall tendency to overshoot the target. Pinch produced a marginally larger overshoot (M = 0.22, SD = 0.90) mm, while Push was closer to zero (M = 0.06, SD = 1.11) mm but with greater variability. Horizontal spacing had little systematic effect: 64 px showed the highest overshoot (M = 0.21, SD = 1.05) mm, with 128 px (M = 0.09, SD = 1.02) mm and 256 px (M = 0.11, SD = 0.97) mm both near zero. Vertical spacing similarly had minimal influence, with all three levels producing comparable means ranging from 0.12 to 0.17 mm.

For the Gesture $\times$ Horizontal interaction, Pinch consistently overshot across all spacing levels (ranging from M = 0.18 to 0.26 mm), whereas Push was effectively at zero for both 128 px and 256 px (both M = 0.00 mm), overshooting only at 64 px (M = 0.17, SD = 1.19) mm. In the Gesture $\times$ Vertical interaction, the most notable result was Push at 128 px vertical spacing, which was the only condition to produce an undershoot (M = -0.07, SD = 1.13) mm, while Pinch at the same spacing recorded the largest overshoot of any combination (M = 0.30, SD = 0.87) mm. The Horizontal $\times$ Vertical interaction showed the largest positive offset at 64 $\times$ 64 (M = 0.45, SD = 1.20) mm, with most other combinations clustered close to zero.

In the three-way Gesture $\times$ Horizontal $\times$ Vertical interaction, both gestures overshot at 64 $\times$ 64 spacing (Pinch: M = 0.42, SD = 1.08 mm; Push: M = 0.48, SD = 1.33 mm), representing the highest error magnitudes in the dataset. The largest undershoot was recorded for Push at 64 $\times$ 128 (M = -0.26, SD = 1.24) mm. Across the majority of conditions, confidence intervals crossed zero, suggesting that error distances were not reliably different from zero at most spacing combinations.

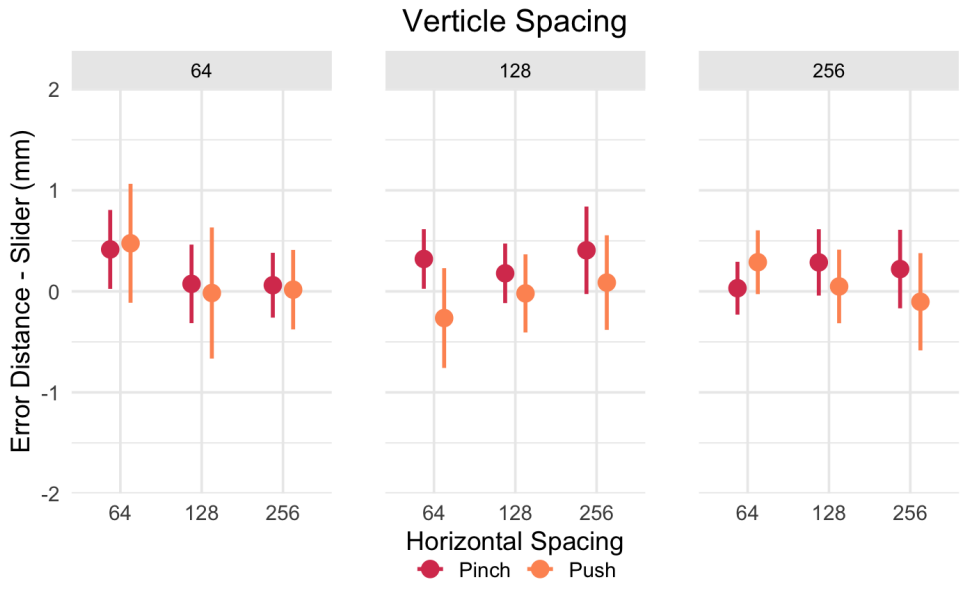


Figure 4.23: Mean slider error distance, per condition. Error bars show 95% CIs.

Slider Activation Distance

Participants activated sliders from a mean distance of 38.93 mm (SD = 25.03 mm) from the widget centre with the proxemic cursor (See Table 4.2 rows 43-49 and Figure 4.24). The two gesture types show near-identical activation distances (Pinch: M = 38.83, SD = 25.44 mm; Push: M = 39.04, SD = 24.68 mm). Horizontal spacing had a modest effect, with activation distance increasing slightly as spacing widened: 64 px (M = 37.68, SD = 24.01) mm, 128 px (M = 38.46, SD = 23.51) mm, and 256 px (M = 40.65, SD = 27.49) mm. A more pronounced trend emerged for Vertical spacing, where participants activated from progressively greater distances as vertical separation increased: 64 px (M = 35.37, SD = 22.28) mm, 128 px (M = 39.77, SD = 25.07) mm, and 256 px (M = 41.65, SD = 27.27) mm.

For the Gesture $\times$ Horizontal interaction, both gestures produced similar activation distances at 64 px and 128 px, though at 256 px Push was slightly higher (M = 41.88, SD = 26.65) mm than Pinch (M = 39.41, SD = 28.47) mm. In the Gesture $\times$ Vertical interaction, both gestures followed the same upward trend across vertical spacing levels, with the two converging closely at 128 px (Pinch: M = 39.73 mm; Push: M = 39.81 mm) and diverging marginally at 256 px (Push: M = 42.25 mm; Pinch: M = 41.05 mm). The Horizontal $\times$ Vertical interaction revealed that the shortest activation distance occurred at 256 $\times$ 64 (M = 33.46, SD = 24.61) mm, while the longest was at 256 $\times$ 256 (M = 45.79, SD = 29.94) mm, suggesting that wider vertical spacing had a stronger influence on activation distance than horizontal spacing.

In the three-way Gesture $\times$ Horizontal $\times$ Vertical interaction, the largest activation distance was recorded for Push at 256 $\times$ 256 (M = 49.12, SD = 28.96) mm, while the smallest was Push at 256 $\times$ 64 (M = 32.23, SD = 25.63) mm. Across all conditions, substantial standard deviations indicate considerable individual differences in how far from the widget centre participants chose to initiate their interactions.

Button Activation Distance

Participants activated buttons from a mean distance of 26.86 mm (SD = 10.65 mm) from the widget centre (See Table 4.2 rows 36-42 and Figure 4.25). Push resulted in slightly greater activation distances (M = 27.81, SD = 11.66) mm than Pinch (M = 25.90, SD = 9.48) mm. Horizontal spacing exhibited a strong linear relationship with activation distance: 64 px (M = 20.28, SD = 9.04) mm, 128 px (M = 25.01, SD = 7.36) mm, and 256 px (M = 35.27, SD = 9.39) mm, indicating that participants engaged buttons from progressively greater distances as horizontal separation increased. By contrast, Vertical spacing had a negligible effect, with all three levels producing similar means: 64 px (M = 26.43, SD = 11.05) mm, 128 px (M = 26.58, SD = 11.41) mm, and 256 px (M = 27.56, SD = 9.45) mm.

For the Gesture $\times$ Horizontal interaction, both gestures followed the same increasing trend across horizontal spacing levels, with Push consistently recording slightly higher distances than Pinch. The gap between gestures was largest at 256 px (Push: M = 36.51, SD = 10.44 mm;

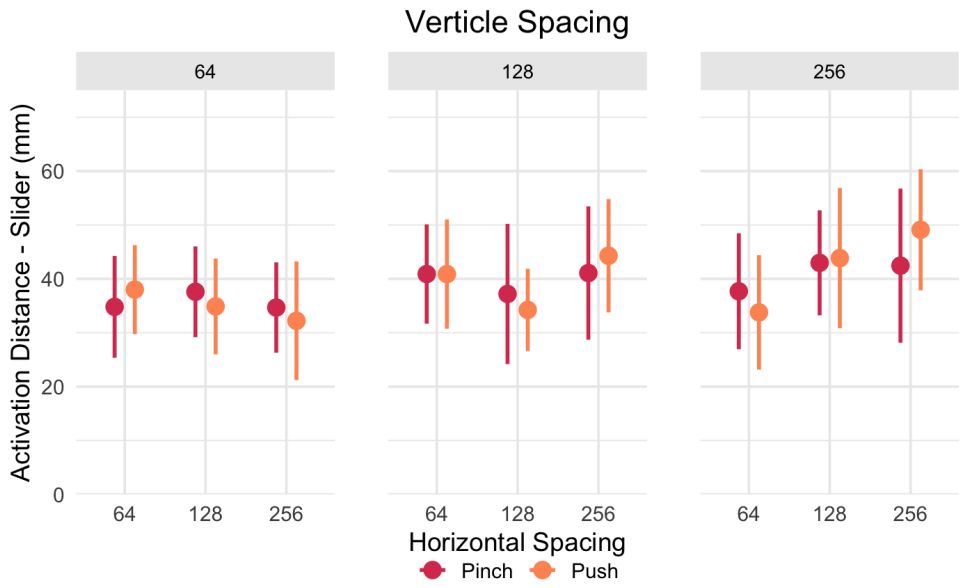


Figure 4.24: Mean distance participants took control over the slider, per condition. Error bars show 95% CIs.

Pinch: M = 34.03, SD = 8.09 mm) and smallest at 128 px (Push: M = 25.65 mm; Pinch: M = 24.38 mm). In the Gesture × Vertical interaction, Push exceeded Pinch at all vertical spacing levels, with the widest difference at 64 px (Push: M = 27.95, SD = 12.75 mm; Pinch: M = 24.90, SD = 8.89 mm). The Horizontal × Vertical interaction confirmed that horizontal spacing was the dominant factor: at each vertical level, activation distance increased substantially from 64 px to 256 px horizontal spacing, while varying vertical spacing within a given horizontal level produced only minor changes. The shortest activation distance was observed at 64 × 128 (M = 19.23, SD = 7.46) mm and the longest at 256 × 128 (M = 36.27, SD = 11.23) mm.

In the three-way Gesture × Horizontal × Vertical interaction, the largest activation distance was recorded for Push at 256 × 256 (M = 37.05, SD = 7.74) mm, while the smallest was Pinch at 64 × 128 (M = 18.54, SD = 6.00) mm. Across conditions, activation distances for buttons were generally lower and less variable than those observed for sliders, suggesting that participants approached button targets more closely before committing to activation.

Heisenberg Displacement

Participants displaced the cursor by a mean of 41.32 mm (SD = 31.88 mm) between activation onset and completion (See Table 4.2 rows 57–63 and Figure 4.27). Push produced far greater displacement (M = 68.14, SD = 23.17) mm than Pinch (M = 14.49, SD = 7.43) mm, almost five times as much. Horizontal spacing had little effect, with all three levels producing similar means: 64 px (M = 39.27, SD = 30.13) mm, 128 px (M = 43.20, SD = 32.68) mm, and 256 px (M = 41.48, SD = 32.91) mm. Vertical spacing was similarly negligible: 64 px (M = 41.15, SD = 32.68) mm, 128 px (M = 40.53, SD = 30.97) mm, and 256 px (M = 42.26, SD = 32.22)

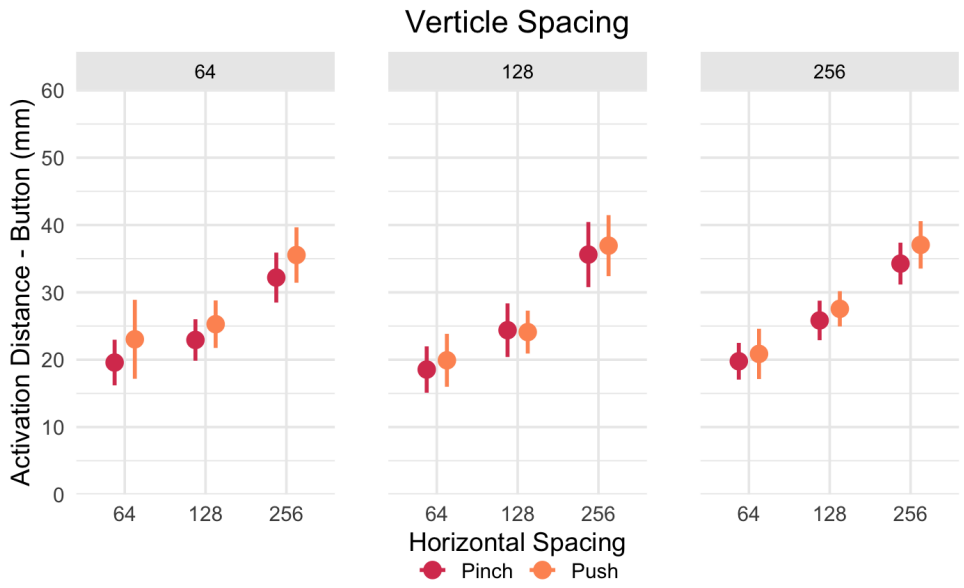


Figure 4.25: Mean distance participants activated the buttons, per condition. Error bars show 95% CIs.

mm. Unlike activation distance, which increased steadily with horizontal spacing, displacement showed no ordered trend across either spacing factor.

For the Gesture × Horizontal interaction, both gestures were highest at the intermediate spacing rather than increasing with separation. Push rose from 64 px (M = 64.27, SD = 22.46) mm to 128 px (M = 71.27, SD = 22.26) mm before falling slightly at 256 px (M = 68.87, SD = 24.55) mm, and Pinch followed the same shape on a much smaller scale: 64 px (M = 14.26, SD = 7.38) mm, 128 px (M = 15.13, SD = 7.51) mm, and 256 px (M = 14.09, SD = 7.48) mm. In the Gesture × Vertical interaction, neither gesture showed an ordered trend; Push varied by only 2.33 mm across the three levels (M = 67.64, 67.22, and 69.55 mm for 64, 128, and 256 px respectively) and Pinch by 1.14 mm (M = 14.66, 13.84, and 14.98 mm). The Horizontal × Vertical interaction produced no systematic pattern either: the shortest displacement was observed at 64 × 128 (M = 36.70, SD = 27.78) mm and the longest at 128 × 128 (M = 44.39, SD = 32.07) mm, a difference of less than 8 mm.

In the three-way Gesture × Horizontal × Vertical interaction, the largest displacement was recorded for Push at 128 × 256 (M = 73.56, SD = 20.66) mm, while the smallest was Pinch at 256 × 128 (M = 12.14, SD = 5.32) mm. Every Push condition (60.31–73.56 mm) was well above every Pinch condition (12.14–16.30 mm), and the two never overlapped at any spacing combination. Push displacement was also far more variable (SDs of 19–27 mm) than Pinch (SDs of 5–9 mm), indicating that Push was both larger and less consistent. Across conditions, the size of the Heisenberg effect depended almost entirely.

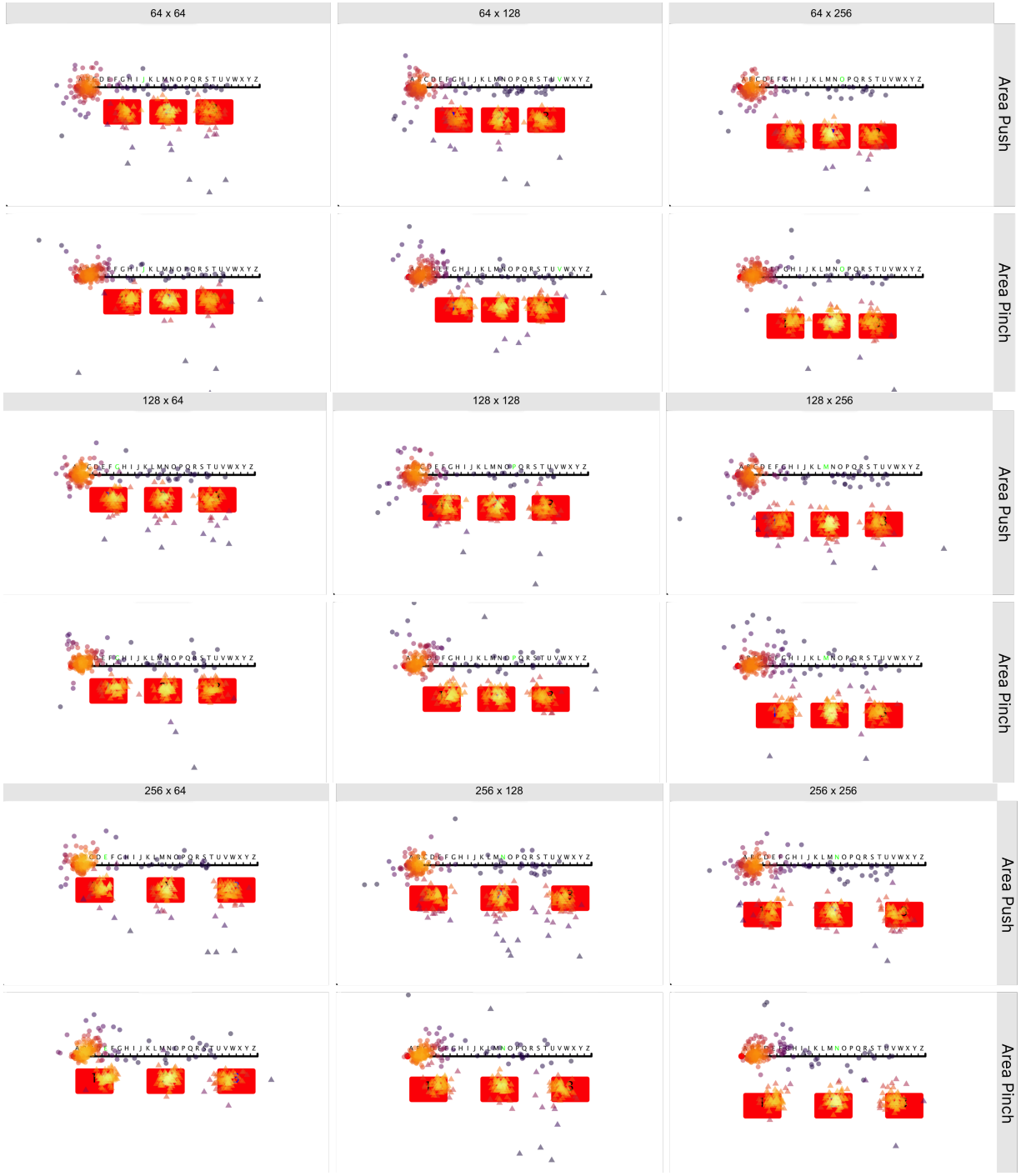


Figure 4.26: Cursor positions at the time of widget activation, with brighter colours marking higher density areas. Slider and button activations are marked by circles and triangles, respectively.

Task Success Rate

Participants achieved an overall success rate of 96.59% (SD = 7.92%) across all conditions (See Table 4.2 rows 50-56 and Figure 4.28). Pinch had a higher success rate (M = 98.04, SD = 5.90)% than Push (M = 95.14, SD = 9.31)%, with Push also exhibiting greater variability. Horizontal

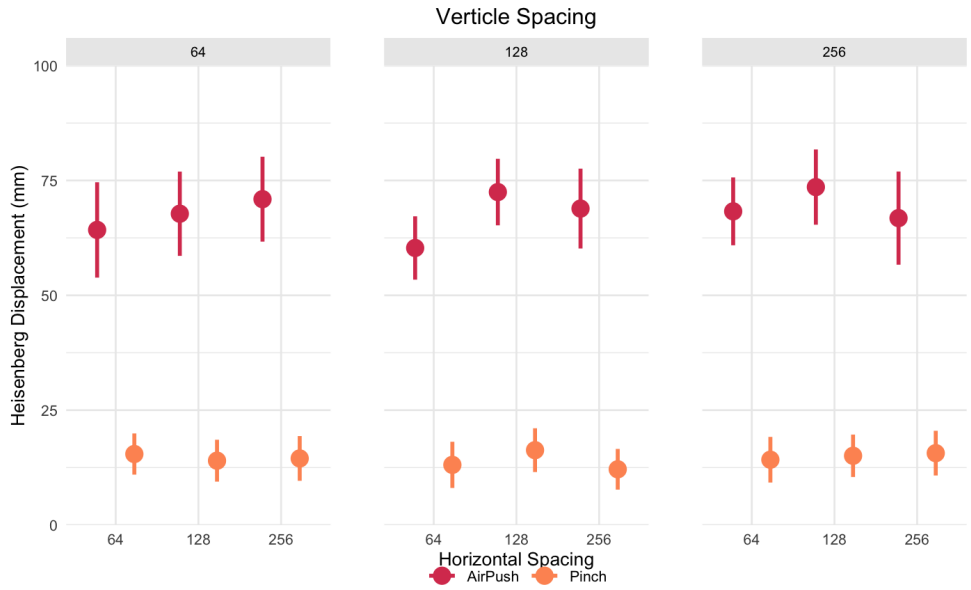


Figure 4.27: Heisenberg displacement per condition. Error bars show 95% CIs.

spacing had a limited effect, with all three levels producing similar means: 64 px (M = 97.05, SD = 8.03)%, 256 px (M = 96.84, SD = 6.77)%, and 128 px (M = 95.89, SD = 8.83)%. Vertical spacing showed a slight upward trend, with success rates increasing from 64 px (M = 95.98, SD = 8.60)% to 256 px (M = 97.29, SD = 6.78)%.

For the Gesture × Horizontal interaction, Pinch maintained consistently high success rates across all spacing levels, peaking at 64 px (M = 99.03, SD = 2.95)% and 256 px (M = 98.54, SD = 4.77)%. Push, by contrast, remained stable around 95% regardless of horizontal spacing, with higher variability throughout. In the Gesture × Vertical interaction, Pinch stayed above 97.7% across all vertical levels, whereas Push improved noticeably at 256 px (M = 96.60, SD = 6.65)% compared to its lowest point at 64 px (M = 94.20, SD = 10.41)%. The Horizontal × Vertical interaction showed that the highest success rate occurred at 64 × 256 (M = 98.92, SD = 3.36)%, while the lowest was at 128 × 64 (M = 95.42, SD = 9.37)%, though differences across combinations were modest.

In the three-way Gesture × Horizontal × Vertical interaction, Pinch at 64 × 256 (M = 99.50, SD = 2.24)% and Pinch at 256 × 256 (M = 99.44, SD = 2.48)% achieved the highest success rates with minimal variability. The lowest success rate was recorded for Push at 64 × 64 (M = 93.06, SD = 12.32)%, which also exhibited the greatest variability of any condition, followed by Push at 64 × 128 (M = 93.83, SD = 12.89)%. Across all conditions, Push consistently produced wider standard deviations than Pinch, suggesting that the latter afforded more reliable task completion.

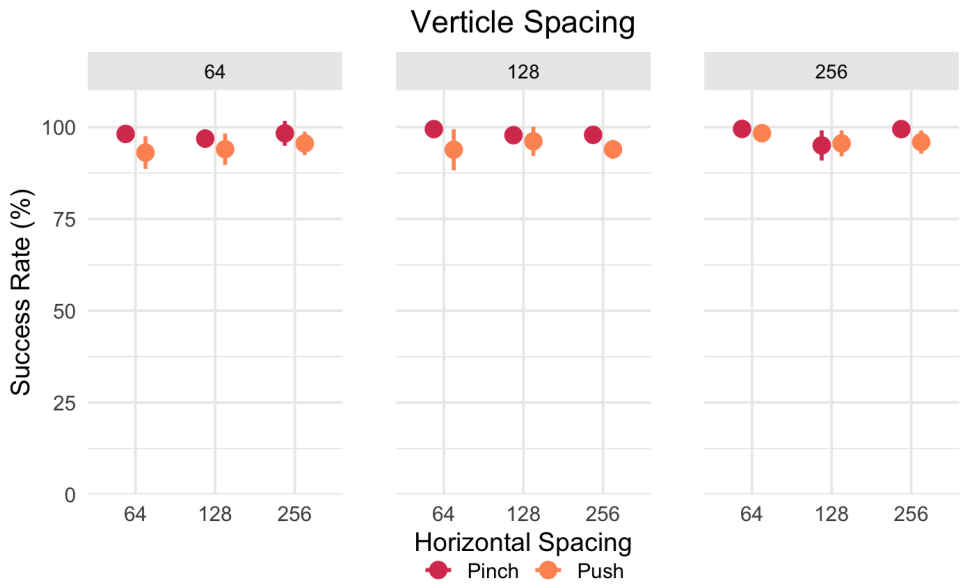


Figure 4.28: Mean task success rate per condition. Error bars show 95% CIs.

Table 4.2: ANOVA results, and post-hoc test results for Experiment 3. Significant comparisons are highlighted*

	Overall task time	**Mean 6057 ms, SD 2061 ms 95% CI [5844, 6271] ms**			
	ANOVA	*Result*	p	η_p^2	*Post-Hoc Comparisons*
1	*Gesture*	F(1, 323) = 46.86	< .001*	.13	• Push > Pinch ($p \leq .001$)
2	*Horizontal (H)*	F(2, 323) = 0.50	.61	.00	
3	*Vertical (V)*	F(2, 323) = 1.18	.31	.01	
4	*Gesture* × *H*	F(2, 323) = 0.81	.44	.01	
5	*Gesture* × *V*	F(2, 323) = 0.78	.46	.00	
6	*H* × *V*	F(4, 323) = 3.93	< .001*	.05	No significant comparisons
7	*Gesture* × *H* × *V*	F(4, 323) = 0.31	.87	.00	
	Time to first widget	**Mean 1178 ms SD 1461 ms, 95% CI [10267, 1330] ms**			
8	*Gesture*	F(1, 323) = 36.85	< .001*	.10	• Push > Pinch ($p \leq .001$)
9	*Horizontal (H)*	F(2, 323) = 0.34	.71	.00	
10	*Vertical (V)*	F(2, 323) = 1.51	.22	.01	
11	*Gesture* × *H*	F(2, 323) = 0.18	.84	.00	
12	*Gesture* × *V*	F(2, 323) = 0.39	.68	.00	
13	*H* × *V*	F(4, 323) = 1.26	.29	.02	
14	*Gesture* × *H* × *V*	F(4, 323) = 0.74	.57	.01	

Continued on next page

Table 4.2: (continued)

	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
	Transition time	**Mean 1247 ms, SD 673 ms, 95% CI [1177, 1317] ms**			
15	*Gesture*	F(1, 323) = 33.53	< .001*	.09	• Push > Pinch ($p \leq .001$)
16	*Horizontal (H)*	F(2, 323) = 1.17	.31	.01	
17	*Vertical (V)*	F(2, 323) = 0.57	.57	.00	
18	*Gesture* × *H*	F(2, 323) = 1.43	.24	.01	
19	*Gesture* × *V*	F(2, 323) = 1.17	.31	.01	
20	*H* × *V*	F(4, 323) = 1.69	.15	.02	
21	*Gesture* × *H* × *V*	F(4, 323) = 1.11	.35	.01	
	Slider time to target	**Mean 983 ms, SD 421 ms 95% CI [940, 1027] ms**			
22	*Gesture*	F(1, 323) = 22.88	< .001*	.07	• Push > Pinch ($p \leq .001$)
23	*Horizontal (H)*	F(2, 323) = 0.03	.97	.00	
24	*Vertical (V)*	F(2, 323) = 0.98	.38	.01	
25	*Gesture* × *H*	F(2, 323) = 0.06	.94	.00	
26	*Gesture* × *V*	F(2, 323) = 0.44	.64	.00	
27	*H* × *V*	F(4, 323) = 1.89	.11	.02	
28	*Gesture* × *H* × *V*	F(4, 323) = 0.27	.90	.00	
	Slider error distance	**Mean 0.14 mm, SD 1.01 mm, 95% CI [0.03, 0.24] mm**			
29	*Gesture*	F(1, 323) = 3.54	.06	.01	
30	*Horizontal (H)*	F(2, 323) = 0.75	.47	.00	
31	*Vertical (V)*	F(2, 323) = 0.16	.86	.00	
32	*Gesture* × *H*	F(2, 323) = 0.15	.86	.00	
33	*Gesture* × *V*	F(2, 323) = 1.25	.29	.01	
34	*H* × *V*	F(4, 323) = 0.97	.42	.01	
35	*Gesture* × *H* × *V*	F(4, 323) = 1.26	.29	.02	
	Button distance	**Mean 26.86 mm, SD 10.65 mm, 95% CI [25.75, 27.96] mm**			
36	*Gesture*	F(1, 323) = 4.48	.03*	.01	• Push > Pinch ($p-.04$)
37	*Horizontal (H)*	F(2, 323) = 181.22	< .001*	.53	• 64 < 128 & 256 ($p \leq .001$) • 128 < 256 ($p \leq .001$)
38	*Vertical (V)*	F(2, 323) = 3.82	.02*	.02	• 256 > 64 ($p = .03$)
39	*Gesture* × *H*	F(2, 323) = 0.18	.83	.00	
40	*Gesture* × *V*	F(2, 323) = 0.61	.55	.00	

Continued on next page

Table 4.2: (continued)

	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
41	$H \times V$	F(4, 323) = 0.90	.46	.01	
42	*Gesture* $\times H \times V$	F(4, 323) = 0.22	.93	.00	
	Slider distance mm	**Mean 38.93 mm, SD 25.03 mm, 95% CI [36.34, 41.53] mm**			
43	*Gesture*	F(1, 323) = 0.07	.79	.00	
44	*Horizontal (H)*	F(2, 323) = 0.31	.73	.00	
45	*Vertical (V)*	F(2, 323) = 1.82	.16	.01	
46	*Gesture* $\times H$	F(2, 323) = 0.50	.61	.00	
47	*Gesture* $\times V$	F(2, 323) = 0.20	.82	.00	
48	$H \times V$	F(4, 323) = 1.85	.12	.02	
49	*Gesture* $\times H \times V$	F(4, 323) = 0.62	.65	.01	
	Task success	**Mean 96%, SD 8%, 95% CI [96, 97]%**			
50	*Gesture*	F(1, 323) = 5.33	.02*	.02	• Pinch > Push (p= .02)
51	*Horizontal (H)*	F(2, 323) = 1.07	.34	.01	
52	*Vertical (V)*	F(2, 323) = 0.20	.82	.00	
53	*Gesture* $\times H$	F(2, 323) = 0.23	.79	.00	
54	*Gesture* $\times V$	F(2, 323) = 0.32	.73	.00	
55	$H \times V$	F(4, 323) = 1.09	.36	.01	
56	*Gesture* $\times H \times V$	F(4, 323) = 1.50	.20	.02	
	Heisenberg	**Mean 41 mm, SD 32 mm, 95% CI [38, 45] mm**			
57	*Gesture*	F(1, 323) = 1263.39	< .001*	.80	• Push > Pinch ($p\leq$ .001)
58	*Horizontal (H)*	F(2, 323) = 3.36	.04*	.02	• 128 > 64 (p= .02)
59	*Vertical (V)*	F(2, 323) = 0.80	.45	.00	
60	*Gesture* $\times H$	F(2, 323) = 1.89	.15	.01	
61	*Gesture* $\times V$	F(2, 323) = 0.27	.76	.00	
62	$H \times V$	F(4, 323) = 1.09	.36	.01	
63	*Gesture* $\times H \times V$	F(4, 323) = 0.93	.45	.01	

4.9 Overall Discussion

4.9.1 Area Cursor Usability

By using the area cursor, participants were able to acquire and transition between widgets with substantially less precision, leading to faster interaction times without a detrimental effect on

accuracy. In Experiment 2, the area cursor reduced the time to acquire the first slider by approximately 700 ms compared to the standard cursor (3021 ms vs. 3706 ms, $p < .001$), and reduced transition times between sliders by over 1000 ms (3381 ms vs. 4400 ms, $p < .001$). These reductions in acquisition and transition time were the primary contributors to the overall task time advantage observed for the area cursor (12549 ms vs. 13543 ms, $p < .001$). Critically, this speed improvement did not come at the cost of accuracy: there was no significant effect of cursor type on error distance ($p = .16$), and the area cursor maintained comparable or higher success rates across conditions.

These findings extend those from Experiment 1, where the Area Pinch technique reduced single-widget acquisition time to 1033 ms compared to 1694 ms for standard Pinch. By generalising the area-based acquisition to multi-widget environments, the proxemic cursor demonstrates that the performance benefits observed with a single slider are not an artefact of the lack of targeting. When two sliders are present, and the proxemic cursor must resolve which widget the user intends to control, the speed advantage is preserved. The transition phase, absent in Experiment 1's single-widget design, represented the interaction phase, during which the proxemic cursor showed its largest effect ($\eta_p^2 = .30$), suggesting that area-based targeting is particularly beneficial when users must release one widget and acquire another in quick succession.

In Experiment 3, where the area cursor was used in all conditions (given the confirmed efficacy from Experiment 2), acquisition and transition times remained low (1456 ms and 1247 ms, respectively) even with a more complex interface containing both sliders and buttons at varying separations. Notably, neither horizontal nor vertical widget spacing had a significant main effect on acquisition time, transition time, or overall task time. This indicates that the proxemic cursor maintained its usability advantages across the range of spatial configurations tested, including at the most constrained spacing (64 px separation), where the boundaries between adjacent activation zones were narrowest. Post-experiment interviews supported this quantitative finding, with participants reporting that the proxemic cursor allowed them to gain control of widgets without precise targeting, regardless of the widget density (e.g. *"I just moved my hand towards where I needed to go and could be less accurate"* [P5].

4.9.2 Targeting Behaviours

A central aim of this chapter was to characterise how users exploited the relaxed targeting affordance of the area cursor. The activation distance data and cursor position distributions (Figures 4.15, 4.26, 4.24, and 4.25) reveal distinct targeting strategies that differed between continuous and discrete widget types.

For sliders, participants in Experiment 2 activated from a mean distance of 26.90 mm from the widget handle when using the area cursor. This was substantially further than the handle radius required for standard cursor activation, confirming that users did take advantage of the proxemic targeting affordance. The cursor position heatmaps (Figure 4.15) further illustrate that

activations were distributed around the slider handles rather than concentrated directly on them, with considerable spread especially in the VSS layout. However, the distribution was not uniform across the entire display; users generally moved the cursor *towards* the slider handle before activating, rather than gesturing from arbitrary positions. This movement tendency towards the widget (despite any requirement to do so) suggests that users adopted a more *lazy* targeting strategy, using the proxemic cursor to reduce the precision demands of the final approach rather than bypassing the approach entirely. This behaviour is consistent with the in-focus visual feedback design, where the nearest-widget highlight provided feedforward about which element would respond to the activation gesture, encouraging users to move close enough to be confident of the targeting assignment.

This targeting pattern contrasts with that observed for the Area Pinch technique in Experiment 1, where users acquired the single slider from positions distributed across the full width of the bar (Figure 3.13, mean start point at 42% of the slider bar width). The difference reflects the additional constraints of multi-widget interfaces; when only one widget is present, there is no ambiguity about which element will be activated, and users are free to gesture from wherever is comfortable. With multiple widgets, users must navigate the boundaries between dynamic activation zones, and this boundary management pulls the cursor closer to the intended target.

The Experiment 3 cursor position heatmap (Figure 4.26) reveals further detail about how participants managed area targeting across a mixed-widget interface. A clear centre-biased pattern is visible for button activations: selections cluster on the right side of button 1, the centre of button 2, and the left side of button 3. Rather than targeting each button in its centre, users kept the cursor in a comfortable central position and exploited the area cursor to activate from there. This behaviour was consistent across all nine spacing combinations, reinforcing the lazy targeting argument: users minimised arm movement by maintaining a central hand position and relying on the area cursor to resolve the intended target. Post-experiment interviews corroborated this observation, with participants reporting that they kept their *"hand down lower"* [P3] and made *"less movement"* [P20] towards the target area, deliberately keeping the cursor near the display centre (*"I feel like the centre of the screen made sense to reach everything"* [P14]) because this had *"less physical effort"* [P17], allowing them to keep their elbows by their torso without raising the arm.

Secondly, the Experiment 3 heatmap also shows that slider activations clustered *below* the slider handle across all spacing conditions, despite the proximity of the button row beneath. Even at the tightest vertical separation of 64 px, the cursor position distributions for slider and button activations remained clearly separated, indicating that users reliably distinguished between the two widget types' activation zones. This is notable because it demonstrates that boundary management remained effective when transitioning between fundamentally different widget types (continuous vs. discrete) at close range. The combination of the in-focus highlight and the handle-based distance metric appeared to give users sufficient confidence to activate the

slider from below without inadvertently triggering a button.

For buttons in Experiment 3, the mean activation distance was 26.86 mm, yet 60% of button selections occurred within the button boundaries. This suggests that, for discrete widgets with relatively large surfaces (as were the buttons), users tended to target the buttons directly rather than rely on proxemic cues. A strong linear relationship emerged between horizontal button spacing and activation distance (20.28 mm at 64 px, 25.01 mm at 128 px, 35.27 mm at 256 px; $\eta_p^2 = .53$), indicating that users activated buttons from progressively greater distances as the space between adjacent buttons widened. By contrast, vertical spacing had a negligible effect on button activation distance.

The slider activation distance data from Experiment 3 tell a similar story. Unlike buttons, slider activation distances showed no significant effect of horizontal spacing ($p = .73$) but a trend towards greater activation distances at wider vertical spacings (35.37 mm at 64 px vs. 41.65 mm at 256 px). Since vertical spacing determined the distance between the slider and the button row beneath it, wider vertical separation gave the slider a larger activation zone. This pattern reinforces that participants adapted their targeting strategies to the available proxemic space, approaching more closely when activation zones were narrow and gesturing from greater distances when the spatial layout permitted.

4.9.3 Activation Gestures

Across both experiments, the Pinch gesture consistently outperformed Push for acquisition, transition, and overall task time. In Experiment 2, Pinch was significantly faster than Push for task time (12326 ms vs. 13762 ms, $p < .001$), acquisition of the first slider (3177 ms vs. 3553 ms, $p < .001$), and transition between sliders (3539 ms vs. 4244 ms, $p < .001$). Experiment 3 was similar: Pinch was faster for task time (5585 ms vs. 6530 ms, $p < .001$), first widget acquisition (1308 ms vs. 1604 ms, $p < .001$), transition (1132 ms vs. 1363 ms, $p < .001$), and time to reach the slider target (924 ms vs. 1042 ms, $p < .001$). These findings are consistent with the results from Experiment 1 (Section 3.5), where pinch-based techniques were similarly faster than Push for single-widget slider control. With Pinch showing advantages across three experiments, with both single and multiple widgets, and with and without an area cursor, provides further evidence that the binary state transition afforded by the proprioceptive feedback of the thumb together is a better method for mid-air widget activation than the depth-based push gesture.

An important question motivating these experiments was whether the area cursor might equalise performance between Push and Pinch by mitigating the target slips that disadvantage Push during precise targeting. The Gesture $\times$ Area interaction in Experiment 2 provides partial support for this hypothesis. The area cursor benefited Push more than Pinch for both acquisition (approximately 1000 ms reduction for Push vs. 400 ms for Pinch) and transition (approximately 1300 ms reduction for Push vs. 700 ms for Pinch), reducing the performance gap between the two gestures. However, Pinch with the area cursor remained the fastest and most preferred

combination. This suggests that while the proxemic cursor alleviates some of the targeting difficulties associated with Push, it may not fully compensate for the Heisenberg effect and the inherent depth-axis ambiguity that remain with the Push activation mechanism.

Pinch also achieved higher task success rates in both experiments (98.79% vs. 97.04% in Experiment 2, $p = .02$; 98.04% vs. 95.14% in Experiment 3, $p = .02$), with Push consistently exhibiting greater variance. As seen in both Experiment 2 and Experiment 3 heatmaps, when using the Push gesture, the activation points are much more spread around the widget handle, especially underneath. This suggests that while the area cursor can counteract the Heisenberg effect it does not remove it. While non-significant, there is a slight trend towards smaller vertical spacing, which is associated with a lower success rate and may be due to the Heisenberg effect.

In the preference rankings for Experiment 2, Area Pinch was the most preferred technique, ranked first by 14 of 21 participants. Push (without the area cursor) was significantly less preferred than all other techniques. However, it is worth noting that Push with the area cursor (Area Push) was ranked more favourably than standard Push, suggesting that the proxemic cursor did improve the subjective experience of push-based activation even if it did not fully close the performance gap with Pinch.

4.9.4 Layout and Widget Spacing

The widget layouts across the two experiments reveal how the arrangement interacts with the area cursor behaviour. In Experiment 2, the HSS layout produced the fastest overall task times (12497 ms), followed by VSS (13229 ms) and HTB (13422 ms). The HTB layout was notably slower, which can be attributed to the vertical distance users had to traverse between the two stacked sliders. The Cursor $\times$ Layout interaction ($p = .03$) indicated that the proxemic cursor's task time advantage was significant for HTB and VSS layouts but not for HSS. This suggests that when widgets are arranged side-by-side along the horizontal axis (the direction of slider manipulation), the transition between widgets is sufficiently short that the proxemic cursor's benefit is less pronounced. By contrast, when transitions require vertical cursor movement (HTB) or when sliders are oriented vertically (VSS), the proxemic cursor provides a more substantial improvement, likely because vertical hand movements in mid-air are less comfortable and less precise, making the relaxed targeting particularly valuable [80].

The time to reach the slider target was primarily influenced by layout rather than cursor type or gesture ($\eta_p^2 = .40$ for layout vs. .03 for cursor), with the VSS layout substantially slower (1905 ms) than HSS (1214 ms) and HTB (1281 ms). This effect reflects the inherent difficulty of vertical hand movement for continuous slider control, consistent with the known ergonomic disadvantage of sustained vertical arm motion in mid-air.

In Experiment 3, where all conditions used the area cursor, spacing had no significant main effects on task time, acquisition time, or transition time. The only significant spacing-related effect was a Horizontal $\times$ Vertical interaction on task time ($p < .01$), though post-hoc comparisons

found no significant pairs. Indicating that the area cursor maintained its effectiveness even at the smallest spacing. This suggests that mid-air touchless interface designers can use compact widget layouts (down to 64 px separation [72]) without degrading the area cursor's usability, enabling more information-dense displays while preserving efficient mid-air input.

4.9.5 Design Implications

Several practical recommendations for touchless interface design emerge from these findings:

Use area-based targeting with Pinch activation. The combination consistently produced the fastest, most accurate, and most preferred interactions across both continuous and discrete tasks. Where platform support allows, Pinch should be the default confirmation gesture when an area cursor is in use.

The proxemic cursor was usable with only a subtle in-focus highlight. Users were able to exploit area-based targeting effectively without the activation region being drawn, which allows a less obtrusive interface design. This thesis did not compare this design against a visible activation region (e.g., the *Bubble Cursor* [71]), so it shows that the minimal design is sufficient, not that visible feedback is unnecessary.

Place high-priority and frequently used widgets towards the centre of the display. The centre-biased targeting pattern observed in Experiment 3 suggests users favour widgets within their comfortable range of motion. Widgets requiring continuous manipulation (sliders) should be positioned so that full traversal remains within reliable tracking bounds; discrete widgets (buttons) can be placed more peripherally, since users are willing to activate them from greater distances when the area cursor is available.

Ensure the full interaction space (including any spatial offset from area-based acquisition) falls within the sensor's reliable tracking volume. Area-based targeting can push the hand towards the edge of reliable tracking, and the 1:1 control-display gain used here means the effective interaction space is directly bounded by the sensor's field of view. The subsequent chapters address this constraint directly, investigating how control-display gain can be used alongside area cursors to reduce the physical movement required for large interaction spaces.

4.10 Chapter Conclusions

The results presented in this chapter demonstrate that area-based targeting significantly improved widget acquisition and transition times without compromising selection accuracy. In Experiment 2, the area cursor reduced the time to acquire the first slider by approximately

700 ms and reduced transition times between sliders by over 1000 ms compared to the standard cursor. This was consistent across all three slider layouts, though the effect was most pronounced for layouts requiring vertical cursor movement (HTB and VSS), where the relaxed targeting was particularly valuable given the ergonomic constraints of sustained vertical hand motion. Experiment 3 extended the investigation to mixed-widget interfaces containing both continuous (slider) and discrete (button) elements at varying spatial separations. The area cursor maintained its effectiveness across all spacing combinations, including the tightest 64 px separation, with no significant effects of horizontal or vertical spacing on overall task time.

The analysis of cursor position revealed that users adopted distinct targeting strategies for continuous and discrete widgets. For sliders, participants exploited the area-based targeting by activating from positions distributed around the handle rather than directly over it, though they generally moved the cursor towards the handle before gesturing. For buttons, 60% of activations occurred within the widget boundaries, with activation distance scaling linearly with horizontal button spacing. Most notably, the Experiment 3 cursor position heatmap revealed a centre-biased targeting pattern in which participants kept the cursor in a comfortable central position and exploited the proxemic cursor to reach peripheral widgets, demonstrating an ergonomically-driven adaptation of targeting behaviour. Despite tight vertical spacing between widget types, users reliably distinguished between the slider's and buttons' activation zones, indicating that the in-focus visual feedback and handle-based distance metric gave users sufficient confidence in boundary management.

The comparison between Pinch and Push activation gestures reinforced the findings from Chapter 3. Pinch consistently outperformed Push across all measures in both experiments, with faster acquisition, transition, and target-reaching times, as well as higher success rates and subjective preference. While the proxemic cursor did reduce the performance gap between the two gestures it did not fully compensate for the depth-axis ambiguity and Heisenberg effect inherent to the push activation mechanism.

These findings are limited by the 1:1 control-display mapping and the 27-inch display size used in both experiments. While the area cursor reduces the precision required for widget acquisition, the physical hand movement required to traverse between widgets or to manipulate sliders across their full extent remains bounded by the sensor's tracking volume. As displays become larger, the interaction space required scales accordingly, potentially pushing users towards the edges of reliable tracking. This constraint is further investigated in Chapter 5, which examines how the ratio between hand and cursor movement (i.e., the control-display gain: CDG) can be tuned in conjunction with area cursors to balance acquisition speed, steering precision, and the physical effort required for large-display interaction.

4.10.1 Chapter Contributions

This chapter makes the following contributions:

1. The proxemic cursor, an adaptation of *Bubble Cursor*-style nearest-target selection [71] to touchless interfaces containing continuous widgets. It uses a handle-based Euclidean distance metric, so that the boundaries between neighbouring activation regions shift as slider handles move, and minimal visual affordance (a subtle in-focus highlight on the nearest widget rather than a rendered bubble).

2. An empirical evaluation demonstrating that area-based targeting significantly reduces widget acquisition and transition times compared to a standard cursor for multi-widget touchless interfaces, while maintaining comparable selection accuracy. The performance benefits persist across different widget types, layouts, and widget spacings.

3. A characterisation of the adaptive targeting strategies employed with area-based targeting, showing that activation behaviour varies with widget type and layout. Users change their targeting precision to the available space, approaching widgets more closely when activation zones are narrow and gesturing from greater distances when the layout permits, whilst also adopting centre-biased cursor positions that minimise arm movement.

4. An extension of the Pinch versus Push gesture comparison from Chapter 3to multi-widget interfaces, showing that while the area cursor reduces the performance gap between the two gestures, Pinch remains the superior activation technique, and the combination of the area cursor with Pinch activation provides the fastest and most preferred interaction.

4.11 Research Questions

The outcomes of Experiments 2 and 3 provide further evidence towards answering the first research question of this thesis, and initial evidence towards the second.

> **RQ1** - How does the choice of target activation technique affect the usability of a mid-air user interface?

Chapter 3 established that pinch-based activation significantly outperformed Push and Dwell for single-widget slider control. The experiments in this chapter extend that finding to multi-widget interfaces with both continuous (slider) and discrete (button) widgets, showing that the Pinch advantage generalises to more complex interactions. Across both experiments, Pinch produced faster acquisition times, faster transitions between widgets, faster time to reach the targets, and higher task success rates than Push. This advantage was consistent with Experiment 1, reinforcing that pinch-based activation is better suited to mid-air widget activation than the depth-based push gesture, regardless of whether the interface contains a single or multiple widgets. An additional finding specific to multi-widget contexts is that Push was more vulnerable to target slips during the transition between widgets, where the forward hand motion required

for activation was more likely to cause incidental cursor displacement across widget boundaries. Pinch, by decoupling the activation gesture from the pointing hand's position, avoided this issue. These results strengthen the case made in Chapter 3 that the target activation technique has a substantial effect on usability, and completes the research question, indicating a clear choice of Pinch over Push.

> **RQ2** - How can we reduce the need for traversal and selection accuracy in a mid-air user interface across different display sizes and widget densities?

The results presented in this chapter demonstrate that a dynamic area cursor can substantially reduce the precision required for widget acquisition in mid-air interfaces. By automatically targeting the nearest widget based on cursor proximity to other widgets, it eliminates the need for precise cursor-to-widget overlap, reducing acquisition times, transition times, and perceived effort while preserving accuracy. These benefits are shown across a range of widget layouts. Users adapt their targeting strategies to exploit area where it is most beneficial, approaching more closely when widget boundaries are narrow and gesturing from further away when the layout permits. Area-based targeting appears to be an effective mechanism for reducing the precision requirements of mid-air input without altering the interface's visual layout or users' established interaction patterns. These findings motivate the subsequent chapters of this thesis for RQ2: Chapter 5 examines how control-display gain can be tuned in conjunction with area cursors to balance speed and precision, while diminishing the likelihood of leaving the optimal tracking area.

Chapter 5

Experiments 4 & 5: Control-Display Gain

5.1 Introduction

Larger displays afford more screen real estate and accommodate richer layouts, and are increasingly deployed across retail kiosks, information terminals, and museum exhibits [102, 109, 158]. Under a 1:1 mapping between hand and cursor movement, traversing the full width of a 50" display requires approximately 120 cm of lateral hand movement, which exceeds the comfortable reach of most seated or standing users and approaches or exceeds the reliable sensing range of typical hand-tracking cameras [57, 108]. Sustained movement of an unsupported arm also consumes endurance [80]: fatigue rises with both the duration and amplitude of motion and progressively degrades fine motor control [87]. A technique that performs well on a 27" display may therefore degrade substantially on a larger one, not because of any property of the technique itself but because of the physical demands the display size imposes on the user. An interaction solution is therefore needed to allow cursor techniques to scale to larger displays without sacrificing accuracy or imposing additional physical strain.

Chapter 4 showed that a dynamic area cursor improves widget acquisition and transition performance in multi-widget touchless interfaces by relaxing the spatial precision required for targeting. It does not, however, drastically reduce the physical extent of the hand movement required to manipulate a widget or to reposition the cursor toward a distant target. The conventional response to that physical-traversal problem is to increase the control-display gain (CDG) [66] so that small hand movements produce proportionally larger cursor displacements; but higher gain amplifies input noise and risks undermining the very accuracy that the area cursor is designed to support. Whether area-based targeting can compensate for the accuracy loss introduced by higher gain, and whether it scales effectively across display sizes, remain open questions. Two research questions motivate this chapter:

> **RQ2** - How can we reduce the need for traversal and selection accuracy in a mid-air user interface across different display sizes and widget densities?

RQ3 - How can we design mid-air interactions such that the required effort is reduced while interaction efficiency is increased?

This chapter presents two experiments that will help answer these questions. **Experiment 4** manipulates display size and CDG across slider and button selection tasks to establish how large the accuracy cost of higher gain is when the pointing technique is an area cursor, and therefore what range of gain levels remains usable. **Experiment 5** adds target density as a third factor to test whether the technique holds when the free space around targets is constrained.

5.1.1 Control-Display Gain

Increasing CDG allows a given hand movement to produce a proportionally larger cursor displacement (as opposed to a direct mapping) and reduces the physical effort required to traverse the display. The trade-off is that higher gain amplifies the unintentional motion inherent to freehand pointing, including sensor noise, muscle tremor [178], and the incidental drift that activation gestures induce (the Heisenberg effect introduced in Section 2.4.3), all of which can contribute to cursor instability and reduce selection accuracy [30, 129]. An appropriate CDG must therefore balance the physical ergonomics of arm movement against the precision demands of the task: i.e., high enough to reduce physical movement, without harming input performance.

CDG has been studied extensively for conventional desktop input using pointing devices (see Section 2.6.2). Two findings are consistent across the literature. First, altering CDG produces a speed-accuracy trade-off: higher gain reduces movement time but raises error rates, particularly for small or densely packed targets, while lower gain yields slower but more accurate selections [30]. Second, the 'optimal' gain is not a fixed constant but depends on the task, the input device, and the interaction context. Constant gain multipliers perform reasonably well for coarse targeting, but adaptive schemes such as pointer acceleration [29], semantic pointing [18], and velocity-dependent dynamic gain have outperformed constant gain in specific settings and could be compelling future work, though are outside the scope of this thesis.

For mid-air input, high CDG linearly amplifies hand tremor and the inherent instability of trying to steady a raised hand, and also increases any displacement caused by activation gestures (i.e., the Heisenberg effect of activation), so instability in the input signal grows more visible on screen as gain increases [129]. Ram et al. [144] found that higher gain reduced movement time in touchless steering tasks but degraded accuracy, especially under time pressure. Nancel et al. [129] showed that the most effective gain level for mid-air pointing on wall-sized displays varied with display size and task. Kota and Young [97] found similar task differences between high gain on free-hand pointing. In each, CDG worked best when matched to the display's scale and the user's physical reach.

Previous work does not consider how CDG interacts with the underlying targeting technique. Each study evaluates CDG against a standard point cursor, where any reduction in precision

caused by higher gain translates directly into a reduction in selection accuracy because users are required to make very precise selections. A dynamic area cursor potentially changes that relationship: by widening the effective target area, it can shift the Fitts' Law trade-off [54], potentially compensating for decreased pointing accuracy. This chapter examines this directly.

5.1.2 Target Density and Area Cursor Behaviour

Display size is not the only spatial factor that interacts with CDG. Target density, meaning how widgets are distributed across the available screen area, affects cursor performance in ways that are partly independent of widget size. Even when individual widgets are large, placing them closely together reduces the space a dynamic area cursor can expand into, constraining its effective activation region. This is a known limitation of the Bubble Cursor [71], whose activation radius scales to enclose the nearest target. Hertzum and Hornbæk [79] reported that the Bubble Cursor's performance advantage over a standard point cursor diminished as target density increased, the two converging when targets were packed tightly together; Mott and Wobbrock [122] reported similar effects for the Bubble Lens.

Density also interacts with CDG in a manner distinct from its interaction with target size. When widgets are packed tightly, the margin for error shrinks: a small overshoot can land the cursor on a neighbouring target, whereas sparse layouts give more targeting room [25]. Higher CDG amplifies input noise, which raises the likelihood of such slips. The behaviour of an area cursor with combined high CDG and high density is therefore not predictable from either factor alone, and characterising it requires explicitly manipulating density. Chapter 4 varied widget spacing at three scales (64, 128, and 256 px) and found that area targeting remained effective across all separations, but did so with a fixed layout of three buttons and one slider on a single display size. A more comprehensive investigation of target layout density is necessary before the technique can be fully understood. This understanding would make it possible to give empirically supported recommendations about how to use CDG to support targeting across a variety of layout configurations, e.g., from sparse public kiosks with a few menu items, to dense list-browsing or keyboard interfaces with numerous close-together targets.

Overall, CDG research has mainly focused on standard point cursors, while prior work on dynamic area cursors has used a 1:1 CDG. Experiments 4 and 5 jointly characterise this interaction. Experiment 4 isolates the interaction between CDG and display size across slider and button tasks; Experiment 5 introduces target density as an additional factor. Together, they establish the range of gain levels over which an area cursor remains usable, and therefore whether the technique can scale to larger displays without requiring proportionally larger arm movements.

5.2 Chapter Structure

Section 5.3 describes the shared experimental design used across both experiments in this chapter. Section 5.4 details the shared interaction design, including the control-display gain mapping and area cursor implementation used in each experiment. Section 5.6 and Section 5.7 discuss Experiments 4 and 5, respectively. Section 5.8 presents an overall discussion of the findings from both experiments. Conclusions are given in Section 5.9 and a revisit to the research questions discussed earlier in the introduction in Section 5.10.

5.3 Chapter Experimental Design

Two within-subjects experiments (Experiments 4 and 5) were used to evaluate the combined effect of increased CDG and area cursor targeting on mid-air input on displays of different sizes. As established in Chapter 4, area-based targeting reduces the precision required for widget acquisition. However, under the 1:1 mapping of hand to cursor movement, the larger the display, the further the user must move an unsupported arm/hand to traverse it. Continuing from Chapter 4, Experiments 4 and 5 focus on characterising how CDG interacts with display size across both continuous (slider) and discrete (button) input tasks.

Across both experiments, the primary aim was to evaluate whether higher CDG could be used in conjunction with an area cursor to avoid degrading task performance. Beyond performance metrics, these experiments also sought to characterise *how* users adapted to the combination of higher gain and area-based targeting: how closely they approached widgets before activation under different gain levels, and whether their targeting strategies changed. Both experiments use the Pinch activation gesture exclusively. Following Chapter 3 and Chapter 4, Pinch consistently outperformed Push across single and multi-widget interactions. Additionally, having a single activation gesture isolates the effects of CDG, display size, and target density without confounding from the depth-axis movement.

5.4 Interaction Design

The interaction mechanics used in Experiments 4 and 5 build directly on those established in Chapter 4 (Section 4.4), now with the introduction of CDG manipulation instead of the 1:1 mapping used in prior experiments. This section describes the shared interaction design; task-specific changes are presented in the relevant experiment sections.

5.4.1 Area Cursor

As in Chapter 3 and Chapter 4, hand translation was mapped to a 2D on-screen cursor using the centre of the palm as the tracked point, decoupling finger movement from the pointing

mechanism to maintain cursor stability during pinch gestures. The proxemic cursor behaviour, introduced in Section 4.4, was preserved; the widget nearest to the cursor was designated as *in-focus* and visually highlighted, providing feedforward about which element would respond to an activation gesture. Euclidean distance from the cursor to the nearest point on each widget's bounding box was used to determine the in-focus widget, with slider handle positions (rather than the full slider track) used for distance calculations on continuous widgets.

No clutching mechanism was implemented. Users could not disengage and reposition their hand without also moving the cursor; all cursor control was direct and continuous whenever the hand was within the sensor's tracking volume. This kept the relationship between hand and cursor movement transparent to the user and avoided introducing an additional mode-switching action that might confound the measurement of CDG effects.

5.4.2 Control Display Gain Mapping

This chapter adds the introduction of a CDG multiplier applied to the hand-to-cursor mapping. Where Chapter 3 and Chapter 4 used a fixed 1:1 ratio, the experiments in this chapter apply a constant gain multiplier g such that a hand displacement of x *mm* produces a cursor displacement of $g \cdot x$ *mm* on the display (see Figure 5.1). Four gain values were used: 1.0 (a direct 1:1 mapping, equivalent to Chapter 4), and three increased gain multipliers (1.5, 2.5, 3.5), e.g., a multiplier of 2.5 would cause 2.5 mm of cursor movement on screen for every 1 mm of hand movement in-air. A lower than 1.0 gain multiplier was not included as it would require users to make greater arm movements which would be impractical and uncomfortable with larger display sizes, requiring a clutch action (to disengage the cursor whilst repositioning the body) or swapping control between both hands to reach the display edges. This necessitates more movement, rather than optimising for less movement and exertion. Constant gain multipliers of 2.5 and 3.5 have been used in prior research in pointing ergonomics [144], 1.5 is also included to give more granularity at the lower end of the scale. A constant multiplier, rather than a velocity-dependent (e.g., pointer acceleration) or adaptive scheme, was chosen to provide a clean characterisation of the interaction between gain, area targeting, display size, and density.

5.4.3 Activation Gesture

Both experiments used the pinch gesture as the sole activation technique. Following the gesture design in Section 4.4.2, participants brought their index finger and thumb together ($\leq$ 20 mm apart) to transition from a pointing state to an activated state and separated them to release. For continuous interactions, such as sliders, activation initiated a mode switch, allowing the widget to be manipulated with corresponding hand movements; for discrete widgets, such as buttons, activation triggered a press event on gesture release.

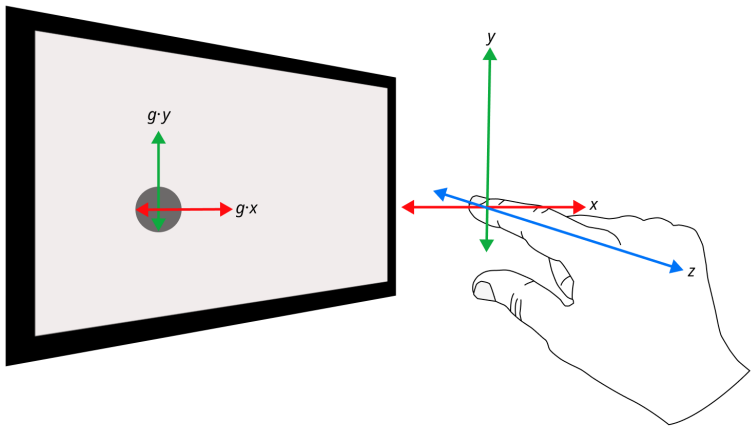


Figure 5.1: Visualisation of the interaction environment, showing the relationship between hand movement in mid-air and cursor movement on-screen for a CDG multiplier g.

5.4.4 Simulating Display Size

Both experiments manipulated display size as a factor. To investigate this without confounds from differing panel hardware (such as variation in pixel density, viewing angle), display sizes were simulated on a single 55” screen, with the surrounding screen area left blank (see Figure 5.2. Smaller display sizes (27”, 32”, and 43”) were created by scaling all interface elements proportionally to maintain a consistent index of difficulty across conditions. This approach preserved consistent pixel density, viewing distance, and physical environment across all conditions, isolating display size as a geometric variable rather than a hardware variable.



Figure 5.2: Example of the simulated display sizes. Black bars are disabled areas of the 55” display

5.4.5 Visual Feedback

The visual feedback design followed the *in-focus* and *active* visual states of Experiments 2 and 3 (Section 4.4.3). The UI was designed for the 27” display, with each element scaled proportionally for increasing the display size. The cursor appeared as a grey circle of fixed size of 30 px (27” display) during pointing, transitioning to a shrunken black circle (25 px) upon entering the active state. The in-focus widget was continuously highlighted, and the cursor

provided continuous feedback on progress toward pinch activation by shrinking as the finger-thumb gap closed. Targets were flat circles without labels or other content, and, as in the earlier experiments, no audio feedback was given.

5.5 Implementation and Apparatus

The experimental software ran full-screen on a 55" display (3840 × 2160 px, 121 × 68 cm) with the visual feedback and GUI rendered using Unity Game Engine (LTS 2021.3).

An Ultraleap Stereo IR 170 was used to track users' hands and fingers for input. The sensor's field of view is 170°x 170°and provides reliable finger tracking at short distances. Gesture detection and hand tracking were facilitated using the Ultraleap Unity SDK[1]. The sensor was placed on a 1 m high table, positioned 2.5 m in front of the display at a 15°angle (as in previous experiments). This sensor placement improved sensing quality for the experiment by keeping hands directly above the sensor at a comfortable height, though in practice such sensors are likely to be integrated as part of the display and facing towards the user. The distance from the display ensured comfortable viewing angles and is a representative interaction distance for this size of display [47, 136]. A wireless button (Microsoft Surface Dial) was positioned next to the sensor so participants could use their other hand to advance the task.

5.6 Experiment 4: Display Size x Gain

Experiment 4 investigates how CDG interacts with display size when used in conjunction with an area cursor. Chapter 4 established that area-based targeting reduces the precision required for widget acquisition, but did so under a fixed 1:1 mapping on a single 27" display. As display size increases, the physical hand movement required to traverse the interface scales correspondingly, raising concerns about both fatigue and the possibility of users moving beyond the sensor's reliable tracking volume (Section 3.6.2 Section 4.9). Increasing the CDG reduces this physical demand, but conventionally introduces a precision cost; the central question of this experiment is how large that cost is when the pointing technique is a proxemic cursor, and how it varies across display sizes.

A secondary aim was to characterise this interaction across two fundamentally different input tasks. Slider manipulation represents a continuous, precision-demanding interaction in which the user must maintain control over a widget while making fine positional adjustments; button selection represents a discrete, ballistic targeting interaction in which the user must rapidly acquire and confirm a target. These two task types are expected to be differentially affected by CDG: continuous control is sensitive to amplified input noise, whereas discrete targeting is more sensitive to acquisition speed. Including both task types provides insight into whether a

[1] https://docs.ultraleap.com/xr-and-tabletop/xr/unity/index.html - Last Accessed March 2026



Figure 5.3: Hardware set-up used for both Experiment 4 and 5: a 55-inch display at head height, with an Ultraleap Stereo IR 170 and wireless button positioned on a table in front of the user.

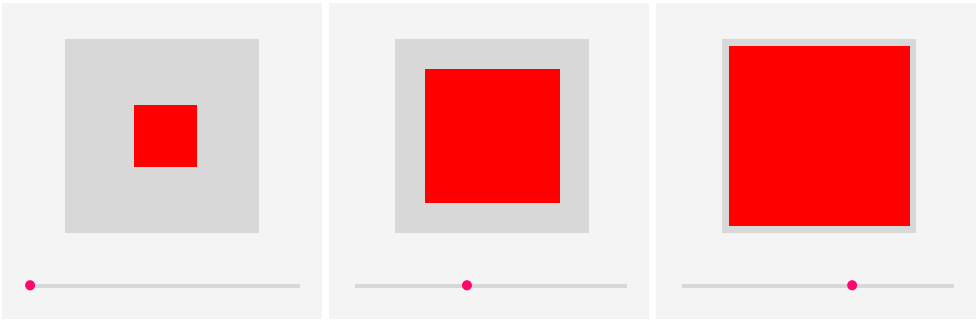
single CDG value is appropriate across interaction modes, or whether different gain levels may be needed to suit different kinds of input operation.

5.6.1 Experimental Design

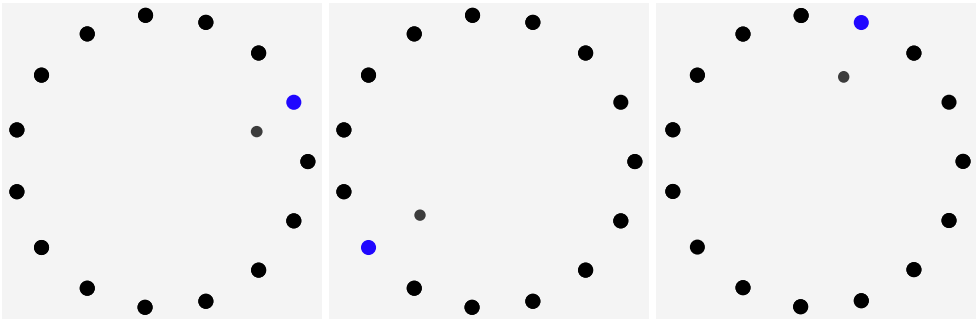
A within-subjects experimental design was used with two independent variables: CDG (1.0, 1.5, 2.5, 3.5) and Display Size (27”, 32”, 43”, 55”), resulting in 16 conditions. Condition order was counterbalanced within each task using a partial Latin square on each condition. Participants completed all conditions for one task before moving to the other, with task order itself counterbalanced across participants. Following both tasks the short semi-structured interview was conducted.

5.6.2 Task Design

Two tasks were used to investigate the impact of increased CDG and display size on input performance:



(a) Slider Task



(b) Button Task

Figure 5.4: Examples of both tasks used in Experiment 4: (a) a slider task that involved matching the size of a target square by moving the slider handle; and (b) a selection task comprising 15 sequential target selections.

Slider Task

The slider task followed the task design of the *scale* task from Experiment 1 (Section 3.4.3). Here, participants adjusted a horizontal slider to match the size of a red square to a grey reference square behind it (Figure 5.4 (a)). The slider's value was mapped linearly to the size of the user-controlled square, such that moving the handle towards the right edge increased the square's size. Participants were instructed to *"Match the size of the red square to the grey square"* as displayed above the task on the display. There were no other widgets on screen during this task, meaning the proxemic cursor would acquire the slider handle from anywhere within the interaction space, isolating the effect of increased CDG on the steering phase rather than the acquisition phase.

Each trial began with the slider handle at the leftmost position. Target square sizes were drawn from a fixed set, randomised per participant. To ensure meaningful traversal distances, target values were positioned at least 5 cm from the start position along the slider bar. The selection was marked with a press of the wireless button using the non-gesturing hand, to reduce the possibility of a confirmation or mode-switch action resulting in a change of cursor position. Participants were permitted to release and re-acquire the slider handle as needed rather than being limited to a single continuous movement.

For Task 1, the following were measured:

- **Task Time:** starting when the hand crossed a threshold above the sensor and continuing until they pressed a physical button on the table with their non-gesturing hand, ensuring consistent timing for the start and end of the task;
- **Time to Target:** the duration between taking control of the slider handle and first reaching the target position (or task end time, if the user stops short of the target without ever fully reaching it);
- **Error Distance:** the pixel difference between final and target slider position;
- **Proximity:** the distance between the slider handle and cursor position when the user performed the pinch gesture.
- **Perceived Fatigue:** fatigue felt following each set of trials, measured using the Borg CR10 scale [21] (Appendix D). The CR10 is a category-ratio scale anchoring numeric responses to verbal descriptors (0 = "nothing at all", 0.5 = "extremely weak", 3 = "moderate", 10 = "extremely strong"), so ratings are comparable across conditions and participants. It was chosen because it is validated for localised muscular exertion rather than whole-body effort, making it appropriate for arm and shoulder load in mid-air pointing; because it is the measure against which Consumed Endurance was validated [80], situating these results against that literature; and because a single item can be administered after each of the 32 conditions without extending the session. The same instrument was used in Experiment 4.(Appendix C).

The *task time* measure includes all phases of a single task, from targeting and taking control of the slider handle, to any final adjustments. *Time to target* was measured separately as this may be indicative of initial ballistic movement [119] towards the target, without including time for more subtle adjustment.

Button Task

The button task used the multi-directional selection layout from ISO 9241-411, in which 15 circular targets were arranged equally spaced around the circumference of a circle (Figure 5.4 (b)). Participants made a sequence of 15 button selections per trial, with the next target in sequence highlighted in blue. Two layout variants were used, varying target width (24 mm, 32 mm) and amplitude (222 mm, 318 mm). All values scaled proportionally with display size to maintain a consistent index of difficulty across conditions.

Participants were instructed to *"Select the highlighted target as quick and precise as possible"* with the proxemic cursor and confirmed selection using the pinch gesture. The next target was highlighted immediately after activation of the previous one, allowing continuous transitions between selections without breaks. The trial ended automatically after the 15th selection.

For Task 2, the following were measured:

- **Task Time:** the total duration between the first to the final target selections;
- **Selection Time:** the duration between completing selection N_{-1} and selection N;
- **Error Rate:** the ratio of incorrect selections;
- **Proximity:** the distance between the target and the cursor position when the user performed the pinch gesture;

5.6.3 Procedure

Before beginning the trials, participants provided informed consent and were introduced to the proxemic cursor, the pinch gesture, and the concept of CDG mapping through practice tasks. Participants completed a minimum of three minutes of practice per task type, with no upper limit, and were free to continue beyond this until they felt comfortable with each interaction. The practice phase used a single CDG value (1.0) and a single display size (27”) so that participants familiarised themselves with the interaction mechanics without exposure to all experimental conditions in advance.

Each trial began with the participant’s non-dominant hand resting on a wireless button placed on the table, pressing to to begin each trial. Participants completed 12 slider trials per condition (192 trials total) and one sequence of 15 button selections per target width × amplitude combination per condition (960 selections total). The full session lasted approximately 90 minutes, including practice and breaks.

5.6.4 Results

21 participants were recruited (10 male, 10 female, 1 other; mean age 29 years, SD 6.4 years; 18 right-handed, 3 ambidextrous) through university mailing lists and poster advertising. Participants were compensated with £15 for taking part in the study, which lasted 90 minutes. Ethics approval was provided by the College of Science and Engineering Ethics Committee.

Analysis was conducted using repeated-measures ANOVA, followed by post-hoc t-test comparisons of estimated marginal means. If data did not have a normal distribution (via the Shapiro-Wilk test), then the Aligned-Rank Transform [180] was applied first, followed by post-hoc comparisons using ART-C [49] where appropriate.

Below are summary statistics and visualisations for each dependent variable. Table 5.2 & Table 5.4 shows the results and analyses across all conditions for Task 1 and Task 2 respectively, including summary statistics for each dependent variable, and results from the repeated-measures ANOVA and post-hoc comparisons.

Task 1 - Slider Manipulations

Task Time

Overall, participants took an average of 6390 ms (SD = 2799 ms) per trial (See Table 5.2 rows 1-3 and Figure 5.5). Across display Sizes, the 27" display produced the fastest mean completion time (M = 6009, SD = 2633) ms, while the 43" display was the slowest (M = 6724, SD = 2709) ms, with the 32" (M = 6173, SD = 2891) ms and 55" (M = 6652, SD = 2939) ms displays falling in between. A CDG multiplier of 1.5 produced the shortest task times (M = 5874, SD = 2459) ms, whereas CDG = 3.5 was associated with the longest (M = 6763, SD = 3195) ms, with CDG = 1 (M = 6486, SD = 2910) ms and CDG = 2.5 (M = 6435, SD = 2551) ms falling in between.

For the Display Size $\times$ CDG interaction, the effect of CDG multiplier was more pronounced on larger displays. On the 43" and 55" screens, task times increased substantially at higher CDG values, with the 55" display at CDG = 3.5 being the slowest overall combination (M = 7344, SD = 3879) ms, and the 43" display at CDG = 2.5 (M = 7159, SD = 2652) ms and CDG = 3.5 (M = 7084, SD = 3091) ms also among the slowest. In contrast, smaller screens remained comparatively stable across CDG values, with the 32" display at CDG = 2.5 (M = 5793, SD = 2573) ms performing comparably to the fastest conditions. The fastest combination overall was the 27" display at CDG = 1.5 (M = 5457, SD = 2304) ms, suggesting that lower gain multipliers benefit smaller screens in particular.

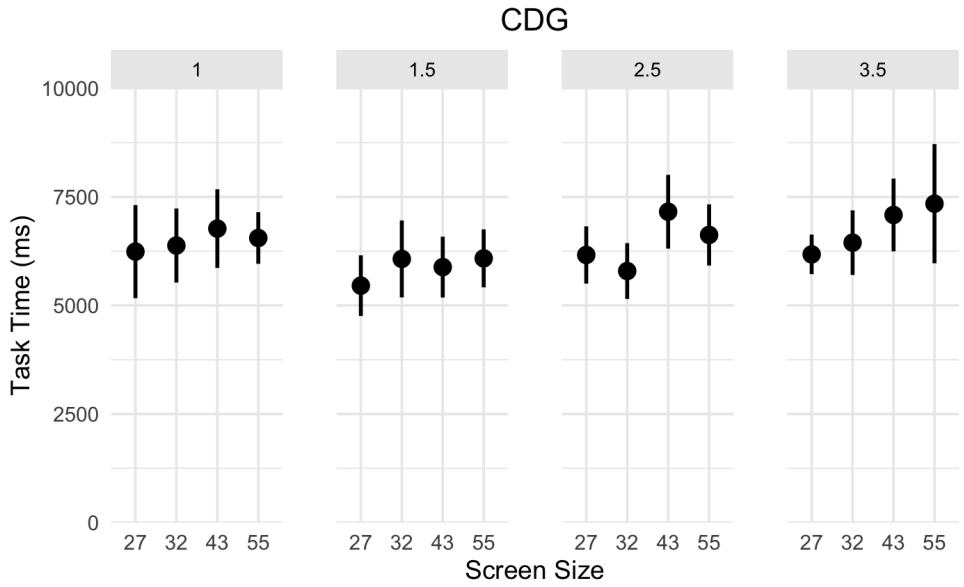


Figure 5.5: Mean task time per condition. Error bars show 95% CIs.

Time to Target

Overall, participants took an average of 1796 ms (SD = 1080 ms) to reach the target per trial (See Table 5.2 rows 4-6 and Figure 5.6). Across display Sizes, the 27" display produced the fastest mean slider time to target (M = 1592, SD = 967) ms, while the 55" display was the slowest (M =

2089, SD = 1150) ms, with the 32" (M = 1663, SD = 1004) ms and 43" (M = 1839, SD = 1139) ms displays falling in between. Higher CDG multipliers were associated with shorter times to reach the target, with CDG = 3.5 producing the fastest times (M = 1582, SD = 980) ms and CDG = 1 the slowest (M = 2122, SD = 1325) ms, with CDG = 2.5 (M = 1690, SD = 887) ms and CDG = 1.5 (M = 1788, SD = 1023) ms falling in between.

For the Display Size × CDG interaction, the benefit of higher CDG multipliers was most apparent on smaller screens. The 27" and 32" displays showed a clear improvement with increasing CDG, with the 27" at CDG = 3.5 being the fastest overall combination (M = 1344, SD = 830) ms and the 32" at CDG = 2.5 (M = 1416, SD = 719) ms also among the fastest. In contrast, the 55" display remained comparatively slow across all CDG values, ranging from (M = 2317, SD = 1358) ms at CDG = 1 to (M = 1994, SD = 939) ms at CDG = 2.5, with little improvement even at higher multipliers. The slowest combination overall was the 55" display at CDG = 1 (M = 2317, SD = 1358) ms.

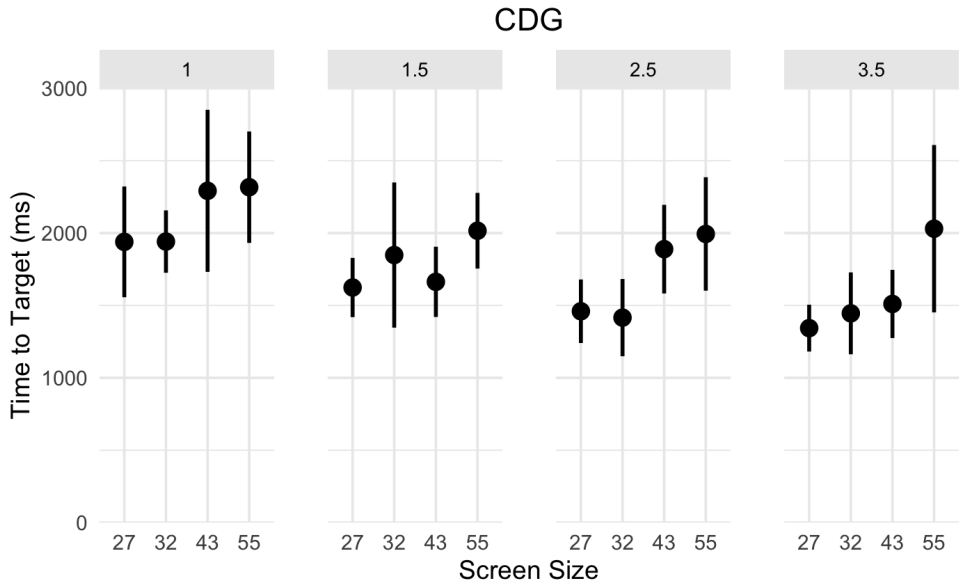


Figure 5.6: Mean time to reach the target per condition. Error bars show 95% CIs.

Slider Error Distance

Overall, participants produced a mean error distance of 3.67 (SD = 2.72) mm per trial (see Table 5.2 rows 7-9 and Figure 5.7). Across display Sizes, the 32" display produced the lowest mean error (M = 3.31, SD = 2.39) mm, while the 43" display was the least accurate (M = 4.07, SD = 2.48) mm, with the 27" (M = 3.47, SD = 2.60) mm and 55" (M = 3.81, SD = 3.30) mm displays falling in between. Higher CDG multipliers were associated with greater error, with CDG = 3.5 producing the largest error distances (M = 4.75, SD = 2.99) mm and CDG = 1.5 the smallest (M = 3.00, SD = 1.60) mm, with CDG = 1 (M = 3.17, SD = 3.34) mm and CDG = 2.5 (M = 3.74, SD = 2.31) mm falling in between.

For the Display Size × CDG interaction, the effect of increasing CDG on error was most pronounced on larger screens. The 43" display showed an increase in error at higher CDG values, peaking at CDG = 3.5 as the worst overall combination (M = 5.46, SD = 2.54) mm, while also showing elevated error at CDG = 2.5 (M = 4.40, SD = 2.42) mm. In contrast, the 32" display remained comparatively accurate across lower CDG values, with the 32" at CDG = 1 being the most accurate combination overall (M = 2.32, SD = 1.20) mm. Notably, at CDG = 1.5, error distances were relatively consistent across all display Sizes, ranging from (M = 2.84, SD = 1.33) mm on the 32" to (M = 3.22, SD = 2.06) mm on the 55".

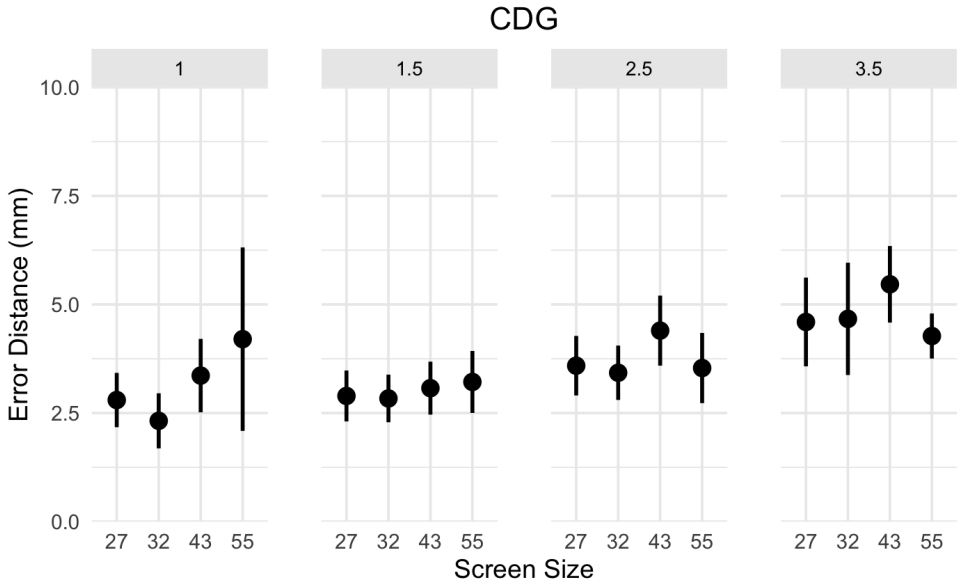


Figure 5.7: Mean error distance per condition. Error bars show 95% CIs.

Hand Movement Distance

Participants moved their hand an average distance of of 198.25 (SD = 109.15) mm (see Table 5.2 rows 10-12, Table 5.1, and Figure 5.8). Across display Sizes, the 27" display required the least hand movement (M = 137.01, SD = 63.06), while the 55" display required the most (M = 269.78, SD = 133.11), with the 32" (M = 164.68, SD = 71.68) and 43" (M = 221.54, SD = 102.96) displays falling in between, reflecting the larger physical extent of bigger screens. Higher CDG multipliers were associated with substantially reduced hand movement, with CDG = 3.5 producing the shortest distances (M = 110.92, SD = 49.74) and CDG = 1 the longest (M = 309.58, SD = 109.94), with CDG = 1.5 (M = 226.71, SD = 81.11) and CDG = 2.5 (M = 145.81, SD = 55.53) falling in between.

For the Display Size × CDG interaction, both factors had influence on hand movement distance. The largest screen paired with the lowest CDG multiplier produced the greatest hand movement overall (55" at CDG = 1: M = 429.64, SD = 108.11), while the smallest screen at the highest multiplier required the least (27" at CDG = 3.5: M = 77.32, SD = 29.87). Notably,

the effect of display Size remained apparent even at higher CDG values, with the 55" display at CDG = 3.5 (M = 151.58, SD = 56.77) still requiring nearly double the hand movement of the 27" display at the same multiplier (M = 77.32, SD = 29.87).

Table 5.1: Percentage decrease in mean hand movement distance across CDG multipliers and screen sizes.

Screen Size	**CDG 1 → 1.5**	**CDG 1.5 → 2.5**	**CDG 2.5 → 3.5**	**CDG 1 → 3.5**
27"	25.5%	32.0%	27.0%	63.0%
32"	28.4%	36.9%	23.9%	65.6%
43"	23.5%	35.8%	24.8%	63.1%
55"	29.1%	36.7%	21.5%	64.7%
Overall	**26.8%**	**35.7%**	**23.9%**	**64.2%**

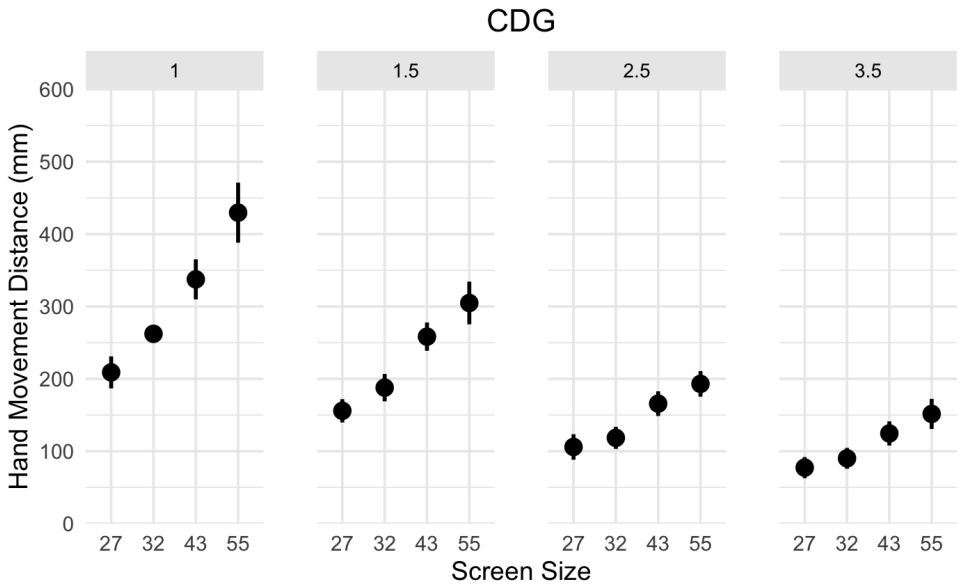


Figure 5.8: Mean time to reach the target per condition. Error bars show 95% CIs.

Proximity

Overall, participants activated the slider distance of 233.16 (SD = 161.57) mm using the area cursor (see Table 5.2 rows 13-15 and Figure 5.9). Across display Sizes, the 32" display produced the shortest activation distance (M = 105.59, SD = 45.40) mm, while the 55" display required the greatest (M = 453.15, SD = 119.20) mm, with the 27" (M = 121.54, SD = 50.67) mm and 43" (M = 252.36, SD = 89.99) mm displays falling in between. Display size on activation distance was far more pronounced than for any other dependent variable, with the 55" display requiring nearly four times the activation distance of the 32" display. Higher CDG multipliers were associated with a modest increase in activation distance, with CDG = 3.5 producing the largest distances

(M = 261.28, SD = 175.89) mm and CDG = 1 the smallest (M = 215.35, SD = 151.28) mm, though the differences across CDG levels were comparatively small.

For the Display Size × CDG interaction, display Size dominated the effect, with CDG playing a relatively minor role. The 55" display consistently produced the largest activation distances across all CDG values, ranging from (M = 441.77, SD = 93.16) mm at CDG = 1 to (M = 486.56, SD = 145.68) mm at CDG = 3.5, with little variation between multipliers. In contrast, the 27" and 32" displays remained consistently low across all CDG levels, with the 32" at CDG = 1.5 being the smallest combination overall (M = 83.89, SD = 35.23) mm. The 43" display showed a more notable increase with higher CDG values, rising from (M = 193.60, SD = 66.15) mm at CDG = 1 to (M = 304.94, SD = 92.48) mm at CDG = 3.5.

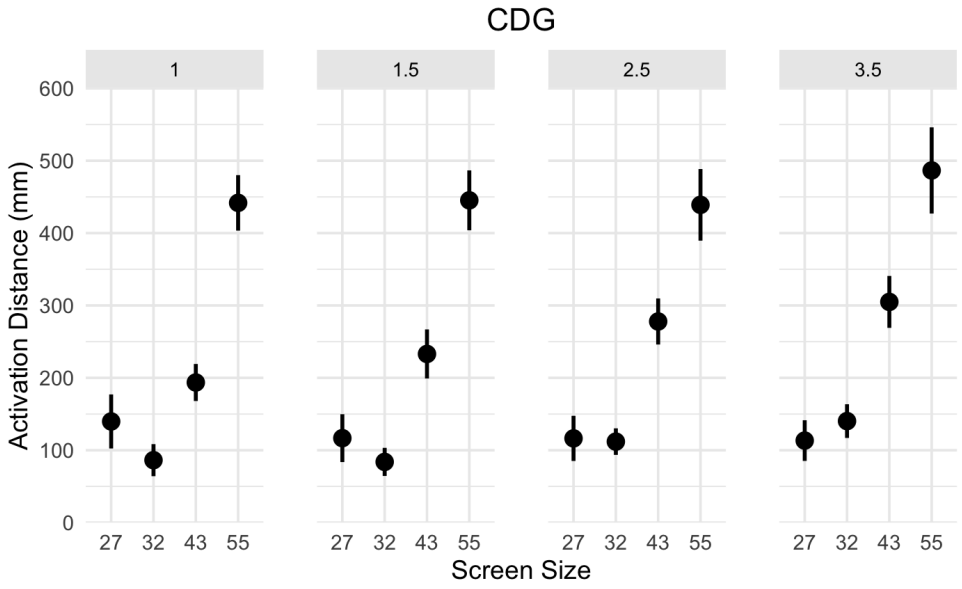


Figure 5.9: Mean slider handle activation distance per condition. Error bars show 95% CIs.

Task Success

Participants achieved a mean task success rate of 90% (SD = 15%) per trial (see Table 5.2 rows 16-18 and Figure 5.10). Across display Sizes, differences were relatively modest, with the 32" display producing the highest success rate (M = 92, SD = 15) % and the 43" the lowest (M = 88, SD = 15) %, with the 27" (M = 91, SD = 17) % and 55" (M = 90, SD = 15) % displays falling in between. In contrast, CDG had a more pronounced effect on success rate, with CDG = 1.5 producing the highest rate (M = 95, SD = 10) % and CDG = 3.5 the lowest (M = 83, SD = 19) %, with CDG = 1 (M = 94, SD = 14) % and CDG = 2.5 (M = 90, SD = 16) %.

For the Display Size × CDG interaction effect, the decline in success rate at higher CDG values was broadly consistent across display Sizes, though with some variation. At lower CDG values, all display Sizes performed well, with the 32" at CDG = 1 being the most successful combination overall (M = 98, SD = 5) %. As CDG increased to 3.5, success rates dropped

across all display Sizes, with the 43" display showing the steepest decline and producing the lowest combination overall (M = 81, SD = 16) %. Notably, at CDG = 1.5 all display Sizes achieved their highest or near-highest success rates, ranging from (M = 93, SD = 13) % on the 55" to (M = 97, SD = 7) % on 32".

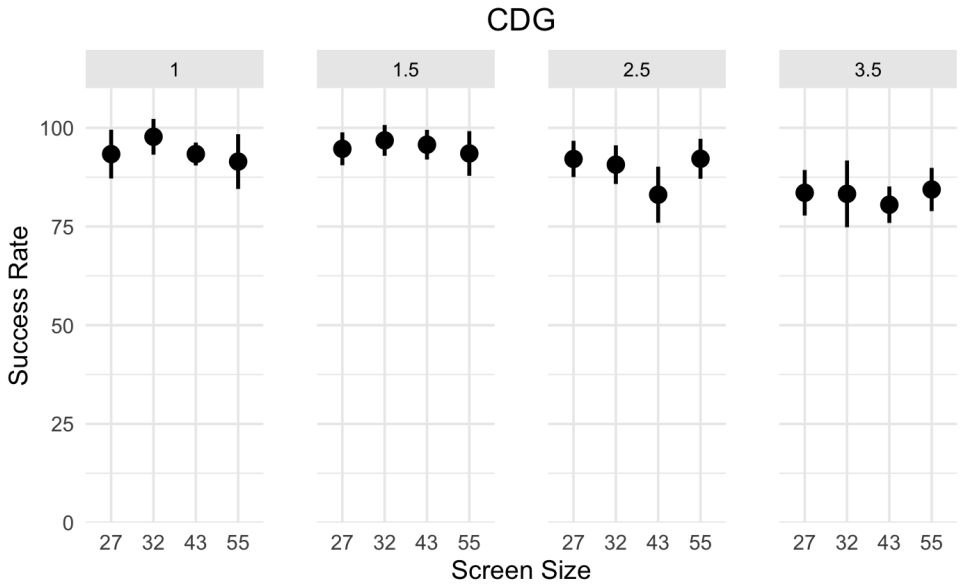


Figure 5.10: Mean success rate per condition. Error bars show 95% CIs.

Perceived Fatigue

Participants reported a mean fatigue rating of 1.62 (SD = 1.35) on the Borg CR10 scale (see Table 5.2 rows 19-21and Figure 5.11). Across display Sizes, differences were modest, with the 43" display producing the highest fatigue rating (M = 1.80, SD = 1.56) and the 55" the lowest (M = 1.47, SD = 1.25), with the 27" (M = 1.63, SD = 1.31) and 32" (M = 1.58, SD = 1.27) displays falling in between. CDG had a comparable effect on fatigue, with CDG = 3.5 producing the highest rating (M = 1.82, SD = 1.48) and CDG = 1.5 the lowest (M = 1.39, SD = 1.23), with CDG = 1 (M = 1.64, SD = 1.40) and CDG = 2.5 (M = 1.64, SD = 1.28) falling in between.

For the Display Size $\times$ CDG interaction effect, the rise in fatigue at higher CDG values was broadly consistent across display Sizes, though with some variation. At CDG = 1.5, all display Sizes produced their lowest or near-lowest fatigue ratings, with the 27" at CDG = 1.5 being the least fatiguing combination overall (M = 1.29, SD = 0.98). As CDG increased to 3.5, fatigue ratings rose across all display Sizes, with the 43" display producing the most fatiguing combination overall (M = 1.98, SD = 1.62). Notably, the 55" display maintained relatively low fatigue across all CDG levels, ranging from (M = 1.31, SD = 1.05) at CDG = 1.5 to (M = 1.69, SD = 1.53) at CDG = 3.5, while the 43" display showed consistently higher fatigue ratings across the CDG range.

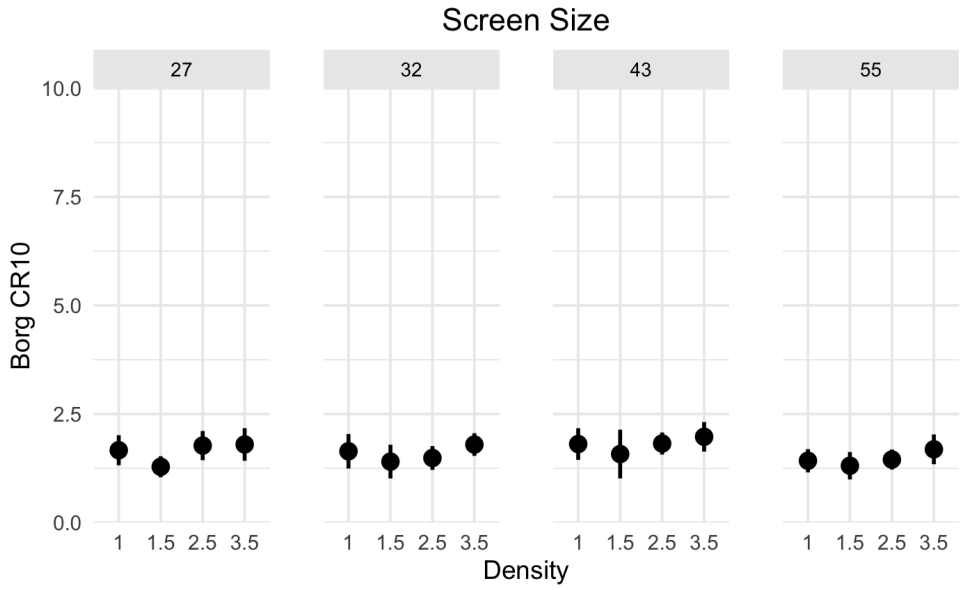


Figure 5.11: Borg CR10 Perceived fatigue scale per condition. Error bars show 95% CIs.

Table 5.2: ANOVA results, and post-hoc test results for task 1. Significant comparisons are highlighted*

	Task Time	**Mean 6390 ms, SD 2799 ms, 95% CI [6082, 6697] ms**			
	ANOVA	*Result*	p	η_p^2	*Post-Hoc Comparisons*
1	*Display Size*	F(3, 285) = 4.93	.001*	.05	• 27" < 43" (p= .01) • 32" < 43" (p= .02)
2	*CDG*	F(3, 285) = 3.89	.01*	.04	• 1.5 < 2.5 (p.03) • 1.5 < 3.5 (p.01)
3	*Display* × *CDG*	F(9, 285) = 0.52	.86	.02	
	Time to Target	**Mean 1795 ms, SD 1080 ms, 95% CI [1677, 1914] ms**			
4	*Display Size*	F(3, 285) = 11.99	< .001*	.11	• 55" > all ($p\leq$ .04) • 43" > 27" (p= .02)
5	*CDG*	F(3, 285) = 8.52	< .001*	.08	• 1 > 2.5 & 3.5 ($p\leq$ .03) • 1.5 > 3.5 (p= .02)
6	*Display* × *CDG*	F(9, 285) = 0.43	.92	.01	
	Error Distance	**Mean 3.67 mm, SD 2.72 mm, 95% CI [3.37, 3.97] mm**			
7	*Display Size*	F(3, 285) = 6.64	< .001*	.07	• 27" < 32" & 43" ($p\leq$ .007) • 43" < 55" (p.03)
8	*CDG*	F(3, 285) = 23.41	< .001*	.20	• 1 & 1.5 < all ($p\leq$ .02)

Continued on next page

Table 5.2: (continued)

	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
9	*Display* × *CDG*	F(9, 285) = 0.98	.45	.03	
	Hand Distance	**Mean 198.25 mm, SD 109.15 mm, 95% CI [186.25, 210.26] mm**			
10	*Display Size*	F(3, 285) = 212.64	< .001*	.69	• larger > smaller ($p \leq .001$)
11	*CDG*	F(3, 285) = 356.13	< .001*	.79	• larger < smaller ($p \leq .001$)
12	*Display* × *CDG*	F(9, 285) = 11.05	< .001*	.26	• smaller < larger at 1.0 ($p \leq .03$) • smaller > larger at 1.0 ($p \leq .04$) • lower < higher at 27" ($p \leq .009$) • lower > higher at 55" ($p \leq .04$)
	Proximity	**Mean 233.16 mm, SD 161.57 mm, 95% CI [215.39, 250.93] mm**			
13	*Display Size*	F(3, 285) = 294.22	< .001*	.76	• 43" > 27" & 32" ($p \leq .001$) • 55" > all ($p \leq .001$)
14	*CDG*	F(3, 285) = 7.68	< .001*	.07	3.5 > 1 & 1.5 ($p \leq .001$)
15	*Display* × *CDG*	F(9, 285) = 1.08	.38	.09	
	Task Success	**Mean 90 %, SD 15 %, 95% CI [89, 92] %**			
16	*Display Size*	F(3, 285) = 6.49	< .001*	.06	• 27" > 43" ($p \leq .001$) • 32" > 43" ($p = .006$) • 55" > 43" ($p = .003$)
17	*CDG*	F(3, 285) = 25.08	< .001*	.21	• 1 > 3.5 ($p \leq .0001$) • 1.5 > 3.5 ($p \leq .001$)
18	*Display* × *CDG*	F(9, 285) = 1.98	.04*	.06	• 2.5 > 1.5 at 27" ($p = .03$)
	Fatigue	**Mean 1.62, SD 1.35, 95% CI [1.47, 1.77]**			
	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
19	*CDG*	F(3, 285) = 5.57	.00*	.05	• 1.5 < 2.5 % 3.5 ($p \leq .02$)
20	*Display Size*	F(3, 285) = 2.00	.11	.02	
21	*CDG* × *Display*	F(9, 285) = 0.49	.88	.01	

Task 2 - Button Selection

Task Time

Participants completed all selections in an average time of 21177 ms (SD = 7219) ms (see Table 5.4 rows 1-3, and Figure 5.12). Across Display Sizes, task completion times were similar, with the 43" display producing the fastest times (M = 20665, SD = 6293) ms and the 27" display the slowest (M = 21622, SD = 7375) ms, and the 55" (M = 21010, SD = 7396) ms and 32" (M = 21404, SD = 7789) ms displays falling in between. Across CDG multipliers, the relationship CDG = 1.5 produced the fastest times (M = 20308, SD = 6937) ms and CDG = 3.5 the slowest (M = 22045, SD = 6899) ms, while CDG = 2.5 (M = 20845, SD = 6853) ms and CDG = 1 (M = 21524, SD = 8128) ms fell in between.

For the Display Size $\times$ CDG interaction, the highest CDG multiplier produced some of the longest task completion times across most screen sizes. The slowest condition overall was the 27" display at CDG = 3.5 (M = 22560, SD = 7221) ms, closely followed by the 32" display at CDG = 3.5 (M = 22526, SD = 6995) ms, while the fastest was the 43" display at CDG = 1.5 (M = 19514, SD = 6001) ms. The slower task times at CDG = 3.5 were consistent across the 27" (M = 22560, SD = 7221) ms, 32" (M = 22526, SD = 6995) ms, and 43" (M = 22164, SD = 5934) ms displays, but was less pronounced on the 55" display (M = 20906, SD = 7722) ms. The lowest CDG multiplier did not always result in the fastest times: the 27" display at CDG = 1 (M = 21764, SD = 8344) ms and the 55" display at CDG = 1 (M = 21822, SD = 7552) ms were among the slower times.

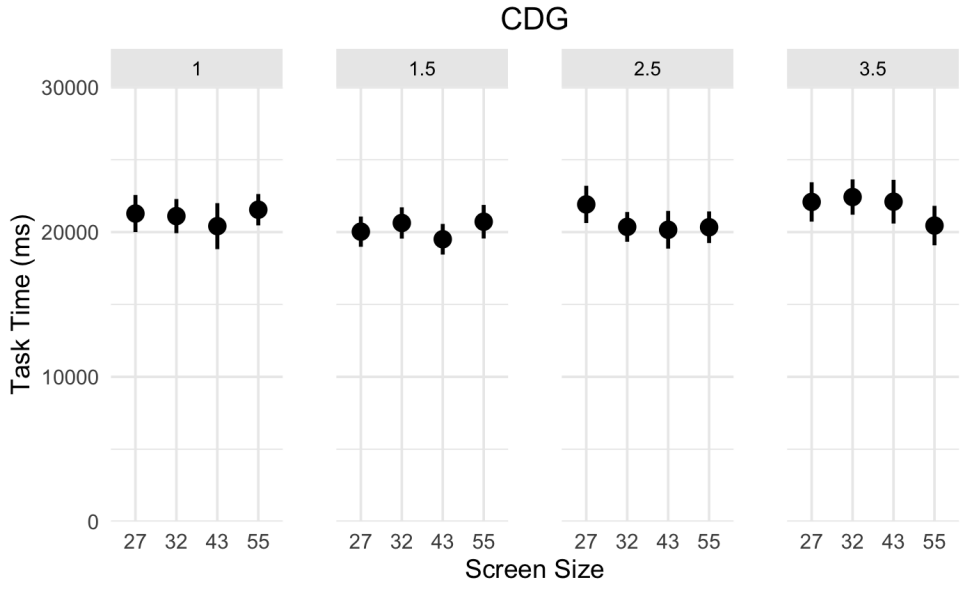


Figure 5.12: Mean total Task Time for all selections per condition. Error bars show 95% CIs.

Selection Time

Participants completed each selection in an average time of 1247 (SD = 424) ms (see Table 5.4 rows 4-6, and Figure 5.13). Across Display Sizes, selection times were broadly comparable, with the 43" display producing the fastest times (M = 1218, SD = 375) ms and the 27" display the slowest (M = 1279, SD = 435) ms, and the 55" (M = 1233, SD = 433) ms and 32" (M = 1259, SD = 455) ms displays falling in between. Across CDG multipliers, CDG = 1.5 produced the fastest times (M = 1196, SD = 404) ms and CDG = 3.5 the slowest (M = 1297, SD = 404) ms, while CDG = 2.5 (M = 1223, SD = 397) ms and CDG = 1 (M = 1274, SD = 488) ms fell in between. For the Display Size $\times$ CDG interaction, the highest CDG multiplier produced the longest completion times across most screen sizes. The slowest condition overall was the 27" display at CDG = 3.5 (M = 1345, SD = 437) ms, while the fastest was the 43" display at CDG = 1.5 (M = 1148, SD = 349) ms. Notably, the slower selection times at CDG = 3.5 were consistent across the 27" (M = 1345, SD = 437) ms, 32" (M = 1322, SD = 410) ms, and 43" (M = 1297, SD = 341) ms displays, but were less pronounced on the 55" display (M = 1224, SD = 439) ms. Conversely, the lowest CDG multiplier did not uniformly produce the fastest times: the 32" display at CDG = 1 (M = 1289, SD = 562) ms and the 55" display at CDG = 1 (M = 1288, SD = 457) ms were among the slower times.

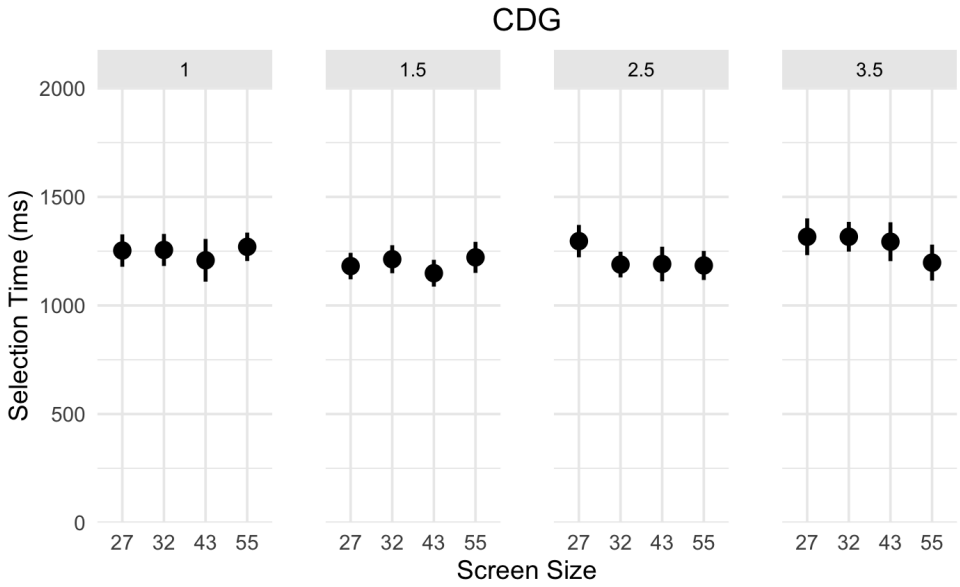


Figure 5.13: Mean selection time per condition. Error bars show 95% CIs.

Error Rate

There was an average error rate of 11.07 %(SD = 12.31) % per task (see Table 5.4 rows 7-9, and Figure 5.14). Display Size had a noticeable effect on accuracy: the 55" display yielded the lowest error rate (M = 9.08 SD = 9.79) %, whereas the 27" display incurred the highest

(M = 14.48, SD = 15.02) %, with the 32" (M = 9.54, SD = 9.33) % and 43" (M = 11.28, SD = 13.77) % falling in between. By contrast, CDG exhibited a clear relationship with error rate: accuracy degraded progressively as gain increased, rising from CDG = 1 (M = 9.07, SD = 10.94) % through CDG = 1.5 (M = 10.69, SD = 11.71) % and CDG = 2.5 (M = 11.59, SD = 12.95) %, and peaking at CDG = 3.5 (M = 12.91, SD = 13.39) %. The Display Size $\times$ CDG interaction revealed that the accuracy penalty associated with high gain was unevenly distributed across screen sizes. The poorest accuracy was observed on the 27" display at CDG = 3.5 (M = 16.75, SD = 14.80) %, whereas the best accuracy occurred on the 32" display at CDG = 1 (M = 6.88, SD = 6.81) %. Crucially, the 55" display appeared to reduce the accuracy cost of high gain values: at CDG = 3.5, the 55" display produced an error rate (M = 8.83, SD = 8.99) % that was substantially lower than that observed on the 27" (M = 16.75, SD = 14.80) %, 32" (M = 11.72, SD = 11.00) %, and 43" (M = 14.39, SD = 17.06) % displays. Similarly, at CDG = 2.5, the 55" display (M = 9.21, SD = 12.50) % outperformed both the 43" display (M = 10.67, SD = 9.75) % and the 27" display (M = 15.36, SD = 17.28) %.

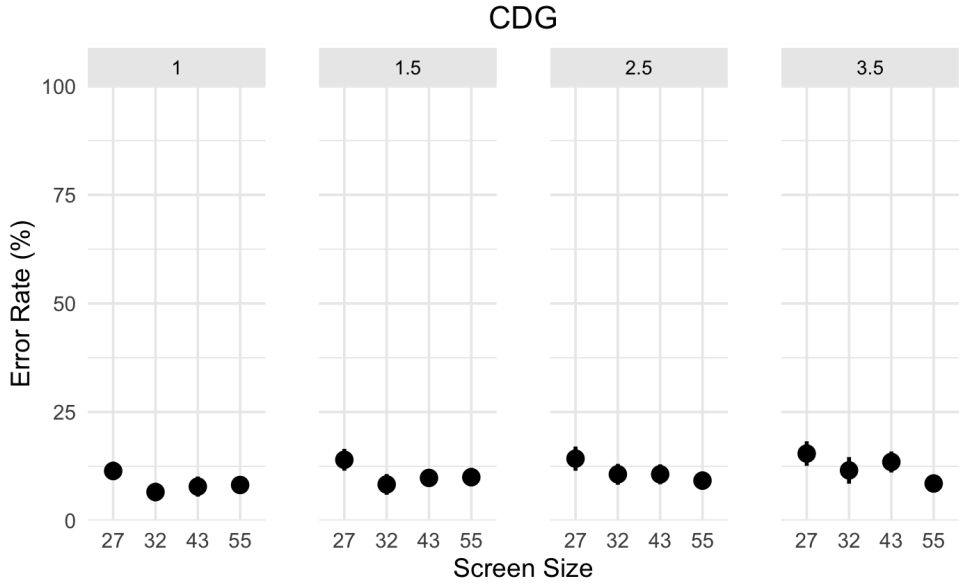


Figure 5.14: Mean error rate per task. Error bars show 95% CIs.

Proximity

Targets were selected using the area cursor at an average distance of 86 (SD = 50) mm (see Table 5.4 rows 10-12, and Figure 5.15). Display Size had the largest effect on selection distance: increasing from the 27" display (M = 62.75, SD = 31.05) through the 32" (M = 68.39, SD = 32.79) and 43" (M = 94.32, SD = 45.84) displays, and reaching its maximum on the 55" display (M = 120.21, SD = 61.06). CDG generally exerted an inverse effect on selection distance, with CDG = 1 producing the longest acquisition distances (M = 97.49, SD = 54.13), dropping to CDG = 1.5 (M = 87.61, SD = 51.88) and reaching its shortest at CDG = 2.5 (M = 78.83, SD =

46.03), before slightly increasing at CDG = 3.5 (M = 81.76, SD = 45.22). The Display Size × CDG interaction revealed that the two factors operated largely independently, with the influence of Display Size remaining dominant across all gain conditions. The greatest selection distance overall was observed on the 55" display at CDG = 1 (M = 150.72, SD = 58.71), whereas the shortest occurred on the 27" display at CDG = 1.5 (M = 59.21, SD = 27.31). The effect of CDG was substantially more pronounced on larger displays: on the 55" display, increasing CDG from 1 (M = 150.72, SD = 58.71) to 3.5 (M = 103.15, SD = 56.24) reduced selection distance by approximately 32%, while on the 27" display the same gain increase resulted in a slight increase in selection distance (from M = 61.96, SD = 27.71 at CDG = 1 to M = 65.98, SD = 35.40 at CDG = 3.5).

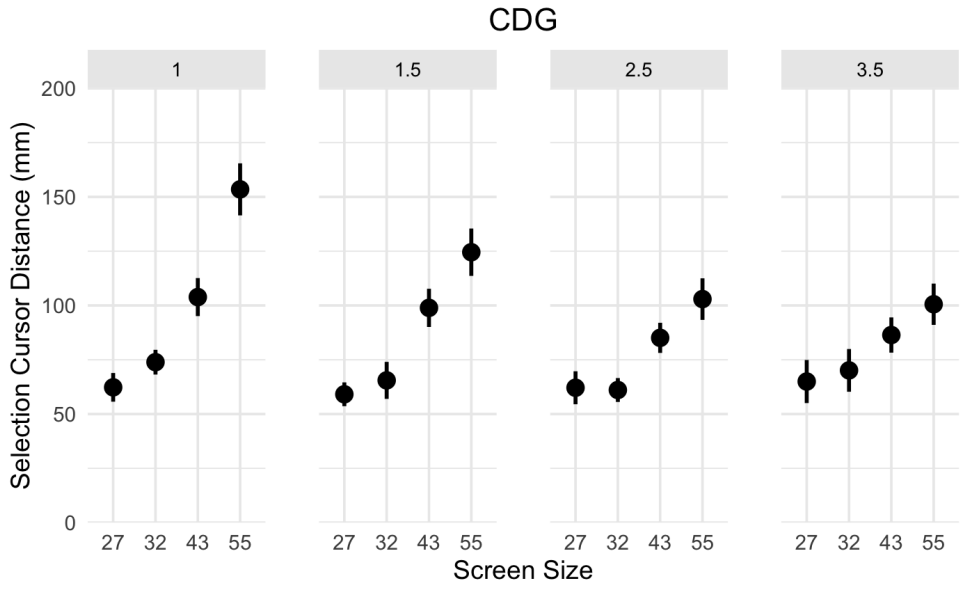


Figure 5.15: Mean Selection Distance using the area cursor per trial. Error bars show 95% CIs.

Hand Movement Distance

Participants moved their hand an average distance of 64.35 (SD = 28.45) mm during each trial (see Table 5.4 rows 13-15, and Figure 5.16). Display Size showed a clear progression: the 27" display showed the smallest mean displacement (M = 58.53, SD = 23.03) mm, followed by the 32" (M = 62.84, SD = 30.20) mm and 43" (M = 65.04, SD = 27.52) mm displays, with the 55" display producing the greatest displacement (M = 71.00, SD = 31.48) mm. CDG also had a consistent effect, where hand displacement was greatest at CDG = 1 (M = 72.93, SD = 31.78) mm and steadily decreased through CDG = 1.5 (M = 66.99, SD = 27.06) mm and CDG = 2.5 (M = 62.48, SD = 25.77) mm, reaching its lowest at CDG = 3.5 (M = 55.02, SD = 26.32) mm.

The Display Size × CDG interaction revealed that the gain-related reduction in hand displacement was unevenly distributed across display sizes. The largest distance overall occurred on the 55" display at CDG = 1 (M = 85.67, SD = 36.50) mm, while the smallest occurred on the

55" display at CDG = 3.5 (M = 51.48, SD = 19.70) mm. This sharp decrease on the 55" display contrasted with the other displays. For instance, the 32" display exhibited considerably weaker gain-related changes, with hand displacement remaining relatively stable across all four CDG levels (ranging from M = 61.53 to M = 64.50) mm. The 27" and 43" displays showed more moderate reductions across the gain levels (decreasing from M = 64.26 to M = 53.54, and M = 77.30 to M = 52.19, respectively) mm.

Table 5.3: Percentage change in mean hand distance across CDG and screen sizes. Negative values indicate an increase in hand distance.

Screen Size	**CDG 1 → 1.5**	**CDG 1.5 → 2.5**	**CDG 2.5 → 3.5**	**CDG 1 → 3.5**
27"	10.3%	−1.8%	8.7%	16.7%
32"	3.1%	1.5%	−2.1%	2.6%
43"	11.4%	9.3%	16.0%	32.5%
55"	7.5%	14.8%	23.8%	39.9%
Overall	**8.1%**	**6.7%**	**11.9%**	**24.6%**

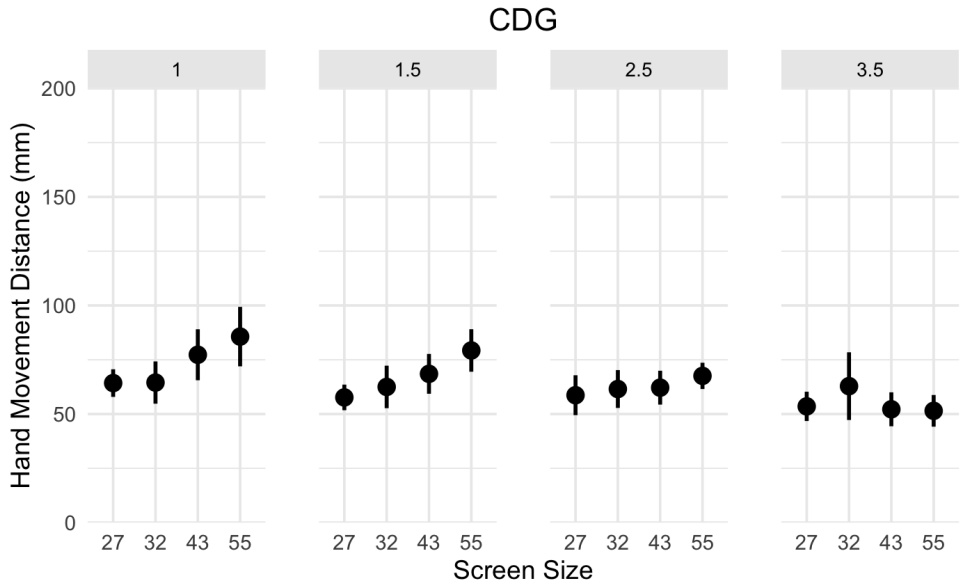


Figure 5.16: Mean hand movement distance per trial. Error bars show 95% CIs.

Perceived Fatigue

Participants reported a mean fatigue rating of 2.28 (SD = 1.88) on the Borg CR10 scale (see Table 5.4 rows 16-18 and Figure 5.17). Across display Sizes, differences were modest, with the 43" display producing the highest fatigue rating (M = 2.56, SD = 2.23) and the 32" the lowest (M = 1.98, SD = 1.55), with the 27" (M = 2.42, SD = 1.87) and 55" (M = 2.16, SD = 1.75) displays falling in between. CDG had a less pronounced effect on fatigue, with CDG = 1 producing the

highest rating (M = 2.45, SD = 2.02) and CDG = 2.5 the lowest (M = 2.16, SD = 1.91), with CDG = 1.5 (M = 2.25, SD = 1.75) and CDG = 3.5 (M = 2.26, SD = 1.84) falling in between.

For the Display Size × CDG interaction effect, fatigue patterns varied considerably across display Sizes rather than following a consistent trend with CDG. The 43" display produced relatively high fatigue ratings across all CDG levels, with the 43" at CDG = 3.5 being the most fatiguing combination overall (M = 2.76, SD = 2.44). In contrast, the 55" display showed a decreasing trend in fatigue as CDG increased, producing the lowest combination overall at CDG = 3.5 (M = 1.85, SD = 1.31). The 32" display maintained consistently low fatigue across all CDG levels, ranging from (M = 1.88, SD = 1.79) at CDG = 2.5 to (M = 2.11, SD = 1.76) at CDG = 3.5, while the 27" display peaked at CDG = 1 (M = 2.69, SD = 2.02) before declining at higher gains.

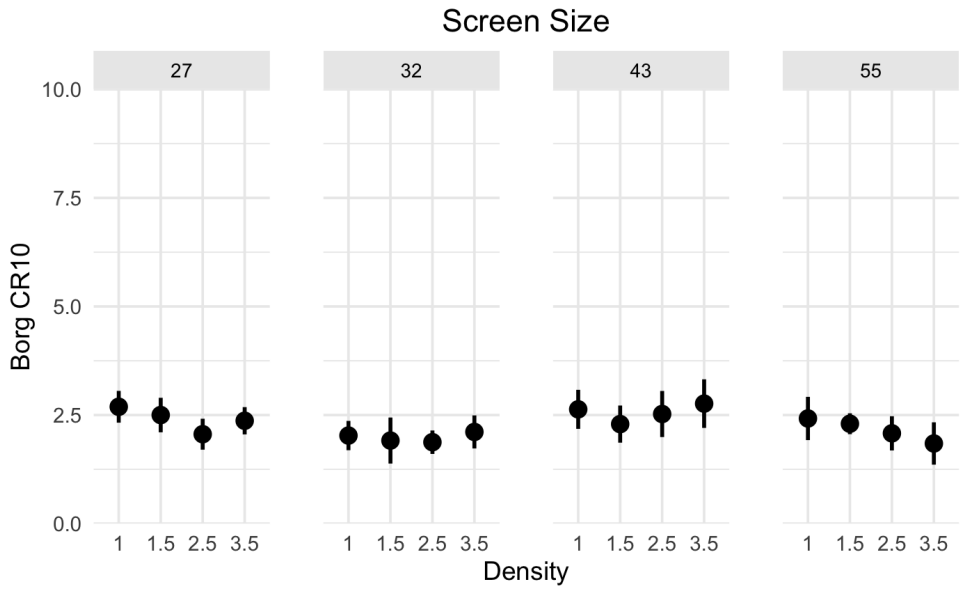


Figure 5.17: Borg CR10 Perceived fatigue scale per condition. Error bars show 95% CIs.

Table 5.4: ANOVA results, and post-hoc test results for task 2. Significant comparisons are highlighted*

	Task Time	**Mean 21177 ms, SD 7218 ms, 95% CI [20390, 21963] ms**			
	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
1	*Display Size*	F(3, 290) = 0.33	.81	.00	
2	*CDG*	F(3, 290) = 3.15	.03*	.03	• 1.5 < 3.5 (p= .01)
3	*Display × CDG*	F(9, 290) = 0.74	.67	.02	
	Selection Time	**Mean 1247 ms, SD 424 ms, 95% CI [1200, 1293] ms**			

Continued on next page

Table 5.4: (continued)

	ANOVA	*Result*	p	η_p^2	*Post-Hoc Comparisons*
4	*Display Size*	F(3, 290) = 0.46	.71	.00	
5	*CDG*	F(3, 290) = 2.93	.03*	.03	• 1.5 < 3.5 (p= .02)
6	*Display* × *CDG*	F(9, 290) = 0.76	.65	.02	
	Error	**Mean 11 %, SD 12 %, 95% CI [9.7, 12.4] %**			
7	*Display Size*	F(3, 290) = 3.81	.01*	.04	• 27” > 43” & 55” ($p\leq$ .04)
8	*CDG*	F(3, 290) = 4.39	.001*	.04	• 1 < 2.5 & 3.5 ($p\leq$ .007)
9	*Display* × *CDG*	F(9, 290) = 0.87	.56	.03	
	Proximity	**Mean 86 mm, SD 49 mm, 95% CI [81, 92] mm**			
10	*Display Size*	F(3, 290) = 55.87	< .001*	.37	• Higher for all larger displays ($p\leq$ .001) • except 27” vs 32” (p= .14)
11	*CDG*	F(3, 290) = 6.99	< .001*	.07	• 1 > 2.5 & 3.5 ($p\leq$ .003)
12	*Display* × *CDG*	F(9, 290) = 2.53	.01*	.07	• 55” > 27” ($p\leq$ .009) • 55” > 32” when gain $\in$ (1, 1.5, 2.5) ($p\leq$ .02) • 55” > 43” when gain $\in$ (1, 1.5) ($p\leq$ .03)
	Hand Movement	**Mean 64 mm, SD 28 mm, 95% CI [60, 68] mm**			
13	*Display Size*	F(3, 290) = 6.14	< .001*	.07	• 55” > 27” & 32” ($p\leq$ .01)
14	*CDG*	F(3, 290) = 12.30	< .001*	.13	• 1 > 2.5 & 3.5 ($p\leq$ .006) • 1.5 > 3.5 ($p\leq$ .001)
15	*Display* × *CDG*	F(9, 290) = 2.10	.03*	.07	No significant comparisons
	Fatigue	**Mean 2.28, SD 1.88, 95% CI [2.07, 2.49]**			
16	*CDG*	F(3, 290) = 0.87	.46	.01	
17	*Display Size*	F(3, 290) = 3.93	.01*	.04	• 43” > 55” (p= .04)
18	*CDG* × *Display*	F(9, 290) = 0.65	.75	.02	

5.7 Experiment 5: Display Size x Gain x Density

Experiment 5 investigates how CDG and area cursor targeting perform under varying levels of target density. Experiment 4 established that increased CDG can be used in conjunction with the proxemic cursor without degrading task performance, particularly on larger displays where higher gain reduces the physical hand movement required to traverse the interface. However, Experiment 4 used widget layouts in which targets were either spaced widely apart (the multi-directional button task) or presented in isolation (the slider task), leaving the effective expansion region of the area cursor unconstrained. Whilst representative of minimal mid-air interface layouts, there remain unanswered questions about how these techniques would perform in more complex designs. In some mid-air interfaces, widgets are often closely grouped together and surrounded by other interface elements, meaning the proxemic cursor must resolve the intended target from amongst nearby distractors. The main question of this experiment is whether the combination of higher CDG and area-based targeting remains effective when target density limits the space available for the area cursor to expand into.

A secondary aim was to characterise how target density interacts with display size. On smaller displays, targets are physically closer together regardless of layout, leaving less margin for cursor overshoot; on larger displays, the same density level produces greater physical separation between targets, which may give the area cursor more room to operate and give the user additional space for controlling the cursor. Including density alongside display size and CDG provides insight into whether the gain levels appropriate for sparse layouts remain appropriate as layouts become more cluttered, or whether dense layouts require a more conservative CDG to maintain selection accuracy. Since touchless interfaces are used across a wide variety of display sizes (e.g., large public displays to smaller personal devices), these insights will help to inform appropriate design choices for different application contexts.

5.7.1 Experimental Design

A within-subjects experimental design was used with three independent variables: CDG (1.0, 1.5, 2.5, 3.5), Display Size (27”, 55”), and Target Density (0.075, 0.15, 0.3, 0.6), giving 32 conditions. Only the smallest and largest display sizes from Experiment 4 were used in order to limit experimental duration and manage participant fatigue, while still capturing the effect of scale across the range tested in Experiment 4. Condition order was counterbalanced using a partial Latin square on each condition. Each participant completed 13 selections per condition, resulting in 416 selections per participant.

5.7.2 Task Design

The task used the distractor-based target acquisition paradigm of Blanch and Ortega [19], an extension of the standard Fitts' Law multi-directional selection task in which the display is populated with additional distractor targets at a specified density level. Target density also follows the definition introduced by Blanch and Ortega [19]: the ratio of occupied to available space, ranging from 0 (a single target on an empty display) to 1 (targets packed with no space between them). The four levels used here (0.075, 0.15, 0.3, 0.6) match those used in their original study, spanning from sparse layouts in which the proxemic cursor has ample space to expand to dense layouts in which surrounding distractors substantially constrain its effective reach.

Targets were arranged radially around a starting position on the left side of the display, with target size increasing with radius from the start position to maintain a consistent index of difficulty across positions. Figure 5.18 shows the four density levels used: from 0.075 (sparse) on the left to 0.6 (dense) on the right.

Each trial began with the participant moving the cursor to a starting position, indicated by a blue circle on the left edge of the display, and resting there for 500 ms. Following this, a screen with several distractors and a single highlighted target appeared. Participants were instructed to *"select the highlighted target as quickly and accurately as possible"* using the proxemic cursor and the pinch gesture. After each selection, the cursor returned to the starting position to begin the next trial. An index of difficulty of 3.0 was used throughout, based on pilot testing (n=5). Following each set of trials, participants ranked their perceived fatigue, control level and accuracy levels. After completing all trials, a small semi-structured interview was conducted.

The following metrics were measured:

- **Selection Time:** duration between activating start position and selecting one of the targets;
- **Error Rate:** the ratio of incorrect selections;
- **Selection Distance:** the distance between target and area cursor position when the user performed the pinch gesture.
- **Sensitivity:** the *index of sensitivity* [19], a measure of the impact that distractor targets had on selection.
- **Perceived Fatigue:** fatigue felt following each set of trials. Measured using the Borg CR10 scale [21]. Chosen due to its strong association and ability to predict consumed endurance [80] (Appendix D).
- **Perceived Control:** how in control the participant felt over the cursor. Measured using 1-10 scale from "not at all" to "completely in control" (Appendix D).
- **Perceived Difficulty:** how difficult the user felt it was to select targets. Measured using 1-10 scale from "not at all difficult" to "very difficult" (Appendix D).

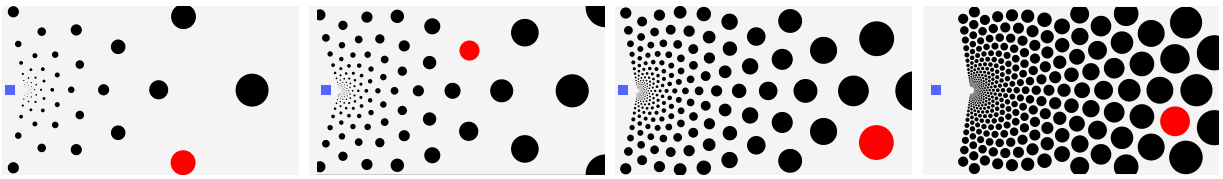


Figure 5.18: Experiment 5 task layouts ranging from density values 0.075 (left) to 0.6 (right)

5.7.3 Procedure

Before beginning the trials, participants provided informed consent and were introduced to the proxemic cursor, the pinch gesture, and the concept of CDG mapping through practice tasks. Participants completed a minimum of three minutes of practice, with no upper limit, and were free to continue beyond this until they felt comfortable with the interaction. The practice phase used a single CDG value (1.0), a single display size (27”), and a single target density (0.15) so that participants familiarised themselves with the interaction mechanics without exposure to all experimental conditions in advance.

Each trial began with the participant’s non-dominant hand resting on a wireless button placed on the table, pressing it to begin each trial. Participants completed 13 selections per condition across 32 conditions, resulting in 416 selections per participant. The full session lasted approximately 60 minutes, including practice and breaks. Upon completing all conditions, participants discussed their preferences in a semi-structured interview. All semi-structured interviews were audio recorded with consent and transcribed. Each transcript was read independently by two readers — the author and a member of the supervision team, who separately highlighted passages relevant to the research questions; the highlighted extracts were then compared and, where both readers had identified the same passage, carried forward and grouped into descriptive categories by consensus. No formal thematic analysis was undertaken and no inter-rater reliability statistic was calculated, so the quotes reported here are illustrative participant commentary that contextualises the quantitative results rather than themes generated through a formal coding process.

5.7.4 Results

18 participants were recruited (10 male, 7 female, 1 not disclosed; mean age 25 years, SD 8.7 years; 16 right-handed, 2 ambidextrous) through university mailing lists and poster advertising. Participants were compensated with £10 for taking part in the study, which lasted 90 minutes. Ethics approval was provided by the College of Science and Engineering Ethics Committee.

Analysis was conducted using repeated-measures ANOVA, followed by post-hoc t-test comparisons of estimated marginal means. If data did not have a normal distribution (via the Shapiro-Wilk test), then the Aligned-Rank Transform [180] was applied first, followed by post-hoc comparisons using ART-C [49] where appropriate.

Below are summary statistics and visualisations for each dependent variable. Table 5.6 shows the results and analyses across all conditions, including summary statistics for each dependent variable, and results from the repeated-measures ANOVA and post-hoc comparisons.

Selection Time

Overall, participants made each selection of 1679 ms ,SD = 392 ms (see Table 5.6 rows 1-7, and Figure 5.19). Mean time increased with density, from 1594 ms (SD = 414) at 0.075, to 1663 ms (SD = 395) at 0.15, 1684 ms (SD = 364) at 0.3, and 1785 ms (SD = 372) at 0.6. For CDG, the slowest value was 1.0 (M = 1767, SD = 419) ms, followed by 3.5 (M = 1687, SD = 377) ms, 1.5 (M = 1652, SD = 366) ms, and 2.5 (M = 1617, SD = 394) ms. The 27-inch display produced faster times (M = 1635, SD = 361) ms than the 55-inch display (M = 1725, SD = 417) ms.

For the Density $\times$ CDG interaction, 0.6 was slowest paired with CDG 1 (M = 1827, SD = 368) ms, while the fastest was 0.075 paired with CDG 2.5 (M = 1492, SD = 452) ms. In the Density $\times$ Display Size interaction, times on the 27-inch display increased at the highest density (M = 1794, SD = 378 ms at density 0.6), whereas on the 55-inch display the values were more closer, ranging from 1632 ms (SD = 437) at 0.075 to 1787 ms (SD = 399) at 0.3. The CDG $\times$ Display Size interaction showed the largest gap at CDG 1, where the 55-inch display (M = 1979, SD = 444) ms was substantially slower than the 27-inch display (M = 1568, SD = 274) ms; the two displays were closer at CDG 3.5 (27-inch: M = 1759, SD = 416 ms; 55-inch: M = 1620, SD = 327 ms).

In the three-way Density $\times$ CDG $\times$ Display Size interaction, the slowest condition was density 0.15 with CDG 1 on the 55-inch display (M = 2008, SD = 434) ms, closely followed by density 0.3 with CDG 1 on the 55-inch display (M = 1992, SD = 496) ms and density 0.075 with CDG 1 on the 55-inch display (M = 1968, SD = 511) ms. The fastest condition was density 0.075 with CDG 1.5 on the 27-inch display (M = 1464, SD = 290) ms, followed by density 0.075 with CDG 3.5 on the 55-inch display (M = 1479, SD = 247) ms and density 0.15 with CDG 2.5 on the 55-inch display (M = 1493, SD = 275) ms.

Error Rate

Overall, participants had an average error rate of 13.22 % SD = 13.07 % (see Table 5.6 rows 8-14, and Figure 5.20). Error rate increased with density, rising from 7.98 % (SD = 9.83 %) at 0.075, to 8.79 % (SD = 9.98 %) at 0.15, 16.62 % (SD = 12.98 %) at 0.3, and 20.18 % (SD = 14.90 %) at 0.6. Across CDG levels, errors were lowest at 1.5 (M = 11.80 %, SD = 12.30 %) and highest at 3.5 (M = 15.32 %, SD = 13.76 %), with CDG 1 (M = 12.94 %, SD = 13.22 %) and 2.5 (M = 12.86 %, SD = 12.89 %) falling between. The 27-inch display produced a higher error rate (M = 14.02 %, SD = 12.86 %) than the 55-inch display (M = 12.40 %, SD = 13.26 %).

For the density $\times$ CDG interaction, the highest error rate occurred at density 0.6 paired with CDG 3.5 (M = 25.90 %, SD = 14.49 %), while the lowest was at density 0.075 paired with CDG

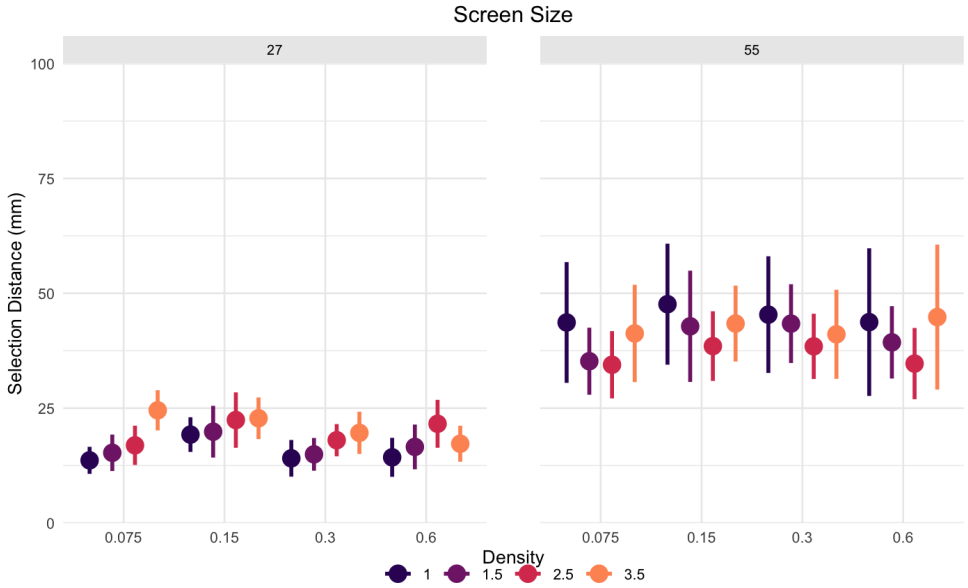


Figure 5.19: Mean Selection Time per trial. Error bars show 95% CIs.

2.5 (M = 6.79 %, SD = 7.52 %). Across all CDG levels, errors increased with density, with the steepest increase occurring at CDG 3.5 (from 9.28 % at density 0.075 to 25.90 % at density 0.6). In the density × Display Size interaction, the 27-inch display produced the highest error rate at density 0.6 (M = 21.63 %, SD = 14.56 %), while the 55-inch display showed the lowest error rate at density 0.075 (M = 5.66 %, SD = 8.68 %). The CDG × Display Size interaction revealed contrasting patterns across sizes: on the 27-inch display, errors increased with CDG from 10.06 % (SD = 10.87 %) at CDG 1 to 19.29 % (SD = 13.80 %) at CDG 3.5, whereas on the 55-inch display, errors were highest at CDG 1 (M = 16.02 %, SD = 14.82 %) and lower at CDG 2.5 (M = 10.45 %, SD = 12.75 %).

In the three-way density × CDG × Display Size interaction, the highest error rate was observed at density 0.6 with CDG 3.5 on the 27-inch display (M = 32.31 %, SD = 10.56 %), followed by density 0.6 with CDG 2.5 on the 27-inch display (M = 23.98 %, SD = 13.01 %) and density 0.3 with CDG 3.5 on the 27-inch display (M = 22.44 %, SD = 11.58 %). The lowest error rate was at density 0.075 with CDG 2.5 on the 55-inch display (M = 3.17 %, SD = 5.48 %), followed by density 0.075 with CDG 3.5 on the 55-inch display (M = 4.52 %, SD = 6.69 %) and density 0.15 with CDG 1 on the 27-inch display (M = 4.98 %, SD = 8.13 %).

Sensitivity

Overall, participants exhibited a mean sensitivity to distractor widgets of 0.18, SD = 0.15 see Table 5.6 rows 15-21, and Figure 5.21. Sensitivity increased with density, rising from 0.08 (SD = 0.03) at 0.075, to 0.10 (SD = 0.04) at 0.15, 0.17 (SD = 0.06) at 0.3, and 0.39 (SD = 0.15) at 0.6. Across CDG levels, CDG 1, 1.5, and 2.5 each showing a mean of 0.19 (SDs of 0.17, 0.14, and 0.14, respectively) and CDG 3.5 producing the lowest mean of 0.17 (SD = 0.14). The

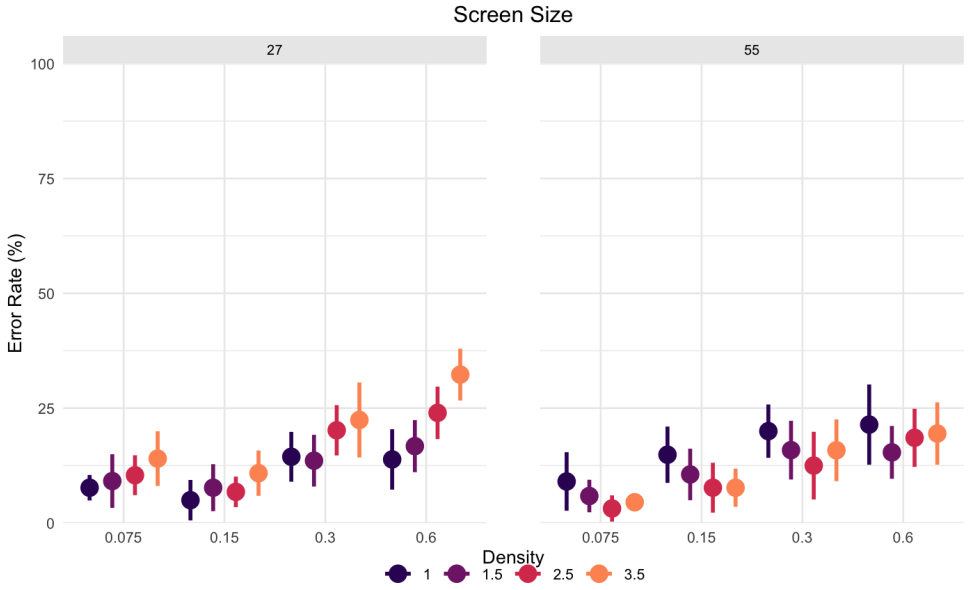


Figure 5.20: Mean Error rate per trial. Error bars show 95% CIs.

27-inch display produced higher sensitivity (M = 0.20, SD = 0.16) than the 55-inch display (M = 0.17, SD = 0.14).

For the density × CDG interaction, the highest sensitivity occurred at density 0.6 paired with CDG 1 (M = 0.42, SD = 0.19), while the lowest was at density 0.075 paired with CDG 3.5 (M = 0.07, SD = 0.03). Sensitivity rose monotonically with density across all CDG levels, with values at density 0.6 ranging from 0.38 (SD = 0.11) at CDG 3.5 to 0.42 (SD = 0.19) at CDG 1. In the density × Display Size interaction, the 27-inch display produced the highest sensitivity at density 0.6 (M = 0.41, SD = 0.15), while both displays yielded the lowest values at density 0.075 (M = 0.08, SD = 0.03 on each). The CDG × Display Size interaction revealed contrasting patterns across displays: on the 27-inch display, sensitivity decreased with increasing CDG, from 0.22 (SD = 0.18) at CDG 1 to 0.17 (SD = 0.15) at CDG 3.5, whereas on the 55-inch display, the lowest value occurred at CDG 1 (M = 0.15, SD = 0.14) and the highest at CDG 2.5 (M = 0.19, SD = 0.15).

In the three-way density × CDG × Display Size interaction, the highest sensitivity was observed at density 0.6 with CDG 1 on the 27-inch display (M = 0.48, SD = 0.19), followed by density 0.6 with CDG 3.5 on the 27-inch display (M = 0.41, SD = 0.10) and density 0.6 with CDG 2.5 on the 55-inch display (M = 0.41, SD = 0.12). The lowest sensitivity was at density 0.075 with CDG 3.5 on the 27-inch display (M = 0.06, SD = 0.02), followed by density 0.075 with CDG 1 on the 55-inch display (M = 0.06, SD = 0.03) and density 0.075 with CDG 1.5 on the 55-inch display (M = 0.07, SD = 0.03).

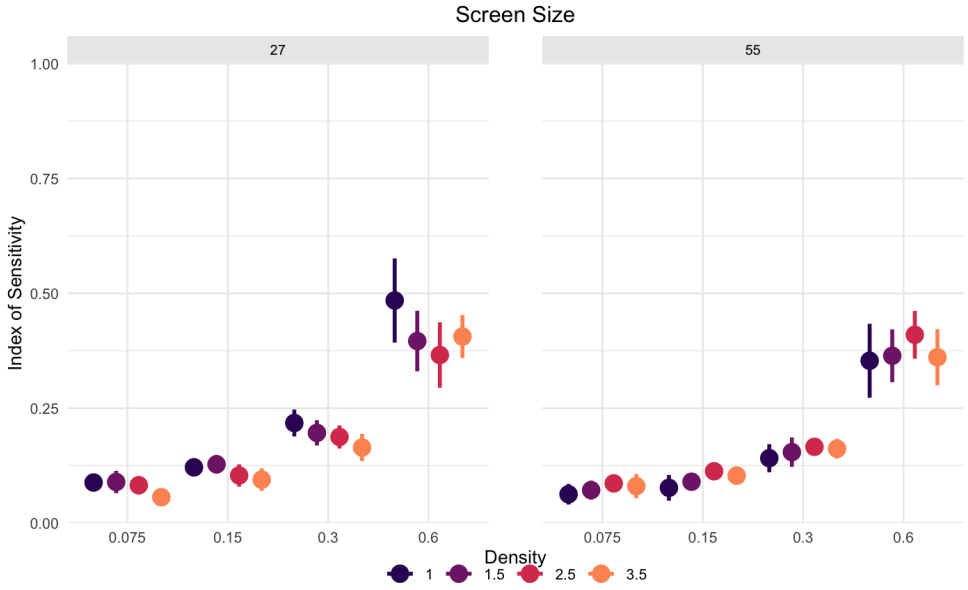


Figure 5.21: Mean Sensitivity to distractors widgets per trial. Error bars show 95% CIs.

Proximity

Participants had a mean selection distance of 29 mm, SD = 19 mm using the area cursor (see Table 5.6 rows 22-28, and Figure 5.22). Selection distance varied modestly with density, with values of 28 mm (SD = 18 mm) at 0.075, 32 mm (SD = 19 mm) at 0.15, 30 mm (SD = 20 mm) at 0.3, and 29 mm (SD = 22 mm) at 0.6. Across CDG levels, the largest distance occurred at CDG 3.5 (M = 32 mm, SD = 20 mm), while CDG 1 (M = 29 mm, SD = 24 mm), 1.5 (M = 28 mm, SD = 19 mm), and 2.5 (M = 28 mm, SD = 16 mm) were similar. The 27-inch display produced markedly shorter distances (M = 18 mm, SD = 8 mm) than the 55-inch display (M = 41 mm, SD = 22 mm).

For the density × CDG interaction, the largest selection distance occurred at density 0.15 paired with CDG 1 (M = 33 mm, SD = 23 mm), while the smallest was at density 0.075 paired with CDG 1.5 (M = 25 mm, SD = 15 mm). Across CDG 3.5, distances remained relatively high across all densities, ranging from 31.05 (SD = 26.22) at density 0.6 to 33 mm (SD = 15 mm) at density 0.15. In the density × Display Size interaction, the 55-inch display produced the largest distance at density 0.15 (M = 43 mm, SD = 21 mm) and the smallest at density 0.075 (M = 39 mm, SD = 20 mm), whereas on the 27-inch display the largest distance occurred at density 0.15 (M = 21 mm, SD = 9 mm) and the smallest at density 0.3 (M = 16 mm, SD = 6 mm). The CDG × Display Size interaction revealed: on the 27-inch display, distance increased with CDG, from 15 mm (SD = 7 mm) at CDG 1 to 21 mm (SD = 7 mm) at CDG 3.5, whereas on the 55-inch display, the largest distance occurred at CDG 1 (M = 45 mm, SD = 27 mm) and the smallest at CDG 2.5 (M = 36 mm, SD = 16 mm).

In the three-way density × CDG × Display Size interaction, the largest selection distance was observed at density 0.15 with CDG 1 on the 55-inch display (M = 47 mm, SD = 26 mm),

followed by density 0.3 with CDG 1 on the 55-inch display (M = 45 mm, SD = 27 mm) and density 0.6 with CDG 3.5 on the 55-inch display (M = 45 mm, SD = 32 mm). The smallest selection distance was at density 0.075 with CDG 1 on the 27-inch display (M = 14 mm, SD = 4 mm), followed by density 0.3 with CDG 1 on the 27-inch display (M = 14 mm, SD = 5 mm) and density 0.6 with CDG 1 on the 27-inch display (M = 14 mm, SD = 7 mm).

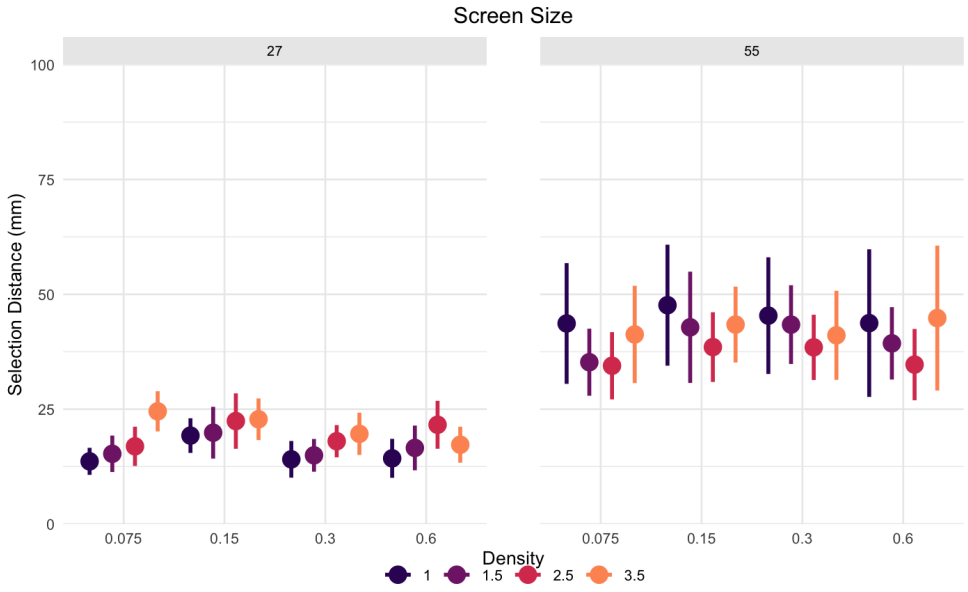


Figure 5.22: Mean selection distance of targets using the area cursor per trial. Error bars show 95% CIs.

Hand Movement Distance

Overall, participant moved their hand a distance of 145.36 mm, SD = 111.24 mm (see Table 5.6, rows 29-36, Table 5.5 and Figure 5.23). Hand movement varied considerably with density, with values of 152.09 mm (SD = 100.20 mm) at 0.075, 146.73 mm (SD = 106.70 mm) at 0.15, 69.44 mm (SD = 38.17 mm) at 0.3, and 212.15 mm (SD = 128.21 mm) at 0.6. Across CDG levels, hand movement decreased as CDG increased, from 228.25 mm (SD = 133.07 mm) at CDG 1, to 164.86 mm (SD = 101.29 mm) at CDG 1.5, 102.08 mm (SD = 57.70 mm) at CDG 2.5, and 85.46 mm (SD = 74.16 mm) at CDG 3.5. The 27-inch display produced shorter hand movement (M = 88.56 mm, SD = 54.39 mm) than the 55-inch display (M = 202.15 mm, SD = 123.94 mm).

For the density $\times$ CDG interaction, the largest distance occurred at density 0.6 paired with CDG 1 (M = 321.72 mm, SD = 140.05 mm), while the smallest was at density 0.3 paired with CDG 3.5 (M = 36.48 mm, SD = 12.88 mm). Across all CDG levels, density 0.3 consistently produced the lowest amount of movement, while density 0.6 produced the longest, with values at density 0.6 ranging from 129.74 mm (SD = 92.57 mm) at CDG 3.5 to 321.72 mm (SD = 140.05 mm) at CDG 1. In the density $\times$ Display Size interaction, the 55-inch display produced

the longest movement at density 0.6 (M = 285.46 mm, SD = 133.31 mm) and the shortest at density 0.3 (M = 86.90 mm, SD = 42.56 mm), whereas on the 27-inch display the longest movement occurred at density 0.6 (M = 137.81 mm, SD = 64.72 mm) and the shortest at density 0.3 (M = 52.23 mm, SD = 22.88 mm). The CDG × Display Size interaction revealed a consistent decrease in hand movement with increasing CDG on both displays: on the 27-inch display, values fell from 140.02 mm (SD = 55.49 mm) at CDG 1 to 48.43 mm (SD = 30.59 mm) at CDG 3.5, while on the 55-inch display, values fell from 317.73 mm (SD = 129.11 mm) at CDG 1 to 121.97 mm (SD = 85.75 mm) at CDG 3.5.

In the three-way density × CDG × Display Size interaction, the largest distance was observed at density 0.6 with CDG 1 on the 55-inch display (M = 422.69 mm, SD = 131.45 mm), followed by density 0.075 with CDG 1 on the 55-inch display (M = 350.55 mm, SD = 39.86 mm) and density 0.15 with CDG 1 on the 55-inch display (M = 339.82 mm, SD = 86.55 mm). The shortest hand movement was at density 0.3 with CDG 3.5 on the 27-inch display (M = 29.41 mm, SD = 13.64 mm), followed by density 0.3 with CDG 2.5 on the 27-inch display (M = 38.90 mm, SD = 8.84 mm) and density 0.3 with CDG 3.5 on the 55-inch display (M = 43.55 mm, SD = 7.04 mm).

Table 5.5: Percentage decrease in mean hand movement distance across CDG multipliers and screen sizes.

Screen Size	**CDG 1 → 1.5**	**CDG 1.5 → 2.5**	**CDG 2.5 → 3.5**	**CDG 1 → 3.5**
27"	25.5%	32.0%	27.0%	63.0%
32"	28.4%	36.9%	23.9%	65.6%
43"	23.5%	35.8%	24.8%	63.1%
55"	29.1%	36.7%	21.5%	64.7%
Overall	**26.8%**	**35.7%**	**23.9%**	**64.2%**

Perceived Fatigue

Perceived fatigue was at floor throughout Experiment 5. Participants reported a mean rating of 0.07 (SD = 0.41, 95% CI [0.03, 0.10]) on the Borg CR10 scale (Table 5.6 rows 36–42 and Figure 5.24), where 0 denotes "nothing at all" and 0.5 (the lowest non-zero point on the scale) denotes an exertion that is just noticeable. The large majority of ratings were exactly zero, and no condition mean (maximum M = 0.33) reached the scale's lowest verbal anchor. The same instrument returned means of 1.62 and 2.28 in Experiment 4; the difference reflects task structure rather than technique, as each Experiment 5 condition comprised only 13 discrete selections separated by a return to the start position and a 500 ms dwell, too short and intermittent to accumulate perceptible muscular fatigue.

Participants reported a mean fatigue rating of 0.07 (SD = 0.41) on the scale (see Table 5.6 rows 36-42 and Figure 5.24), indicating minimal perceived fatigue across the experiment. Across

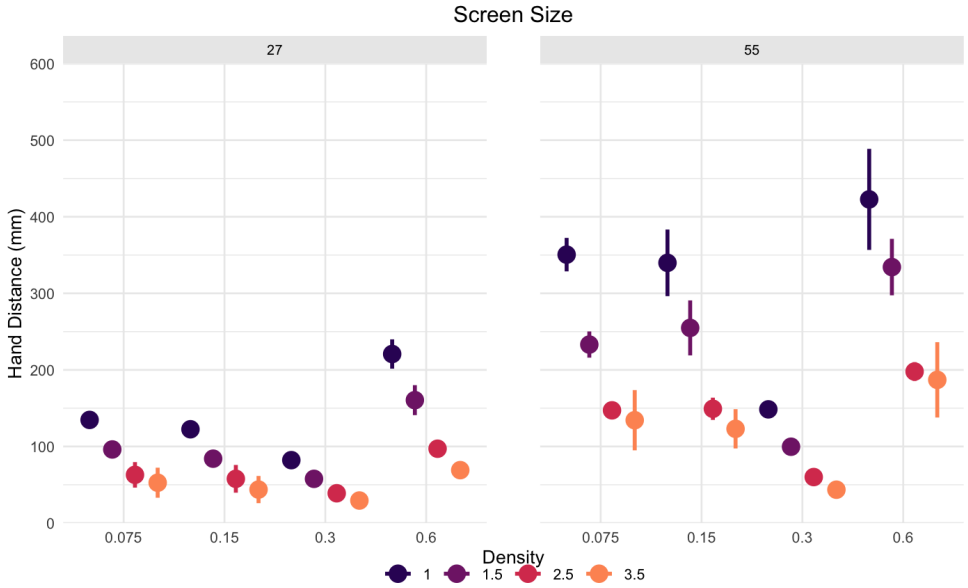


Figure 5.23: Mean distance the hand moved per trial. Error bars show 95% CIs.

Densities, differences were modest, with density = 0.15 producing the highest fatigue rating (M = 0.11, SD = 0.65) and density = 0.075 the lowest (M = 0.04, SD = 0.18), with density = 0.3 (M = 0.06, SD = 0.34) and density = 0.6 (M = 0.06, SD = 0.30) falling in between. CDG had a comparable effect, with CDG = 3.5 producing the highest rating (M = 0.10, SD = 0.56) and CDG = 1.5 the lowest (M = 0.04, SD = 0.23), with CDG = 1 (M = 0.07, SD = 0.47) and CDG = 2.5 (M = 0.06, SD = 0.31) falling in between. Display Size showed almost no effect, with the 55" (M = 0.07, SD = 0.48) and 27" (M = 0.06, SD = 0.33) displays producing nearly identical ratings.

For the two-way interaction effects, the Density × CDG interaction was driven primarily by density = 0.15, which produced the highest fatigue at both CDG = 1 (M = 0.18, SD = 0.90) and CDG = 3.5 (M = 0.22, SD = 0.94), while remaining low at intermediate CDG values. The Density × Display Size interaction showed a similar pattern, with density = 0.15 on the 55" display producing the highest combination (M = 0.17, SD = 0.90), while all other density-display combinations remained at or below M = 0.07. The CDG × Display Size interaction was less pronounced, with CDG = 1 on the 55" (M = 0.11, SD = 0.64) and CDG = 3.5 on the 27" (M = 0.10, SD = 0.46) producing the highest ratings.

For the three-way Density × CDG × Display Size interaction, the most fatiguing combination overall was density = 0.15 at CDG = 3.5 on the 55" display (M = 0.33, SD = 1.29), followed by density = 0.15 at CDG = 1 on the 55" display (M = 0.31, SD = 1.25). In contrast, several combinations produced no reported fatigue at all (M = 0.00, SD = 0.00), including density = 0.3 at CDG = 1 on the 27" display and density = 0.075 at CDG = 3.5 on the 55" display. Notably, the density = 0.15 condition appeared to drive most of the elevated fatigue ratings across the design, with its effect most pronounced on the 55" display at extreme CDG values.

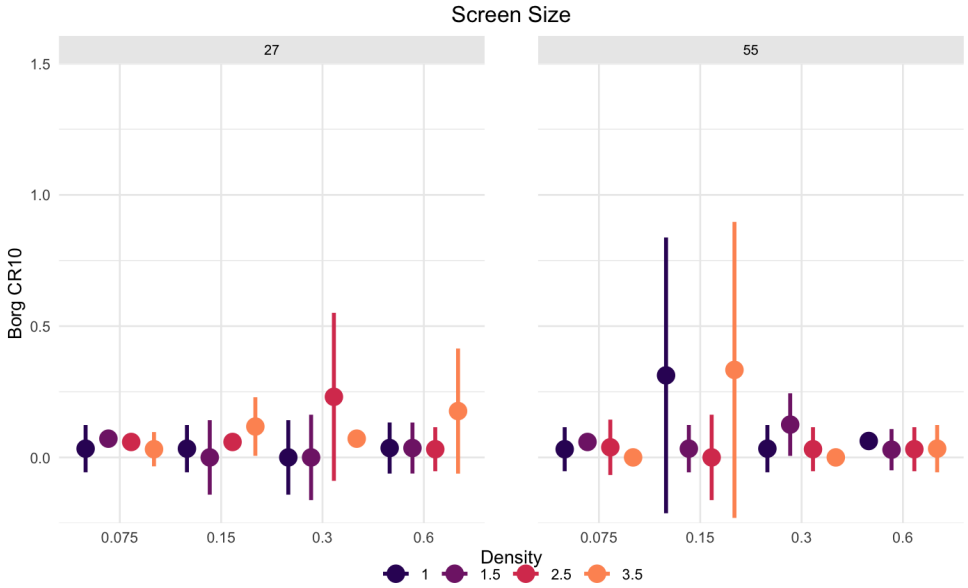


Figure 5.24: Borg CR10 Perceived fatigue scale per condition. Error bars show 95% CIs.

Perceived Control

Participants reported a mean perceived control rating of 6.63 (SD = 1.71) on a 10-point scale (see Table 5.6 rows 43-39 and Figure 5.25). Across Densities, differences were modest, with density = 0.075 producing the highest control rating (M = 6.77, SD = 1.68) and density = 0.3 the lowest (M = 6.46, SD = 1.72), with density = 0.15 (M = 6.64, SD = 1.71) and density = 0.6 (M = 6.64, SD = 1.73) falling in between. In contrast, CDG had a more pronounced effect on perceived control, with CDG = 1.5 producing the highest rating (M = 7.05, SD = 1.40) and CDG = 3.5 the lowest (M = 5.78, SD = 1.91), with CDG = 1 (M = 6.93, SD = 1.73) and CDG = 2.5 (M = 6.75, SD = 1.48) falling in between. Display Size showed minimal effect, with the 55" (M = 6.70, SD = 1.58) and 27" (M = 6.56, SD = 1.83) displays producing similar ratings.

For the two-way interaction effects, the Density $\times$ CDG interaction showed that the decline in perceived control at CDG = 3.5 was broadly consistent across Densities, ranging from (M = 5.70, SD = 1.75) at density = 0.075 to (M = 5.84, SD = 1.97) at densities = 0.15 and 0.6. The highest combination was density = 0.075 at CDG = 1 (M = 7.37, SD = 1.54), closely followed by density = 0.6 at CDG = 1.5 (M = 7.31, SD = 1.38). The Density $\times$ Display Size interaction was modest, with density = 0.075 on the 55" display producing the highest combination (M = 6.92, SD = 1.43) and density = 0.6 on the 27" display the lowest (M = 6.39, SD = 1.92). The CDG $\times$ Display Size interaction was more pronounced: on the 27" display, perceived control declined steadily from (M = 7.38, SD = 1.62) at CDG = 1 to (M = 5.34, SD = 2.05) at CDG = 3.5, while on the 55" display the pattern was less monotonic, peaking at CDG = 2.5 (M = 7.03, SD = 1.41) and dropping only at CDG = 3.5 (M = 6.25, SD = 1.64).

For the three-way Density $\times$ CDG $\times$ Display Size interaction, the highest combination overall was density = 0.15 at CDG = 1 on the 27" display (M = 7.87, SD = 1.13), followed by density

= 0.6 at CDG = 1.5 on the 27" display (M = 7.54, SD = 1.01). In contrast, the lowest combination was density = 0.3 at CDG = 3.5 on the 27" display (M = 5.04, SD = 1.84), with the 27" display producing the four lowest ratings overall — all at CDG = 3.5 across Densities (range M = 5.04 to 5.56). Notably, on the 55" display the drop at CDG = 3.5 was less severe (range M = 6.00 to 6.47), suggesting that the larger display partially mitigated the loss of perceived control at high CDG values.

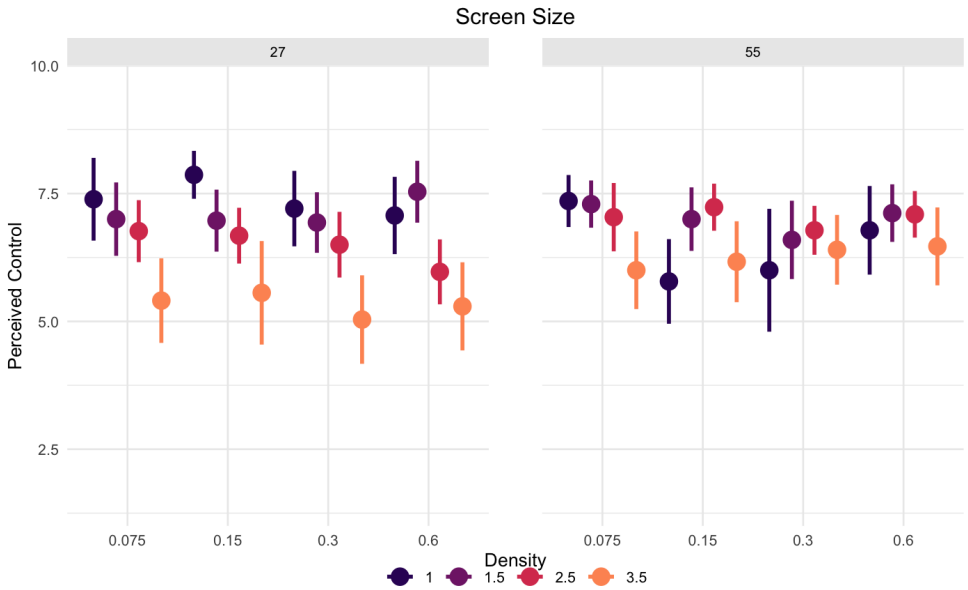


Figure 5.25: Perceived control of the cursor per condition. Error bars show 95% CIs.

Perceived Difficulty

Participants reported a mean perceived difficulty rating of 4.23 (SD = 1.84) on a 10-point scale (see Table 5.6 rows 50-56 and Figure 5.26). Across Densities, differences were modest, with density = 0.6 producing the highest difficulty rating (M = 4.36, SD = 1.86) and density = 0.075 the lowest (M = 4.07, SD = 1.83), with density = 0.15 (M = 4.20, SD = 1.85) and density = 0.3 (M = 4.28, SD = 1.82) falling in between. In contrast, CDG had a more pronounced effect on perceived difficulty, with CDG = 3.5 producing the highest rating (M = 4.82, SD = 1.91) and CDG = 1.5 the lowest (M = 3.89, SD = 1.72), with CDG = 1 (M = 4.07, SD = 1.83) and CDG = 2.5 (M = 4.14, SD = 1.77) falling in between. Display Size showed a modest effect, with the 27" display producing slightly higher difficulty ratings (M = 4.37, SD = 1.91) than the 55" (M = 4.09, SD = 1.75).

For the two-way interaction effects, the Density × CDG interaction showed that the rise in difficulty at CDG = 3.5 was broadly consistent across Densities, ranging from (M = 4.75, SD = 1.79) at density = 0.6 to (M = 4.88, SD = 1.91) at density = 0.075. The lowest combination was density = 0.075 at CDG = 1.5 (M = 3.55, SD = 1.61), while at CDG = 2.5 difficulty rose with Density, peaking at density = 0.6 (M = 4.59, SD = 1.68). The Density × Display Size interaction

was modest, with density = 0.3 on the 27" display producing the highest combination (M = 4.61, SD = 1.91) and density = 0.075 on the 55" the lowest (M = 4.01, SD = 1.75); on the 55" display difficulty was relatively flat across Densities (range M = 3.97 to 4.19), while on the 27" display it rose with Density (range M = 4.14 to 4.61). The CDG × Display Size interaction was the most pronounced: on the 27" display, difficulty rose steadily from (M = 3.70, SD = 1.66) at CDG = 1.5 to (M = 5.45, SD = 1.99) at CDG = 3.5, while on the 55" display the pattern was flatter and non-monotonic, dipping at CDG = 2.5 (M = 3.79, SD = 1.81) and only modestly rising at CDG = 3.5 (M = 4.16, SD = 1.58).

For the three-way Density × CDG × Display Size interaction, the highest combination overall was density = 0.3 at CDG = 3.5 on the 27" display (M = 5.61, SD = 2.08), followed by density = 0.075 at CDG = 3.5 on the 27" display (M = 5.56, SD = 2.03), with the 27" display producing the four highest ratings overall — all at CDG = 3.5 across Densities (range M = 5.29 to 5.61). In contrast, the lowest combination was density = 0.075 at CDG = 1.5 on the 27" display (M = 3.11, SD = 1.27), and on the 55" display difficulty at CDG = 3.5 was substantially lower (range M = 4.07 to 4.27), suggesting that the larger display partially mitigated the increase in perceived difficulty at high CDG values.

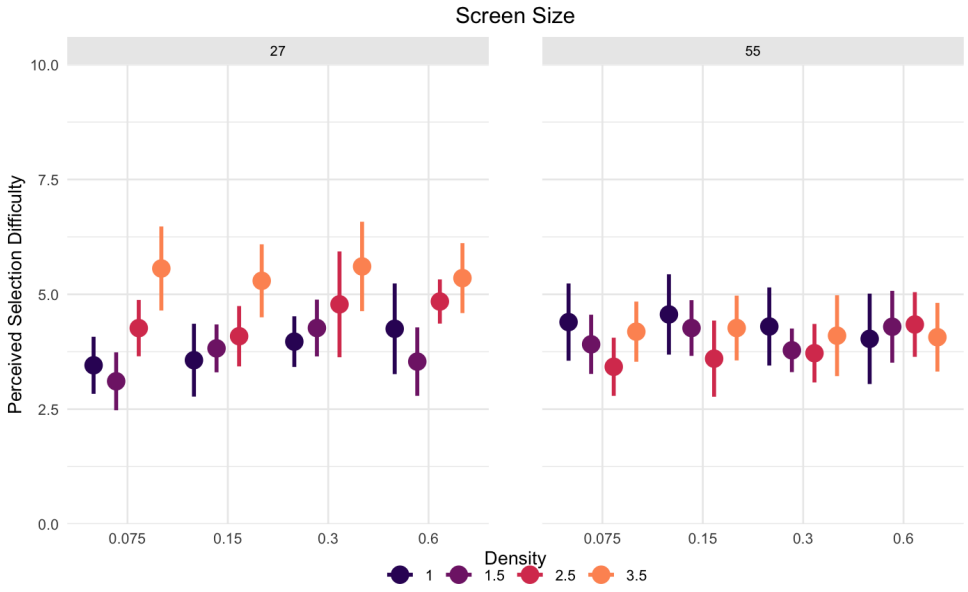


Figure 5.26: Perceived selection difficulty per condition. Error bars show 95% CIs.

Table 5.6: ANOVA results, and post-hoc test results for Experiment 5. Significant comparisons are highlighted*

	Selection Time	**Mean 1679 ms, SD 392 ms, 95% CI [1645, 1713] ms**			
	ANOVA	*Result*	p	η_p^2	*Post-Hoc Comparisons*
1	*Density*	F(3, 470) = 7.8	< .001*	.07	• 0.6 > 0.075 ($p \leq .001$) • 0.6 > 0.15 ($p = .02$)
2	*CDG*	F(3, 470) = 4.9	< .001*	.05	• 1 > 1.5 ($p = .01$) • 1 > 2.5 ($p \leq .001$)
3	*Display*	F(1, 470) = 9.8	.001*	.02	• 55" > 27" ($p = .02$)
4	*Density* × *CDG*	F(9, 470) = 0.42	.75	.01	
5	*Density* × *Display*	F(3, 470) = 2.1	.10	.01	
6	*CDG* × *Display*	F(3, 470) = 14.8	< .001*	.10	• 1 > 1.5, 2.5, 3.5 for 55" ($p \leq .001$) • 55" > 27" at 1.0 ($p \leq .001$)
7	*Density* × *CDG* × *Display*	F(9, 470) = 1.7	.07	.03	
	Error Rate	**Mean 13 %, SD 13 %, 95% CI [12, 14] %**			
8	*Density*	F(3, 470) = 43.4	< .001*	.21	• Higher > lower ($p < .05$)
9	*CDG*	F(3, 470) = 4.0	.00*	.03	• 1.5 < 3.5 ($p = .005$)
10	*Display*	F(1, 470) = 3.1	.08	.01	
11	*Density* × *CDG*	F(9, 470) = 1.3	.10	.03	
12	*Density* × *Display*	F(3, 470) = 2.9	.05*	.02	Corroborates Density main effect
13	*CDG* × *Display*	F(3, 470) = 11.5	< .001*	.07	• 3.5 > 1 & 1.5 for 27" ($p \leq .001$) • 55" > 27" for 1 ($p = .02$) • 27" > 55" for 3.5($p \leq .001$)
14	*Density* × *CDG* × *Display*	F(9, 470) = 0.35	.63	.01	
	Sensitivity	**Mean 0.18, SD 0.15, 95% CI [0.17, 0.20]**			
15	*Density*	F(3, 470) = 505.3	< .001*	.75	Higher > lower ($p < .001$)
16	*CDG*	F(3, 470) = 1.1	.06	.02	
17	*Display*	F(1, 470) = 14.4	< .001*	.07	• 55" > 27" ($p < .001$)
18	*Density* × *CDG*	F(9, 470) = 0.58	.93	.01	
19	*Density* × *Display*	F(3, 470) = 1.86	.13	.03	

Continued on next page

Table 5.6: (continued)

	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
20	*CDG* × *Display*	F(3, 470) = 8.0	< .001*	.09	• 27" > 55" for 1 (p< .001)
21	*Density* × *CDG* × *Display*	F(9, 470) = 1.4	.23	.02	
	Proximity	**Mean 29 mm, SD 20 mm, 95% CI [27, 31] mm**			
22	*Density*	F(3, 470) = 1.67	.17	.03	
23	*CDG*	F(3, 470) = 2.0	.11	.03	
24	*Display*	F(1, 470) = 303.1	< .001*	.53	• 55" > 27" (p< .001)
25	*Density* × *CDG*	F(9, 470) = 0.22	.97	.01	
26	*Density* × *Display*	F(3, 470) = 0.48	.35	.01	
27	*CDG* × *Display*	F(3, 470) = 4.4	< .001*	.04	• Corroborates Display main effect • 2.5 < 1 for 55" (p= .02)
28	*Density* × *CDG* × *Display*	F(9, 470) = 0.35	.92	.01	
	Hand Distance	**Mean 145 mm, SD 111 mm, 95% CI [136, 154] mm**			
29	*Density*	F(3, 470) = 414.86	< .001*	.71	• 0.6 > {0.075, 0.15} > 0.3 (p≤ .001)
30	*CDG*	F(3, 470) = 384.70	< .001*	.69	• 1 > 1.5 > 2.5 > 3.5 (p≤ .001)
31	*Display*	F(1, 470) = 997.78	< .001*	.66	• 55" > 27" (p≤ .001)
32	*Density* × *CDG*	F(9, 470) = 55.21	< .001*	.49	• 0.3 lowest at CDG 1, highest at CDG 2.5, 3.5 (p≤ .001)
33	*Density* × *Display*	F(3, 470) = 215.70	< .001*	.55	• 55" > 27" except at 0.3 (p≤ .001) • 27" > 55" 0.3 (p≤ .001)
34	*CDG* × *Display*	F(3, 470) = 240.23	< .001*	.58	• 55" > 27" at {1, 1.5} (p≤ .001) • 27" > 55" at {2.5, 3.5} (p≤ .001)
35	*Density* × *CDG* × *Display*	F(9, 470) = 21.68	< .001*	.27	Corroborates 2-way interactions

Continued on next page

Table 5.6: (continued)

	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
	Perceived Fatigue	**Mean 0.07, SD 0.41, 95% CI [0.03, 0.10]**			
	ANOVA	*Result*	*p*	η_p^2	*Post-Hoc Comparisons*
36	*Density*	F(3, 470) = 149.44	< .001*	.50	• 0.15 > others ($p<$ .001)
37	*CDG*	F(3, 470) = 151.71	< .001*	.50	• 1 > 1.5 ($p<$ .001) • 1 > 2.5 ($p<$ .001) • 3.5 > 1.5 ($p<$ .001) • 3.5 > 2.5 ($p<$ .001)
38	*Display Size*	F(1, 470) = 431.95	< .001*	.49	• 55" > 27" ($p<$ .001)
39	*Density* × *CDG*	F(9, 470) = 66.33	< .001*	.57	• 0.15 lowest at CDG 1.5 & 2.5 ($p<$ .001)
40	*Density* × *Display*	F(3, 470) = 182.51	< .001*	.55	• 55" > 27" only at 0.15 ($p<$ .001)
41	*CDG* × *Display*	F(3, 470) = 211.58	< .001*	.59	• 55" > 27" at CDG 1 & 3.5 ($p<$ .001) • 27" > 55" at CDG 2.5 ($p<$ .001)
42	*Density* × *CDG* × *Display*	F(9, 470) = 61.57	< .001*	.55	Corroborates 2-way interactions
	Perceived Control	**Mean 6.63, SD 1.71, 95% CI [6.48, 6.78]**			
43	*Density*	F(3, 470) = 1.12	.34	.01	
44	*CDG*	F(3, 470) = 23.61	< .001*	.14	• 3.5 < all ($p\leq$ .001)
45	*Display Size*	F(1, 470) = 1.89	.17	.00	
46	*Density* × *CDG*	F(9, 470) = 0.90	.52	.02	
47	*Density* × *Display*	F(3, 470) = 1.31	.27	.01	
48	*CDG* × *Display*	F(3, 470) = 12.09	< .001*	.07	• 1 > 2.5 & 3.5 on 27" ($p=$ .001) • 27" > 55" & at 1.0" ($p<$ .001) • High CDG > on larger display ($p\leq$ .02)
49	*Density* × *CDG* × *Display*	F(9, 470) = 1.83	.06	.04	
	Perceived Difficulty	**Mean 4.23, SD 1.84, 95% CI [4.07, 4.39]**			

Continued on next page

Table 5.6: (continued)

	ANOVA	*Result*	p	η_p^2	*Post-Hoc Comparisons*
50	*Density*	F(3, 470) = 0.89	.45	.01	
51	*CDG*	F(3, 470) = 13.15	< .001*	.08	• 3.5 > all ($p \leq$ .001)
52	*Display Size*	F(1, 470) = 7.98	.00*	.02	• 27" > 55" (p= .005)
53	*Density* × *CDG*	F(9, 470) = 0.75	.66	.01	
54	*Density* × *Display*	F(3, 470) = 1.35	.26	.01	
55	*CDG* × *Display*	F(3, 470) = 10.53	< .001*	.07	• 3.5 > 1 on 27" (p= .005) • 3.5 27" > 3.5 55" (p= .007)
56	*Density* × *CDG* × *Display*	F(9, 470) = 0.47	.90	.01	

5.8 Overall Discussion

Experiments 4 and 5 jointly characterise how CDG interacts with dynamic area-cursor targeting across a range of display sizes, input tasks, and target densities. The starting question for this chapter was whether the relaxed precision requirement of an area cursor could absorb the input instability that higher gain introduces, and therefore allow the technique to scale to larger displays without requiring larger arm movements. The combined results provide evidence towards this for dynamic area cursor interactions the case: increased gain reduced the physical demand of mid-air input substantially, area-based targeting preserved acceptable accuracy across most of the conditions tested, and the combination performed best on displays where a 1:1 mapping was most costly. At the same time, the results also reveal that the appropriate gain level is not a single fixed value: it depends on display size, on whether the task is continuous or discrete, and on how densely targets are packed across the interface.

5.8.1 Dynamic Area Cursors Support Increased Gain Levels

The most consistent finding across both experiments is that the area cursor's relaxed precision requirement and increased CDG are mutually beneficial. Usually, higher gain produces a speed–accuracy trade-off: small hand movements allow larger cursor displacements and shorter traversal distances, but amplified input noise raising selection error, particularly for small or densely packed targets. As area-based targeting widens the effective activation region of a widget, the cost of small cursor displacements caused by sensor noise, muscle tremor, and the Heisenberg effect is reduced. The cursor does not have to land precisely on the target, only close enough for the area cursor to enclose the intended target. Users can also make smaller cursor movements knowing that fully moving to the target may not be necessary, giving a greater level of control over interaction and potentially reducing fatigue via reduced range of motion,

especially if the arm does not need to be raised as high [10, 80]. This reframes the trade-off Casiez et al. [30] characterised for point cursors: gain still amplifies instability, but the area cursor absorbs much of that instability before it translates into a selection error or target slip.

In the slider task of Experiment 4, increasing the gain multiplier from 1.0 to 3.5 reduced mean hand movement distance by 64.2% overall, with comparable significant reductions across all four display sizes (Table 5.1, ($p \leq .001$)). On the 55" display, this reduction corresponded to a difference of approximately 280 mm of lateral hand movement per trial. Similar reductions were observed in the button task of Experiment 4 and in Experiment 5, where increasing CDG from 1.0 to 3.5 reduced hand movement by approximately 63% on average. Despite reduced hand movement, CDG produced a slight increase in error rates across all experiments ($p \leq .001$) with participants having to be *"more careful"* [P9] when navigating. In Experiment 4, slider error distance increased from a mean of 3.08 mm at CDG 1.0 and 1.5 to 4.24 mm at CDG 2.5 and 3.5, a difference of just over 1 mm, that is unlikely to be functionally meaningful for most slider applications; this could particularly be true for notched sliders with discrete steps. Button selection error increased more steeply with gain, but the increase was concentrated on the smallest displays: at CDG 3.5, the 27" display incurred an error rate of 16.75%, compared with only 8.83% on the 55" display. The same gain that destabilised interaction on the smallest screen was absorbed effectively on the largest.

The hand-movement and accuracy results together suggest that high gain combined with area-based targeting is more effective on larger displays than on smaller ones. On the 27" display, targets are physically close enough that a 1:1 mapping already places the cursor within reach with minimal arm movement, so the additional amplification provided by gain offers a smaller physical-traversal benefit while still amplifying noise. On the 55" display, the 1:1 mapping required hand movements that approached the comfort limits of unsupported arm motion (mean 309.58 mm at CDG 1, with peaks of nearly 430 mm in the slider task), and the area cursor provided flexibility for absorbing the instability that higher gain introduced. This is consistent with prior work on mid-air pointing on large displays [128], where the optimal gain level was found to depend on display size and task, but extends that finding by showing that the underlying targeting technique materially changes the trade-off. Previous studies have evaluated CDG against standard point cursors, where any reduction in pointing precision translates directly into reduced selection accuracy. A dynamic area cursor changes that by widening the effective target area, shifting the Fitts' Law trade-off in favour of higher gain for a given display size. Hand movement distance at density 0.3 was markedly lower than at the other densities; this was rechecked against the logged data and reflects the task rather than participant behaviour, as the targets used at this density lay closer to the start position. Absolute hand movement is therefore not compared across density levels, and the result of interest is the effect of CDG, which reduced hand movement at every density.

On the smaller displays, participants approached widgets *"more closely"* [P6] before acti-

vation than on the larger displays (($p \leq .001$)), and the influence of CDG on activation distance was small ($\eta_p^2 = .03, .07$). On the 55" display, activation was done farther than on the 27" display and varied more with gain, indicating that participants made greater use of area-based targeting on larger screens. This behaviour is not a property of the cursor technique itself, which always targets the nearest widget, but of how participants chose to interact with it. On smaller displays, where the entire interface is within comfortable reach, participants tended to gravitate towards the visible region of their intended widget (also seen in Chapter 4), whereas on larger displays the physical cost of doing so encouraged selection from a greater distance. This suggests that area-based targeting may be more impactful as display size increases, provided gain is set appropriately. Display sizes were simulated at a fixed 2.5 m viewing distance, so physical size and visual angle scaled together (13.5° at 27" against 27.2° at 55"), whereas a user free to approach a smaller display would recover the visual angle but not the hand movement it demands; part of the display size effect reported here may therefore be visual rather than motor. Interactions also began from a fixed start position with the hand raised over a sensor in a known location, whereas a user approaching a display in practice raises their hand at an arbitrary offset that increased gain amplifies as readily as deliberate movement.

Subjective fatigue ratings reinforce the same. In the slider task of Experiment 4, mean Borg CR10 ratings increased modestly with gain (from 1.43 at CDG 1 to 1.71 at CDG 3.5) and were broadly comparable across display sizes, with the 55" display producing the lowest mean rating overall (1.47). In the button task, the 43" display produced the highest mean rating (2.56) and the 55" display the lowest (2.16). This finding appears counter-intuitive given that the 55" display required the greatest hand movement. The pattern is, however, consistent with the interaction of the two effects. First, participants exhibited greater reliance on area-based targeting when interacting with the 55" display, reducing the necessity of traversing the full extent of the interface. Second, participants used higher CDG values with greater confidence on the 55" display, stating the ‘*‘larger display made it easier”* [P4]. In Experiment 5, perceived fatigue was very low overall (mean 0.07 on the Borg CR10 scale) but the same effect was apparent. The 55" display did not produce higher fatigue than the 27" display despite requiring more than twice the hand movement on average. The combination of higher gain and area-based targeting therefore appears to scale with display size. While the Borg CR10 scale has been shown to have a strong association with consumed endurance [80], it would be preferable to objectively calculate fatigue using computational metrics like Consumed Endurance or the more comprehensive NICER model [101], due to hardware limitations this was not possible and is left for future work.

The error rates in this chapter are high for a Fitts-style task. Mean error was 11.07% for the button task of Experiment 4 (6.88% to 16.75% across conditions) and 13.22% in Experiment 5 (3.17% to 32.31%), against the roughly 4% conventionally expected in ISO 9241-411 evaluations [150]. Much of this gap is a property of the input modality rather than of the techniques

studied here. Direct comparisons report error rates of around 13% for hand-tracked mid-air pointing against 3–4% for a touchpad on the same task [52, 88], a consequence of tracking jitter [154] and the unsupported hand [168], so the overall means reported here sit at, rather than above, what is typical for mid-air input. The way errors are counted under an area cursor also inflates the value. As the proxemic cursor always has a widget in focus, a near miss that would land on the background with a point cursor instead selects a neighbouring target and is recorded as an error, and this matters most at high density, where little background remains. This is likely to have pushed participants towards speed as well, since errors were neither corrected nor penalised and participants were asked to be quick as well as accurate. These tasks are therefore best read as controlled comparisons using a Fitts-derived layout rather than as a formal Fitts' law characterisation; effective-width throughput was not computed, and Experiment 5 used a single index of difficulty.

In both experiments, CDG 3.5 was associated with the highest error rates, the lowest perceived control (*"it was more difficult to control"* [P11]), and the highest perceived difficulty ratings , but small effect on selection/task times (η_p^2 = .04, .03, .05). In Experiment 4, success rates dropped at CDG 3.5 across all display sizes (($p \leq .001$)), and in Experiment 5 perceived control decreased on the 27" display from a mean of 7.38 at CDG 1 to 5.34 at CDG 3.5, while only declining modestly on the 55" display (from 7.03 at CDG 2.5 to 6.25 at CDG 3.5). Taken together, these results suggest that smaller displays benefit more from conservative gain multipliers ($\sim$1.0–1.5), which strike a reliable balance between the physical-traversal benefit and the precision cost. Larger displays, however, will benefit from increased gain ($\sim$1.5–3.5), as the accuracy cost of higher CDG is effectively mitigated by the affordances of the area cursor alongside the additional visual space. Furthermore, while here, horizontal aspect ratios were used, these recommendations can be generalised to vertical orientation displays since the distribution of targets across the available screen space is likely to be similar [106].

Although CDG was held constant within each condition, the slider and button tasks of Experiment 4 imposed different demands on cursor control, and the results suggest that different gain levels are appropriate for each. Slider manipulation is a continuous, precision-demanding interaction in which the user must maintain control over a widget while making fine positional adjustments; once the slider handle has been acquired, the affordances of the area cursor no longer apply, and CDG governs cursor behaviour directly. In this, higher gain amplifies the input noise that subtle positional adjustments are most sensitive to, and the data reflects this: slider error distance increased with CDG, and the two highest gain levels produced reliably greater error than the two lowest. Button selection, by contrast, is a discrete, ballistic targeting interaction in which the user must rapidly acquire and confirm a target. Here, higher gain reduced the physical movement required to traverse between targets and the area cursor compensated for moderate overshoots, so CDG had a smaller and less consistent effect on selection time and a smaller effect on accuracy than in the slider task.

The implication is that a single CDG value is unlikely to be optimal across the full range of operations a mid-air interface might support. A gain multiplier of 1.0 to 1.5 favours fine continuous control; a gain multiplier of 2.5 favours discrete coarse selection on larger displays. Adjusting gain dynamically to suit different input tasks could help to better accommodate these competing cursor demands. Adaptive control-display gain mechanisms have been successfully used for other in-air gesture techniques [60,129] to allow users to take advantage of high and low gain where appropriate. For example, for the proxemic cursor, switching gain on a per-widget basis: when the proxemic cursor takes control of a continuous widget such as a slider, the gain multiplier could be reduced for the duration of the manipulation, and restored when the widget is released. This would preserve the physical-traversal benefit of higher gain during the acquisition phase, when the area cursor is helping to absorb noise, while restoring the precision needed during the steering phase, when it is not. Such a scheme would not require the user to perform an additional mode-switching action, since the change in gain would be tied to the existing activation gesture. Whether the resulting change in cursor speed during a single interaction would remain transparent to the user, or would introduce a new source of confusion, is a question that warrants follow-up work.

A more general alternative is velocity-dependent or acceleration-based gain, where the multiplier is determined by the speed of hand movement rather than by the current widget. Pointer acceleration [44] has been shown to improve input performance across various settings [18, 29, 30, 51, 94, 129], albeit with an increased risk of overshooting distant targets [30]. Others have adapted gain based on additional inputs, such as *semantic pointing* [18, 29], which considers the 'importance' of a user interface element. The constant multipliers used in this chapter were chosen specifically to provide a clean characterisation of the interaction between gain, area-based targeting, display size, and density, free of the additional dynamics that an adaptive scheme would introduce. The clean baseline this has established now provides the reference against which an adaptive scheme would need to be evaluated. Such schemes lie outside the scope of this thesis but are a natural direction for future work.

Additional alternatives worth further investigation include adapting the coordinate space based on what users can comfortably reach rather than assuming the use of a rectangular input plane [72], manipulating gain so that the cursor is attracted towards the centre of targets [100], or varying gain systematically based on vertical targeting, making movements towards the upper portions of the display less costly than lower portions where the arm naturally rests. However, a critical consideration for any of these more complex mappings is that a non-linear relationship between hand and cursor movements could have detrimental effects on usability by breaking the direct spatial mapping between hand position in mid-air and cursor position on screen.

5.8.2 Designing Mid-Air Cursor Interactions and User Interfaces

The combined results of Experiments 4 and 5 give designers a clearer picture of how to set CDG for a area cursor based interface, and how the answer depends on the display and on the layout of widgets within it.

Display size is the strongest moderator. On smaller displays in the 27"–32" range, targets are physically close together, and a 1:1 mapping or modest gain (CDG 1.0–1.5) is sufficient to keep the interface within comfortable reach. Higher gain on these displays reduces the small physical-traversal benefit available while introducing instability that the area cursor cannot fully absorb, particularly for dense layouts. On larger displays in the 43"–55" range, the 1:1 mapping requires hand movements that approach the comfortable limit of arm motion and that, in some configurations, exceed the reliable sensing range of typical hand-tracking cameras. Higher gain (CDG 2.5–3.5) becomes beneficial on these displays, as the area cursor has more visual space in which to absorb instability and because the alternative is to require the user to sustain large arm movements causing fatigue, progressively degrade fine motor control [103].

Target density acts as a second moderator. The Experiment 5 results show that as density increases, the effective region into which the proxemic cursor can expand shrinks, and the chance that input noise will land the cursor on a neighbouring widget rises. At the lowest density tested (0.075), area-based targeting performed strongly across all gain levels and on both display sizes, and participants benefited fully from higher gain. At intermediate density (0.15), the technique remained effective and selection times were among the fastest observed, suggesting that moderate clustering of widgets does not undermine the combination of high gain and area targeting. At high density (0.6), error rates rose substantially, sensitivity to distractor widgets more than tripled relative to the lower density levels, and the accuracy cost of high gain on the 27" display became pronounced (CDG 3.5 produced a 32.31% error rate at this density). The 55" display absorbed this cost more effectively, with error rates remaining below 14% at the same density and gain combination. These results are consistent with prior reports for the Bubble Cursor [71] and the Bubble Lens [122], where the advantage of dynamic area cursors diminished as targets were placed more tightly. The proxemic cursor is subject to the same fundamental limitation, but the additional finding here is that display size partly compensates for it: a larger display provides the area cursor with more physical room around each widget, even when the relative density is held constant.

A practical recommendation that emerges from these results is to think about widget layout in terms of localised groupings and the amount of empty space surrounding each grouping, rather than in terms of an absolute density figure. This measure of density can be directly applied to assist with the design of target layouts [19]. Sparse layouts, common in public information kiosks, retail menus, and museum exhibits sit clearly within the scope in which higher gain and area-based targeting work well. Mid-air user interface layouts should leave sufficient space around and between adjacent controls, as this creates more space for the area cursor to com-

pensate for targeting inaccuracy. This permits a more casual [141] approach to interaction by relaxing the demands on precise physical control, and is more forgiving for less experienced users, who are more likely to benefit from the assistive effects of area cursor targeting.

Even within denser target layouts, leaving generous whitespace at the edges of the interface, rather than placing widgets close to the display boundaries, gives the area cursor visible room to absorb overshoot. This appears to be an important contributor to the technique's robustness. In more locally dense layouts, extra space on the 'outside' edge of the interface might even encourage users to overshoot on purpose to avoid distractors and give the area cursor more room to expand, allowing users to disambiguate between targets more reliably.

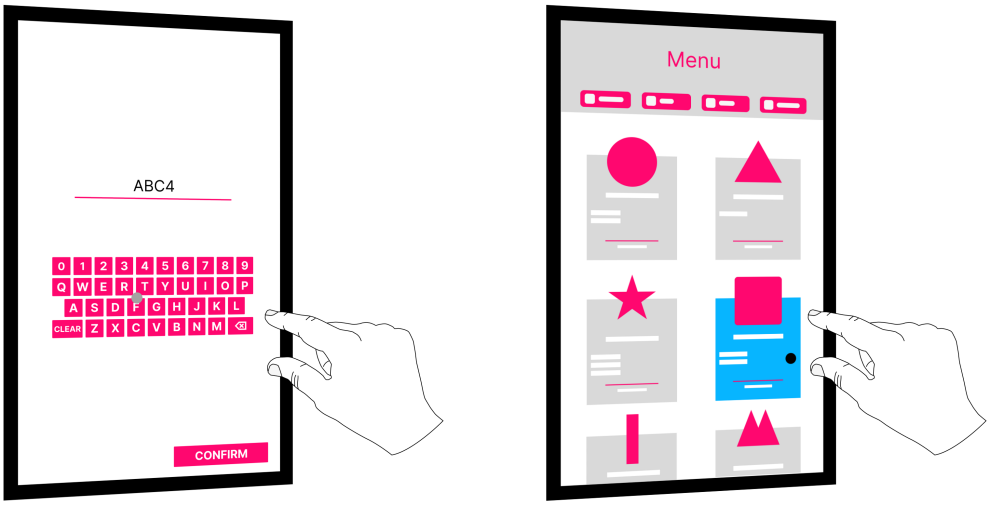


Figure 5.27: Example touchless user interface layouts inspired by real-world examples [142, 157]: text-entry interface with high-density small targets (left); and menu interface with low density, and both small and large targets (right).

To illustrate how these findings apply to real-world touchless display deployments [142,157], two contrasting exemplar user interfaces (Figure 5.27) are provided. Touchless interfaces for public displays are less likely to feature dense grid or radial layouts that fill the available space, but there are situations where this is necessary, such as text-entry keyboards (Figure 5.27, left) or choosing letters from A–Z for filtering transit destinations. Keyboard layouts are typically not rearranged to make more optimal use of screen space because users rely on familiar structures for effective input. Adopting a more conservative gain (CDG 1.0–1.5) for text entry will preserve cursor stability and selection accuracy within the dense cluster. However, this has implications for selecting other targets, such as navigation buttons at the bottom of the screen. Where a single interface mixes dense and sparse layouts, gain may need to be modulated regionally, enabling low CDG when the cursor is within the dense cluster and restoring high CDG elsewhere. This is a generalisation of the per-task adaptive scheme described in Section 5.8.2 and would allow a single touchless interface to host both layouts without compromising either.

In contrast, the menu kiosk (Figure 5.27, right) is representative of use cases where users

are presented with a limited number of options. The low target density and predominantly large target sizes mean higher CDG (2.5–3.5) can be confidently adopted to reduce hand movement. Furthermore, the smaller buttons located towards the top of the interface additionally benefit from higher CDG, as users do not need to raise their hands as high to target these elevated elements. Area cursor targeting is highly effective here because the empty space above the buttons provides room for the cursor to expand, though rearranging these top buttons into a 2×2 grid would yield even more horizontal space for coarse pointing.

These two examples are illustrative rather than evaluated. All five experiments in this thesis used abstract tasks: labelled sliders, ISO 9241-411 target rings and fields of identical distractors. This was deliberate, as it allowed target size, separation, density and movement amplitude to be controlled and follows standard practice for pointing evaluations [19, 150], but it limits what can be claimed about real interfaces. Abstract tasks remove visual search, meaningful content, targets that differ in importance or likelihood, scrolling and navigation, and any real consequence of a wrong selection. What they retain is the geometry of the interface, and the findings of this thesis are motor-level findings that depend on geometry: the stability of pinch at the moment of activation, the tolerance an area cursor gains from empty space, and the hand movement that gain removes on a given display size. These findings should therefore transfer to a real interface to the extent that its geometry falls within the ranges tested, while task times and error rates in absolute terms should not be expected to.

A self-service ordering kiosk, such as those used in fast-food restaurants, shows how this reasoning applies. Its main screen is a grid of large image tiles separated by gutters on a large portrait display: a sparse-to-moderate density layout with large targets, which is where the proxemic cursor and higher gain (CDG 2.5–3.5) performed best, and where gain has the added benefit of reducing how high the arm must be raised to reach the top of a tall screen. The row of category tabs resembles the closely spaced buttons of Experiment 3, where the area cursor remained effective but users tended to target buttons directly. Isolated controls such as the basket or payment button have the most empty space around them and would benefit most from area-based targeting.

The keyboard (Figure 5.27, left) represents the opposite extreme and marks the limit of these techniques. With keys, there is almost no empty space for the area cursor to use: nearest-target selection reduces to removing the gaps between keys, and the layout sits at the high-density end of Experiment 5, where error rose sharply with gain. These are predictions from the experimental results rather than findings, and evaluating the techniques on complete, realistic interfaces is the most important next step for this work.

These recommendations have been distilled into Table 5.7, which provides structured guidance on cursor configuration and user interface layout. These recommendations are split by display size (small versus large), reflecting the relative trade-off of increasing gain. They also present distinct guidelines for *coarse* and *fine* pointing, recognising that certain applications

Table 5.7: Summary of design recommendations for mid-air cursor interfaces.

	Smaller Displays (∼27")		**Larger Displays (∼55")**	
	Coarse Pointing Tasks	*Fine Pointing Tasks*	*Coarse Pointing Tasks*	*Fine Pointing Tasks*
Gain Adaptation	∼1–2.5 Lower gain for sliding	∼1–1.5	∼2.5–3.5 Lower gain for sliding	∼1.5–2.5
Density Layout	Low (≤0.15–0.3) Targets towards edges, extra space outside	Low (≤0.1) Targets towards centre	Moderate (≤0.45) Targets towards edges, extra space outside	Moderate (≤0.3) Targets towards centre
Benefits	Fast targeting, more space for cursor expansion	Enhanced precision, more controlled hand movement	Improved ergonomics, more space for cursor expansion	Balances precision with ergonomics

necessitate different levels of screen utilisation. For coarse pointing tasks involving a reduced number of targets, higher gain can be easily accommodated via area cursor expansion, and UI targets should be positioned to allow this. Conversely, fine-pointing tasks involving numerous or smaller targets benefit more from lower gain ratios. To reduce the necessary physical movement during these low-gain, fine-pointing tasks, targets should ideally be positioned towards the centre of the screen or directly in front of the user's torso.

The subjective measures collected in Experiment 5 add a further design consideration. Perceived control declined with gain on the 27" display (from 7.38 at CDG 1 to 5.34 at CDG 3.5) but remained relatively stable on the 55" display (from 6.92 at CDG 1 to 6.25 at CDG 3.5). Perceived difficulty increased correspondingly on the 27" display and remained more constant on the 55" display. The larger display appears to mitigate not only the accuracy cost of high gain, as discussed above, but also the user's experience of the interaction: participants felt more in control of higher-gain interactions when they had more visual space in which to recover from instability.

5.9 Chapter Conclusions

This chapter presented Experiments 4 and 5, which jointly characterised the interaction between CDG and the proxemic cursor across different display sizes, input tasks, and target densities. Building on Chapter 4, where area-based targeting reduced the precision required for widget acquisition under a 1:1 mapping on a 27" display, this chapter examined whether the same technique could scale to larger displays once CDG was introduced as a means of reducing the physical extent of mid-air hand movement. Experiment 4 manipulated CDG (1.0, 1.5, 2.5, 3.5) and display size (27", 32", 43", 55") across slider and button tasks, while Experiment 5

introduced target density (0.075, 0.15, 0.3, 0.6) as a third factor on the smallest and largest of those displays.

The results presented in this chapter demonstrate that the relaxed precision requirement of the proxemic cursor and increased CDG are mutually beneficial, allowing users to interact effectively with smaller and more controlled hand and arm movements. Across both experiments, increasing the gain multiplier from 1.0 to 3.5 reduced mean hand movement by approximately 63% on average, with the largest absolute reductions occurring on the 55" display where the 1:1 mapping otherwise required hand movements approaching the limits of comfortable arm motion and reliable sensing. Despite this substantial reduction in physical effort, selection accuracy was largely preserved: slider error distance increased by only $\sim$1 mm between the two lowest and two highest gain levels, and the button task, error with the highest gain (3.5) was concentrated on the smallest display rather than the largest, where the area cursor had more visual space in which to absorb instability. The technique's benefits were therefore most pronounced on those displays where a 1:1 mapping was most costly, supporting that an area cursor's relaxed precision can help absorb the input instability that higher gain introduces for area cursor interfaces.

The slider and button tasks of Experiment 4 imposed different demands on cursor control, and the results suggest that no single CDG value is appropriate across the full range of operations a touchless interface might support. Conservative gain (1.0–1.5) preserved the fine positional control required during continuous slider manipulation, where the area cursor's affordances no longer apply once the handle has been acquired; higher gain (2.5–3.5) was better suited to discrete button selection on larger displays, where ballistic targeting was supported by the area cursor and the physical-traversal benefit of higher gain was greatest. Experiment 5 added target density as a factor and showed that the combination of higher gain and area-based targeting remained effective across sparse and moderately tight layouts, but degraded at the highest density (0.6) where the surrounding distractor widgets substantially constrained the proxemic cursor's effective expansion region. As with the accuracy cost of high gain, the larger display absorbed density more effectively than the smaller one, with error rates remaining below 14% on the 55" display at density 0.6 and CDG 3.5, compared with over 32% on the 27" display under the same configuration.

These findings give designers of touchless cursor interfaces a clearer picture of how display size, target density, and CDG interact, and they motivate a set of layout and CDG recommendations distilled into Table 5.7: smaller displays favour conservative gain ($\sim$1.0–1.5), particularly for fine pointing tasks, whereas larger displays accommodate higher gain ($\sim$1.5–3.5) provided that target layouts leave sufficient empty space around clusters of widgets and at the edges of the interface for the area cursor to expand into.

5.9.1 Chapter Contributions

This chapter makes the following contributions:

1. Empirical characterisation of how CDG modification interacts with dynamic area-cursor targeting in mid-air input, demonstrating evidence towards that the area cursor's relaxed precision requirement can potentially absorb much of the input instability that higher gain introduces. The combination is shown to reduce the physical extent of hand movement substantially while preserving selection accuracy across a wide range of conditions.

2. An evaluation across four display sizes (27", 32", 43", 55") and two task types (continuous slider control and discrete button selection), showing that the appropriate gain level depends on both the display scale and the nature of the input operation. Conservative gain best supports fine continuous control, whereas higher gain becomes increasingly beneficial as display size grows and as the task shifts from continuous steering to discrete acquisition.

3. An evaluation across four levels of target density, showing that area-based targeting and increased CDG remain effective across sparse and tight UI layouts but degrade in the densest layouts, where the surrounding distractor widgets constrain the area cursor's effective expansion region. The results also show that display size partially compensates for high density, with larger displays absorbing the imprecision of high gain more effectively than smaller ones.

4. A characterisation of how users adapt their targeting behaviour to the combination of higher gain and area-based targeting, including a tendency to activate widgets from greater distances on larger displays and to gravitate towards the visible region of the intended widget on smaller displays. These behavioural findings extend the targeting strategies described in Chapter 4.

5. A set of design recommendations for touchless cursor interfaces (Table 5.7), split by display size and pointing granularity, that describe how to configure CDG and arrange widgets so that the combined benefits of area-based targeting and increased gain can be realised in practical mid-air deployments.

5.10 Research Questions

The outcomes of Experiments 4 and 5 provide further evidence that informs complete answers to the second and third research questions of this thesis, building on the initial evidence presented in Chapter 4.

> **RQ2** - How can we reduce the need for traversal and selection accuracy in a mid-air user interface across different display sizes and widget densities?

Chapter 4 established that an area cursor mitigates the precision required for widget acquisition in multi-widget mid-air interfaces, this effect was initially demonstrated under a 1:1 hand to cursor mapping on a single 27" display. The experiments presented in this chapter extend these findings to demonstrate that area-based targeting continues this mitigation when paired with an increased CDG across varying Display Sizes (27" to 55") and widget densities (sparse to dense layouts). Notably, increasing the CDG successfully reduced the broad physical hand movement, whilst maintaining selection accuracy. However, this dynamic varied. The accuracy costs of a high CDG were most evident on the smallest displays (i.e., where the physical-traversal benefit of higher gain is inherently modest) and within the densest layouts (i.e., where the area cursor has the least spatial room to expand before inadvertently encapsulating adjacent widgets). In contrast, larger displays appear to absorb the precision penalties of high gain more effectively, primarily due to the area cursor having more visual space in which to operate. Together, these results suggest that the precision requirements and physical traversal demands of a mid-air interfaces can be jointly mitigated. Crucially, not through a single applied technique, but through the combination of area-based targeting and a CDG that is matched to both the display's scale and the layout's density.

> **RQ3** - How can we design mid-air interactions such that the required effort is reduced while interaction efficiency is increased?

The experiments in this chapter also provide evidence and insights that help answer the third research question. The combination of area-based targeting and increased CDG reduces the required physical effort of mid-air interaction (in terms of hand and arm movement) without a corresponding reduction in interaction efficiency. On the largest display, mean hand movement at CDG 3.5 was less than half that at CDG 1.0 across both the slider and button tasks, while task and selection times either remained stable or improved at the moderate gain levels. Subjective fatigue ratings did not rise in line with the longer hand movements that the larger displays would typically cause, and were lowest on the 55" display with high gain. Participants also adapted their targeting behaviour to the available space, activating widgets from greater distances when the layout permitted and only approaching the widget closely when the surrounding density required it. The recommendations distilled in Table 5.7 translate this into concrete guidance for designers, addressing the question of how mid-air interactions can be designed so that the required effort is reduced while interaction efficiency is increased.

Chapter 6

Thesis Discussion

6.1 Introduction

The experimental chapters of this thesis each closed with a discussion of their own results. What they do not do is step back from the individual experiments and consider the thesis as a whole: how the three lines of work depend on one another, where the work sits relative to the interaction paradigms that have come to dominate mid-air input since the research began, and which of its findings are properties of the human hand rather than properties of the sensors and displays that were available at the time. This chapter takes up those questions.

6.2 Three Techniques, Three Residual Problems

The three techniques investigated within this thesis are presented in the preceding chapters as answers to three research questions, which makes them appear as independent improvements stitched together to create a thesis. They are better read as a chain. Each technique resolves one obstacle to efficient mid-air cursor interaction and, in resolving it, leaves a residual cost that is the obstacle the next technique addresses:

- **Pinch stabilises activation, but the user must still point precisely.** Experiment 1 found that pinch acquired a slider approximately 600 ms faster than push, and Experiments 2 and 3 reproduced the advantage across acquisition, transition, and task time, with higher success rates throughout. The mechanism is decoupling: the finger and thumb articulate while the palm to which the cursor is mapped does not, so the state change does not displace the cursor and the target slips that push produces are avoided. What pinch removes is the cost of *confirming* a selection. What it leaves is the cost of *aiming*: the cursor must still be brought inside the widget's bounds before there is anything to confirm, and pinch does nothing to make that easier. An activation gesture that cannot slip is of limited value if acquiring the target is difficult itself.

- **An area cursor relaxes pointing, but the user must still traverse.** The use of a dynamic area cursor addresses that residual directly, replacing the requirement of *widget overlap* with *closest to the cursor*. Experiment 2 recorded a reduction of approximately 700 ms in acquisition time and over 1000 ms in transition time against a standard cursor, with no accuracy penalty. However, the hand must still travel far enough to make the intended widget the nearest one, and on a large display that is a substantial physical movement. Experiment 1 made the point concretely: acquiring a slider from wherever the hand happened to be could leave it near the edge of reliable tracking for the remainder of the manipulation, since the workspace a user needs is the initial offset plus the widget's full traversal distance (Section 3.6.2). What the area cursor removes is the cost of *precision*, what it does not address is the cost of *amplitude*, and amplitude is what makes mid-air interaction fatiguing.

- **Increased gain removes traversal, but it amplifies instability.** Raising the gain multiplier from 1.0 to 3.5 reduced mean hand movement by approximately 63% across Experiments 4 and 5, which on the 55" display amounted to around 280 mm of lateral hand movement per trial. That addresses the amplitude residual directly. However, a constant multiplier is applied to the whole interaction and cannot distinguish intended displacement from tremor, sensor jitter, or the incidental movement of a hand held unsupported, so every source of instability is amplified alongside the intended motion. What gain removes is the cost of *amplitude*. What it introduces is a higher demand on *precision*.

Read in this way, the sequence closes on itself: the accuracy cost that increased CDG introduces is the one that area-based targeting relieves, and the movement cost that area-based targeting leaves is the one that increased CDG removes; the area cursor widens the target that gain destabilises, whilst pinch holds the cursor steady at the moment that the target is activated.

6.2.1 Findings Across The Experiments

Several findings emerged across the experiments that were not designed into any of them, and these are in some ways more informative than the planned comparisons. The first is *lazy targeting*: in the area-cursor conditions of Experiments 2 through 5, participants kept the cursor near a comfortable resting position, moved towards a widget only far enough to disambiguate it from its neighbours, and committed from a distance. This behaviour appeared without instruction and adapted to layout, with participants approaching more closely where widgets were dense and committing from further away where they were sparse. It is the mechanism by which the reduction in hand movement was found, and it indicates that participants formed an accurate model of how the area cursor resolved intent despite its minimal visual footprint. The second is the density bound. The same behaviour that produces the benefit depends on there being empty space to commit from, and Experiment 5 located the point at which that space runs out: at density 0.6

the surrounding distractors constrained the cursor's expansion region and the accuracy cost of gain re-emerged, most sharply on the smallest display. The third is the level of the error rates. Mean selection error in the button tasks of Chapter 5 was 11.07% in Experiment 4 and 13.22% in Experiment 5, and reached 32.31% in the worst condition. Those figures are in line with what other hand-tracked mid-air pointing studies report for comparable optical sensing [52, 88], but they are several times what a mouse or touchscreen achieves. None of the techniques in this thesis set out to remove the tremor and instability that produce that baseline; they were designed to let users work around it, and the error rates are a reminder that the underlying imprecision is still there.

6.3 Where This Work Sits: Other Modalities and Paradigms

6.3.1 The Commercial Introduction of Pinch Interactions

The most consequential development in mid-air interaction during the course of this research was the adoption of pinch (often combined with gaze) as one of the primary input paradigm of consumer XR headsets, most visibly in Apple's visionOS and in the hand-tracked mode of the Meta Quest. The paradigm has a long research background, running from early gaze-assisted pointing through Chatterjee et al.'s gaze-plus-gesture combinations [33] to Pfeuffer et al.'s formulation of *Gaze + Pinch* for virtual reality [140], in which the eyes indicate the target and a finger-to-thumb pinch confirms it. Here, pinch is chosen for the reasons outlined in this thesis: pinch is decoupled from pointing, it produces a clearly perceptible binary state through finger-to-thumb contact, and it does not disturb the pointer at the moment of activation. With targeting, gaze-and-pinch does something this thesis could not: it removes the traversal cost altogether. The eyes reach any target on the display at effectively no physical cost, so the hand no longer has to travel, and the problem that Chapter 5 addresses with control-display gain does not arise.

It would be easy to conclude from this that hand-cursor targeting is a transitional technology, useful only until eye tracking becomes readily available. This thesis takes a different position: the user still has to target, and relaxing the demand that targeting places on the user is beneficial whatever method does the targeting. Gaze pointing has a precision problem of its own. Eye-tracker accuracy, the constant small movements of fixation, and the difficulty of distinguishing looking-to-select from looking-to-read make gaze can make imprecise pointer for small or closely spaced targets [113, 172]. The response has been to relax the targeting demand at the interface: visionOS snaps the gaze pointer to the nearest interactive element and imposes generous minimum target sizes and spacing, and the hover effect that highlights the element about to be selected serves the same role as the in-focus highlight of the proxemic cursor. Sidenmark et al.'s weighted pointer [147] and cone-and-bubble techniques [148] carry the area-cursor idea directly into gaze and hand-ray pointing. The contribution of Chapter 4 can so be read not as

just a technique for hand-tracked cursors specifically but as further evidence for a principle that gaze-based systems have arrived at by a different route: when the pointer is noisy, let the system resolve intent to the nearest plausible target and keep the visual interface unchanged while it does so. The density bound of Chapter 5 is the cost of that principle, and it applies to gaze snapping in the same way; the minimum spacing rules in headset design guidelines are the same constraint expressed as a recommendation rather than a measurement.

Where hand-only cursor interaction retains its place is where this thesis was designed for. Planar public and shared displays are used at a distance, by many people in succession, often for a few seconds each, without per-user calibration and without a worn device (as is required by current gaze systems). Accurate remote eye tracking under those conditions, at distance, for uninstructed walk-up users, is not available in the way it is inside a headset that sits at a fixed distance from a calibrated pair of eyes. A cursor driven by the hand also has properties gaze lacks that matter for shared displays: it is visible to bystanders, so a second person can follow what the first is doing; it can be stopped, whereas gaze cannot be told to *stop looking*; and it maps onto the same interface that touch and controller users see, preserving the cross-modal parity discussed in Section 2.2. The two paradigms are therefore best understood as sharing an activation type and a targeting principle while making different decisions about which channel points, and those decisions follow from the deployment context rather than from one being a stopgap for the other.

6.3.2 Multimodal Interactions

Section 2.5 discounted multimodal activation, including voice, gaze, and ultrasonic haptics [20, 55], on the grounds that the additional hardware complicated touchless display deployment. Inside a headset that argument no longer holds: gaze tracking and hand tracking are both built in, and the multimodal combination is the default rather than the augmentation. On planar displays it still holds, and the techniques here were chosen so that they need nothing beyond a single hand-tracking sensor. The multimodal literature does, however, sharpen the argument for pinch. The reason gaze-and-pinch, voice-and-point, and haptic-confirmed selection all improve on push is that each supplies a confirmation signal that is separate from the pointing movement, and the sense-of-agency work on touchless selection [41] suggests this separation is what users experience as control. Pinch is the cheapest such interaction available to a hand-tracking system, as the confirmation is generated by the user's own fingers. Therefore, what this thesis does is add to continued evidence of separating targeting from activation, ensuring that targeting can remain stable.

6.3.3 Towards a More General Theory of Interaction

This thesis does not offer a theory of mid-air gesture interaction, and it does not have the evidence to claim one. It does offer a step towards one. Every cursor-based selection, whatever the hardware, has three parts: the *activation* that commits a selection, the *targeting* rule that decides which widget a pointer position refers to, and the *traversal* mapping that turns body movement into pointer movement. Each part places a precision demand on someone, and the demand can sit with the user or with the system. The commercial default (push gestures, point cursor, 1:1 mapping) this thesis started from leaves all three with the user: a point cursor that must overlap its target, a push or dwell that must be held without drifting, and a direct mapping that makes the arm cover the whole display. The techniques outlined in this thesis each move one of those demands to the system. Gaze-and-pinch relaxes targeting by snapping to the nearest element, removes traversal by letting the eyes do it, and stabilises activation with the same pinch. The general claim is that mid-air interaction improves when precision is taken from the user and given to the system, and that this holds regardless of which sensor is doing the tracking or which channel is doing the pointing.

Each of these moves has a cost. Decoupled activation needs a gesture the sensor can see and the user can feel. Relaxed targeting needs empty space around targets, which is why density bounds it. Reduced traversal costs precision, needing relaxed targeting and selection. A design that makes all three moves at once is what this thesis recommends; a design that makes none is what it set out to improve. What a theory would add, and what this thesis cannot provide, is a quantitative account of how those costs trade against one another, of the kind Fitts' law supplies for traversal alone. That would need comparisons this thesis did not run: an area cursor and a point cursor measured at the same gain levels, so that the interaction between relaxed targeting and amplified traversal is estimated rather than assumed.

6.4 Looking Into The Future: Reflecting On What Stands

An empirical thesis using interaction technology is built on hardware that will be obsolete before the thesis is old, and it is worth being explicit about which of its findings will go with the hardware.

What will change is everything tied to the apparatus. The sensors, displays, and tracking used here will be superseded, and as hardware improves so will sensing: lower latency, less positional noise, more reliable finger tracking, larger tracking volumes, and better performance at distance. Every threshold, gain range, spacing rule, and error rate reported in this thesis is a measurement of what one generation of hand tracking allowed, and each should be expected to shift as that generation is replaced. The commercial landscape will move too; push and dwell were the industry standard when Chapter 3 was designed and have already given way to pinch in the headset market.

This thesis does contain findings which are expected to outlast the current hardware: The first is that pinch, or more generally a passive-haptic, binary activation decoupled from the pointing movement, is the better activation method for hand-based mid-air input. The reasons pinch outperformed push and dwell are not due to sensing: finger-to-thumb contact gives the user tactile and proprioceptive confirmation of the state change, the gesture has an unambiguous on and off, and it can be performed without displacing the hand that is pointing. Those are properties of the human hand, and they will be as true of future sensors as they were of the one used here. The independent convergence of the XR industry on pinch is the strongest available evidence that the finding generalises, and the emergence of wrist-worn sensing that detects pinch from muscle activity rather than from a camera suggests the technique will outlive optical hand tracking altogether.

The second is that the precision problem of mid-air pointing should be relaxed at the interface, as it cannot be sensed away. The instability that produces target slips, selection errors, and lazy targeting comes from the arm, not the sensor: an unsupported hand trembles, drifts, and accumulates fatigue the longer it is held up. Better hardware will remove the noise the sensor adds, but it will still report the body's movement. Therefore, techniques that allow for relaxed pointing, whether the area cursor, gaze snapping, or whatever comes next, will remain necessary however good tracking becomes. Their need for empty space around targets, which produced the density bound in Experiment 5, will remain with them, and future layouts will have to design around it. The particular mechanism used here, resolving to the nearest widget, will likely be replaced by something that infers intent from more than proximity to widgets, however the principle that the system helps the user target should stand.

Finally, whether the cursor interaction metaphor itself survives on planar touchless displays is also an open question: this thesis kept it for reasons of parity and retrofit, and if such displays come to be driven by gaze or by direct spatial manipulation the cursor may become a legacy of the time this thesis was written.

Chapter 7

Conclusions & Recommendations

7.1 Introduction

This thesis made the following assertion in its Introduction (Section 1.3):

> This thesis demonstrates that using finger-to-thumb pinch gestures for state change activation, in conjunction with increased control-display gain and dynamic area cursor pointing, can improve input efficiency in cursor-based touchless interfaces. This combination of interaction techniques reduces the need for pointing precision and supports relaxed targeting behaviours, whilst also mitigating concerns relating to depth ambiguity often seen in spatial interfaces. The benefits of these techniques scale across a range of interface configurations, e.g., from sparse to dense target layouts, from small to large displays, and for both activation and manipulation operations. As a result, users can interact more quickly and with less physical effort, making touchless input a more viable option for everyday computing tasks on planar displays.

In the chapters that followed, experiment results were presented as evidence in support of this statement. Chapter 3 investigated activation techniques for mid-air cursor interfaces, comparing the commercial standard push and dwell gestures against finger-to-thumb pinch, demonstrating that pinch reduces acquisition time and target slips during continuous control tasks, contributing a better understanding of how the choice of activation mechanism shapes selection. Chapter 4 introduced the proxemic cursor, a dynamic area cursor that automatically targets the nearest widget with minimal visual change to the interface, and showed across multi-widget continuous and mixed continuous/discrete task layouts that area-based targeting reduces both widget acquisition and inter-widget transition time without sacrificing accuracy. Chapter 5 then situated these techniques in the broader design space of control-display gain, display size, and target density, demonstrating that increased control display gain and an area cursor are mutually beneficial techniques that can enhance the efficiency of mid-air cursor control. The experiments

in these chapters found users could use these interaction techniques successfully and that the combinations recommended substantially reduced the physical demand of mid-air interaction while preserving its accuracy. This chapter now summarises this research and revisits each of the research questions, discussing how they were addressed and summarising their answers. It also summarises the main contributions of this research and discusses areas for future work, with limitations and future directions appearing alongside the recommendations and conclusions to which they are most relevant rather than being given a separate section.

7.2 Research Questions

This section will address the three research questions stated in Section 1.4, by summarising the key experiment findings presented in this thesis. **RQ1** considers the activation method itself and was evaluated first so that the following experiments (which take a more thorough look at cursor movement) could be paired with a usable and reliable activation gesture. **RQ2** considers the act of controlling cursor position across a range of task configurations (e.g., different displays, different target densities, selection vs manipulation tasks), with an aim of adapting facets of cursor functionality to enable more efficient selection. Finally, **RQ3** considers how these new interaction techniques affect the physical demand of mid-air interaction, since balancing interaction efficiency against cognitive and physical demand is desirable for user experience. The following discussions answer these questions and, collectively, can inform future interaction design for mid-air cursor interfaces.

RQ1 - How does the choice of target activation technique affect the usability of a mid-air user interface?

Target activation techniques impact the usability of mid-air interfaces because of their effect on interaction time and cursor stability, and because they influence the targeting strategies users adopt (e.g., to mitigate limitations of an activation gesture). Experiment 1 (Chapter 3) showed that two variations of Pinch activation significantly outperformed Push and Dwell in a single-slider control task: target acquisition times were faster for both Pinch conditions than for Push, and Dwell was inherently slower because of the additional dwell period. Experiments 2 & 3 (Chapter 4) compared Push and Pinch across continuous control and discrete selection tasks, with similar findings.

In these experiments, Pinch gestures consistently reduced target acquisition time, accelerated transitions between widgets, and showed higher task success rates while being the most preferred method. By separating selection from hand translation, Pinch reduces the depth-axis ambiguity inherent in Push activation and reduces vulnerability to target slips from incidental cursor movement; Dwell, by contrast, suffers a cumulative re-dwell cost whenever continuous

control is incidentally lost, making it poorly suited to continuous manipulation. Push and Dwell have been the default activation methods used across most touchless input platforms (e.g., UltraLeap TouchFree and Intel RealSense used Push gestures, and Xbox Kinect primarily used Dwell), though these findings show Pinch to be a more compelling alternative. Based on these results, Pinch was used as the main activation gesture in the remainder of the thesis, and the area cursor nature of the Area Pinch method motivated further study of the dynamic area cursor approach investigated later in the thesis.

RQ2 - How can we reduce the need for traversal and selection accuracy in a mid-air user interface across different display sizes and widget densities?

Area cursor targeting reduces the need to fully move the cursor within the precise bounds of a target, which has a positive effect on input efficiency. Experiment 1 (Chapter 3) first introduced the principle through the *Area Pinch* condition, which produced the fastest acquisition times and was most preferred by users. Experiments 2 & 3 (Chapter 4) continued the investigation of area cursor targeting, this time introducing the *proxemic cursor* as a more general instance of a dynamic area cursor that reduces the motor precision required for target acquisition. This was validated across a range of interface layout conditions and task types, including discrete button selection and continuous slider control. Experiments 4 & 5 (Chapter 5) then investigated area cursor use across a range of display sizes and target densities, coupled with increased control-display gain in search of increased throughput.

In these experiments, results showed that dynamic area cursors reduce widget acquisition times and inter-widget transition times while preserving accuracy. Findings suggest that users adapt their targeting strategy based on the layout configuration, approaching targets more closely when widget boundaries are narrow and from further away when the layout permits, all without altering the interface's visual layout or cursor-and-click interaction patterns. When paired with increased control-display gain, the area cursor absorbed the imprecision and jitter that high gain would otherwise amplify while maintaining accuracy across widget densities and screen sizes. However, accuracy still diminished on the smaller displays and within the densest layouts, where the area cursor has the least room to expand; larger displays absorbed the precision penalties more effectively due to the additional space available for the area cursor to expand into. These findings answer RQ2 by showing that selection and traversal accuracy of mid-air cursor interaction can be mitigated through area-based targeting.

RQ3 - How can we design mid-air interactions such that the required effort is reduced while interaction efficiency is increased?

This thesis explored mechanics to reduce the physical effort required for mid-air interaction, by aiming to reduce necessary physical movement and decrease interaction time and error. Chap-

ter 3 showed that pinch activation gestures mitigate depth ambiguity and reduce the amount of physical effort needed to activate the cursor. Chapter 4 demonstrated that dynamic area cursors can successfully alleviate the precision required to select targets with the cursor. Chapter 5 then integrated these techniques with increased control-display gain, evaluating their performance across varying display sizes and widget densities, with higher gain levels reducing the necessary physical hand movement needed to reach targets.

In these experiments, results demonstrated that combining area-based targeting with increased control-display gain minimises physical effort while maintaining or improving task and selection times. The area cursor absorbed the input noise and jitter that higher gain would otherwise result in selection errors, resulting in a net reduction in hand travel distance, marking the mutually beneficial combination of dynamic area cursors and increased Control-Display Gain. This reduction was driven by a consistent *lazy targeting* behaviour, in which users kept the cursor near a comfortable resting position and relied on the area cursor and small wrist/hand movements to target and activate the intended widget from a distance rather than fully traversing to it. These physical benefits were most pronounced on the largest display, where mean hand movement at high gain was less than half that of the baseline, effectively offsetting the longer arm movements typically required by larger screens. Further, perceived fatigue did not rise on the larger displays under high gain and was in fact lowest on the larger display with high gain, indicating that the combination reduces required effort even in the configurations traditionally considered most fatiguing, such as interacting with high-density widget layouts on large displays. These findings answer RQ3 by showing that combining the techniques developed in answering RQ1 and RQ2 reduces physical effort while increasing interaction efficiency, provided that control-display gain, display size, and target density are designed to work together.

7.3 Thesis Contributions

This thesis makes novel contributions which inform the design of cursor-based mid-air gesture interfaces on planar displays. Its main contributions are: (1) an empirical comparison of activation gestures for mid-air cursor interaction; (2) the proxemic cursor, a dynamic area cursor for mid-air pointing; and (3) a characterisation of the relationship between control-display gain, display size, and target density when these are combined with the proxemic cursor. This section summarises these contributions.

C1: Activation Gestures for Mid-Air Cursor Interfaces

This thesis contributes experimental comparisons of how target activation techniques (push, dwell, pinch) affect usability in mid-air cursor-based interfaces. While push gestures are a common industry standard, users often experience uncertainty about state changes and struggle to maintain a consistently activated state. The ambiguous use of depth frequently leads to target

slips and a diminished sense of control. The experimental findings demonstrate that pinch gestures are consistently faster to acquire, more accurate to maintain, and more preferred by users. Pinch proves to be a highly effective alternative because it decouples activation from cursor position and provides a clearly defined binary state transition via the passive haptics and proprioception of the finger and thumb touching. This mechanism enables the hand, and therefore the cursor, to remain steady during activation, successfully mitigating target slips and overcoming the physical instabilities inherent in the push gesture.

C2: Dynamic Area Cursors and the Proxemic Cursor

This thesis contributes an investigation into improving targeting efficiency and accuracy while reducing the physical effort required for mid-air interaction. It presents the *proxemic cursor*, which uses the same nearest-target selection as the *Bubble Cursor* [71] without rendering the bubble. Its contribution is not the targeting mechanism itself but its application to continuous widgets, whose activation boundaries move as they are manipulated, and its pairing with increased control-display gain (C3). The experimental findings demonstrate that the dynamic area cursor significantly enhances usability by reducing both initial widget acquisition time and inter-widget transition time without sacrificing accuracy. Furthermore, users benefited from area-based targeting even though the proxemic cursor's activation region was never drawn: they were able to acquire control more quickly and via more relaxed movements, adapting their targeting strategy to the available spatial layout rather than reverting to traditional overlap-based selection.

C3: Control-Display Gain in Mid-Air Cursor Interaction

This thesis contributes an investigation into the optimisation of cursor control for mid-air interfaces, specifically characterising the relationship between control-display gain, area-based targeting, display size, and target density. While dynamic area cursors reduce the need for precise aiming, traversing a spatial environment remains physically demanding; however, increasing control-display gain to reduce arm movement typically degrades precision. The experimental findings demonstrate that a dynamic area cursor and elevated control-display gain form a mutually beneficial combination. The area cursor effectively absorbs gain-amplified noise, allowing for efficient and accurate target selection with smaller hand movements, even on large displays with dense target layouts.

7.4 Designing Efficient Interaction Techniques for Mid-Air Gesture Interfaces

The goal of this thesis, as reflected in its title, was to gain an improved understanding of how to design efficient interaction techniques for mid-air cursor interfaces. Five experiments gave insights into how cursor movement (towards a target) and activation (of a 'selected' target) can be improved through novel interaction design. The experimental chapters presented in the thesis contribute a set of design recommendations for cursor-based mid-air interfaces on planar displays. This thesis explored hand-based mid-air interaction; while the recommendations below may extend to peripheral-based input (e.g., controller or stylus) or to immersive XR environments, such extensions are outwith the scope of this thesis and are left to future work. The design recommendations presented below consolidate the empirically supported design guidelines given in Chapter 3, Chapter 4, and Chapter 5, and include explicit notes on the scope within which each recommendation has been validated. Each recommendation is followed by a statement of the scope within which it was validated, and two limits that apply to all of them, participant handedness and the fixed interface design, are set out at the end of this section.

DR1: Use Pinch Gestures for Hand-Based Activation

Where a hand-tracking sensor reliably detects finger-to-thumb pinch, pinch should be the default activation gesture for cursor-based mid-air interfaces. Push should be considered only when sensor limitations or accessibility constraints prevent pinch detection, and where push must be used, it should be combined with a dynamic area cursor to mitigate the Heisenberg effect; this recovers a portion of the performance lost to depth-axis ambiguity but does not eliminate it. Dwell is appropriate only for discrete activation in low-throughput interfaces; it should not be used for continuous control because incidental release imposes a cumulative re-dwell cost that compounds over the task.

Implementation Notes: Pinch should be defined as a two-state interaction, with state change occurring using a threshold of finger-to-thumb distance (since finger-to-finger contact is difficult to establish via typical optical sensing). The experiments here used a $\leq$20 mm threshold for activation and $>$20 mm for release (Section 4.4.2); a more relaxed 35 mm threshold was used in Chapter 3 but produced occasional false activations and should be regarded as an upper bound. When implementing the cursor interaction, the position of the on-screen cursor should ideally be mapped to the palm position rather than a fingertip position; this allows the cursor to remain stable during the pinch gesture, since pinching causes finger movement, even in fingers not directly involved in the pinch. Continuous feedforward on activation progress should be provided by shrinking the cursor proportionally as the finger-thumb gap closes, and on registered activation, the cursor should change to a distinct visual state (the experiments here used grey-to-black with a small size reduction from 30 px to 25 px on a 27" display) so users receive confirmation

of the binary state transition. Where push must be used as a fallback, the activation threshold should be a *fixed* forward displacement from the resting hand depth (100 mm in Chapter 4) rather than a percentage offset from a moving-window depth average; the percentage scheme used in Chapter 3 shifted during interaction and proved less predictable for participants. Dwell, where used for low-throughput discrete activation, should provide visible countdown feedback (e.g., a circular progress ring around the cursor) with a dwell period in the range used here (800 ms) so users can confirm or abandon a pending activation before it triggers.

Scope: The pinch advantage has been characterised in this thesis for cursor-based 2D interaction on planar displays using an Ultraleap hand tracking sensors. Extending the recommendation to sensors with reduced finger-tracking fidelity, to gloved interaction, and to 3D manipulation tasks requires further empirical work. Pinch also tested on participants with the physical capability to do so; an accessible deployment should retain a non-pinch fallback.

DR2: Use a Dynamic Area Cursor for Widget Targeting

A dynamic area cursor that automatically targets the widget nearest to the cursor should be the default pointing technique for cursor-based mid-air interfaces. The proxemic cursor investigated in this thesis allowed for more relaxed spatial targeting resulting in less selection error and faster interaction times. The use of the dynamic area cursor also alleviated instability at increased control-display gain allowing for precise selections.

Implementation notes: The 'nearest' widget should be defined as the widget whose bounding box has the minimum Euclidean distance from the current cursor position. For widgets containing both a static and a movable element (e.g., sliders with a track and a handle), the distance should be computed to the *handle* bounding box, rather than the full track; Chapter 4 found this aligned with users' expectations of how neighbouring sliders should respond as handles move, and produced a boundary between adjacent widgets that shifts dynamically with handle position. Exactly one widget should be in-focus at all times: there is no idle "no widget" state, and the cursor should always have a designated target regardless of where it sits in the interaction space. Compact widget arrangements remain workable, down to approximately 64 px between adjacent widgets, enabling information-dense interfaces without sacrificing acquisition or transition performance. No clutching mechanism is required: cursor control should remain direct and continuous whenever the hand is in the sensor's tracking volume, preserving the spatial mapping between hand and cursor.

Scope: The proxemic cursor has been validated for cursor-based 2D interaction on planar displays using Ultraleap hand-tracking sensors, across slider and button widgets, single and multi-widget layouts, four display sizes (27"–55"), and target densities. At the highest density tested the surrounding distractor widgets begin to constrain the cursor's expansion region; use beyond this, or with substantially different widget layouts, warrants further evaluation. The proxemic cursor adopted a minimal visual design that resembled regular point cursor. This kept

the standard cursor-click interaction metaphor while relaxing precision. While this is a departure from other dynamic area cursors [71] where the visual size of the cursor is continually altered to expand towards the nearest target, the results in this thesis indicate that the minimal design worked well (in terms of, task time, transition time, success rate and preference). However, this thesis did not compare the visual design of the proxemic cursor to alternatives, so it cannot claim that visible feedback about the activation region is unnecessary, only that the minimal design was sufficient. Substantiating the stronger claim would require a within-subjects comparison that holds the targeting mechanism constant and varies only its rendering: no visible region (as here), a small translucent spot as in *DynaSpot* [32], and a fully visible bubble as in the *Bubble Cursor* [71], across a range of target densities. Task time, error rate and activation distance would show whether seeing the region changes performance or how much of it users exploit, alongside subjective ratings of workload and preference.

DR3: Tailor Control-Display Gain towards Display Size and interface complexity

Choice of an appropriate control-display gain should be informed by the physical display size and the requirements of the pointing task. Smaller displays (e.g., $\leq$ 32" horizontal diameter) favour lower gain ratios that approximate a direct mapping of cursor to physical hand movement (CDG $\sim$1.0–1.5 $\times$), since the extremities of the smaller display size remain within comfortable reach. Larger displays benefit from higher gain (CDG $\sim$1.5–3.5 $\times$) as the area cursor has more visual space in which to absorb gain amplified instability, and because the alternative is to require fatiguing arm motion approaching the limits of reliable sensing and ergonomic reach. Discrete ballistic selection allows for increased control-display gain as the area cursor can absorb overshoot, whereas fine continuous control tasks such as slider manipulation favour conservative multipliers, since once the handle has been acquired the area cursor no longer apply and gain governs cursor behaviour directly. This mapping was consolidated in Table 5.7.

Implementation notes: Where the interface mixes continuous-control widgets (e.g., sliders) with discrete-selection widgets (e.g., buttons), gain can be switched per-widget: lowered while a continuous widget is held and restored on release, tying the change to the activation gesture so users are not asked to perform a separate mode switch. The same per-widget approach extends to regional gain changes in interfaces that mix dense (low-gain) and sparse (high-gain) regions, such as keyboards.

Scope: The gain ranges above were derived from constant-multipliers deliberately chosen to give a clean characterisation of the gain $\times$ display $\times$ density design space; they represent a single-value compromise rather than an optimum. A constant multiplier was used throughout to preserve the direct spatial mapping between hand and cursor position; non-linear, velocity, or acceleration-based mappings, semantic-pointing variants that bias gain by widget importance, and adaptive coordinate-space mappings that reflect what users can comfortably reach all lie outside the scope of this thesis and are left for future work. Whether such mappings remain

transparent to users in practice, or introduce a new source of confusion as the cursor speed changes mid interaction, is itself an open research question. Multipliers below 1.0 were likewise not investigated. As arm reach is a proportion of a user's height, a less than 1:1 mapping does not necessarily force a clutching mechanism; it only does so once the display is large enough that arm movements can no longer span it. The conditions under which sub 1:1 mapping become useful and the trade-off against the increased arm motion they demand are left for future work.

DR4: Design Widget Layouts to Give the Area Cursor Room to Expand

When designing new user interface layouts for use with mid-air cursor input, widget layouts should be designed with the area cursor configuration in mind. Designers should consider widget placement in terms of localised groupings and the empty space that surrounds them rather than in terms of absolute target density. Target density acts as a moderator of how much room the area cursor has to expand before encountering a another widget. Within denser clusters of targets, generous empty space at the edges of the interface gives the area cursor room to absorb overshoot, and in mixed-density layouts, empty space can encourage deliberate overshoot as a disambiguation strategy between closely spaced targets. Target placement should follow the targeting patterns users adopt under area-based targeting. Frequently used or high-priority widgets, and widgets used in fine-pointing tasks, should sit towards the centre of the display, within the user's comfortable range of motion and consistent with the centre-biased cursor positions observed throughout this thesis. Discrete widgets (e.g., buttons) can be placed more peripherally because users are willing to activate them from greater distances when an area cursor is available; widgets requiring continuous manipulation (e.g., sliders) should instead be positioned so their full traversal distance stays within the sensor's reliable tracking bounds.

Implementation notes: Inter-widget spacing should be kept above the cursor's visual size to preserve the area cursor's effectiveness. Density should be computed using a measure such as Blanch's [19] ratio of total widget area to available screen or by approximating density based on the size and number of targets available on screen (e.g., an on-screen keyboard has higher density than an on-screen numberpad); guidance can be found in Table 5.7. Sensor placement contributes to the available interaction space and should be considered alongside UI layout: the experiments in this thesis positioned the Ultraleap tracker 15 cm below and 10 cm in front of the display for desk setups, or on a table approximately 2.5 m in front of a 55" display for distal setups, in both cases angled $\sim$15° towards the user. This angling (in conjunction with the sensor field of view and operating range) extends the comfortable tracking volume by allowing a relaxed posture rather than requiring the user to hold their hand directly above the sensor. Independent of density, the physical extent of the interface plus any spatial offset introduced by area-based acquisition must remain inside the sensor's tracking volume; where the size of a display would leave this volume, higher control-display gain (DR3) should be used.

Scope: The density and layout recommendations above have been validated for the wid-

get types (sliders, buttons), display sizes (27"–55"), and density range (0.075–0.6) studied in this thesis. The density-0.6 ceiling appears as a practical limit: at this density the area cursor's expansion is constrained by adjacent distractors, and error rates rise on the smallest displays even though larger displays absorb the cost more effectively. Layouts that mix dense and sparse regions within a single interface are particularly likely to benefit from the adaptive gain mechanisms flagged as future work in DR3.

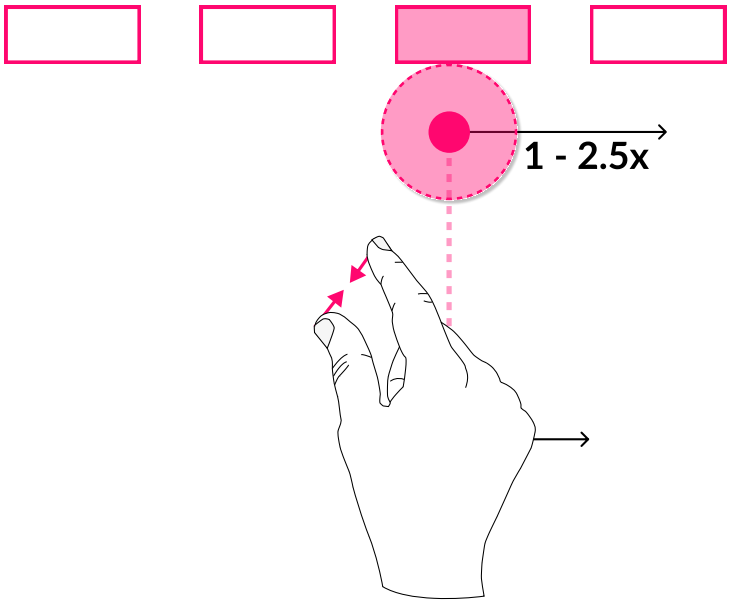


Figure 7.1: An illustration of the main design recommendations supported by this thesis. Mid-air cursor interfaces should use a control-display gain proportional to the display size and target density (DR3), a dynamic area cursor that allows proximal target selection (DR2), and a pinch activation gesture to confirm selection or change user interface state (DR1). Note the magnitude of the area cursor is only shown here for illustration

Scope Common To All Recommendations

Across the five experiments there were 95 participant sessions, of which 84 (88%) involved right-handed participants, four left-handed (one in Experiment 1, two in Experiment 2 and one in Experiment 3) and seven ambidextrous; Experiments 4 and 5 included no left-handed participants. All participants interacted with their dominant hand (ambidextrous participants used whichever hand they preferred), and the apparatus and on-screen layouts were identical for everyone. This distribution is close to that of the general population, but it leaves too few left-handed participants to analyse handedness as a factor.. Prior work suggests the consequences are limited: touchless pointing shows much smaller performance differences between hands than a mouse or stylus [76, 89], as it places less demand on fine feedback control. The more likely influence is directional rather than motor. Some tasks were laterally asymmetric (sliders in Experiment 4 started at their leftmost position and the start position in Experiment 5 was at the left

edge of the display), so most movements were outward for a right hand but would be across the body for a left hand, which may differ in comfort and fatigue. Layout recommendations (DR4) should therefore not assume a particular hand, and interfaces may benefit from being mirrored for left-handed use; confirming this requires a sample balanced by handedness.

The interface design was also held constant, and different choices could plausibly have changed the size of some effects. Widgets were large: slider handles were more than twice the diameter of the cursor and buttons were larger still. This favours precise targeting, so the advantage of the area cursor over a standard cursor (DR2) is likely conservative and would be expected to grow with smaller widgets; equally, the finding that 60% of button activations fell inside the button reflects how large the buttons were and may not hold for small targets. The styling was plain and high-contrast, with no surrounding content. Real interfaces add visual search and clutter, which would lengthen task times in all conditions and could make the in-focus outline less salient; the relative differences between techniques concern motor behaviour and are less likely to change, but this was not tested. Feedback was visual only. Audio feedback can compensate for the missing tactile cues of mid-air input [59, 135].

7.5 Thesis Conclusion

Mid-air gesture interaction enables users to control touchless displays without physical contact. Touchless interfaces predominantly inherit the cursor-and-click interaction style from conventional input modalities, which requires a level of precision that is difficult to sustain with an unsupported hand. Pointing accuracy is reduced by hand instability and the fatigue of sustained mid-air pointing, and activation is undermined by the absence of a clear way to both initiate and perceive a state change during selection. This thesis has demonstrated that finger-to-thumb pinch activation, dynamic area cursor pointing, and increased control-display gain can together improve the efficiency of cursor-based mid-air interaction on planar displays. Pinch was shown to be a faster, more stable, and more preferred activation technique than the push and dwell gestures that dominate commercial touchless platforms, decoupling activation from cursor position and giving users a clear binary state transition through the passive haptics of finger-to-thumb contact. This thesis introduced the proxemic cursor, a dynamic area cursor that automatically targets the widget nearest to the cursor, and was shown to reduce both widget acquisition and inter-widget transition times without sacrificing accuracy, and to do so without altering the established interaction metaphors of cursor-and-click interaction. Increasing control-display gain and the proxemic cursor were shown to be a mutually beneficial combination: the area cursor absorbs the imprecision that higher gain would otherwise amplify, while higher gain reduces the physical hand movement required to traverse the display. The benefits scaled across small and large displays, sparse and dense target layouts, and both continuous and discrete pointing tasks, reducing acquisition time, transition time, and hand movement with no corresponding increase

in perceived fatigue. Together, these contributions provide a clearer understanding of how mid-air cursor interaction can be designed for efficiency, and offer a set of empirically supported design recommendations on which future touchless interfaces can build to make mid-air input a more viable option for everyday computing tasks on planar displays.

Appendices

Appendix A

Experiment 1

Figure A.1 shows the agreement questions and Figure A.2 shows the ranking question used in Experiment 1. See Section 3.4.5 for a full description.

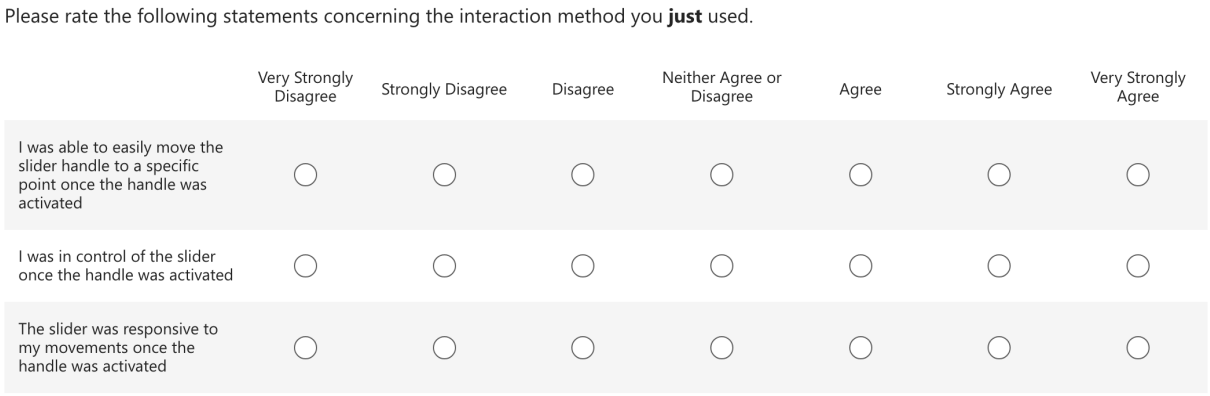


Figure A.1: Agreement survey used in Experiment 1



Figure A.2: Technique rank question used in Experiment 1

Appendix B

Experiment 2

Figure B.1 shows the ranking question used in Experiment 2. See Section 4.5 for a full description.



Figure B.1: Technique rank question used in Experiment 2

Appendix C

Experiment 4

Figure C.1 shows the Borg CR10 [21] scale used in Experiment 4. See Section 5.6 for a full description.

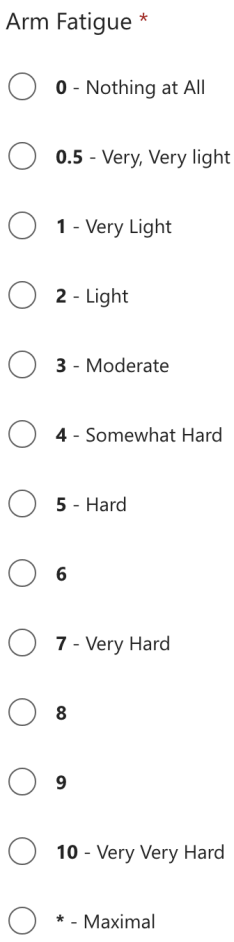


Figure C.1: Borg CR10 scale question used in Experiment 4

Appendix D

Experiment 5

Figure D.1 shows the perceived control and difficulty survey, and Figure D.2 shows the Borg CR10 [21] scale used in Experiment 4. See Section 5.7 for a full description.

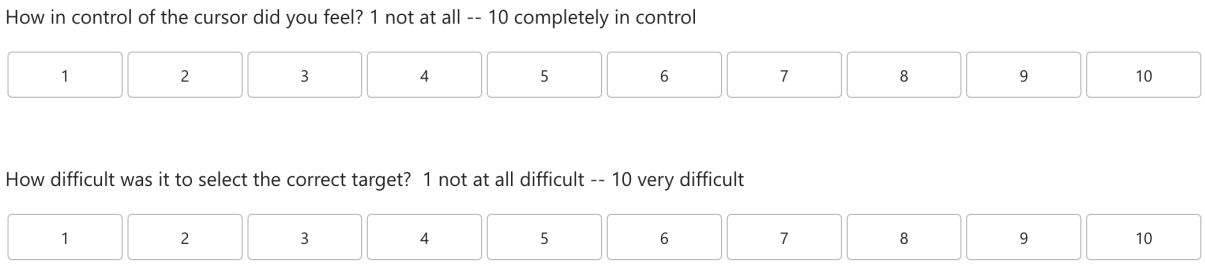


Figure D.1: Perceived control and difficulty survey used in Experiment 5

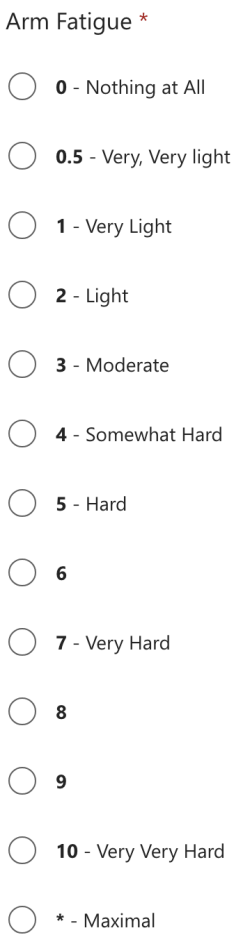


Figure D.2: Borg CR10 scale question used in Experiment 4

Bibliography

[1] A, N., AND NIELSENJAKOB. Gestural interfaces. *Interactions* (Sept. 2010).

[2] ACCOT, J., AND ZHAI, S. Beyond Fitts' law: models for trajectory-based HCI tasks. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (New York, NY, USA, Mar. 1997), CHI '97, Association for Computing Machinery, pp. 295–302.

[3] ACCOT, J., AND ZHAI, S. Refining Fitts' law models for bivariate pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2003), CHI '03, Association for Computing Machinery, pp. 193–200.

[4] ACKAD, C., CLAYPHAN, A., TOMITSCH, M., AND KAY, J. An in-the-wild study of learning mid-air gestures to browse hierarchical information at a large interactive public display. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (New York, NY, USA, Sept. 2015), UbiComp '15, Association for Computing Machinery, pp. 1227–1238.

[5] ADKINS, A., CANALES, R., AND JÖRG, S. Hands or Controllers? How Input Devices and Audio Impact Collaborative Virtual Reality. In *Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, Oct. 2024), VRST '24, Association for Computing Machinery, pp. 1–12.

[6] AIGNER, R., WIGDOR, D., BENKO, H., HALLER, M., LINDBAUER, D., ION, A., ZHAO, S., AND KOH, J. Understanding mid-air hand gestures: A study of human preferences in usage of gesture types for hci. *Microsoft Research TechReport MSR-TR-2012-111 2*, MSR-TR-2012-111 (2012), 30.

[7] ARDITO, C., BUONO, P., COSTABILE, M. F., AND DESOLDA, G. Interaction with Large Displays: A Survey. *ACM Computing Surveys 47*, 3 (Feb. 2015), 46:1–46:38.

[8] ARGELAGUET, F., AND ANDUJAR, C. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics 37*, 3 (May 2013), 121–136.

[9] BACHMANN, D., WEICHERT, F., RINKENAUER, G., BACHMANN, D., WEICHERT, F., AND RINKENAUER, G. Review of Three-Dimensional Human-Computer Interaction with Focus on the Leap Motion Controller. *Sensors 18*, 7 (July 2018). Company: Multidisciplinary Digital Publishing Institute Distributor: Multidisciplinary Digital Publishing Institute Institution: Multidisciplinary Digital Publishing Institute Label: Multidisciplinary Digital Publishing Institute.

[10] BACHYNSKYI, M., PALMAS, G., OULASVIRTA, A., AND WEINKAUF, T. Informing the Design of Novel Input Methods with Muscle Coactivation Clustering. *ACM Trans. Comput.-Hum. Interact. 21*, 6 (Jan. 2015), 30:1–30:25.

[11] BALOUP, M., PIETRZAK, T., AND CASIEZ, G. RayCursor: A 3D Pointing Facilitation Technique based on Raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2019), CHI '19, Association for Computing Machinery, pp. 1–12.

[12] BAN, R., HIRAO, Y., AND NARUMI, T. Determining the Target Point of the Mid-Air Pinch Gesture. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (Mar. 2021), pp. 492–493.

[13] BATMAZ, A. U., AND STUERZLINGER, W. The Effect of Rotational Jitter on 3D Pointing Tasks. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2019), CHI EA '19, Association for Computing Machinery, pp. 1–6.

[14] BAUDEL, T., AND BEAUDOUIN-LAFON, M. Charade: Remote Control of Objects Using Free-Hand Gestures. *Communications of the ACM 36*, 7 (1993), 28–35.

[15] BAUDISCH, P., CUTRELL, E., ROBBINS, D., CZERWINSKI, M., TANDLER, P., BEDERSON, B., AND ZIERLINGER, A. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-Operated Systems. In *Human-Computer Interaction–INTERACT '03* (Jan. 2003), IOS Press, pp. 57–64. Edition: Human-Computer Interaction–INTERACT '03.

[16] BELLOTTI, V., BACK, M., EDWARDS, W. K., GRINTER, R. E., HENDERSON, A., AND LOPES, C. Making sense of sensing systems: five questions for designers and researchers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2002), CHI '02, Association for Computing Machinery, pp. 415–422.

[17] BESSA SEIXAS, M. C., CARDOSO, J. C. S., AND DIAS, M. T. G. The Leap Motion movement for 2D pointing tasks: Characterisation and comparison to other devices. In

2015 International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS) (Feb. 2015), pp. 15–24.

[18] BLANCH, R., GUIARD, Y., AND BEAUDOUIN-LAFON, M. Semantic pointing: improving target acquisition with control-display ratio adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2004), CHI '04, Association for Computing Machinery, pp. 519–526.

[19] BLANCH, R., AND ORTEGA, M. Benchmarking pointing techniques with distractors: adding a density factor to Fitts' pointing paradigm. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2011), CHI '11, Association for Computing Machinery, pp. 1629–1638.

[20] BOLT, R. A. "Put-that-there": Voice and gesture at the graphics interface. *SIGGRAPH Comput. Graph. 14*, 3 (July 1980), 262–270.

[21] BORG, G. *Borg's perceived exertion and pain scales*. Borg's perceived exertion and pain scales. Human Kinetics, Champaign, IL, US, 1998. Pages: viii, 104.

[22] BOWMAN, D., WINGRAVE, C., CAMPBELL, J., AND LY, V. Using pinch gloves(tm) for both natural and abstract interaction techniques in virtual environments. Technical report tr-01-23,, Computer Science, Virginia Tech.

[23] BOWMAN, D. A., COQUILLART, S., FROEHLICH, B., HIROSE, M., KITAMURA, Y., KIYOKAWA, K., AND STUERZLINGER, W. 3D User Interfaces: New Directions and Perspectives. *IEEE Computer Graphics and Applications 28*, 6 (Nov. 2008), 20–36.

[24] BOWMAN, D. A., KRUIJFF, E., LAVIOLA, J. J., AND POUPYREV, I. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., USA, June 2004.

[25] BREWSTER, S. Overcoming the lack of screen space on mobile computers. *Personal and Ubiquitous computing 6*, 3 (2002), 188–205.

[26] BUCKINGHAM, G. Hand Tracking for Immersive Virtual Reality: Opportunities and Challenges. *Frontiers in Virtual Reality 2* (Oct. 2021).

[27] BUXTON, W. A three-state model of graphical input. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction* (NLD, Aug. 1990), INTERACT '90, North-Holland Publishing Co., pp. 449–456.

[28] CARD, S. K., ENGLISH, W. K., AND BURR, B. J. Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text

Selection on a CRT. *Ergonomics 21*, 8 (Aug. 1978), 601–613. _eprint: https://doi.org/10.1080/00140137808931762.

[29] CASIEZ, G., AND ROUSSEL, N. No more bricolage! methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (New York, NY, USA, Oct. 2011), UIST '11, Association for Computing Machinery, pp. 603–614.

[30] CASIEZ, G., VOGEL, D., BALAKRISHNAN, R., AND COCKBURN, A. The Impact of Control-Display Gain on User Performance in Pointing Tasks. *Human–Computer Interaction 23*, 3 (Aug. 2008), 215–250. _eprint: https://www.tandfonline.com/doi/pdf/10.1080/07370020802278163.

[31] CHAN, L.-W., KAO, H.-S., CHEN, M. Y., LEE, M.-S., HSU, J., AND HUNG, Y.-P. Touching the void: direct-touch interaction for intangible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2010), CHI '10, Association for Computing Machinery, pp. 2625–2634.

[32] CHAPUIS, O., LABRUNE, J.-B., AND PIETRIGA, E. DynaSpot: speed-dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2009), CHI '09, Association for Computing Machinery, pp. 1391–1400.

[33] CHATTERJEE, I., XIAO, R., AND HARRISON, C. Gaze+Gesture: Expressive, Precise and Targeted Free-Space Interactions. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (New York, NY, USA, Nov. 2015), ICMI '15, Association for Computing Machinery, pp. 131–138.

[34] CHATTOPADHYAY, D., AND BOLCHINI, D. Touchless circular menus: toward an intuitive UI for touchless interactions with large displays. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces* (New York, NY, USA, May 2014), AVI '14, Association for Computing Machinery, pp. 33–40.

[35] CHATTOPADHYAY, D., AND BOLCHINI, D. Understanding Visual Feedback in Large-Display Touchless Interactions: An Exploratory Study. Accepted: 2014-11-03T18:53:40Z.

[36] CHEN, C.-H., AND WANG, J.-L. The Semantic Meaning of Hand Shapes and Z-Dimension Movements of Freehand Distal Pointing on Large Displays. *Symmetry 12*, 3 (Mar. 2020), 329. Number: 3.

[37] CHEN, H.-J., LIN, C. J., AND LIN, P.-H. Effects of control-display gain and postural control method on distal pointing performance. *International Journal of Industrial Ergonomics 72* (July 2019), 45–53.

[38] CLARK, L. D., BHAGAT, A. B., AND RIGGS, S. L. Extending Fitts' law in three-dimensional virtual environments with current low-cost virtual reality technology. *International Journal of Human-Computer Studies 139* (July 2020), 102413.

[39] CLARKE, C., AND GELLERSEN, H. MatchPoint: Spontaneous Spatial Coupling of Body Movement for Touchless Pointing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, Oct. 2017), UIST '17, Association for Computing Machinery, pp. 179–192.

[40] COCKBURN, A., AHLSTRÖM, D., AND GUTWIN, C. Understanding performance in touch selections: Tap, drag and radial pointing drag with finger, stylus and mouse. *International Journal of Human-Computer Studies 70*, 3 (Mar. 2012), 218–233.

[41] CORNELIO MARTINEZ, P. I., DE PIRRO, S., VI, C. T., AND SUBRAMANIAN, S. Agency in Mid-air Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2017), CHI '17, Association for Computing Machinery, pp. 2426–2439.

[42] CRONIN, S., FREEMAN, E., AND DOHERTY, G. Investigating Clutching Interactions for Touchless Medical Imaging Systems. In *CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2022), CHI '22, Association for Computing Machinery, pp. 1–14.

[43] CZERWINSKI, M., ROBERTSON, G., MEYERS, B., SMITH, G., ROBBINS, D., AND TAN, D. Large display research overview. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2006), CHI EA '06, Association for Computing Machinery, pp. 69–74.

[44] DALSGAARD, T.-S., KNIBBE, J., AND BERGSTRÖM, J. Modeling Pointing for 3D Target Selection in VR. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, Dec. 2021), VRST '21, Association for Computing Machinery, pp. 1–10.

[45] DE LA BARRÉ, R., CHOJECKI, P., LEINER, U., MÜHLBACH, L., AND RUSCHIN, D. Touchless Interaction-Novel Chances and Challenges. In *Human-Computer Interaction. Novel Interaction Methods and Techniques* (Berlin, Heidelberg, 2009), J. A. Jacko, Ed., Lecture Notes in Computer Science, Springer, pp. 161–169.

[46] DELAMARE, W., DANIEL, M., AND HASAN, K. MultiFingerBubble: A 3D Bubble Cursor Variation for Dense Environments. CHI EA '22, Association for Computing Machinery, pp. 1–6.

[47] DOSTAL, J., HINRICHS, U., KRISTENSSON, P. O., AND QUIGLEY, A. SpiderEyes: designing attention- and proximity-aware collaborative interfaces for wall-sized displays. In *Proceedings of the 19th international conference on Intelligent User Interfaces* (New York, NY, USA, Feb. 2014), IUI '14, Association for Computing Machinery, pp. 143–152.

[48] DOVER, A., POOR, G. M., GUINNESS, D., AND JUDE, A. Improving Gestural Interaction With Augmented Cursors. In *Proceedings of the 2016 Symposium on Spatial User Interaction* (New York, NY, USA, Oct. 2016), SUI '16, Association for Computing Machinery, pp. 135–138.

[49] ELKIN, L. A., KAY, M., HIGGINS, J. J., AND WOBBROCK, J. O. An Aligned Rank Transform Procedure for Multifactor Contrast Tests. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, Oct. 2021), UIST '21, Association for Computing Machinery, pp. 754–768.

[50] ENGLISH, W., ENGELBART, D., AND BERMAN, M. Display-Selection Techniques for Text Manipulation. *IEEE Transactions on Human Factors in Electronics HFE-8*, 1 (Mar. 1967), 5–15.

[51] ESAKIA, A., ENDERT, A., AND NORTH, C. Large Display Interaction via Multiple Acceleration Curves and Multifinger Pointer Control. *Advances in Human-Computer Interaction 2014* (Nov. 2014), e691507.

[52] FALLAH, S., AND MACKENZIE, I. S. LeapBoard: Integrating a Leap Motion Controller with a Physical Keyboard for Gesture-Enhanced Interactions. *Journal on Multimodal User Interfaces* (Sept. 2025).

[53] FISCHER, F., YU, Z., JOHNS, C. A., KRISTENSSON, P. O., AND DUDLEY, J. J. Probabilistic In-Plane Detection for Mid-Air Virtual Surface Interactions. In *Proceedings of the 2025 ACM Symposium on Spatial User Interaction* (New York, NY, USA, Nov. 2025), SUI '25, Association for Computing Machinery, pp. 1–11.

[54] FITTS, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology 47*, 6 (1954), 381–391.

[55] FREEMAN, E., BREWSTER, S., AND LANTZ, V. Tactile Feedback for Above-Device Gesture Interfaces: Adding Touch to Touchless Interactions. In *Proceedings of the International Conference on Multimodal Interaction - ICMI '14* (2014), ACM, pp. 419–426.

[56] FREEMAN, E., BREWSTER, S., AND LANTZ, V. Do That, There: An Interaction Technique for Addressing In-Air Gesture Systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (2016), ACM Press, pp. 2319–2331.

[57] FREEMAN, E., VO, D.-B., AND BREWSTER, S. HaptiGlow: Helping Users Position their Hands for Better Mid-Air Gestures and Ultrasound Haptic Feedback. In *2019 IEEE World Haptics Conference (WHC)* (July 2019), IEEE, pp. 289–294.

[58] FREEMAN, E., WILLIAMSON, J., SUBRAMANIAN, S., AND BREWSTER, S. Point-and-Shake: Selecting from Levitating Object Displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2018), CHI '18, Association for Computing Machinery, pp. 1–10.

[59] FREEMAN, E., WILSON, G., VO, D.-B., NG, A., POLITIS, I., AND BREWSTER, S. Multimodal feedback in HCI: haptics, non-speech audio, and their applications. In *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations - Volume 1*, vol. 14. Association for Computing Machinery and Morgan & Claypool, Apr. 2017, pp. 277–317.

[60] FREES, S., KESSLER, G. D., AND KAY, E. PRISM interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction 14*, 1 (May 2007), 2–es.

[61] GALLO, L. A study on the degrees of freedom in touchless interaction. In *SIGGRAPH Asia 2013 Technical Briefs* (New York, NY, USA, Nov. 2013), SA '13, Association for Computing Machinery, pp. 1–4.

[62] GARZOTTO, F., AND VALORIANI, M. Touchless gestural interaction with small displays: a case study. In *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI* (New York, NY, USA, Sept. 2013), CHItaly '13, Association for Computing Machinery, pp. 1–10.

[63] GENTILE, V., FARZAND, H., BONACCORSO, S., ROCCHESSO, D., MALIZIA, A., KHAMIS, M., AND SORCE, S. User-Centered Evaluation of Different Configurations of a Touchless Gestural Interface for Interactive Displays. In *Human-Computer Interaction – INTERACT 2023* (Cham, 2023), J. Abdelnour Nocera, M. Kristín Lárusdóttir, H. Petrie, A. Piccinno, and M. Winckler, Eds., Lecture Notes in Computer Science, Springer Nature Switzerland, pp. 501–520.

[64] GENTILE, V., SORCE, S., MALIZIA, A., PIRRELLO, D., AND GENTILE, A. Touchless Interfaces For Public Displays: Can We Deliver Interface Designers From Introducing

Artificial Push Button Gestures? In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (New York, NY, USA, June 2016), AVI '16, Association for Computing Machinery, pp. 40–43.

[65] GHAEMI, Z., ENGELKE, U., ENS, B., AND JENNY, B. Proxemic maps for immersive visualization. *Cartography and Geographic Information Science 49*, 3 (May 2022), 205–219. _eprint: https://doi.org/10.1080/15230406.2021.2013946.

[66] GIBBS, C. B. Controller Design: Interactions of Controlling Limbs, Time-Lags and Gains in Positional and Velocity Systems. *Ergonomics 5*, 2 (Apr. 1962), 385–402. _eprint: https://doi.org/10.1080/00140136208930602.

[67] GRANDHI, S. A., JOUE, G., AND MITTELBERG, I. Understanding naturalness and intuitiveness in gesture production: insights for touchless gestural interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2011), CHI '11, Association for Computing Machinery, pp. 821–824.

[68] GREENWOLD, S. Spatial computing. Master's thesis, Massachusetts Institute of Technology, 2003.

[69] GROENEWALD, C., ANSLOW, C., ISLAM, J., ROONEY, C., PASSMORE, P., AND WONG, W. Understanding 3D Mid-Air Hand Gestures with Interactive Surfaces and Displays: A Systematic Literature Review. BCS Learning & Development.

[70] GROSSMAN, T., AND BALAKRISHNAN, R. Pointing at trivariate targets in 3D environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2004), CHI '04, Association for Computing Machinery, pp. 447–454.

[71] GROSSMAN, T., AND BALAKRISHNAN, R. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2005), CHI '05, Association for Computing Machinery, pp. 281–290.

[72] GUINNESS, D., JUDE, A., POOR, G. M., AND DOVER, A. Models for Rested Touchless Gestural Interaction. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction* (New York, NY, USA, Aug. 2015), SUI '15, Association for Computing Machinery, pp. 34–43.

[73] GUNA, J., JAKUS, G., POGAČNIK, M., TOMAŽIČ, S., AND SODNIK, J. An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. *Sensors 14*, 2 (Feb. 2014), 3702–3720.

[74] GUPTA, A., PIETRZAK, T., YAU, C., ROUSSEL, N., AND BALAKRISHNAN, R. Summon and Select: Rapid Interaction with Interface Controls in Mid-air. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton United Kingdom, Oct. 2017), ACM, pp. 52–61.

[75] GUTWIN, C. Improving focus targeting in interactive fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2002), CHI '02, Association for Computing Machinery, pp. 267–274.

[76] HABIBI, P., AND CHATTOPADHYAY, D. The impact of handedness on user performance in touchless input. *International Journal of Human-Computer Studies 149* (May 2021), 102600.

[77] HARRINGTON, K., LARGE, D. R., BURNETT, G., AND GEORGIOU, O. Exploring the Use of Mid-Air Ultrasonic Feedback to Enhance Automotive User Interfaces. In *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (New York, NY, USA, Sept. 2018), AutomotiveUI '18, Association for Computing Machinery, pp. 11–20.

[78] HASAN, K., AHLSTRÖM, D., AND IRANI, P. Ad-binning: Leveraging around device space for storing, browsing and retrieving mobile device content. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI '13, Association for Computing Machinery, p. 899–908.

[79] HERTZUM, M., AND HORNBÆK, K. Input techniques that dynamically change their cursor activation area: A comparison of bubble and cell cursors. *International Journal of Human-Computer Studies 65*, 10 (Oct. 2007), 833–851.

[80] HINCAPIÉ-RAMOS, J. D., GUO, X., MOGHADASIAN, P., AND IRANI, P. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2014), CHI '14, Association for Computing Machinery, pp. 1063–1072.

[81] HINCKLEY, K., PAUSCH, R., GOBLE, J. C., AND KASSELL, N. F. Passive real-world interface props for neurosurgical visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 1994), CHI '94, Association for Computing Machinery, pp. 452–458.

[82] HURSTEL, A., AND BECHMANN, D. Approach for Intuitive and Touchless Interaction in the Operating Room. *J 2*, 1 (Mar. 2019), 50–64. Number: 1.

[83] HUTCHINS, E. L., HOLLAN, J. D., AND NORMAN, D. A. Direct Manipulation Interfaces. *Human–Computer Interaction 1*, 4 (Dec. 1985), 311–338. _eprint: https://doi.org/10.1207/s15327051hci0104_2.

[84] INTEL. Intel RealSense Touchless Control Software, March 2021. Last Accessed 24/04/2022.

[85] JACOB, R. J. Eye movement-based human-computer interaction techniques: Toward non-command interfaces. *Advances in human-computer interaction 4* (1993), 151–190.

[86] JAKOBSEN, M. R., JANSEN, Y., BORING, S., AND HORNBÆK, K. Should I Stay or Should I Go? Selecting Between Touch and Mid-Air Gestures for Large-Display Interaction. In *Human-Computer Interaction – INTERACT 2015* (Cham, 2015), J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler, Eds., Lecture Notes in Computer Science, Springer International Publishing, pp. 455–473.

[87] JANG, S., STUERZLINGER, W., AMBIKE, S., AND RAMANI, K. Modeling Cumulative Arm Fatigue in Mid-Air Interaction based on Perceived Exertion and Kinetics of Arm Motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2017), CHI '17, Association for Computing Machinery, pp. 3328–3339.

[88] JONES, K. S., MCINTYRE, T. J., AND HARRIS, D. J. Leap Motion- and Mouse-Based Target Selection: Productivity, Perceived Comfort and Fatigue, User Preference, and Perceived Usability. *International Journal of Human–Computer Interaction 36*, 7 (Apr. 2020), 621–630. _eprint: https://doi.org/10.1080/10447318.2019.1666511.

[89] JUDE, A., POOR, G. M., AND GUINNESS, D. An evaluation of touchless hand gestural interaction for pointing tasks with preferred and non-preferred hands. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational* (New York, NY, USA, Oct. 2014), NordiCHI '14, Association for Computing Machinery, pp. 668–676.

[90] JUDE, A., POOR, G. M., AND GUINNESS, D. Grasp, Grab or Pinch? Identifying User Preference for In-Air Gestural Manipulation. In *Proceedings of the 2016 Symposium on Spatial User Interaction* (New York, NY, USA, Oct. 2016), SUI '16, Association for Computing Machinery, p. 219.

[91] KAMEL BOULOS, M. N., BLANCHARD, B. J., WALKER, C., MONTERO, J., TRIPATHY, A., AND GUTIERREZ-OSUNA, R. Web GIS in practice X: a Microsoft Kinect natural user interface for Google Earth navigation. *International Journal of Health Geographics 10*, 1 (July 2011), 45.

[92] KARAM, M., AND SCHRAEFEL, M. C. A Taxonomy of Gestures in Human Computer Interactions, 2005. ISBN: 9780854328338.

[93] KATTINAKERE, R. S., GROSSMAN, T., AND SUBRAMANIAN, S. Modeling steering within above-the-surface interaction layers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), CHI '07, Association for Computing Machinery, p. 317–326.

[94] KIM, H., OH, S., HAN, S. H., AND CHUNG, M. K. Motion–Display Gain: A New Control–Display Mapping Reflecting Natural Human Pointing Gesture to Enhance Interaction with Large Displays at a Distance. *International Journal of Human–Computer Interaction 35*, 2 (Jan. 2019), 180–195. _eprint: https://doi.org/10.1080/10447318.2018.1447422.

[95] KIM, J., SON, J., AND OAKLEY, I. Cross, Dwell, or Pinch: Designing and Evaluating Around-Device Selection Methods for Unmodified Smartwatches. CHI '25, Association for Computing Machinery, pp. 1–11.

[96] KOPPER, R., BOWMAN, D. A., SILVA, M. G., AND MCMAHAN, R. P. A human motor behavior model for distal pointing tasks. *International Journal of Human-Computer Studies 68*, 10 (Oct. 2010), 603–615.

[97] KOTA, S., AND YOUNG, J. G. Effects of Control-Display Mapping and Spatially Dependent Gain on Supported Free-Hand Gesture Pointing Performance. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 63*, 1 (Nov. 2019), 391–395.

[98] KOUTSABASIS, P., AND VOGIATZIDAKIS, P. Empirical Research in Mid-Air Interaction: A Systematic Review. *International Journal of Human–Computer Interaction 35*, 18 (Nov. 2019), 1747–1768. _eprint: https://doi.org/10.1080/10447318.2019.1572352.

[99] LAFUENTE, M., ELIZONDO, S., FERNÁNDEZ, U. J., AND MARZO, A. Comparing a Mid-air Two-Hand Pinching Point-and-Click Technique with Mouse, Keyboard and TouchFree. In *Proceedings of the XXIII International Conference on Human Computer Interaction* (New York, NY, USA, Jan. 2024), Interaccion '23, Association for Computing Machinery, pp. 1–4.

[100] LÉCUYER, A., BURKHARDT, J.-M., AND ETIENNE, L. Feeling bumps and holes without a haptic interface: the perception of pseudo-haptic textures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '04* (2004), vol. 6, pp. 239–246.

[101] LI, Y., TAG, B., DAI, S., CROWTHER, R., DWYER, T., IRANI, P., AND ENS, B. NICER: A New and Improved Consumed Endurance and Recovery Metric to Quantify

Muscle Fatigue of Mid-Air Interactions. *ACM Transactions on Graphics (TOG) 43*, 4 (July 2024), 102:1–102:14.

[102] LIMERICK, H. Call to interact: communicating interactivity and affordances for contactless gesture controlled public displays. In *Proceedings of the 9TH ACM International Symposium on Pervasive Displays* (New York, NY, USA, June 2020), PerDis '20, Association for Computing Machinery, pp. 63–70.

[103] LIN, C. J., CHEN, H.-J., AND CHOI, J.-H. The postural and control-display gain effects of distal pointing on upper extremity fatigue. *Ergonomics 59*, 1 (Jan. 2016), 73–84. _eprint: https://doi.org/10.1080/00140139.2015.1055824.

[104] LIU, L., AND LIERE, R. V. *The Effect of Varying Path Properties in Path Steering Tasks*. The Eurographics Association, 2010. Accepted: 2014-01-27T11:13:43Z.

[105] LIU, L., MARTENS, J.-B., AND VAN LIERE, R. Revisiting path steering for 3D manipulation tasks. *International Journal of Human-Computer Studies 69*, 3 (Mar. 2011), 170–181.

[106] LJUBIC, S., GLAVINIC, V., AND KUKEC, M. Finger-Based Pointing Performance on Mobile Touchscreen Devices: Fitts' Law Fits. In *Universal Access in Human-Computer Interaction. Access to Today's Technologies* (Cham, 2015), M. Antona and C. Stephanidis, Eds., Springer International Publishing, pp. 318–329.

[107] LOPES, D. S., PARREIRA, P. D. D. F., PAULO, S. F., NUNES, V., REGO, P. A., NEVES, M. C., RODRIGUES, P. S., AND JORGE, J. A. On the utility of 3D hand cursors to explore medical volume datasets with a touchless interface. *Journal of Biomedical Informatics 72* (Aug. 2017), 140–149.

[108] LTD., U. Camera placement, 2021. Last Accessed 25/04/2022.

[109] LTD., U. Touch free exhibit arrives at the aquarium of the pacific. `https://www.ultraleap.com/company/news/press-release/touchfree-aquarium-pacific-exhibit/`, Jul 2021.

[110] LTD., U. TouchFree For Kiosks and Digital Signage, 2021. Last Accessed 25/04/2022.

[111] LUBOS, P., BRUDER, G., AND STEINICKE, F. Analysis of direct selection in head-mounted display environments. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)* (Mar. 2014), pp. 11–18.

[112] LUO, Y., AND VOGEL, D. Crossing-based selection with direct touch input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2014), CHI '14, Association for Computing Machinery, pp. 2627–2636.

[113] LYSTBÆK, M. N., ROSENBERG, P., PFEUFFER, K., GRØNBÆK, J. E., AND GELLERSEN, H. Gaze-Hand Alignment: Combining Eye Gaze and Mid-Air Pointing for Interacting with Menus in Augmented Reality. *Proc. ACM Hum.-Comput. Interact. 6*, ETRA (May 2022), 145:1–145:18.

[114] MACKENZIE, I. S., AND BUXTON, W. Extending Fitts' law to two-dimensional tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, June 1992), CHI '92, Association for Computing Machinery, pp. 219–226.

[115] MAIKE, V. R. M. L., DE SOUSA BRITTO NETO, L., BARANAUSKAS, M. C. C., AND GOLDENSTEIN, S. K. Seeing through the Kinect: A Survey on Heuristics for Building Natural User Interfaces Environments. In *Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access* (Cham, 2014), C. Stephanidis and M. Antona, Eds., Springer International Publishing, pp. 407–418.

[116] MARKUSSEN, A., JAKOBSEN, M. R., AND HORNBÆK, K. Selection-Based Mid-Air Text Entry on Large Displays. In *Human-Computer Interaction – INTERACT 2013* (Berlin, Heidelberg, 2013), P. Kotzé, G. Marsden, G. Lindgaard, J. Wesson, and M. Winckler, Eds., Lecture Notes in Computer Science, Springer, pp. 401–418.

[117] MCGUFFIN, M., AND BALAKRISHNAN, R. Acquisition of expanding targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2002), CHI '02, Association for Computing Machinery, pp. 57–64.

[118] MCNEILL, D. *Gesture and Thought*. University Of Chicago Press, 2007.

[119] MEYER, D. E., ABRAMS, R. A., KORNBLUM, S., WRIGHT, C. E., AND KEITH SMITH, J. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological review 95*, 3 (1988).

[120] MINE, M. R., BROOKS, F. P., AND SEQUIN, C. H. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (USA, Aug. 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 19–26.

[121] MORRIS, M. R., DANIELESCU, A., DRUCKER, S., FISHER, D., LEE, B., SCHRAEFEL, M. C., AND WOBBROCK, J. O. Reducing legacy bias in gesture elicitation studies. *Interactions 21*, 3 (May 2014), 40–45.

[122] MOTT, M. E., AND WOBBROCK, J. O. Beating the bubble: using kinematic triggering in the bubble lens for acquiring small, dense targets. In *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems (New York, NY, USA, Apr. 2014), CHI '14, Association for Computing Machinery, pp. 733–742.

[123] MUTASIM, A. K., BATMAZ, A. U., AND STUERZLINGER, W. Pinch, Click, or Dwell: Comparing Different Selection Techniques for Eye-Gaze-Based Pointing in Virtual Reality. In *ACM Symposium on Eye Tracking Research and Applications* (New York, NY, USA, May 2021), ETRA '21 Short Papers, Association for Computing Machinery, pp. 1–7.

[124] MÄKELÄ, V., HEIMONEN, T., LUHTALA, M., AND TURUNEN, M. Information wall: evaluation of a gesture-controlled public display. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia* (New York, NY, USA, Nov. 2014), MUM '14, Association for Computing Machinery, pp. 228–231.

[125] MÄKELÄ, V., WINTER, J., SCHWAB, J., KOCH, M., AND ALT, F. Pandemic Displays: Considering Hygiene on Public Touchscreens in the Post-Pandemic Era. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2022), CHI '22, Association for Computing Machinery, pp. 1–12.

[126] MÜLLER, J., WALTER, R., BAILLY, G., NISCHT, M., AND ALT, F. Looking glass: a field study on noticing interactivity of a shop window. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2012), CHI '12, Association for Computing Machinery, pp. 297–306.

[127] MÜLLER-TOMFELDE, C. Dwell-Based Pointing in Applications of Human Computer Interaction. In *Human-Computer Interaction – INTERACT 2007* (Berlin, Heidelberg, 2007), C. Baranauskas, P. Palanque, J. Abascal, and S. D. J. Barbosa, Eds., Lecture Notes in Computer Science, Springer, pp. 560–573.

[128] NANCEL, M., CHAPUIS, O., PIETRIGA, E., YANG, X.-D., IRANI, P. P., AND BEAUDOUIN-LAFON, M. High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2013), CHI '13, Association for Computing Machinery, pp. 831–840.

[129] NANCEL, M., PIETRIGA, E., CHAPUIS, O., AND BEAUDOUIN-LAFON, M. Mid-Air Pointing on Ultra-Walls. *ACM Transactions on Computer-Human Interaction 22*, 5 (Aug. 2015), 21:1–21:62.

[130] NANCEL, M., WAGNER, J., PIETRIGA, E., CHAPUIS, O., AND MACKAY, W. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2011), CHI '11, Association for Computing Machinery, pp. 177–186.

[131] NORMAN, D. A. Affordance, conventions, and design. *interactions 6*, 3 (1999), 38–43.

[132] NORMAN, D. A. The way i see it signifiers, not affordances. *interactions 15*, 6 (2008), 18–19.

[133] NORMAN, D. A. Natural user interfaces are not natural. *interactions 17*, 3 (May 2010), 6–10.

[134] O'HARA, K., HARPER, R., MENTIS, H., SELLEN, A., AND TAYLOR, A. On the naturalness of touchless: Putting the interaction back into NUI. *ACM Transactions on Computer-Human Interaction 20*, 1 (Apr. 2013), 5:1–5:25.

[135] PARK, Y., KIM, J., AND LEE, K. Effects of Auditory Feedback on Menu Selection in Hand-Gesture Interfaces. *IEEE MultiMedia 22*, 1 (Jan. 2015), 32–40. Conference Name: IEEE MultiMedia.

[136] PAUL, C. L., AND BRADEL, L. Size matters: the effects of interactive display size on interaction zone expectations. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces* (New York, NY, USA, May 2018), AVI '18, Association for Computing Machinery, pp. 1–5.

[137] PAVLOVIC, V., SHARMA, R., AND HUANG, T. Visual interpretation of hand gestures for human-computer interaction: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 7 (July 1997), 677–695.

[138] PEARSON, J., BAILEY, G., ROBINSON, S., JONES, M., OWEN, T., ZHANG, C., REITMAIER, T., STEER, C., CARTER, A., SAHOO, D. R., AND RAJU, D. K. Can't Touch This: Rethinking Public Technology in a COVID-19 Era. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2022), CHI '22, Association for Computing Machinery, pp. 1–14.

[139] PFEUFFER, K., ALEXANDER, J., CHONG, M. K., AND GELLERSEN, H. Gaze-touch: combining gaze with multi-touch for interaction on the same surface. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (New York, NY, USA, Oct. 2014), UIST '14, Association for Computing Machinery, pp. 509–518.

[140] PFEUFFER, K., MAYER, B., MARDANBEGI, D., AND GELLERSEN, H. Gaze + pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (Brighton United Kingdom, Oct. 2017), ACM, pp. 99–108.

[141] POHL, H., AND MURRAY-SMITH, R. Focused and casual interactions: allowing users to vary their level of engagement. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2013), CHI '13, Association for Computing Machinery, pp. 2223–2232.

[142] QUALCOMM. Icymi: Qualcomm brought the future of retail to life at nrf 2022 [video], Apr. 2022.

[143] RACHAVETI, D., CHAKRABHAVI, N., SHANKAR, V., AND SKM, V. Thumbs up: movements made by the thumb are smoother and larger than fingers in finger-thumb opposition tasks. *PeerJ 6* (Oct. 2018), e5763.

[144] RAM, S., MAHADEVAN, A., RAHMAT-KHAH, H., TURINI, G., AND YOUNG, J. G. Effect of control-display gain and mapping and use of armrests on accuracy in temporally limited touchless gestural steering tasks. *Proceedings of the Human Factors and Ergonomic Society annual meeting 61*, 1 (2017), 380–384.

[145] RIYAL, R., PATEL, A. D., MURUGESAN, T., TURINI, G., AND YOUNG, J. G. Effect of control-display transfer function on pointing performance for a hand/finger based touchless gestural controls: a preliminary investigation. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 59*, 1 (Sept. 2015), 1085–1089.

[146] SHNEIDERMAN, B. Direct manipulation: A step beyond programming languages. *Computer 16*, 8 (Aug. 1983), 57–69.

[147] SIDENMARK, L., PARENT, M., WU, C.-H., CHAN, J., GLUECK, M., WIGDOR, D., GROSSMAN, T., AND GIORDANO, M. Weighted Pointer: Error-aware Gaze-based Interaction through Fallback Modalities. *IEEE Transactions on Visualization and Computer Graphics 28*, 11 (Nov. 2022), 3585–3595. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[148] SIDENMARK, L., SUN, Z., AND GELLERSEN, H. Cone&Bubble: Evaluating Combinations of Gaze, Head and Hand Pointing for Target Selection in Dense 3D Environments. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (Mar. 2024), pp. 642–649.

[149] SIGNORI, A., SORMANI, M. P., SCHIAVETTI, I., BISIO, A., BOVE, M., AND BONZANO, L. Quantitative assessment of finger motor performance: Normative data. *PLOS ONE 12*, 10 (Oct. 2017), e0186524.

[150] SOUKOREFF, R. W., AND MACKENZIE, I. S. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *International journal of human-computer studies 61*, 6 (2004), 751–789.

[151] SPADACCINI, J. Touchless gesture-based exhibits, part two: Full-body interaction, Apr 2020.

[152] SRIDHAR, S., FEIT, A. M., THEOBALT, C., AND OULASVIRTA, A. Investigating the dexterity of multi-finger input for mid-air text entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2015), CHI '15, Association for Computing Machinery, p. 3643–3652.

[153] SURALE, H. B., MATULIC, F., AND VOGEL, D. Experimental Analysis of Barehand Mid-air Mode-Switching Techniques in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow Scotland Uk, May 2019), ACM, pp. 1–14.

[154] TEATHER, R. J., AND STUERZLINGER, W. Pointing at 3d targets in a stereo head-tracked virtual environment. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)* (2011), IEEE, pp. 87–94.

[155] TENG, Z. F. Next Generation of Human-Computer Interaction Technology Based on Kinect. *Applied Mechanics and Materials 329* (2013), 478–482.

[156] THOMPSON, S., SLOCUM, J., AND BOHAN, M. Gain and Angle of Approach Effects on Cursor-Positioning Time with a Mouse in Consideration of Fitts' Law. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 48*, 5 (Sept. 2004), 823–827.

[157] ULTRALEAP. Touchless ticket machine | powered by ultraleap, Oct. 2020.

[158] ULTRALEAP LTD. LEGO: Touchless Interactive Play Experience, 2021.

[159] ULTRALEAP LTD. Touch free exhibit arrives at the Aquarium of the Pacific, 2021.

[160] ULTRALEAP LTD. PepsiCo innovation: Touch less, smile more!, 2022.

[161] ULTRALEAP LTD. UST Integrates Ultraleap Tech To Power Faster Checkout Experiences, 2022.

[162] VAN DEN BOGAERT, L., AND GEERTS, D. User-Defined Mid-Air Haptic Sensations for Interacting with an AR Menu Environment. In *Haptics: Science, Technology, Applications* (Cham, 2020), I. Nisky, J. Hartcher-O'Brien, M. Wiertlewski, and J. Smeets, Eds., Springer International Publishing, pp. 25–32.

[163] VANACKEN, L., GROSSMAN, T., AND CONINX, K. Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments. In *2007 IEEE Symposium on 3D User Interfaces* (Mar. 2007).

[164] VATAVU, R.-D., UNGUREAN, O.-C., AND BILIUS, L.-B. Interactive Public Displays and Wheelchair Users: Between Direct, Personal and Indirect, Assisted Interaction. In

Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (New York, NY, USA, Oct. 2022), UIST '22, Association for Computing Machinery, pp. 1–17.

[165] VELLOSO, E., CARTER, M., NEWN, J., ESTEVES, A., CLARKE, C., AND GELLERSEN, H. Motion Correlation: Selecting Objects by Matching Their Movement. *ACM Transactions on Computer-Human Interaction 24*, 3 (Apr. 2017), 22:1–22:35.

[166] VERMEULEN, J., LUYTEN, K., VAN DEN HOVEN, E., AND CONINX, K. Crossing the bridge over Norman's Gulf of Execution: revealing feedforward's true identity. CHI '13, Association for Computing Machinery, pp. 1931–1940.

[167] VO, D.-B., AND BREWSTER, S. Touching the Invisible: Localizing Ultrasonic Haptic Cues. In *Proceedings of World Haptics Conference 2015 - WHC '15* (2015), IEEE, pp. 368 – 373.

[168] VOGEL, D., AND BALAKRISHNAN, R. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th annual ACM symposium on User interface software and technology* (New York, NY, USA, Oct. 2005), UIST '05, Association for Computing Machinery, pp. 33–42.

[169] VOGIATZIDAKIS, P., AND KOUTSABASIS, P. Gesture Elicitation Studies for Mid-Air Interaction: A Review. *Multimodal Technologies and Interaction 2*, 4 (Dec. 2018), 65. Number: 4.

[170] VYSOCKÝ, A., GRUSHKO, S., OŠČÁDAL, P., KOT, T., BABJAK, J., JÁNOŠ, R., SUKOP, M., AND BOBOVSKÝ, Z. Analysis of Precision and Stability of Hand Tracking with Leap Motion Sensor. *Sensors 20*, 15 (Jan. 2020), 4088.

[171] WACHS, J. P., KÖLSCH, M., STERN, H., AND EDAN, Y. Vision-based hand-gesture applications. *Commun. ACM 54*, 2 (Feb. 2011), 60–71.

[172] WAGNER, U., LYSTBÆK, M. N., MANAKHOV, P., GRØNBÆK, J. E. S., PFEUFFER, K., AND GELLERSEN, H. A Fitts' Law Study of Gaze-Hand Alignment for Selection in 3D User Interfaces. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2023), CHI '23, Association for Computing Machinery, pp. 1–15.

[173] WAUGH, K., GUTIERREZ, M., AND KEHOE, A. Could the Stylus Be Mightier Than the Controller? An Investigation Into the Heisenberg Effect of Spatial Interaction for 3D Tracked Styluses. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2026), CHI EA '26, Association for Computing Machinery, pp. 1–6.

[174] WAUGH, K., AND ROBERTSON, J. Don't Touch Me! A Comparison of Usability on Touch and Non-touch Inputs. In *Human-Computer Interaction – INTERACT 2021* (Cham, 2021), C. Ardito, R. Lanzilotti, A. Malizia, H. Petrie, A. Piccinno, G. Desolda, and K. Inkpen, Eds., Lecture Notes in Computer Science, Springer International Publishing, pp. 400–404.

[175] WHIFFING, J., LANGLOTZ, T., LUTTEROTH, C., SHARMA, A., AND CLARKE, C. Understanding Freehand Cursorless Pointing Variability and Its Impact on Selection Performance. *ACM Trans. Comput.-Hum. Interact.* (Oct. 2025). Just Accepted.

[176] WHITLOCK, M., HARNNER, E., BRUBAKER, J. R., KANE, S., AND SZAFIR, D. A. Interacting with Distant Objects in Augmented Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (Mar. 2018), pp. 41–48.

[177] WIGDOR, D., AND WIXON, D. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*, 1 ed. Morgan Kaufmann Publishers Inc., 2011.

[178] WILLIAMSON, J. Fingers of a Hand Oscillate Together: Phase Syncronisation of Tremor in Hover Touch Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2016), CHI '16, Association for Computing Machinery, pp. 3433–3437.

[179] WILSON, A. D. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2006), UIST '06, Association for Computing Machinery, p. 255–258.

[180] WOBBROCK, J. O., FINDLATER, L., GERGLE, D., AND HIGGINS, J. J. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, Association for Computing Machinery, p. 143–146.

[181] WOLF, D., GUGENHEIMER, J., COMBOSCH, M., AND RUKZIO, E. Understanding the Heisenberg Effect of Spatial Interaction: A Selection Induced Error for Spatially Tracked Input Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2020), CHI '20, Association for Computing Machinery, pp. 1–10.

[182] XU, W., MENG, X., YU, K., SARCAR, S., AND LIANG, H.-N. Evaluation of Text Selection Techniques in Virtual Reality Head-Mounted Displays. IEEE Computer Society, pp. 131–140.

[183] YAMANAKA, S. Steering Performance with Error-accepting Delays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2019), CHI '19, Association for Computing Machinery, pp. 1–9.

[184] YOO, S., PARKER, C., KAY, J., AND TOMITSCH, M. To Dwell or Not to Dwell: An Evaluation of Mid-Air Gestures for Large Information Displays. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction* (New York, NY, USA, Dec. 2015), OzCHI '15, Association for Computing Machinery, pp. 187–191.

[185] YU, D., SYIEM, B. V., IRLITTI, A., DINGLER, T., VELLOSO, E., AND GONCALVES, J. Modeling Temporal Target Selection: A Perspective from Its Spatial Correspondence. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2023), CHI '23, Association for Computing Machinery, pp. 1–14.

[186] ZHAI, S. User performance in relation to 3D input device design. *SIGGRAPH Comput. Graph. 32*, 4 (Nov. 1998), 50–54.

[187] ZIMMERMAN, T. G., LANIER, J., BLANCHARD, C., BRYSON, S., AND HARVILL, Y. A hand gesture interface device. *SIGCHI Bull. 17*, SI (May 1986), 189–192.